



47
2e1.

Universidad Nacional Autónoma de México
Facultad de Ingeniería

**Diseño y construcción de un equipo para medición de
permeabilidad en textiles**

Tesis
que para obtener el título de
Ingeniero Mecánico Electricista
(área eléctrica-electrónica)
presenta
Tomás Covarrubias Velasco

Dirección de tesis: Ing. Ma. Jaqueline Guadarrama Lihó

1997

TESIS CON
FALLA DE ORIGEN

Ciudad Universitaria, México D. F.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos.

Indudablemente sería imposible mencionar a todos aquellos amigos, profesores y familiares que colaboraron con el trabajo de esta tesis, sin embargo, quiero hacer mención de aquellos que colaboraron directamente con asesorías y revisiones a este trabajo:

- Ing. Jaqueline Guadarrama Liho, Departamento de Mecatrónica, DIMEI.
- Ing. Jesús Manuel Dorador, C.D.M.
- Mto. Víctor González Villcla, Departamento de Mecatrónica, DIMEI.
- Mto. Ernesto Ríos Patrón, School of Chemics, University of Illinois at Urbana Champaign.
- Dr. Antonio del Río, C.I.E. en Temixco, Morelos.
- Dr. Federico Méndez Lavielle, Laboratorio de Aerodinámica, F. I.
- Prof. Ing. Gabriel Castillo Hernández, Sistemas, I.I.
- Ing. José Manuel Covarrubias Solís, Dirección, F. I.
- Mto. Luis Marcial Hernández, Centro de Diseño Electrónico, F. I.
- Dra. Julia Tagüña, C.I.E. en Temixco, Morelos.
- Mto. Lauro Santiago Cruz, Instrumentación, I.I.
- Prof. Ing. Roberto Macías Pérez, Centro de Diseño Electrónico, F.I.
- Prof. Ing. Rogelio Gloria Arredondo, Laboratorio de Equipo Eléctrico, F. I.
- Prof. Ing. Antonio Salva Calleja, Departamento de Control, F.I.
- Prof. Ing. Eduardo Ramírez, Centro de Diseño Electrónico, F. I.
- Prof. Ing. Moisés Rucda, F. I.
- Mr. Bode Buckley, contacto en la ASTM.
- Mr. Robert Holcombe, D13.59 Textils Subcommittee Chairman.
- Lic. Enrique Lask Marcoshamer, Director de Operaciones, ORBITEL S. A. de C. V.
- Lic. Margarita Magriña Ocampo, Diseño Gráfico, ORBITEL S. A. de C. V.
- Ing. Fernando Ramos Aguilar, MACODISA S. A.
- Lic. María Elisa Covarrubias Velasco.
- Lic. Alejandro Gutiérrez Otero.
- Lic. María Citlalli Castillo Béjar.
- Al personal del Laboratorio de Equipo Eléctrico.
- Al Sr. Rafael Paredes Alvarado y al personal de los talleres del Centro de Diseño y Manufactura.

Índice.

Introducción.	iv
Capítulo 1.	1
<i>Antecedentes.</i>	
1.1 Porosidad, ley y experimentos de Darcy sobre la permeabilidad.	1
1.2 La norma ASTM D-737.	7
1.3 Observaciones y aclaraciones a la norma.	9
1.4 Discusión sobre las ecuaciones de Darcy y la norma D-737.	9
1.5 Conclusiones sobre la norma y Darcy.	12
1.6 Método propuesto.	13
Capítulo 2.	14
<i>Modelo Mecánico.</i>	
2.1 Términos utilizados en mecánica de fluidos.	14
2.2 Medición de flujo.	15
2.3 Modelo mecánico. Determinación de parámetros de diseño.	26
2.4 Observaciones sobre la selección de elementos del sistema mecánico.	35
2.5 Elementos seleccionados.	36
2.6 Nuevo cálculo del intervalo de operación.	37
Capítulo 3.	38
<i>El extractor de aire. Control del motor.</i>	
Parte I. El extractor de aire y la electrónica de control.	39
3.1 El motor universal.	39
3.2 Pruebas con regulador lineal aplicadas al motor universal.	42
3.3 Pruebas con regulador conmutado.	44
3.4 Algunas observaciones y resultados en la prueba del regulador conmutado.	49
3.5 Descripción del circuito controlador.	49
Parte II. Los transductores de presión.	73

Capítulo 4.

<i>Diseño del control.</i>	77
4.1 La necesidad del control en el aparato medidor de permabilidad.	77
4.2 Definiciones.	77
4.3 Tipos de control.	78
4.4 Observaciones al sistema medidor de permabilidad.	83
4.5 Experimentación como método para encontrar un método de control.	85
4.6 Curvas de respuesta y experimentación con el aparato medidor.	94
4.7 El modelo matemático correlativo.	98
4.8 El modelo de control.	104
4.9 El modelo de control digital.	108
4.10 Experimentación y pruebas para mejorar la respuesta del sistema.	108

Capítulo 5.

<i>Arquitectura del sistema.</i>	124
5.1 Los elementos de la arquitectura.	124
5.2 El microcontrolador HC11.	126
5.3 Mapa de memoria y decodificación de direcciones.	126
5.4 La pantalla de cristal líquido.	127
5.5 Memorias ROM y RAM.	128
5.6 El reloj de tiempo real (RTC).	128
5.7 La interfase RS-232.	131
5.8 Regulación de voltaje y circuito de supervisión.	131
5.9 Conversión D/A.	132
5.10 Conversión A/D.	132

Capítulo 6.

<i>Programación.</i>	133
6.1 Definición de sistema operativo.	134
6.2 Conceptos generales de sistemas operativos.	134
6.3 Modelo de programación para el sistema basado en el HC11.	137
6.4 Programación en Windows.	161
6.5 Comunicación.	162

Capítulo 7.**Conclusiones y recomendaciones.**

7.1 Sobre la norma.	164
7.2 Sobre el diseño del aparato.	164
7.3 Sobre la electrónica de control.	165
7.4 Sobre la arquitectura y la programación del HC11.	165
7.5 Sobre el control.	167
7.6 Sobre el programa en PC.	167

Bibliografía.

Apéndice A. Norma ASTM D-737.	A1
Apéndice B. Coeficientes de descarga para Vena Contracta.	B1
Apéndice C. Hojas de datos de los circuitos utilizados.	C1
Apéndice D. Curvas de permeabilidad y mediciones.	D1
Apéndice E. Sistema operativo para el sistema HC11.	E1
Apéndice F. Programa para sistema HC11 para la obtención de las curvas de permeabilidad	F1
Apéndice G. Programa para Windows.	G1
Apéndice H. Programas de control Proporcional y Proporcional-integral.	H1
H.1 Control proporcional.	H1
H.2 Control proporcional-integral.	H10

Introducción.

A principios del año pasado se propuso al Centro de Diseño y Manufactura de la Facultad de Ingeniería el desarrollo de un proyecto destinado a la medición de porosidad en textiles. En un principio el proyecto estaría realizado por dos personas, una de ellas encargada del desarrollo de los principios teóricos del modelo mecánico y la otra persona desarrollaría la parte electrónica para el control del aparato medidor de porosidad. De esta manera se me ofreció el desarrollo del sistema electrónico de captura de datos y control.

Desafortunadamente el proyecto se canceló y el desarrollo del modelo se detuvo. Para poder continuar con el trabajo que me fue asignado se necesitaba que la parte mecánica estuviera completamente definida; por ello tomé esta parte del trabajo.

Resultado de estas investigaciones es que existe una diferencia conceptual de lo que se suponía se iba a medir y lo que realmente se estaba midiendo. El aparato no mide porosidad, sino permeabilidad; estos conceptos se describen en el primer capítulo con claridad. También se encontró que el procedimiento para medir permeabilidad está normalizado y estandarizado por la "American Society of Testing Materials" (ASTM), por lo que se procedió a estudiar la norma D-737, que es la referente a pruebas de permeabilidad con aire en textiles, también se describe esta norma en el primer capítulo.

La permeabilidad es un fenómeno físico de la materia que fue estudiado el siglo pasado por Henry Darcy (1808-1858), ingeniero francés dedicado a obras hidráulicas en Dijon. Más adelante se describen los experimentos de Darcy y sus conclusiones.

Conforme se desarrolló el modelo teórico basado en Darcy se encontraron varias incongruencias y vacíos respecto a la norma D-737. Probablemente se debe a que la norma D-737 parte de un procedimiento empírico e industrial. Se pidió más información a la ASTM sobre las relaciones matemáticas del modelo que utilizan, pero no hubo respuesta, por lo que se decidió limitar el alcance de esta tesis y se propone un método alternativo para medir la permeabilidad.

En los siguientes capítulos se establecen las dimensiones físicas del dispositivo medidor y los requerimientos de control. También se habla de control de motores de AC, y el tipo de control seleccionado. El procesador utilizado es el HC11 de Motorola.

En esta tesis se encontrará que se destinó mucho trabajo a las investigaciones del modelo mecánico y a la parte de programación. Como en todo campo de la ingeniería, siempre se presenta una interrelación de áreas. en este caso provino de la interacción de la ingeniería mecánica y técnicas de programación con la ingeniería electrónica. La programación tiene un enfoque especial, se trató de utilizar algunos conceptos de programación que hicieron de esta parte una gran labor.

La magnitud de la parte mecánica se debe fundamentalmente al hecho de que se tuvo que realizar esta parte para poder completar el proyecto; de ahí que se hable del fenómeno de permeabilidad, porosidad, y dinámica de fluidos. Sin embargo, es precisamente esta interrelación de áreas, lo que le da a la ingeniería un carácter especial dentro de las ciencias, pues si bien no se concentra en la investigación de un fenómeno específico, trata de que varios problemas o fenómenos físicos se interrelacionen o integren de manera armoniosa.

Usos de la permeabilidad en textiles.

La permeabilidad con aire es un factor importante en la calidad de los textiles como filtros para gases, en la fabricación de ropa, mosquiteros, paracaídas, tiendas de campaña, etc...

La medición de permeabilidad cobró especial importancia durante las guerras mundiales, pues arroja datos sobre la calidad o resistencia de textiles militares al medio ambiente. Se desarrollaron muchos métodos para medir permeabilidad utilizando distintos fluidos. Este es uno de los métodos comerciales utilizados comúnmente.

Objetivos.

El objetivo de este trabajo es el de llevar a la práctica un proyecto de ingeniería utilizando microcontroladores. Existe un objetivo personal en la programación del microcontrolador HC11: el desarrollo de un sistema operativo completo, enfocado a objetos y de alguna manera flexible. Como se verá más adelante, se retomaron varios conceptos de la programación moderna: objetos, multiproceso, generación de eventos, manejo de memoria de manera dinámica, etc... El alcance de este desarrollo se limitó por razones de tiempo, especialmente si no se cuenta con herramientas de programación y depuración que agilicen esta labor, las cuales, para el procesador HC11, no se encontraron. Sin embargo, se llegó a un modelo de programación muy flexible y que pudiera ser completado posteriormente.

Capítulo 1.- Antecedentes.

En este capítulo se tratan las definiciones y teoría necesarias para la descripción del problema de la permeabilidad en medios porosos. Empieza con la descripción del experimento de Darcy y las ecuaciones que describen esta propiedad de la materia. Las ecuaciones de Darcy y las que se derivan de ésta, siguen un procedimiento totalmente empírico, sin que esto les quite valor científico, aunque existen deducciones teóricas¹. Continúa con la descripción de la norma D737 de la ASTM relativa a la medición de la permeabilidad con aire de textiles. Esta norma es la que establece el procedimiento comercial e industrial para determinar esta propiedad en las telas. El proyecto, en consecuencia, debería cumplir con esta norma para darle uso industrial.

Hay que destacar que la permeabilidad puede ser vista desde dos puntos de vista, a nivel microscópico o molecular (medio discontinuo), o a un nivel macroscópico (medio continuo). Es precisamente el experimento de Darcy el que establece un procedimiento a nivel macro que cumple con las expectativas de este problema de ingeniería.

1.1 Porosidad, ley y experimentos de Darcy sobre la permeabilidad.

El estudio de la permeabilidad es a veces confuso con el estudio de los medios porosos. Esto es porque normalmente se asocia el flujo de alguna sustancia a través de un material como flujo en medios porosos. Sin embargo hay que hacer la distinción entre ambos términos; a lo largo de esta parte se habla indistintamente de permeabilidad y de flujo en medios porosos, aclarando que si existe flujo en un medio poroso, existe permeabilidad.

1.1.1 Medio poroso.

Se define como porosidad a la porción de volumen de espacio vacío en relación al total del volumen analizado de materia. En cierto sentido, se puede decir que la porosidad es equivalente a la densidad: masa por unidad de volumen. Al igual que la densidad, el estudio de la porosidad es extremadamente complejo, pues abarca desde un nivel molecular o microscópico hasta un nivel macroscópico. Partiendo de la definición de porosidad podemos encontrar dos casos: porosidad en un medio homogéneo y porosidad en un medio heterogéneo. Si se analiza un elemento de volumen se encuentra que la porosidad se comporta de la siguiente manera, al igual que la densidad:

¹ Stephen Whitaker, revista "Transport in Porous Media", volumen primero, página 3, 1986.

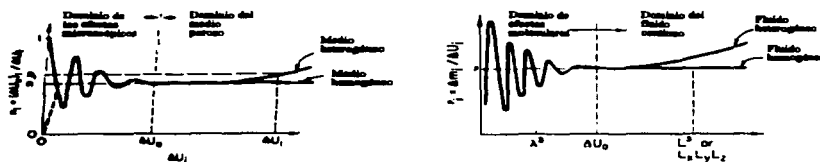


Fig. 1.1. Definiciones de porosidad y densidad.

Puede observarse que mientras más pequeño es el elemento de volumen de estudio, la porosidad, al igual que la densidad, tiene variaciones significativas, esto se debe a que el volumen es tan pequeño que las variaciones en el material tienen un impacto significativo en el volumen analizado (de esta forma puede verse que, para el caso de la porosidad, el volumen de estudio es tan pequeño que puede ser un hueco, línea punteada, o sólo materia, línea continua). Por el contrario mientras más grande es el volumen analizado, las variaciones en el material tienen un impacto menor en el total del volumen. Se puede observar que para cierto intervalo de volúmenes de estudio entre materiales homogéneos² y heterogéneos la porosidad y la densidad pueden coincidir, pero cuando se excede cierto volumen de estudio divergen los resultados.

Porosidad se refiere exclusivamente a la cantidad de volumen vacío respecto al volumen total analizado, esto significa que podemos tener medios porosos que no permitan el paso de fluidos a través de ellos. Esto se debe a dos posibles razones: que no existan conductos entre un extremo y otro del material o que el diámetro de los conductos es tan estrecho que las moléculas del fluido no pueden pasar a través de estos conductos. La siguiente figura ilustra esto:

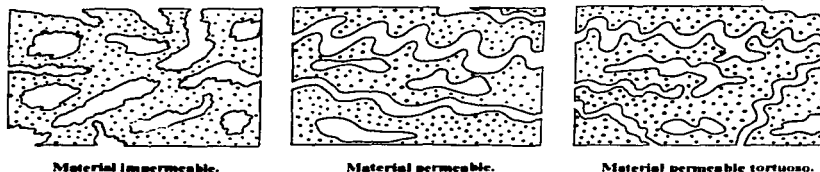


Fig. 1.2.

² El término homogéneo se refiere a que una propiedad es independiente de la posición dentro del medio; heterogéneo es lo contrario.

De esta forma, se puede decir que si un material presenta conductos de un extremo a otro hay percolación y el material presenta cierta permeabilidad; si estos conductos son extremadamente tortuosos, como la figura de la extrema derecha, entonces el material presentará una permeabilidad baja. Algunos investigadores han encontrado una relación entre permeabilidad y porosidad, ésta es válida únicamente en experimentos con ciertos materiales y ciertos fluidos; no existe una relación universal o general entre ambas propiedades³.

1.1.2 Experimento de Darcy, Ley de Darcy y Coeficiente de Permeabilidad.

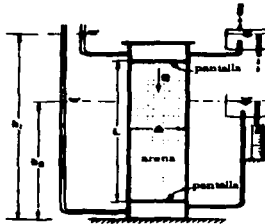


Fig 1.3. Experimento de Darcy con filtro de arena.

Las deducciones de Darcy son particularmente interesantes para casos prácticos de ingeniería o hidrólogos del suelo cuyo interés radica en flujos a través de un medio permeable.

En 1856, Henry Darcy investigó el flujo de agua en filtros de arena verticales en conexión con las fuentes de la ciudad de Dijon en Francia, como se muestra en la figura 1.3. De sus experimentos se concluyó que el gasto de agua Q es proporcional al área de la sección del filtro A , proporcional a la diferencia de alturas $(h_1 - h_2)$ e inversamente proporcional a la longitud L , o espesor del filtro. De mancha combinada queda:

$$Q = \frac{KA(h_1 - h_2)}{L}$$

donde K es un coeficiente de proporcionalidad que se discutirá más adelante.

Al factor $(h_1 - h_2)/L$ se le denomina gradiente hidráulico J , por realizar estos experimentos con agua, y describe la suma de energía de presión y energía potencial del fluido por unidad de longitud. Esta ecuación se puede extender a cualquier medio poroso homogéneo.

³ Existen estudios en medios no continuos que tratan de asociar estos aspectos.

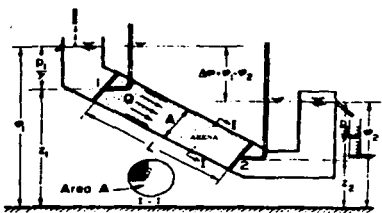


Fig. 1.4. Experimento de Darcy con filtro inclinado.

Si el flujo es inclinado, como se muestra en la figura 1.4, se tiene la ecuación:

$$Q = \frac{KA(\varphi_1 - \varphi_2)}{L}, \quad \varphi_1 = z + p_1 / g.$$

Si el flujo es horizontal, se puede reescribir el gasto de la siguiente manera:

$$Q = \frac{KA(P_1 - P_2)}{gL}.$$

Estas ecuaciones describen el movimiento del fluido del punto de mayor energía al de menor energía.

La velocidad, en estos filtros, está dada por $V=Q/A$.

Haciendo análisis dimensional, las unidades de las variables involucradas son:

$$[Q]=L^3/T; \quad [V]=L/T; \quad [K]=L/T.$$

1.1.3 Generalización de la ley de Darcy.

La ecuación obtenida por Darcy tiene como característica que se utilizó un medio poroso homogéneo e isotrópico⁴, un fluido incompresible, y no hubo cambios de fase (interacciones térmicas). Generalizando para tres dimensiones, suponiendo medio homogéneo e isotrópico y sin interacciones térmicas:

$$q = -K \cdot \text{grad}(\varphi); \quad q = Q / A \quad (\text{eq 1.1}).$$

Esta ecuación es válida aún para un medio heterogéneo e isotrópico, donde K es un tensor.

Existen varias correcciones a la ley de Darcy, básicamente se hicieron porque se consideraron más variables en el experimento. De esta forma se llegó a otra generalización de la ley de Darcy, en la que se

⁴ El término isotrópico se refiere a que una propiedad es independiente de la dirección en la que se hace el análisis, de lo contrario se dice que es anisotrópico.

involucran las características del fluido, densidad y viscosidad. Procurando expresarla en términos de energía:

$$q = -(k\rho / \mu) \cdot \text{grad}(\varphi). \quad (\text{ec. 1.2})$$

En esta ecuación, al coeficiente k se le denomina permeabilidad, diferente del coeficiente de proporcionalidad K , que se le llama coeficiente de permeabilidad. Para un medio anisotrópico existe el problema del término $\text{grad}(\varphi)$, donde se obtendrían distintos valores de K (tensor) para las distintas direcciones y se tendrían componentes de q para las tres direcciones del espacio vectorial:

$$q_x = K_{xx}J_x + K_{xy}J_y + K_{xz}J_z$$

$$q_y = K_{yx}J_x + K_{yy}J_y + K_{yz}J_z$$

$$q_z = K_{zx}J_x + K_{zy}J_y + K_{zz}J_z$$

El cálculo se hace extremadamente complejo. Sin embargo, investigaciones posteriores han demostrado que la ecuación, que representa una relación lineal entre el gradiente y la descarga, es inválida. Por ello varios investigadores han aplicado restricciones a la ley de Darcy.

Estas restricciones son:

- 1.- Es válida para medios isotrópicos.
- 2.- El número de Reynolds (N_{Rc}) para medios porosos⁵ debe estar entre 1 y 10 (frontera entre flujo laminar y turbulento), es decir, que Darcy está basado en flujos laminares en el medio poroso.

Esta serie de ecuaciones fueron aplicadas con fluidos líquidos, concretamente el agua. Sin embargo al hacer experimentos con aire se obtuvieron resultados dispares⁶. Al medir la permeabilidad con aire, se obtuvo un valor mayor al obtenido con agua para un mismo medio poroso. Esto es especialmente cierto cuando los diámetros o secciones del medio poroso son semejantes al diámetro molecular del fluido; si el fluido es gas, las moléculas circularán con mayor facilidad que si fuera un líquido. También sucede algo similar cuando el flujo se da a bajas presiones, el flujo es tan lento que las moléculas de aire no tienen impactos predominantes sobre los conductos debido al movimiento del fluido, sin embargo, su movimiento libre las hace moverse dentro de los conductos de manera aleatoria, afectando el flujo lento debido a la baja presión. Este fenómeno es conocido como efecto Klinkenberg⁷.

⁵ De igual manera que se define el número de Reynolds para tuberías, se ha definido un número de Reynolds para medios porosos. Esta dado por: $N_{Rc} = qd/\nu$, donde d es el radio promedio del hueco en la matriz porosa, ν es la viscosidad cinemática del fluido y q la descarga. El problema en esta ecuación radica en encontrar un diámetro representativo de la matriz, tarea un tanto difícil.

⁶ Jacob Bear, "Dynamics of Fluids in Porous Media", parte 5.2 y 5.3.

⁷ Jacob Bear, "Dynamics of Fluids in Porous Media", parte 5.3.3.

1.1.4 Coeficiente de Permeabilidad y la Permeabilidad.

El coeficiente de permeabilidad que aparece en las distintas formas de la ley de Darcy es también llamado Conductividad Hidráulica (dado que el fluido en el experimento fue agua). En un medio isotrópico puede ser definido como la descarga específica por unidad de gradiente hidráulico. Es un escalar con dimensión L/T , que expresa la facilidad con que un fluido es transportado a través de un medio poroso. Es un término que depende de la matriz porosa y de las propiedades del fluido.

Las propiedades relevantes de un fluido son la densidad y la viscosidad, que se definen como:

- Densidad:** la cantidad de masa por unidad de volumen.
- Viscosidad:** la característica mecánica de un fluido que consiste en deformarse continuamente bajo la acción de un esfuerzo cortante. Es una propiedad independiente del tiempo. Es la razón con la que un fluido se deforma dada una razón de cambio de la distancia entre dos puntos cercanos, moviéndose con el fluido entre ambos puntos. Para fluidos newtonianos es una relación lineal, y es llamada coeficiente de viscosidad o viscosidad simplemente.

Las propiedades relevantes de la materia son el tamaño de los orificios (o tamaño de grano), el perfil del grano o conductos, tortuosidad, superficie específica y porosidad.

De un análisis de las ecuaciones de Darcy se concluye que el coeficiente de permeabilidad está definido como:

$$K = k \gamma / \mu,$$

donde k es dimensionalmente L^2 , llamada permeabilidad intrínseca de la matriz porosa y que depende exclusivamente de las propiedades ésta, y γ/μ representa la influencia de las propiedades del fluido, peso específico y viscosidad. De modo que la ecuación de Darcy puede ser reescrita como:

$$q_i = \frac{\partial P}{\partial x_i}, \quad i = 1, 2, 3, \dots, n \quad (\text{ec. 1.3}).$$

Se han encontrado muchas fórmulas teóricas y otras tantas empíricas para la permeabilidad, esto es debido fundamentalmente a la dificultad de modelar el medio poroso. Es preferible obtener ecuaciones empíricas para un material y fluidos dados, y obtener relaciones entre diámetros de partículas o conductos, que tratar de entender al fluido y al medio poroso a nivel microscópico (medio no continuo). Es por ello que Darcy ofrece una solución empírica al problema de la permeabilidad, es decir, sin grandes desarrollos matemáticos o análisis físico-químicos.

De acuerdo a los experimentos de Darcy, las unidades son:

K es dimensionalmente $[L/T]$ ó $[L^3/T/L^2]$, entendiendo a K como una descarga total Q sobre una sección A cuando se aplica un gradiente de presión. Sus unidades en el sistema internacional son $[m/s]$ ó en $[m^3/s/m^2]$; en el sistema inglés son $[gal/s/ft^2]$.

La constante k es dimensionalmente $[L^2]$ y refleja las propiedades del medio poroso. Sus unidades en el sistema internacional son $[m^2]$ y en el sistema inglés en $[ft^2]$.

1.2 La norma ASTM D-737: "Air Permeability of Textile Fabrics"⁸.

A continuación se describe brevemente la norma D-737, que se puede consultar en el apéndice A.

1.2.1 Definiciones.

Permeabilidad: es el flujo de aire que fluye a través de un material cuando se aplica una diferencia de presiones entre las dos caras del material.

La permeabilidad está expresada en el sistema estadounidense en pies cúbicos por minuto por unidad de área en pies [$\text{ft}^3/\text{min}/\text{ft}^2$].

1.2.2 Aparatos.

La norma establece el procedimiento para medir la permeabilidad con máquinas ya construidas. Estas máquinas consisten en: un extractor de aire que disminuye la presión sobre una muestra colocada en un orificio de diámetro de 2.75 pulgadas (7 cm), un manómetro vertical para medir el gasto, una prensa para colocar la muestra sobre el orificio del tubo de manera que se establezca una diferencia de 0.5 pulgadas de agua entre las caras del textil, una serie de orificios de distintos diámetros (de 1 mm. a 16 mm.) para facilitar las lecturas del manómetro vertical. También se incluyen en esta máquina: una placa calibradora (o varias, según el fabricante) y una tabla de conversiones de gasto a permeabilidad. En la figura 1.1 se muestra el esquema del aparato medidor. En primera instancia se tiene la placa de orificio, cuya diferencia de presiones se utiliza para determinar el gasto. Le sigue, en dirección del flujo, el medio poroso, cuyas caras deben alcanzar una diferencia de presión de 0.5 pulgada de agua; y finalmente el extractor de aire, controlado por un reóstato.

El esquema del aparato medidor es:

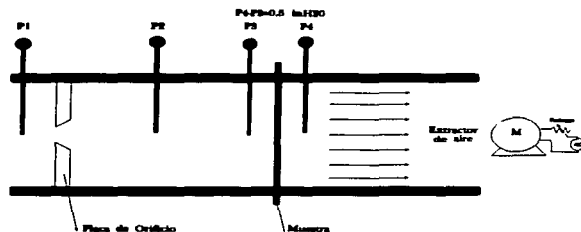


Fig. 1.1. El aparato medidor de la norma ASTM D-737.

⁸ ANSI/ASTM D 737-75: "Standard Method for Air Permeability of Textile Fabrics". Primer tomo del volumen 7 del conjunto de normas ANSI/ASTM, p. 171.

1.2.3 Método.

Se mide el gasto de aire a través de la muestra de área conocida a la que se le aplica una diferencia de presión de 0.5 pulgadas de agua (aproximadamente 124.42 [Pa] o 0.0012442 [Bar]), y a partir de este gasto se determina o se le asocia una permeabilidad⁹.

La norma establece que se debe aplicar una diferencia de presiones de 0.5 pulgadas de agua como valor normalizado, esto se debe a que la permeabilidad de la tela no es una función lineal de la diferencia de presiones¹⁰. Si el textil es demasiado abierto como para alcanzar esta diferencia de presiones, entonces se puede establecer una diferencia menor, pero deberán aclararse en el reporte las condiciones en las que se realizó la prueba.

1.2.4 Condiciones.

No deben existir corrientes de aire en el lugar donde se realicen las pruebas.

No deben existir partículas de polvo que afecten la muestra en más de 0.2% del peso de la muestra antes de iniciar la prueba, durante 2 horas

Humedad relativa entre 2% y 25%.

Temperatura no superior a los 50 °C.

Existen otras condiciones particulares que se establecen para muestras de textiles especiales, y están especificadas en la norma D-1776 de pruebas en textiles¹¹.

1.2.5 Procedimiento.

La norma establece el procedimiento para este aparato.

- Calibración del aparato: una vez establecidas las condiciones de prueba explicadas anteriormente, se asegura que la máquina esté totalmente horizontal con la ayuda del nivel localizado en la base del aparato. Se ajusta el nivel del manómetro inclinado para que dé una lectura de cero, pudiéndose agregar o retirar aceite del mismo. Se debe ajustar el manómetro vertical a cero.
- Seleccionar la placa de prueba e introducirla en el gancho de muestra y colocar el orificio de 4 mm.
- Acelerar el motor hasta alcanzar una diferencia de presión de 0.5 pulgadas en el manómetro inclinado. Tomar la lectura del manómetro vertical. Buscar en la tabla su equivalente en

⁹ La norma dice: "...from this rate of flow the air permeability of the fabric is determined.", sin embargo la norma nunca establece la relación entre flujo de aire y permeabilidad. Por lo que entender la permeabilidad como flujo de aire o gasto por unidad de área es, en rigor, incorrecto.

¹⁰ Esto está explícitamente indicado en el texto de la norma. Más adelante se explica este hecho.

¹¹ ANSI/ASTM Standards, "Standard Recommended Practice for Conditioning Textiles and Textile Products for Testing", Textile, Part 7, tomo 2.

permeabilidad y compararlo con la permeabilidad establecida en la placa de prueba, las variaciones deben estar dentro del rango del 2%.

- d) Si las variaciones están por encima del 2%, verificar que no existan fugas. Las mediciones sólo se pueden realizar si se cumplen los puntos anteriores.
- e) Colocar la muestra en el orificio de muestras. Acelerar al motor hasta alcanzar 0.5 pulgadas en el manómetro inclinado. Verificar la lectura en el manómetro vertical, de manera que la lectura sea mayor a 3 pulgadas, si ésto no puede lograrse, seleccionar un orificio menor hasta sobrepasar las 3 pulgadas.
- f) Con la lectura del manómetro vertical y el orificio seleccionado, consultar la tabla para encontrar la permeabilidad del material.
- g) Utilizar un método estadístico para asegurar que la certeza esté en 95%.

1.3 Observaciones y aclaraciones a la norma.

Como se ha visto, la norma establece un procedimiento para un dispositivo ya fabricado, es decir, no propone un método independiente del aparato. Esto es un problema para el proyecto. La idea del proyecto es diseñar un aparato que entregue los mismos resultados que si se utilizara la máquina establecida por la norma. Ello implica que las lecturas del nuevo aparato tienen que ser validadas con el de la norma, lo que presenta el primer problema: no se tiene acceso a la máquina. Este problema podría ser resuelto si la base teórica del procedimiento estuviera aclarada o explicada en la norma, lo que no es así; se presenta el segundo problema: la norma no establece la relación del gasto con permeabilidad que se obtiene con el aparato.

Un último obstáculo en el uso de la norma es que la automatización del mecanismo ofrece varios aspectos a considerar: primero, que se utiliza el método de orificio para medir gastos, y dadas las variaciones entre un textil cerrado y uno muy abierto, se requieren de varios tipos de orificios para que el manómetro vertical entregue una lectura visible y precisa. No hay transductor que registre tal variación con la resolución del manómetro, pues si observamos los límites de permeabilidad establecidos por la norma veremos que para la muestra más abierta se tiene una permeabilidad de $700 [ft^3/ft^2 \cdot min]$ y para la más cerrada de $1 [ft^3/ft^2 \cdot min]$, esto es, una variación de varios órdenes de magnitud; lo que daría como resultado una variación de los mismos ordenes para el gasto. Si se utilizara un mismo orificio para todo el intervalo, no hay transductor de presión que pueda registrar este rango; segundo, pensar en la automatización del cambio de orificios implica una labor que sale del alcance de esta tesis.

1.4 Discusión sobre las ecuaciones de Darcy la Norma D-737.

Una vez explicados ambos aspectos teóricos, se encontraron varias incongruencias entre la norma y las ecuaciones de Darcy. Primero hay que describir las propiedades del medio poroso, la tela. La tela tiene un comportamiento interesante por estar constituida por varios elementos, una matriz porosa en los

elementos que forman los hilos que la constituyen, una matriz porosa en los mismos hilos, y otra matriz porosa en el conjunto tejido. Por lo tanto la tela es un medio poroso:

- a) Heterogéneo: pues la matriz porosa, la densidad y permeabilidad, son propiedades que cambian de acuerdo a la posición del elemento de estudio dentro de la tela.
- b) Anisotrópico: pues la permeabilidad cambia de acuerdo a la dirección en la que se analiza el medio.

A pesar de que el textil es un medio anisotrópico, las ecuaciones de Darcy pueden aplicarse porque la dirección del flujo en este estudio siempre es la misma, nos interesa la permeabilidad en una sola dirección. La ecuación de Darcy en su forma más sencilla es (ecuación 1.1):

$$q = -K \cdot \text{grad}(\varphi); \quad q = Q / A$$

Dadas estas puntualizaciones, las ecuaciones de Darcy podrían aplicarse al textil, sin embargo, al ser un medio heterogéneo, el término $\text{grad}(\varphi)$ se vuelve difícil de determinar, además la permeabilidad es no lineal con respecto a la presión, lo cual daría distintos valores de permeabilidad para distintos valores de presión. Sin embargo, es por ello que la norma estandariza la presión aplicada a 0.5 pulgadas de agua, para que todas las muestras estén referenciadas a esta presión.

Desde el punto de vista conceptual, tanto la norma como Darcy coinciden en el entendimiento de lo que pretenden determinar: qué tan fácilmente pasa un fluido a través del textil. Sin embargo, si tomamos el concepto rigurosamente encontramos diferencia: la norma define permeabilidad como la cantidad de fluido, gasto, que pasa a través de una sección conocida, área, cuando se aplica una diferencia de presiones, media pulgada de agua; Darcy define el coeficiente de permeabilidad¹² como un coeficiente que expresa la facilidad con que un fluido es transportado a través de la matriz porosa. Es decir, uno mide el gasto por unidad de área, y el otro mide facilidad para el paso del fluido, dos cosas distintas. Es como medir corriente cuando se aplica un voltaje y medir resistencia cuando se aplica tal voltaje; dos conceptos relacionados, pero no iguales. La energía potencial, que es el voltaje, hace circular una corriente a través de un medio conductor. De igual forma, la presión del fluido hace circular un fluido a través de un medio poroso. En ambos casos, la resistencia y el medio poroso, reciben energía del fluido por fricción. Es de esperarse que el flujo a través del conductor y del medio poroso provoque una pérdida de energía en el movimiento del flujo, que se manifiesta en forma de una caída de potencial. Podríamos expresar a la resistencia como la dificultad del paso de la corriente al aplicar un potencial, que es lo que Darcy mide, o bien medir la resistencia como la corriente por unidad de área (densidad de corriente J) que circula por el conductor al aplicar un voltaje estandarizado, que es lo que la norma hace. Realmente, en electricidad, resulta pobre expresar de esta manera a la resistencia, aunque conceptualmente se pretende medir lo mismo.

¹² Y aquí encontramos otra divergencia: la norma hace referencia a la constante de proporcionalidad como permeabilidad, Darcy se refiere a ella como Coeficiente de Permeabilidad, ambas describen a la matriz

Desde el punto de vista dimensional, las unidades de la norma son correctas y corresponden al coeficiente de permeabilidad de Darcy. Sin embargo, la norma establece una presión constante, lo que en Darcy no significa un gradiente constante. Para este caso la ecuación puede resumirse a:

$$q_i = -K \frac{\partial p}{\partial x}.$$

Se puede, entonces, tratar de expresar la ley de Darcy en términos de una presión a través de un espesor de muestra. Por la naturaleza del textil se tendría un comportamiento para el gradiente como una función no lineal. La siguiente gráfica muestra el comportamiento del gasto a través de una muestra de tejido de algodón (franela) cuando se varía la presión entre ambas caras del textil. Para realizar esta prueba se utilizó un tubo en el que se colocó la muestra y se hicieron varios incrementos de presión entre las caras del textil. Con un método de obstrucción (placa de orificio) se determinó el gasto en la tela. Los valores inferiores a 270 [Pa] tienen un error significativo, pues la velocidad en el tubo no es lo suficientemente alta para observar un flujo turbulento. Sin embargo, dan idea del comportamiento real. Esta gráfica muestra que la relación presión-gasto no es lineal para el textil:

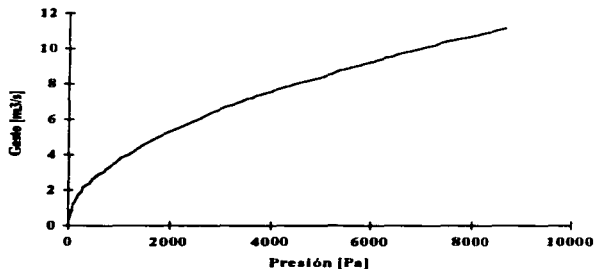


Fig 1.6. Comportamiento no lineal del gasto en el medio poroso al variar la presión.

La idea de describir al comportamiento de la presión de esta manera, partiendo de entender al textil como una caja negra a la cual se le aplica la presión establecida por la norma, parece atractiva. Se entendería al gradiente de presión como la variación de la presión entre dos puntos de la tela, antes y después. El problema ahora es determinar la longitud o espesor del textil. Otra tarea no tanto difícil, ¿cómo determinar el espesor de una tela?, ¿qué criterios se pueden utilizar?. Se tendría, entonces, que normalizar el espesor de un textil, que es algo que por la misma elaboración del textil es imposible, además es en parte determinante en la característica que se pretende medir (es característico de los textiles sus distintos espesores). Esto puede sugerir que en lugar de determinar la permeabilidad de un textil se podría caracterizar a los textiles a través de su gradiente de presión a lo largo de la muestra. Todo esto no es práctico para fines comerciales o industriales. Este método sería válido para textiles cuya

permeabilidad no cambie conforme se aumenta la presión, pero el textil tiende a ofrecer menor resistencia al paso de un fluido cuando se incrementa la presión, y que es el problema de origen, la permeabilidad no es una función lineal de la presión.

Según se lee en la norma y fue confirmado por la ASTM, jamás se hace referencia a espesores de las telas.

Si se analiza la otra expresión de la ley de Darcy (ec. 1.3):

$$q_i = -\left(\frac{k}{\mu}\right) \frac{\partial P}{\partial x_i}, \quad i = 1, 2, 3, \dots, n.$$

En esta ecuación se habla de un nuevo término que involucra exclusivamente las propiedades del medio poroso, la permeabilidad k ($K=k/\mu$). Sin embargo, el análisis dimensional de las unidades de la norma con la permeabilidad de Darcy muestra incompatibilidad; además de que no se ha librado el problema de tratar con espesores de tela.

Con todos estos puntos de análisis entre la norma y la teoría de Darcy surge una gran interrogante: ¿de qué manera se relacionan uno y otro? Como se explicó al principio de este trabajo, se pidió la información a la ASTM y no hubo respuesta.

1.5 Conclusiones sobre la norma y Darcy.

La asesoría consultada¹³ en este tema sugirió que probablemente el procedimiento utilizado en la norma es un método empírico e industrial, lo que implica que la información podría ser privilegiada. Pueden estar utilizando un espesor normalizado, que no indican expresamente; o simplemente calculan el gasto por unidad de área, como lo indican en su definición de permeabilidad, al cual le corresponde una permeabilidad (determinada de otra manera), y que es un procedimiento muy sencillo para aplicaciones comerciales.

Existen entonces dos posibles propuestas: una es determinar exclusivamente el gasto por unidad de área bajo la presión establecida en la norma, o bien proponer un espesor medio y utilizar Darcy para calcular el coeficiente de permeabilidad o la permeabilidad intrínseca.

¹³ Dr. Federico Méndez Laviclle, consulta oral, Laboratorio de Aerodinámica, F. I.
Dr. Antonio del Río, consulta oral, C. I. E., Temixco, Morelos.

1.6 Método propuesto.

El método propuesto es el de determinar la permeabilidad como el flujo de aire que pasa a través del textil cuando se le aplica una diferencia de presiones de 0.5 pulgadas de agua. Es decir, se sigue la definición de la norma. No se espera obtener los mismos resultados, pues habría que determinar la relación que existe entre el gasto dado y la permeabilidad asociada en la norma (validación de resultados).

El procedimiento para determinar esta propiedad es muy similar al propuesto por la norma, pues trata de apearse a la norma lo más posible. La figura 1.6 es un esquema del aparato medidor:

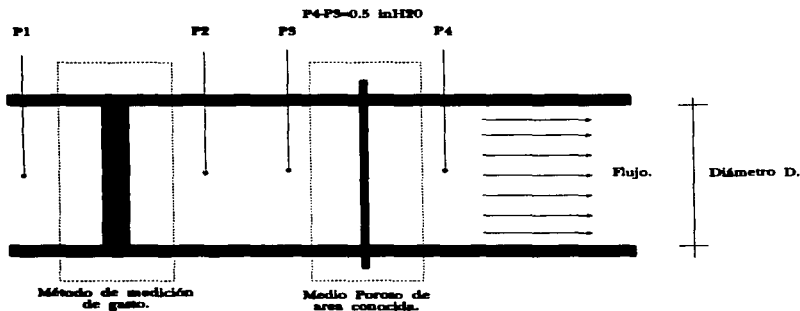


Fig. 1.7. Esquema del aparato medidor.

Para medir la permeabilidad se hace pasar aire a través de una muestra del textil de área conocida. Para ello se utiliza la presión normalizada de 0.5 pulgadas de agua entre las caras del material.

El aparato medidor determina el flujo de aire que pasa a través de la muestra utilizando alguno de los métodos de medición de gasto por caída de presión comúnmente utilizados en la industria y laboratorios: Venturi, placas de orificio o tubo de Pitot.

Para provocar la diferencia de presión se utilizará un extractor o ventilador de aire.

Se pretende automatizar todo el proceso y utilizar herramientas matemáticas y de análisis disponibles para PC, esto implica que el sistema debe transferir la información a la computadora.

En el siguiente capítulo se describe el procedimiento para determinar las dimensiones y parámetros de operación del aparato medidor, también se habla comparativamente de los requerimientos del aparato y lo que se encontró en el mercado.

Capítulo 2.- Modelo Mecánico.

Una vez definido el método, en este capítulo se describe el aparato para determinar el gasto y las ecuaciones que lo gobiernan. Se explican los métodos para medir gasto y el método seleccionado. Se establecen las necesidades para la selección del aparato, transductores y extractor.

2.1 Términos utilizados en mecánica de fluidos.

Antes de analizar los métodos de medición de flujo conviene definir algunos términos utilizados en mecánica de fluidos.

2.1.1 Flujo laminar y turbulento.

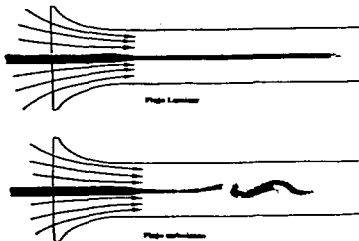


Fig. 2.1. Flujo laminar y flujo turbulento.

El flujo laminar se caracteriza por un movimiento suave de una lámina de fluido sobre otra, mientras que el flujo turbulento se caracteriza por un movimiento irregular y casi alcaatorio superpuesto al movimiento principal del fluido.

Normalmente se fija como punto de referencia para distinguirlos al número de Reynolds.

$$N_{Re} = \frac{\rho v D}{\mu},$$

donde: ρ es la densidad, v es la velocidad, D es el diámetro del tubo y μ es la viscosidad dinámica del fluido.

El número de Reynolds relaciona las fuerzas inerciales con las fuerzas viscosas. Reynolds encontró que la frontera se halla en el valor de 2100, en conductos circulares rectos. Es un parámetro que se utiliza para distinguir los dos tipos de flujo.

2.1.2 Flujo permanente.

Se dice que el flujo es permanente cuando tanto la presión y la velocidad son independientes del tiempo¹.

2.1.3 Problemas de flujo.

Para la solución de la mecánica de fluidos, se utilizan el principio de mecánica de Newton o, lo que es lo mismo, las dos leyes de Euler de mecánica²:

- 1.- La razón de cambio en el momentum de un cuerpo es igual a la fuerza que actúa en él.
- 2.- La razón de cambio en el momento angular (momentum de momentum) es igual al torque que actúa en el cuerpo, visto desde el mismo punto de referencia.

Además de estas leyes, se hace uso de los principios de la Conservación de la Masa y Conservación de la Energía. Haciendo una similitud con las leyes de la mecánica:

- 1.- La razón de cambio de energía de un cuerpo es igual a la razón con la que la energía es entregada al cuerpo.
- 2.- La razón de cambio de masa de un cuerpo es igual a la razón de masa que se le entrega a tal cuerpo.

2.2 Medición de flujo.

Existen cuatro métodos para medición de gasto comúnmente utilizados en el campo de la ingeniería. Ellos son: Venturi, tubo de Pitot, orificio de borde suave y orificio de borde rectangular.

A continuación se describen los cuatro métodos. Se comienza con el método de Venturi porque en éste se hacen las deducciones para llegar al modelo matemático de los otros.

Se toma como premisa, en las deducciones de las ecuaciones, un flujo incompresible. Esto es porque el rango de velocidades que se manejan en el procedimiento de la norma, entra dentro del comportamiento de flujos incompresibles³.

¹ En dinámica de sistemas se dice que el sistema es invariante con el tiempo o en estado estable.

² En mecánica de fluidos estos principios están expresados en las ecuaciones de Navier-Stokes.

³ Se afirma que se debe tratar al aire como un fluido compresible cuando la velocidad esté cercana o sea superior a la velocidad del sonido en el aire. Como referencia para determinar esta frontera se utiliza el número de Mach, que establece una relación entre la velocidad del fluido y la velocidad del sonido:

2.2.1 Medidor Venturi.

El medidor Venturi consiste en una relativa reducción de sección que acelera el flujo produciendo una caída de presión; este proceso continúa hasta una gradual expansión permitiendo que las pérdidas se minimicen. Haciendo un análisis de balances de energía y masa se puede encontrar la relación presión-flujo.

El medidor Venturi tiene la forma

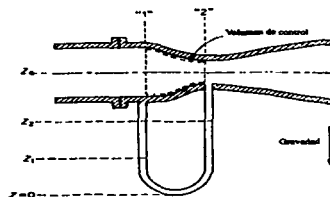


Fig. 2.2. Medidor Venturi.

Se trata de un proceso adiabático y en el que se supone al fluido como incompresible. Si se hace el análisis en estado permanente, se tiene la ecuación de balance de energía:

$$\int_A \frac{1}{2} \rho v^2 v \cdot n \, dA = \int_A V \cdot t \, dA - \int_A \rho \phi v \cdot n \, dA + E\mu.$$

Esta ecuación describe en el término izquierdo de la igualdad a la energía cinética dentro del volumen de control (líneas discontinuas, entre las secciones 1 y 2) y en el lado derecho al trabajo total aplicado al control de volumen en sus entradas, trabajo realizado por la fuerza gravitacional y finalmente la energía disipada debido a la viscosidad. En este caso, dado que se trata de un flujo horizontal en el que no intervienen los efectos de la gravedad y los efectos viscosos son despreciables por tratarse de aire, la simplificación de esta ecuación de balance de energía, utilizando las ecuaciones del gas ideal, es la conocida ecuación de Bernoulli para un fluido incompresible:

$$\frac{P_1}{\rho_1} + \frac{v_1^2}{2g} = \frac{P_2}{\rho_2} + \frac{v_2^2}{2g}.$$

$$N_{Match} = \frac{v_{flujo}}{v_{sonido}}.$$

Si el número de Match es menor a 0.3, se trata de flujo incompresible. Cabe destacar que dependiendo del tipo de análisis y de la experiencia se debe suponer al fluido como compresible.

La ecuación de Bernoulli es una expresión que se obtiene de este balance de energía y supone un fluido incompresible. Aplicando el balance de energía y el balance macroscópico de masa $v_2 A_2 = v_1 A_1$, queda finalmente la ecuación que define el gasto:

$$Q = A_2 v_2 = A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho(1 - (d_2/d_1)^2)}}$$

donde d_1 y d_2 son los diámetros de las secciones 1 y 2 respectivamente.

Varios experimentos han demostrado que esta ecuación no es lo suficientemente exacta para fines de ingeniería, por lo que se ha introducido un factor de corrección denominado coeficiente de descarga, y depende del número de Reynolds⁴ para el medidor. En la gráfica se muestra la gráfica del coeficiente de descarga para una relación de áreas $A_1/A_2=0.25$. Esta corrección se debe a que el fluido se comporta laminarmente para Reynolds inferiores a 2300 haciendo que el perfil de velocidades no sea plano y el coeficiente de descarga decrezca considerablemente.

$$Q = A_2 v_2 = Cd A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho(1 - (d_2/d_1)^2)}}$$

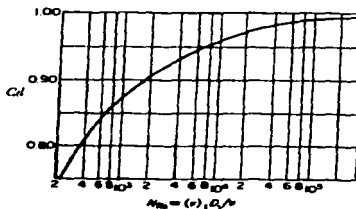


Fig. 2.3. Coeficiente de descarga para Venturi.

Venturi ofrece una medición de flujo de manera muy precisa. El principal problema de utilizar Venturi es su manufactura. ASME propone los parámetros de construcción para poder utilizar los coeficientes y obtener resultados muy precisos. La pérdida total de presión en un medidor Venturi es menor al 20% de la presión diferencial máxima (ver figura 2.9).

⁴ El número de Reynolds permite tener una frontera entre flujos turbulentos y laminares. En un flujo laminar, las fuerzas viscosas predominan sobre las fuerzas que provocan el movimiento del fluido. Cuando las fuerzas viscosas predominan, el perfil de velocidades se vuelve curvo, mientras que con un flujo

2.2.2 Medidor con placa de orificio de borde rectangular.

Este medidor de flujo utiliza una placa de orificio y tiene la siguiente construcción:

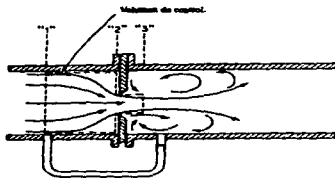


Fig. 2.4. Medidor de placa de orificio.

La solución para el problema no es tan sencilla como para el Venturi, debido a lo poco suave de la contracción del orificio. El volumen de control se toma como se indica en la figura (con las líneas discontinuas). Aplicando balances de energía, despreciando efectos viscosos, y suponiendo un fluido incompresible, se llega a una ecuación similar a la de Venturi, sólo que en ésta se introduce un nuevo factor de corrección denominado coeficiente de contracción y trata de expresar un equivalente del método de orificio para tratarlo como Venturi:

$$A_3 = CcA_2,$$

$$Q = A_2 v_2 = Cc A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho(1 - Cc(A_2/A_1)^2)}}.$$

Para fines prácticos se expresa la ecuación en función de un coeficiente de flujo K_f , que se determina de manera experimental.

$$Q = A_2 v_2 = K_f A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho}}.$$

donde:

$$K_f = \frac{Cc}{\sqrt{1 - Cc(A_2/A_1)^2}}.$$

Los resultados de estas ecuaciones son satisfactorios para números de Reynolds grandes, en la sección del tubo medidor, conforme se analiza en la gráfica:

turbulento, es plano. La idea de un medidor tipo Venturi es que el flujo sea turbulento, para que la distribución de las presiones sea uniforme.

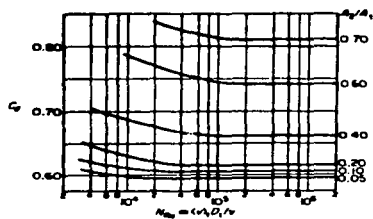


Fig. 2.5. Coeficiente de flujo para orificios.

2.2.3 Medidor de embudo o placa de orificio de borde suave.

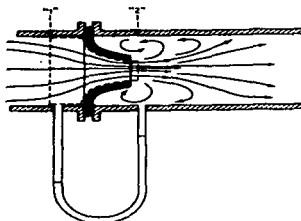


Fig. 2.6. Método de placa de orificio con borde suave.

Con las mismas consideraciones que en los métodos anteriores, se encuentra la ecuación de este método. También existe un factor de corrección obtenido empíricamente:

$$Q = A_2 v_2 = Cd A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho(1 - (A_2/A_1)^2)}}.$$

El coeficiente de descarga es:

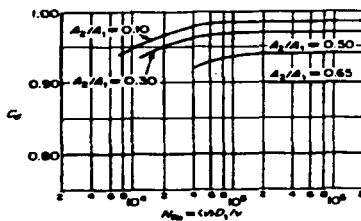


Fig. 2.7. Coeficientes de descarga.

Este método es de una eficiencia mayor que el de placa de orificio con bordes bruscos o no suavizados. La dificultad para el uso de este orificio es la construcción del embudo, éste requiere de trabajo mecánico especial. ASME e ISO tienen estandarizadas las formas del orificio.

2.2.4 Tubo de Pitot.

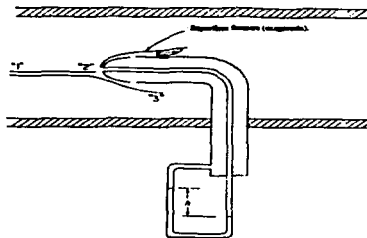


Fig. 2.8 Tubo de Pitot.

Este método es substancialmente diferente a los otros tres. Mientras que los otros métodos miden un gasto en función de presiones suponiendo que el flujo es lo suficientemente turbulento para hacer al perfil de velocidades plano, el tubo de Pitot está enfocado a la medición de velocidades laminares. Utilizando Bernoulli se llega a la ecuación que describe el comportamiento del tubo de Pitot. Para que este dispositivo opere correctamente, se debe tener un número de Reynolds alto, para la sección del tubo, así se

evita que la región cercana al tubo no muestre un perfil grueso, como se muestra en el dibujo, y se pueda suponer que $v_3 = v_j$.

$$v_1 = \sqrt{\frac{2(p_2 - p_3) + \rho g(z_2 - z_3)}{\rho}}$$

Despreciando los efectos gravitatorios:

$$v_1 = \sqrt{\frac{2(p_2 - p_3)}{\rho}}$$

Como en los demás métodos, se introduce un factor de corrección llamado también coeficiente de descarga y es muy cercano a 1.

Las pérdidas totales para los distintos tipos de medidores han sido determinadas también de manera experimental por ASME, la siguiente tabla muestra las pérdidas para los distintos tipos de medidores:

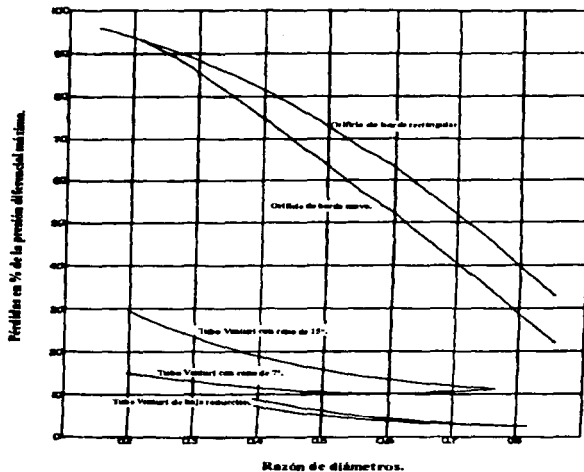


Fig. 2.9. Caída total de presión para los distintos métodos.

2.2.5 Método de medición de flujo seleccionado.

En la norma se utilizan distintos tipos de orificios para la medición de gasto. El método de orificio es el más fácil de desarrollar, pues tanto Venturi, orificio de borde suave y Pitot, requieren de trabajo mecánico preciso en la elaboración de sus partes. Aunque el método de orificio de borde rectangular es el que tiene más errores de precisión y pérdidas, se puede utilizar si se tienen los valores del coeficiente de flujo para hacer las correcciones y si el sistema medidor tiene las semejanzas simétricas y dinámicas propuestas por ASME. El sistema medidor completo incluye un microprocesador que puede perfectamente, en función de las presiones medidas, seleccionar el factor de corrección adecuado. Como se señaló, las dimensiones de este tipo de medidores han sido estandarizadas por la American Society of Mechanical Engineering (ASME)⁵. Esta estandarización permite colocar los transductores de presión en el punto óptimo de manera que se puedan medir las presiones donde las corrientes turbulentas provocadas por el método de orificio se han estabilizado, además de asegurar que el factor de corrección dará los resultados de manera muy aproximada.

El análisis de estos métodos partió de la premisa de utilizar un fluido incompresible. Para las velocidades del aire que se manejan se puede asegurar que se trabaja en la zona de incompresibilidad⁶.

Un punto que debe ser recalcado es que la presión diferencial medida mediante el método de contracción de sección (especialmente para el medidor de orificio) no depende exclusivamente de la contracción del dispositivo, sino también de la localización de los orificios de medición. La localización del orificio o transductor corriente abajo es de particular importancia. Si el orificio de medición se localiza entre el disco del orificio y el plano de vena contracta (es el plano en el que el chorro que sale del orificio tiene la menor sección, ver figura 2.11), el flujo efectivo en la sección donde se encuentra el medidor es menor que el flujo en la sección del orificio de la placa. Se intuye que la velocidad del fluido es mayor que en la placa de orificio; consecuentemente, la diferencia de presiones será mayor también. Dado que el plano de vena contracta es la sección donde el flujo tiene su menor área, la presión diferencial máxima se presentará si se mide en este punto (vena contracta). La figura 2.10 muestra el comportamiento de las presiones a lo largo del tubo, para el método de placa de orificio⁷, d corresponde al diámetro del orificio y D es el diámetro del tubo:

⁵ "Fluid Meters. Their theory and Application", ASME, New York, 1971

⁶ Esto quedará comprobado al determinar los rangos de operación en los apartados siguientes.

⁷ Una descripción precisa de los puntos de medición a lo largo del método está en "Principles and Practice of Flow Meter Engineering". Consultar la bibliografía.

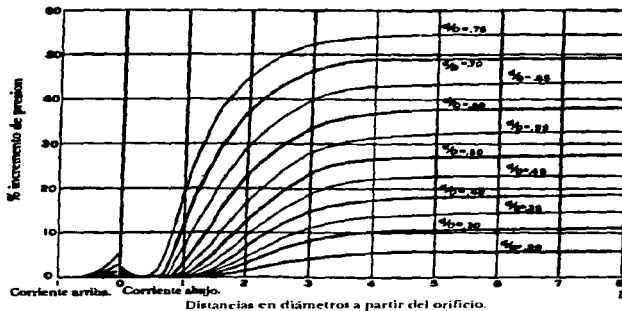


Fig. 2.10. Variación de la presión a lo largo del tubo.

La disminución en la presión estática entre dos secciones da como resultado una medición del cambio de velocidad dentro de estas mismas secciones. Estas cantidades están relacionadas a través del flujo efectivo en las áreas de estas secciones de medición, no necesariamente de la sección del tubo en las secciones o en el área de la vena contracta.

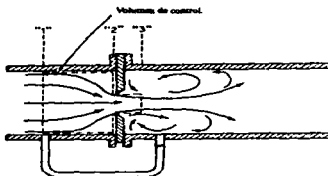


Fig. 2.11 Comportamiento del chorro en placa de orificio.

Si la presión se mide más allá de la sección 3 del dibujo anterior, aproximadamente localizada a 8 veces el diámetro del tubo a partir de la placa de orificio, la disminución de la presión estática es una medida del cambio en la energía cinética resultante de la fricción a través de todo el volumen de control.

A través del coeficiente de flujo, determinado empíricamente, se relacionan la presión diferencial y la velocidad del fluido.

Las características del fluido en movimiento no dependen solamente de las propiedades del fluido ni de las dimensiones geométricas de las secciones del tubo, sino también de todos los agentes externos al

volumen analizado, como comportamientos previos a la entrada del medidor y posteriores al medidor. El estudiar el efecto de cada uno de estos factores generaría un número de cálculos exagerado. Es por esta razón que se han establecido las dimensiones de estos métodos de medición como estándares. Esto implica que al diseñar el método de medición (sea Venturi, orificios, Pitot, etc.) se deben buscar dos condiciones para lograr similitudes cinemáticas:

- 1.- Los sistemas deben ser simétricamente similares.
- 2.- Los flujos deben ser dinámicamente similares.

Es por ello que se toman las recomendaciones y tabulaciones ASME para medidores diferenciales de orificio. De acuerdo a la posición de los orificios de medición se han tabulado los valores experimentales de los métodos de medición mencionados. Para el método de orificio existen tres posibles posiciones de los orificios de medición:

1. "Flange Taps": localizados, en tuberías de diámetro mayor o igual a 2 pulgadas, a una pulgada corriente arriba y a una pulgada corriente abajo. Esta localización simétrica es útil cuando se desea medir flujos bidireccionales. Su precisión es deficiente para los requerimientos comerciales.
2. "Corner Taps". Los puntos de medición están localizados justo en ambas caras de la placa de orificio. Para razones de diámetro d/D altas, se producen turbulencias para las cuales los coeficientes tabulados no se pueden aplicar.

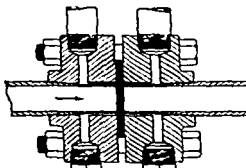


Fig. 2.12. Localización de los puntos de medición en "Corner Taps".

3. Vena Contracta y "Radius Taps": en Vena Contracta están colocados un diámetro corriente arriba y en el punto de menor presión corriente abajo (plano de vena contracta). Es el que comúnmente se utiliza.
4. En "radius taps" se localizan 1 diámetro del tubo corriente arriba y 1/2 diámetro corriente abajo. Se aproxima mucho a vena contracta hasta una razón d/D de 0.72. La diferencia en el uso de estos dos tipos radica en la facilidad para obtener la posición del punto de medición. Dado que muestran los mismos resultados para relación de diámetros hasta de 0.72, prácticamente es más fácil ubicar el punto de medición a medio diámetro corriente abajo que calcular y obtener los puntos de medición en vena contracta. Esto, claro, si la precisión y exactitud no son de vital importancia. (Fig. 2.12).

5. "Pipe taps": localizados a $2\frac{1}{2}$ diámetros corriente arriba y 8 diámetros corriente abajo. Esta localización, como se explicó anteriormente, no es la más adecuada, pues la presión diferencial medida involucra también las pérdidas por fricción. (Fig. 2.14).

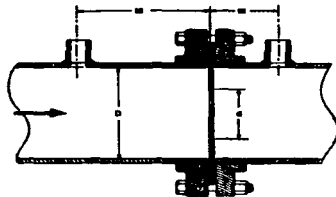


Fig. 2.13. Puntos de medición en Vena Contracta.

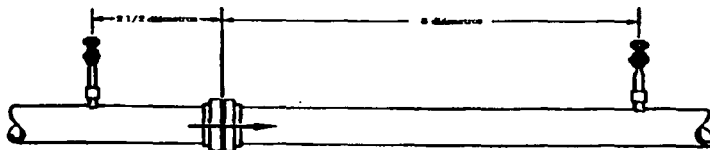


Fig. 2.14. Puntos de medición en "Pipe taps".

La localización elegida para este proyecto es la de tipo Vena Contracta. La razón para seleccionarlo es que con el método de vena contracta se obtiene la mejor exactitud y su localización es fácil de determinar y construir. Además se tienen los coeficientes de descarga para este método (proporcionados por ASME).

2.3 Modelo mecánico. Determinación de parámetros de diseño.

Los parámetros de diseño están determinados en gran parte por los elementos disponibles para la fabricación: tipos de tubo, tipos de transductores, extractores, etc. A continuación se describe el procedimiento para determinar cada una de las partes que componen el medidor.

2.3.1 Dimensiones físicas del medidor.

Para que el método de orificio pueda utilizarse, como se apuntó anteriormente, debe procurarse que en el flujo predominen los efectos turbulentos en la región de medición. El criterio para asegurar un flujo turbulento es con el número de Reynolds y establece como valor frontera a 2300. Este valor de Reynolds fija el flujo mínimo de este aparato medidor; es decir, suponiendo un $N_{Re}=2300$ como valor mínimo para que el método de orificio sea válido se puede determinar la velocidad mínima del fluido y el gasto que le corresponde:

$$N_{Re} = \frac{\rho v D}{\mu}; \quad v = \frac{N_{Re} \mu}{\rho D}.$$

donde D es el diámetro del tubo. El primer paso es la selección de un tubo. Se investigó en el mercado algún tipo de tubo que cumpliera con los siguientes requisitos:

- 1.- Diámetro de 70 mm.
- 2.- Superficie interior lo mejor pulida posible.
- 3.- Lo más fácil posible de cortar y perforar.

Según se consultó la lista de tubos estandarizados por ASME en los Estados Unidos, existe un tubo con un diámetro de 70 mm y construcción en acero. Sin embargo, en México no se encontró. Lo más cercano a 70 mm. es un tubo tipo PVC sanitario de diámetro exterior de 75 mm. e interior de 71.5 mm. Aunque no cumple con el diámetro exacto⁸, se tienen las otras dos características: un pulido bueno y es muy fácil de cortar y manipular.

ASME propone como localización de los puntos de medición para el método de orificio en vena contracta la siguiente gráfica:

⁸ Además, dado que se entregará la permeabilidad como gasto por unidad de área, la sección del tubo que establece la norma no es un parámetro fijo; aunque, desde luego, se trata de aproximarse a la norma lo más posible.

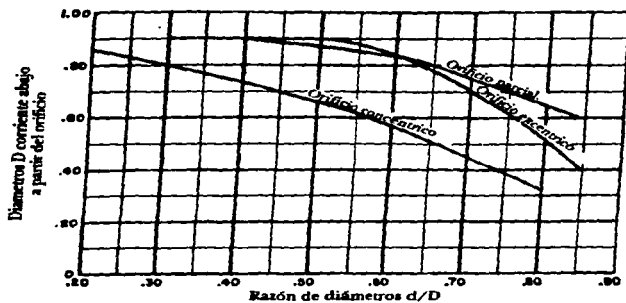


Fig. 2.15. Localización de puntos de medición.

De acuerdo con esta gráfica, la localización de los transductores es:

Corriente arriba: 1 diámetro, 71.5 mm.

Corriente abajo: 0.833 diámetros, 59.55 mm.

La localización del extractor depende mucho de las características de éste. Si entrega un flujo estable puede localizarse, corriente abajo, a distancias menores de 5 veces el diámetro. Sin embargo, como se utilizará un extractor comercial sin un control de calidad óptimo, se prefirió colocarlo a 5 diámetros corriente abajo. La distribución de las presiones en el método de orificio, mostradas anteriormente, sugieren que a 8 diámetros la caída de presión está estabilizada; sin embargo, se presentan pérdidas por fricción. Es por esto que se decidió colocarlo a 5 diámetros del orificio, corriente abajo.

De esta manera, quedan definidas las dimensiones del aparato medidor y se muestran en la siguiente hoja, fig. 2.16.

2.3.2 Orificio.

Los valores ASME tabulados son para tubos específicos. En el apéndice B se muestran los valores para tubos de 2 y 3 pulgadas. El tubo sugerido tiene un diámetro interno de 71.5 mm. que son 2.81 pulgadas. Para un tubo de 3 pulgadas, ASME sugiere una relación de diámetros para valores de Reynolds entre 2,000 y 3,000 de 0.2 a 0.25 ⁽⁹⁾. Utilizando interpolaciones exponenciales o lineales se puede determinar el valor de orificio que resulta de un tubo de 2.81 pulgadas de diámetro y un Reynolds de 2300. Sin embargo, resultaría un poco idóneo hacer este orificio con la medida exacta, pues las herramientas de trabajo no tienen la precisión suficiente. Para una razón de diámetros de 0.2 (y que estaría dentro del rango sugerido por ASME) se tiene un diámetro del orificio de 14.3 mm. (0.562 pulgadas). Una broca comercial cercana a este valor es la de 9/16" (0.5625").

Con estas consideraciones se puede definir el diámetro del orificio: 9/16" o 14.3 mm. que es una relación de diámetros de 0.2.

En cuanto al espesor de la placa con orificio, se debe estar en el rango: $0.02D < 0.005 D^{10}$, donde D es el diámetro del tubo.

2.3.3 Parámetros de operación.

Las condiciones de entrada antes del orificio son: el diámetro es de 71.5 mm, la viscosidad del aire oscila entre 0.00018 [Poises] y 0.0002 [Poises] para intervalos de temperatura de entre -10 [°C] y 50 [°C]. Se utiliza comúnmente el valor de 0.0002 [Poises] para una temperatura ambiente de 25 [°C], y la densidad del aire es de aproximadamente 1.29 [g/l] a una temperatura ambiente de 25 [°C]. Dados estos parámetros se puede determinar la velocidad mínima del aire a la entrada del dispositivo (el valor de Reynolds mínimo debe ser de 2300):

$$N_{Re} = \frac{\rho v d}{\mu}; \quad v = \frac{N_{Re} \mu}{\rho d} = \frac{2300 * 0.0002 [Nm / s]}{1.29 [kg / m^3] * 0.0715 [m]} = 0.4987 [m / s].$$

Es decir, la velocidad mínima a la que debe entrar el aire para asegurar lecturas confiables con el método de orificio es de 49.87 [cm/s]. Para esta velocidad el gasto que le corresponde es de:

⁹ Puede observarse en las tablas que existe una línea que separa los valores en dos grupos. Los valores por encima de esta línea son valores determinados experimentalmente y que aseguran cierta exactitud, según los criterios ASME. Los valores por debajo de esta línea son extrapolaciones, no determinados experimentalmente, y que duplican el margen de error de este procedimiento.

¹⁰ ISO/DIS 5167-1, "Measurement of fluid flow by means of orifice plates, nozzles and Venturi tubes inserted in circular cross-section conduits running full", (1989).

$$Q = vA$$

$$Q = 0.4987[\text{m} / \text{s}] \cdot \left(\frac{0.0715}{2}\right)^2 \pi = 0.002[\text{m}^3 / \text{s}]$$

Esto significa que la permeabilidad mínima a medir está dada por:

$$K = \frac{Q}{A} = \frac{0.002[\text{m}^3 / \text{s}]}{0.004015[\text{m}^2]} = 0.4987 \left[\frac{\text{m}^3 / \text{s}}{\text{m}^2} \right].$$

Para determinar la presión que le corresponde a este valor mínimo de gasto se debe determinar el coeficiente de flujo, para ello se consultan las tabulaciones de ASME y se interpola¹¹ para un diámetro de 2.81" y el valor de Reynolds de 2300. De ahora en adelante se trabaja con la relación de diámetros $\beta = d/D$, donde d es el diámetro del orificio y D es el diámetro del tubo.

$\beta \backslash \text{Re}$	2,000	2,300	3,000
0.2	0.6103	0.6094	0.6074

Coficiente de flujo para tubo de 2".

$\beta \backslash \text{Re}$	2,000	2,300	3,000
0.2	0.6096	0.6087	0.6067

Coficiente de flujo para tubo de 3".

De estos dos valores de coeficiente de flujo se determina el coeficiente para el tubo de 2.81":

$\text{Re} \backslash d$	2"	2.81"	3"
2300	0.6094	0.6088	0.6087

La caída de presión en la placa de orificio será entonces de (referirse a la figura 2.16):

$$P_2 - P_3 = \frac{\rho}{2} \left(\frac{Q}{K_f \cdot A_2} \right)^2 = \frac{1.29 [\text{kg} / \text{m}^3]}{2} \left[\frac{0.002 [\text{m}^3 / \text{s}]}{0.6088 \cdot \left(\frac{0.0143[\text{m}]}{2} \right)^2 \cdot \pi} \right]^2$$

¹¹ De aquí en adelante se utilizan interpolaciones lineales. Anteriormente se mostró el comportamiento del coeficiente de descarga para el método de orificio; podrá verse que no son lineales. Para este tipo de cálculos basta con hacer interpolaciones lineales, pues los valores son muy aproximados. Las variaciones son del orden de varias centenas de milésimas, menos del 1%.

$$P_2 - P_3 = 269.86 \text{ [Pa]}.$$

Este es el valor de presión mínimo a medir. Es decir, que la tela más cerrada que se medirá con el medidor de permeabilidad dará una diferencia de presiones de 269.86 [Pa]. Para determinar cuál será la diferencia de presión máxima a medir por el aparato (límite superior de medición) hay que analizar tres puntos: los valores que normalmente se manejan en este tipo de pruebas, un transductor que no sea de una resolución mayor al 1% de la escala total y finalmente la resolución del convertidor A/D.

Analizando la norma (apéndice A), se puede observar que para una relación de diámetros de 0.2, y de hecho, para casi cualquier relación de diámetros, se tiene un valor máximo del doble del valor mínimo. Llevando esto al medidor, se podría suponer un valor del doble y esto nos daría un rango de valores de:

$$49.87 < K < 99.74 \text{ [cm}^3\text{/s/cm}^2\text{]}.$$

Se puede deducir otro límite superior a partir del siguiente razonamiento: se desea medir un mínimo de 269.86 [Pa] con 1% de resolución, esto significa que la resolución es de 2.69 [Pa]. Para los 255 niveles de conversión del HC11, se tendrían 2.69 [Pa] para cada nivel de conversión, lo que da un intervalo total de 0 a 685.95 unidades de medición. Es decir, que bajo este criterio, el rango de medición del aparato sería de 0 a 685.95 [Pa] con una resolución del 1% de la escala total: esto tiene un equivalente en permeabilidad de 0 a:

$$K = \frac{Q}{A_1} = \frac{K_f \cdot A_2}{A_1} \sqrt{\frac{2(P_1 - P_2)}{\rho}} = \frac{0.6 \cdot 0.00016 \text{ [m}^2\text{]}}{0.00405 \text{ [m}^2\text{]}} \sqrt{\frac{2 \cdot 685.95 \text{ [Pa]}}{1.29 \text{ [kg / m}^3\text{]}}} = 0.773 \text{ [m}^3 \text{ / s]}$$

Lo cual no es del todo cierto, pues se está tomando un error del 1% sobre una lectura intermedia y no realmente a plena escala.

Sin embargo, la máquina medidora de la norma tiene una escala de 0 a 16 pulgadas, la cual presenta una resolución de 5 divisiones por pulgada (0.2 de pulgada). Es decir, para una lectura dada se tienen 13 pulgadas (pues la norma pide que se rebasen 3 pulgadas para tener precisión) con cinco divisiones por cada pulgada, lo cual da un total de 65 divisiones. Esto sugiere, que para lecturas sobre una permeabilidad máxima de un orificio dado, se tiene una resolución de la [permeabilidad máxima]/65 o 1.5% de la escala total. El convertidor A/D del HC11 permite 256 niveles de conversión, es decir, casi 4 veces.

Como propuesta para este medidor se sugiere un intervalo de medición de cero a tres veces la permeabilidad mínima, con una resolución del 0.3% de la escala total. Los transductores normalmente ofrecen resoluciones del 0.1% al 1% de la escala total.

Para determinar la presión máxima a medir en la placa de orificio se debe determinar primero el coeficiente de flujo máximo: para una permeabilidad de 1.4961 [m³/s/m²], se tiene un Reynolds de:

$$N_{Re} = \frac{\rho v d}{\mu} = \frac{1.29 \text{ [kg / m}^3\text{]} \cdot 1.4961 \text{ [m / s]} \cdot 0.0715 \text{ [m]}}{0.00002 \text{ [N} \cdot \text{m / s]}} = 6899.63.$$

$\beta \sqrt{N_{Re}}$	6,000	6,899.63	8,000
0.2	0.6035	0.6029	0.6022

Coefficientes de flujo para tubo de 2".

$\beta \sqrt{N_{Re}}$	6,000	6,899.63	8,000
0.2	0.6028	0.6022	0.6016

Coefficientes de flujo para tubo de 3".

El coeficiente de descarga para un tubo de 2.81" (71.5mm) será entonces de:

$N_{Re} D$	2"	2.81"	3"
6,899.63	0.6029	0.6023	0.6022

Coefficiente de flujo para tubo de 2.81".

La diferencia de presiones máxima en la placa de orificio está dada por:

$$P_2 - P_3 = \frac{\rho}{2} \left(\frac{Q}{K_f \cdot A_2} \right)^2 = \frac{1.29 \text{ [kg / m}^3\text{]}}{2} \left(\frac{0.006 \text{ [m}^3\text{ / s]}}{0.6022 \cdot \left(\frac{0.0143 \text{ [m]}}{2} \right)^2 \cdot \pi} \right)^2.$$

$$P_2 - P_3 = 2482.31 \text{ [Pa]}$$

Los intervalos de funcionamiento del medidor quedan establecidos en la siguiente tabla¹²:

Permeabilidad K. [m ³ /s/m ²]	Coefficiente de flujo K _f	N _{Re}	Gasto [m ³ /s] Q.	P2-P3 [Pa]. Placa de orificio.	P1-P2 [Pa]. Textil.
0.4987	0.6094	2300	0.002	269.86	124.42
1.4961	0.6022	6899.63	0.006	2482.31	124.42

¹² Las presiones se refieren a la figura 2.16.

Estos límites han sido determinados de manera teórica, de acuerdo a los criterios antes explicados. Pueden sufrir alteraciones debido a la disponibilidad de transductores de presión en el mercado y los elementos de construcción del aparato (tubo, extractor de aire, etc.). Estos valores, pues, son la primera aproximación al intervalo de operación del aparato medidor. Posteriormente se discuten los elementos seleccionados y las modificaciones a este intervalo.

Los valores de presión en sus equivalentes en otras unidades de presión son:

$\Delta P=(P_3-P_2)$ [Pa]	psi	inH ₂ O	bar
269.86	0.03913	1.0844	0.00269
2482.42	0.36004	9.9759	0.2482

Aclarando que para lecturas menores a $K=49.87$ [$\text{cm}^3/\text{s}/\text{cm}^2$] no se tiene confiabilidad, pues no satisface el criterio de Reynolds.

2.3.4 Extractor de aire.

Para seleccionar el extractor de aire es necesario determinar la potencia máxima requerida para desplazar el máximo gasto. La potencia necesaria está dada por la expresión:

$$W = \text{Caída total de presión} \cdot \text{Gasto}$$

La potencia máxima necesaria será para el caso de un textil muy abierto, en el que se presentará el máximo gasto y la máxima caída de presión en el orificio además de la caída de presión en la tela. El problema ahora se centra en el cálculo de la caída de presión a través del método de orificio. Anteriormente se expusieron las gráficas del comportamiento de la presión a lo largo del tubo utilizando el método de orificio. ASME ha calculado la caída total de la presión cuando se utilizan distintos orificios. La pérdida de presión para los distintos métodos está en función del tamaño de la obstrucción, es decir, de la relación de diámetros. La siguiente gráfica muestra la caída total para los distintos métodos:

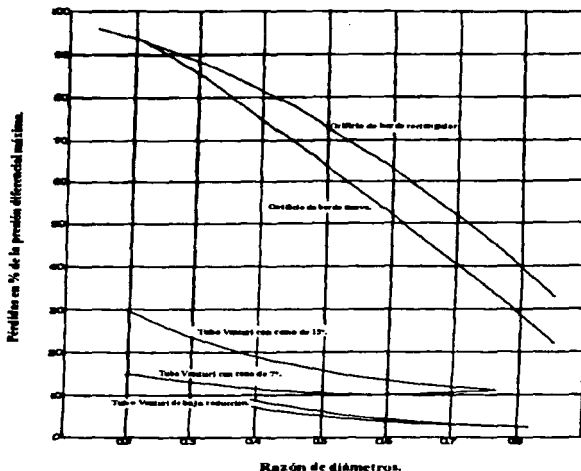


Fig. 2.17 Pérdidas totales de presión para los distintos métodos.

Para una relación de diámetros de 0.2 y para el método de orificio se tiene una caída total de presión de aproximadamente del 94% de la máxima diferencia de presiones medida (que se da a un diámetro corriente arriba y el plano de vena contracta corriente abajo). El método de medición utilizado es precisamente el de vena contracta, que da la diferencia máxima de presiones. De esta forma, la potencia necesaria está dada por la siguiente ecuación:

$$W = (0.006 \text{ fm}^3/\text{s})(124.42 [\text{Pa}] + 0.94 \cdot 2482.31 [\text{Pa}]) = 14.74 [W].$$

Esta es la potencia máxima necesaria. Esto no significa que el motor está determinado por esta potencia. En un ventilador o extractor de aire existe cierta eficiencia. La caracterización matemática de un ventilador o extractor centrífugo es extremadamente difícil¹³, más aún para un ventilador comercial cuya construcción no es del todo controlada. La asesoría consultada¹⁴ sugiere suponer una eficiencia del

¹³ Puede encontrarse una breve descripción de bombas centrífugas en: Robert H. Perry, "Perry's Chemical Engineer's Handbook", Mc. Graw-Hill 1989. Donde se definen los términos y propiedades comúnmente utilizados en bombas centrífugas comerciales o industriales.

¹⁴ Dr. Federico Méndez Lavielle, F. I., consulta oral.

ventilador de aproximadamente 50 % (que es, de hecho, baja para los ventiladores comunes) además de la eficiencia del motor. Se ha sugerido utilizar una eficiencia del 60%. Bajo estas consideraciones se requiere de un ventilador cuyo motor sea de una potencia de aproximadamente 24.5 [W]. La potencia más cercana, en valores cerrados, sería de 30 [W].

Además de la potencia del motor, el ventilador o extractor debe ser capaz de generar un gasto de 0.006 [m³/s] (12.72 [cfm]) a una presión de 2606.73 [Pa]. Las características del ventilador son entonces:

Potencia nominal: 30 [W].

Gasto: 6 [cm³/s] = 12.72 [cfm].

Presión: 2457.79 [Pa] = 0.378 [psi] = 9.37 [inH₂O].

2.4 Observaciones sobre la selección de elementos del sistema mecánico.

Como se mencionó al inicio de este trabajo, el proyecto no tiene presupuesto propio. Por esta razón, se ha buscado en lo mayor posible, el uso de materiales y dispositivos lo más económico y accesible posible. Estas observaciones surgen a partir de este criterio y de la confrontación entre las exigencias de la teoría y lo que los fabricantes ofrecen en el mercado.

El tipo de tubo más adecuado puede ser de PVC sanitario de 75 [mm] de diámetro, cobre o aluminio (ambos en medida estándar), pues son fácilmente manipulables. La fijación de la placa de orificio y del sujetador de la tela debe ser lo más hermética posible. Para el caso de la placa conviene que quede perfectamente sellada e inmóvil. Para la sujeción de la muestra de tela debe procurarse que el sujetador la mantenga lo más presionada posible o hacer el camino de fuga lo más tortuoso posible, para evitar la fuga de aire. La construcción requiere trabajo mecánico en torno, fresadoras, taladros, etc.; elementos que pueden ser fácilmente encontrados en un taller mecánico.

Se puede observar que el orden de la presión diferencial en la tela es muy baja. La diferencia de presión que se debe alcanzar entre las dos caras del textil es de media pulgada de agua. Si se supone que esta media pulgada de agua se mide a un 75% de la escala total del transductor, resulta que se requiere un transductor de 0 a 0.66 pulgadas de agua. Se consultaron varios fabricantes de transductores y el valor más próximo es de cero a una pulgada de agua. Entonces, la lectura de la presión diferencial de la tela estaría en el 50% de la escala total, que es aceptable.

La diferencia de presiones que resulta de la placa de orificio está en el intervalo de 1 pulgada de agua a 10 pulgadas aproximadamente. El transductor que se aproxima más a este intervalo es, precisamente, uno de 0 a 10 pulgadas de agua. Existen varios fabricantes que tienen estos valores o muy cercanos a ellos, lo que permitiría la selección del más económico. Todos los transductores tienen resoluciones menores al 1% de la escala total.

Lo que rige la selección de transductores es el costo de los mismos, por las razones que se explicaron anteriormente. El transductor que presenta inconvenientes económicos es el de una pulgada de agua, pues sólo se encontraron dos fabricantes con representación en México.

Para el extractor de aire se requiere que sea capaz de entregar una presión estática de 10 pulgadas de agua con un gasto de 12.72 [cfm]. Se consultaron varias casas especializadas en fabricación y distribución de ventiladores y extractores y no se encontró ninguno con estas características. Los extractores con mayor caída de presión están en el orden de 2.5 pulgadas de agua. Todos los extractores vistos tienen motores de c. d. o jaula de ardilla. Para lograr la presión de 10 pulgadas se debe utilizar un extractor con un motor universal, pues con éste se pueden alcanzar las revoluciones necesarias para tal presión.

En la selección de los elementos (tubos, mangueras, etc) que conforman el aparato medidor se debe tener cuidado de no utilizar materiales muy flexibles o elásticos, pues éstos se comportarían como almacenadores de presión. El impacto que esto tiene en el sistema es que al comportarse como elementos almacenadores de presión, éstos son almacenadores de energía, lo que puede generar oscilaciones o inestabilidad.

2.5 Elementos seleccionados.

Para la construcción del medidor de permeabilidad se utilizó tubo tipo PVC sanitario de 75 mm. de diámetro exterior, 71.3 mm. de diámetro interior con acabado liso. La construcción de la placa de orificio es con lámina de acero de calibre de 1 [mm], su sujeción es con bridas hidráulicas de PVC con empaques de hule. La fijación de la muestra de tela también con bridas hidráulicas de PVC con empaque de poliestireno.

El extractor utilizado es un motor con turbina de aspiradora. El motor es de 1.2 caballos tipo universal, de 16,000 [r.p.m.], 120 [V] a. c.. No se tiene más información del motor por ser de segunda mano y reconstruido (el criterio de selección fue el costo). Los transductores de presión a voltaje son marca MAMAC, uno de ellos es de 0 a 1 pulgadas de agua y el otro de 0 a 15 pulgadas de agua. Ambos están compensados contra variaciones de temperatura y calibrados. La salida de los transductores es en corriente, 4 [mA] para 0 pulgadas y 20 [mA] para el rango más alto. Se incluye la información técnica de los transductores en los apéndices.

Dado que no se tiene la información técnica del motor universal, se hicieron pruebas para determinar sus características eléctricas. Esto fue necesario, pues la selección de la electrónica de control para el motor requiere de esta información para su diseño. En el siguiente capítulo se habla del motor universal y la circuitería utilizada para su control.

Las tomas de presión son con derivadores de bronce para instalaciones hidráulicas tipo cónicas de 1/4" de diámetro. Las fueron colocadas en las bridas y tubo de PVC con cuerda, utilizando un machuelo tipo cónico. Se fijaron, al igual que las bridas al tubo, con pegamento especial para PVC.

La sujeción del motor también fue usando una brida de PVC con empaque de hule.

Las mangueras para transmitir la presión son de plástico rígido.

2.6 Nuevo cálculo del intervalo de operación.

Debido a que el transductor de presión que se utiliza en la placa de orificio puede medir hasta 15 pulgadas de agua, el intervalo de operación del aparato medidor puede ser ajustado a este nuevo valor de presión máxima. El gasto que corresponde a una presión en la placa de orificio es:

$$Q = K_f A_2 \sqrt{\frac{2(\Delta P)}{\rho}}$$

$$Q = 0.6022 \cdot 0.00016 [m^2] \sqrt{\frac{2 \cdot 15 [inH_2O] \cdot 248.84 [Pa / inH_2O]}{1.29 [kg / m^3]}} = 0.0073 [m^3 / s]$$

La velocidad (del mismo valor que la permeabilidad) será:

$$v = Q / A = 0.0073 [m^3 / s] / 0.00405 [m^2] = 1.8181 [m / s]$$

$$N_{Re} = \frac{\rho v d}{\mu} = \frac{1.29 [kg / m^3] \cdot 1.8181 [m / s] \cdot 0.0715 [m]}{0.00002 [N \cdot m / s]} = 8384.62$$

De esta forma, el nuevo intervalo de operación del aparato medidor es:

	Permeabilidad K [m ³ /s/m ²]	Coefficiente de flujo K _F	N _{Re}	Gasto [m ³ /s] Q.	P2-P3 [Pa]. Placa de orificio.	P1-P2 [Pa]. Textil.
Límite inferior.	0.4987	0.6094	2300	0.002	269.86	124.42
Límite superior teórico.	1.4961	0.6022	6899.63	0.006	2482.31	124.42
Nuevo límite.	1.8181	0.6020	8384.62	0.0073	3732.60	124.42

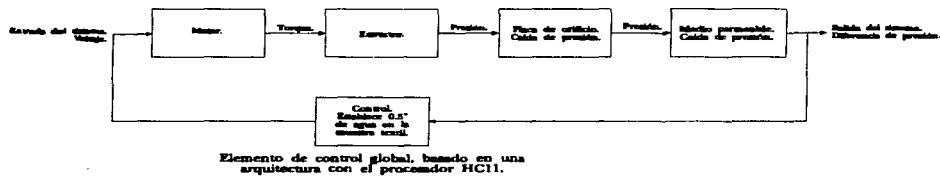
El nuevo intervalo de operación es: de 0.4987 [m³/s/m²] a 1.8181 [m³/s/m²].

En otras unidades de presión:

ΔP=(P3-P2) [Pa]	psi	inH2O	bar
269.86	0.03913	1.0844	0.00269
3732.6	0.54136	15	0.037326

Capítulo 3. El extractor de aire. Control del motor. Transductores de presión.

Este capítulo se divide en dos secciones. La primera de ellas habla del motor del extractor de aire y su controlador; la segunda habla de los transductores de presión y de la electrónica asociada a estos medidores. La siguiente gráfica ayuda a ubicar el motor, el controlador y los medidores de presión dentro del aparato medidor de permeabilidad (visto como subsistemas):



Elemento de control global, basado en una arquitectura con el procesador HC11.

Fig. 3.1 El sistema medidor.

Este capítulo describe la construcción del controlador electrónico del motor y la circuitería para el acondicionamiento de las señales de los medidores de presión.

El controlador está diseñado para manejar al motor universal en función de una señal de corriente directa (DC) que es proporcionada por el elemento de control global, que es el que se encarga de medir la diferencia de presiones entre ambas caras del textil; esta presión es la establecida por la norma. En este capítulo se hace referencia a una arquitectura HC11, es esta arquitectura la encargada de establecer el control en lazo cerrado del sistema medidor de permeabilidad, es decir, proporciona la señal de control a la electrónica de control del motor universal.

El acondicionamiento de la salida de los transductores se debe a que éstos entregan su salida como una variación lineal de la corriente respecto de la presión. El rango de voltajes que convierte el microcontrolador HC11 es de 0 [V] a 5 [V]. Es por ello que la corriente que entrega el transductor debe ser convertida en una señal de 0 [V] a 5 [V]. En la segunda parte de este capítulo se describe el circuito acondicionador.

Parte I. El extractor de aire y la electrónica de control.

Como se describió en el capítulo anterior, se requiere de un motor tipo universal para alcanzar la diferencia de presión del aparato medidor. En este capítulo se describe el motor universal, las pruebas realizadas y la circuitería utilizada para el control del motor. Para establecer el tipo de circuito para el control del motor del extractor, conviene hacer una caracterización previa. Indudablemente, los circuitos comerciales para el control de motores universales proporcionan un mecanismo eficaz de control; sin embargo, conviene hacer algunas pruebas para determinar si no se presentarán resultados inesperados y así satisfacer los parámetros de operación del circuito.

3.1 El motor universal.

Los motores pueden clasificarse en tres grupos principalmente, atendiendo al tipo de alimentación que se utiliza: motores de corriente alterna (AC), motores de corriente directa (DC) y motores de pasos. Sin embargo, existe un tipo de motor, el universal, que permite conectarse, sin hacer ninguna modificación, indistintamente a AC ó DC.

El motor universal es un tipo de motor que permite alcanzar altas revoluciones. Su construcción es casi idéntica a la de un motor de corriente directa con campo serie, de hecho, un motor de corriente directa con campo serie puede ser utilizado con corriente alterna.

3.1.1 Principio de operación.

Un motor de DC, cuyo campo no sea de imán permanente, puede conectarse a una fuente de AC debido a que la dirección del torque depende de la polaridad del campo y la polaridad de la armadura, dado que en AC se alternan ambos, el torque conserva su giro. La conexión de un motor de DC en serie define lo que se llama motor universal:

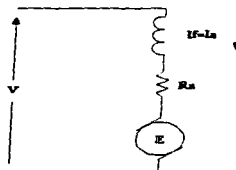


Fig. 3.2 El motor universal.

A diferencia de un motor síncrono o de un motor con campo en derivación, el motor universal o serie cambia su velocidad considerablemente con la carga. Para cargas fuertes la velocidad es baja, para cargas ligeras la velocidad es alta.

La característica de alta velocidad se debe a que mientras el motor tiende a subir sus revoluciones, la fuerza electromotriz inducida disminuirá la corriente demandada, y la velocidad en este motor es función directamente proporcional al voltaje aplicado e inversamente proporcional a la corriente que circula por la armadura. Dado que la f.e.m. inducida tiende a disminuir la corriente demandada, entonces la función de velocidad tiende a incrementarse de manera hiperbólica.

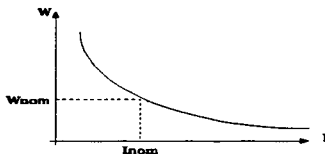


Fig. 3.3 Velocidad vs. corriente.

Teóricamente la velocidad tendería a infinito, sin embargo las fuerzas de fricción en los rodamientos y con el aire impiden que se acelere infinitamente. De hecho, esto es un peligro para los motores industriales de DC que se conectan en serie. A este fenómeno se le llama desbocamiento. Los motores universales están contruidos para soportar las fuerzas centrífugas producidas por las altas revoluciones.

Al comparar pruebas de velocidad contra carga, entre alimentación AC y DC en un motor, se puede ver que no describen la misma curva hiperbólica, pero se cruzan en un mismo punto. Normalmente en AC alcanzan velocidades más grandes en vacío que en DC; para una misma carga, en DC tienen una velocidad más alta que en AC; pero, se puede presentar lo contrario, es decir, que en DC alcance mayores revoluciones que en AC al estar en vacío y viceversa al estar cargado. Esto se debe fundamentalmente a los efectos de la reactancia de la armadura y del campo en AC y a los efectos resistivos en DC. En DC, con carga, los efectos resistivos tienden a disminuir la velocidad y la reacción de armadura tiende a subir la velocidad; de modo que, dependiendo cuál efecto es el que predomine cuando se construye un motor, se logrará mayor o menor velocidad al estar cargado. En AC se presenta un tercer factor, que es la caída de voltaje por reactancia que tiende a disminuir la velocidad cuando se carga. El efecto de la reacción de armadura es el de disminuir el flujo magnético en el entrehierro. Cuando se utiliza AC la caída de voltaje por efectos reactivos disminuye el voltaje efectivo en el motor, consecuentemente disminuye la velocidad del motor; si el voltaje reactivo se incrementa más que lo que el flujo magnético disminuye, entonces el motor tiende a una menor velocidad cuando se utiliza AC, si esto sucede al revés, entonces lo contrario es

verdadero. Cuando se carga demasiado el motor, el flujo en el entrehierro crece y el metal se satura, entonces el flujo magnético efectivo por amper es menor que el producido por una corriente DC. El resultado de esto es que el motor girará más rápido cuando se utiliza AC. Con todo esto, si el voltaje reactivo no actuara en contra de la reacción de armadura, la velocidad en AC sería siempre mayor que en DC, bajo carga. Aquel factor que predomine es el que determinará si el motor va más rápido en DC que en AC, bajo carga. Normalmente al funcionar en vacío, la caída por voltaje reactivo es despreciable, lo que hace al motor girar más rápido en corriente alterna que en directa.

Estos factores pueden combinarse para lograr mayor revoluciones en una u otra alimentación. El motor utilizado en este trabajo, dado que es de una aspiradora, está diseñado para trabajar en AC bajo cargas ligeras (aire), por lo que se espera lograr mayor velocidad en DC que en AC.

3.1.2 Características del motor universal.

El motor universal tiene las características del motor DC con campo serie:

- 1.- Altas revoluciones.
- 2.- Se puede controlar su velocidad.
- 3.- Par de arranque muy alto.
- 4.- Velocidad inestable.

Dado que el embobinado del campo está en serie con la armadura, el campo magnético producido por el embobinado del campo debe provocar la menor caída posible para que el rotor gire más rápidamente. Es por ello que el campo tiene un conductor mucho más grueso que el del rotor y con un número menor de vueltas (resistencia baja). De esta manera se obtiene un campo fuerte debido fundamentalmente a la cantidad de corriente y no al número de vueltas (lo contrario sucede en un motor con el campo paralelo).

La velocidad del motor universal es muy sensible a cambios en la carga, es decir, que para un voltaje constante en la entrada no se mantiene la misma velocidad para variaciones despreciables de carga. Esto no sucede en el motor con campo paralelo, donde la velocidad tiende a mantenerse aunque se varíe la carga considerablemente.

En los siguientes apartados se explican las pruebas realizadas al motor y sus resultados. Se aplicaron dos pruebas básicamente: con regulador lineal y con regulador conmutado. El objetivo de estas dos pruebas es, primero, obtener los datos de placa del motor universal y segundo analizar el comportamiento inductivo del motor con el tipo de control que se pretende utilizar, que es conmutado (a base de tiristores).

3.2 Pruebas con regulador lineal aplicadas al motor universal.

En la prueba realizada con regulador lineal se analizaron dos casos: uno con la entrada de aire obstruida (en vacío) y el otro con la entrada abierta (con carga). Se hizo la siguiente conexión para medición de corriente, voltaje y ángulo de fase:

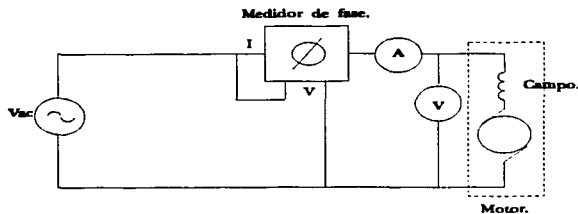


Fig. 3.4 Diagrama de conexiones para las pruebas.

De acuerdo al diagrama, se midió la corriente demandada por el motor para distintos valores de voltaje, al igual que el factor de potencia del motor para estas variaciones de corriente. El voltaje se reguló con el reóstato del regulador.

3.2.1 Pruebas en vacío.

La siguiente gráfica muestra el comportamiento de la corriente cuando se varía el voltaje en incrementos de 10 [V]:

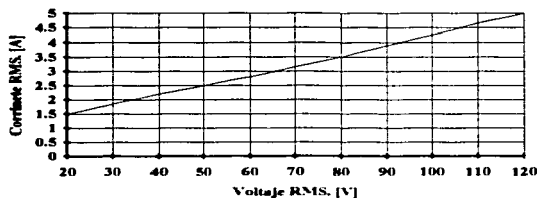


Fig. 3.5 Corriente vs. voltaje.

Es un comportamiento casi lineal de la corriente con respecto del voltaje, es decir, la impedancia del motor no varía.

El comportamiento de la fase es exactamente igual con el motor cargado. (Ver la figura 3.7 en el siguiente apartado).

3.2.2 Pruebas con carga.

La carga consistió en permitir el paso del aire por las turbinas del motor.

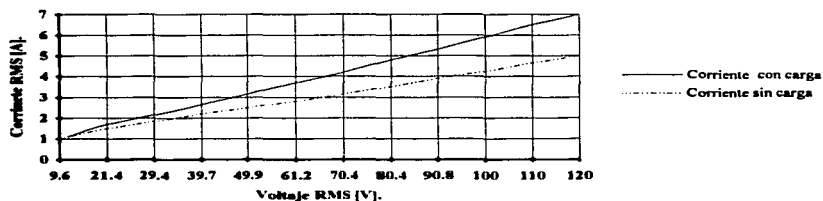


Fig. 3.6 Corriente vs. corriente con carga.

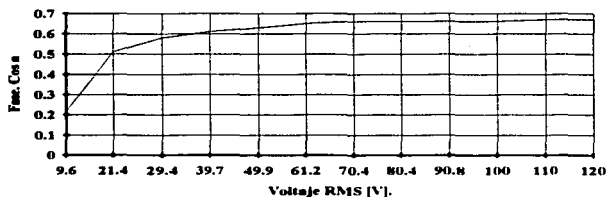


Fig. 3.7 Fase vs. voltaje aplicado.

Para las pruebas que se realizaron, se partió de un voltaje de 10 [V] rms, voltaje al que el motor comenzó a girar, es decir, se venció la inercia. El factor de potencia al empezar a girar es de 0.22 (un elemento muy inductivo), mientras que conforme el motor alcanza su velocidad máxima a pleno voltaje el motor presenta un factor de potencia de 0.67, que aún está fuera del 0.8 que normalmente se exige a un circuito reactivo conectado a la red eléctrica.

3.2.3 Observaciones sobre las pruebas con regulador lineal.

De los datos arrojados en las pruebas se deducen las características del motor:

Voltaje nominal: 120 [V].

Corriente nominal: 5 [A] sin carga, 7[A] con carga.

Potencia aparente: 120 [V] * 7 [A] = 840 [VA].

Potencia real: 840 [VA] * 0.67= 562.8 [W].

Factor de potencia: de 0.2 a 0.67.

Resistencia medida entre las terminales del motor: 5 [Ω].

De aquí se concluye las características del controlador deben ser tales que permitan el paso de 7 [A] de corriente (suponiendo un arranque gradual) y que contemple el defasamiento de la corriente respecto al voltaje. Hay que tener en cuenta que si el arranque es de golpe, la demanda de corriente puede llegar a ser mayor que el doble de la corriente nominal, pues sólo intervienen, en el preciso instante del arranque, los efectos resistivos del motor.

3.3 Pruebas con regulador conmutado.

Estas pruebas consisten en operar al motor mediante un controlador conmutado para observar las formas de onda y fase de corriente.

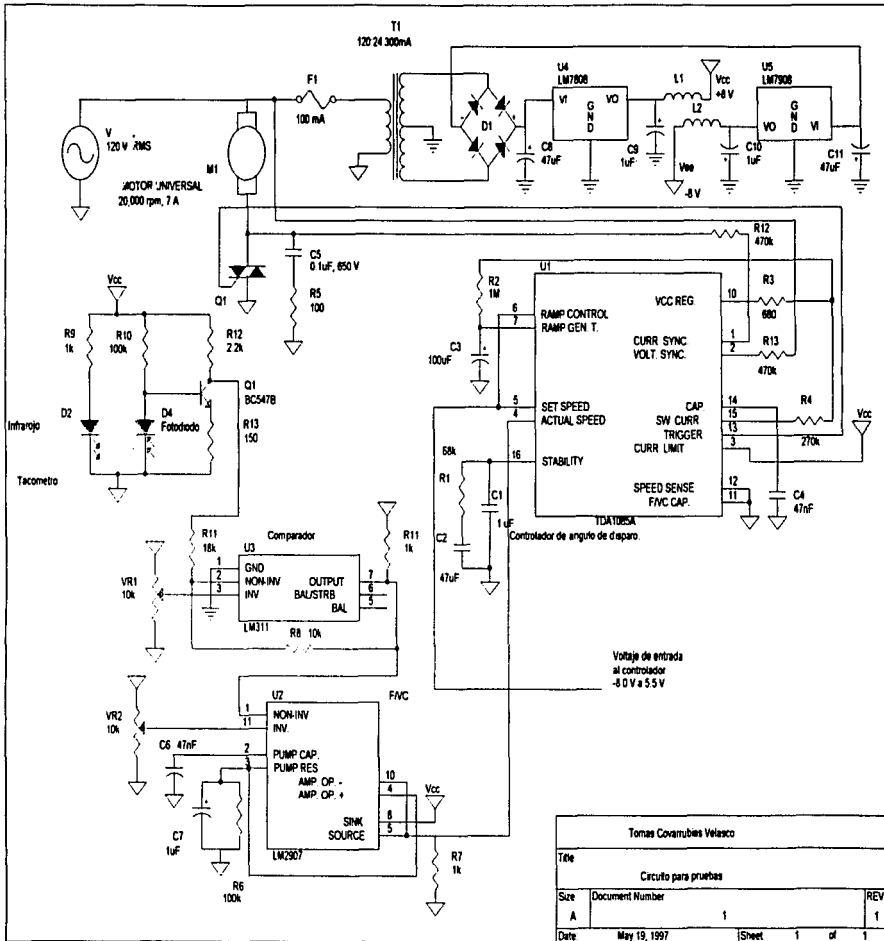
Por controlador conmutado se entiende a un circuito que regula la energía entregada al motor conectando el voltaje al motor por un cierto tiempo (o ángulo de disparo). Normalmente se hace el control cada medio ciclo de la frecuencia de alimentación (60 Hz). La potencia que entrega el controlador no es una función lineal del tiempo de encendido o del ángulo de disparo.

Al revisar las pruebas del motor con el regulador lineal se ve la importancia de realizar pruebas con un circuito conmutado, pues estos controladores operan con tiristores que son componentes electrónicos que cambian de un estado de conducción (cuando se ha disparado) a un estado de alta impedancia cuando la corriente que pasa por ellos es cero, y en un circuito inductivo como el motor la corriente está desfasada respecto al voltaje en las terminales del motor. Si este defasamiento es muy grande, el tiristor se abrirá un tiempo considerable después del cruce por cero de la señal de voltaje. Esto puede representar un problema para el control del motor con controladores conmutados.

3.3.1 Circuito para controlar el motor.

Se consultaron varios fabricantes y Motorola ofrece un circuito comercial diseñado exclusivamente para control en lazo cerrado de velocidad de motores universales. El número de parte es el TDA1085A¹. El circuito utilizado para regular la velocidad se muestra en la siguiente hoja.

¹Motorola, "Linear/Interface IC's Device Data", tomo primero.



Tomas Covarrubias Velasco		
Título		
Circuito para pruebas		
Size	Document Number	REV
A	1	1
Date	May 19, 1997	Sheet 1 of 1

El control de la velocidad es de acuerdo a una señal de voltaje que alimenta al circuito controlador, que varía el ángulo de disparo del tiristor. Las revoluciones se obtienen a partir de un tacómetro digital, cuya señal se convierte en una señal de DC que varía linealmente de acuerdo a la frecuencia del tacómetro digital. Cada una de las partes de este controlador se explican posteriormente con detalle.

La finalidad de este circuito es que opere en lazo cerrado para entregar las revoluciones como una función lineal del voltaje de entrada. Es por esta razón que se hacen las pruebas operando al circuito en lazo cerrado. El circuito TDA1085 puede operar en lazo abierto, pero hay que hacer algunas conexiones entre terminales del mismo circuito que permitan esta operación, pues si se opera en lazo abierto sin ningún tipo de realimentación entre sus partes, el circuito se satura².

Las pruebas consistieron en medir voltaje promedio entregado al motor, ángulos de disparo para el semiciclo positivo y para el semiciclo negativo, ángulos de desfase para el semiciclo positivo y para el semiciclo negativo y finalmente, las revoluciones del motor.

Los ángulos de disparo y fase están definidos de la siguiente forma, la forma de onda continua corresponde a la corriente en el motor:

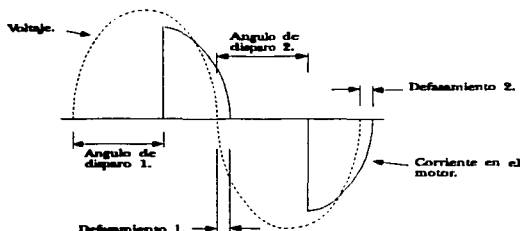


Fig. 3.8 Definición de ángulos de disparo y desfases.

²En las hojas de especificaciones técnicas pueden encontrarse las conexiones necesarias que permiten operar al circuito en lazo abierto.

Las siguientes gráficas muestran las mediciones hechas.

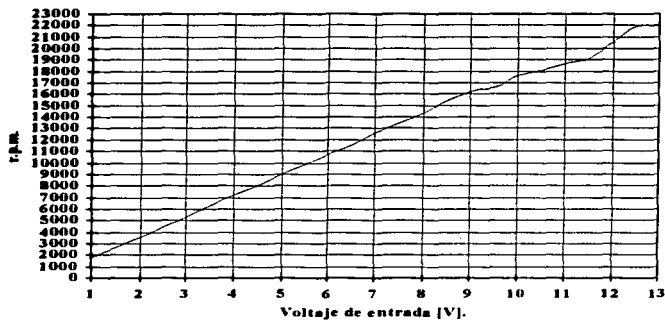


Fig. 3.9 Revoluciones vs. voltaje de entrada.

En esta prueba se confirma el objetivo del controlador en lazo cerrado: la dependencia lineal de las revoluciones respecto al voltaje de entrada.

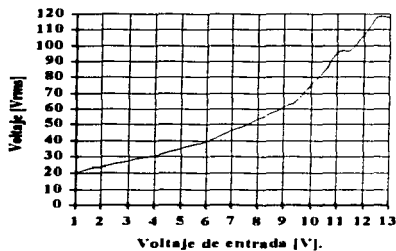


Fig. 3.10 Voltaje promedio vs. voltaje de entrada.

Esta prueba muestra la no linealidad del circuito conmutado. El voltaje promedio entregado al motor no es una función lineal del voltaje entregado al controlador.

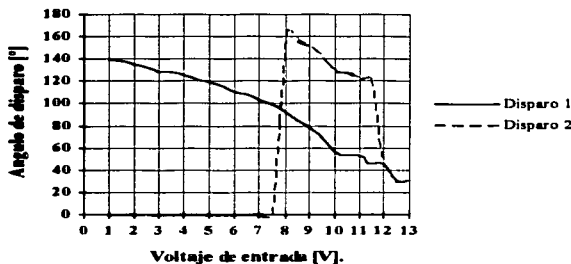


Fig. 3.11 Ángulos de disparo vs. voltaje de entrada.

Un aspecto a destacar del controlador es que no son simétricos sus ángulos de disparo. Desde cero revoluciones hasta aproximadamente 13,000 r.p.m. sólo activa el ángulo de disparo del semiciclo positivo de la señal de AC, a partir de las 13,000 r.p.m. comienza a conducir en el semiciclo negativo. Esto podría ser un problema para el controlador, pues está actuando solamente sobre medio ciclo en bajas revoluciones. Los efectos de este problema son pequeños golpes en el rotor; sin embargo, al hacer las pruebas fue imperceptible este problema y el objetivo, que es controlar la velocidad sobre todo el rango de voltaje en la entrada del controlador, se cumplió.

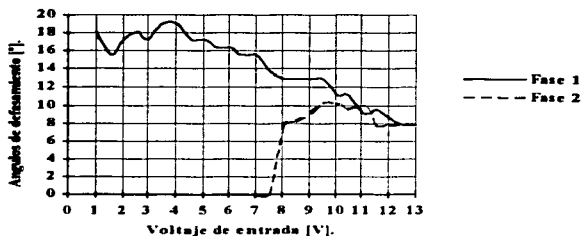


Fig. 3.12 Ángulos de defasamiento vs. voltaje de entrada.

Esta es la prueba que más importancia tiene, pues muestra el comportamiento inductivo del motor sobre el circuito conmutado. Contrario a lo que entregó el análisis con circuito lineal, con un circuito de control conmutado se tienen defasamientos despreciables. El máximo factor de potencia medido es de 0.9 ($\cos 18^\circ$), mientras que el menor, que es a pleno voltaje, es de 0.99 ($\cos 8^\circ$). Los problemas de un factor

de potencia bajo se manifiestan en el control a baja velocidad, pues al haber un desfase grande en la corriente, el tiristor podría apagarse mucho tiempo después de que se hizo el cruce por cero. Es por esta razón que los circuitos deben monitorear la señal de corriente, y de esta forma se tiene información precisa del momento en que se apaga el TRIAC.

3.4 Algunas observaciones y resultados en la prueba del regulador conmutado.

Con el diseño propuesto se hicieron las pruebas para todo el rango de voltaje de entrada. Resulta interesante hacer algunas observaciones.

1. El ángulo de disparo no es simétrico para los dos semiciclos. Al variar las resistencias de monitoreo de voltaje y corriente se obtuvieron variaciones en los ángulos, pero nunca fueron iguales.
2. El desfase de la corriente fue menor que cuando se utilizó un regulador lineal. Este nunca fue significativo, de tal forma que la malla de protección para el TRIAC puede ser eliminada, sin embargo, por razones de protección, se dejó.
3. Se presentaron oscilaciones para revoluciones muy bajas y para revoluciones cercanas a las 20,000 r.p.m. El impacto de este tipo de inestabilidad en el sistema en conjunto se desconoce hasta no realizar las pruebas de control.
4. La potencia disipada por el tiristor fue considerable para ángulos de disparo menores a 90°. Se colocó un disipador para evitar sobretemperatura.

3.5 Descripción del circuito controlador de velocidad en lazo cerrado.

Las razones para utilizar el controlador del motor en lazo cerrado se deben fundamentalmente al modelo de control del sistema global (el sistema medidor de permeabilidad). Al utilizar el esquema de lazo cerrado, esta parte del sistema se hace lineal o se puede modelar como un sistema de primer o segundo orden; esto quiere decir, para un incremento en voltaje en la entrada del controlador se pretende un incremento en las revoluciones en la misma proporción. Esto simplifica mucho el esquema de control general, pues esta parte puede ser tratada como un sistema estable, de lo contrario, habría que incluir en el estudio del sistema en su conjunto los efectos de la no linealidad del controlador conmutado y del motor.

Dentro del esquema de control global de este proyecto, existe el problema del modelado de cada una de las partes que conforman al sistema. El motor y su controlador son elementos no lineales, esto significa que se deben tomar, dentro del esquema de control, en cuenta los siguientes aspectos:

1. La potencia entregada al motor no es linealmente dependiente de la señal de entrada al controlador de ángulo de disparo.
2. Las revoluciones no son una función lineal del voltaje aplicado al motor.
3. La presión o gasto que entrega la turbina no es una función lineal de las revoluciones del motor.

Estos tres puntos llevan a la necesidad de establecer un modelo de control sofisticado para el sistema medidor (control global). Esta deficiencia en el controlador del motor puede ser resuelto en el modelado del sistema en conjunto. De esta forma el controlador general o global absorbería la no linealidad del controlador.

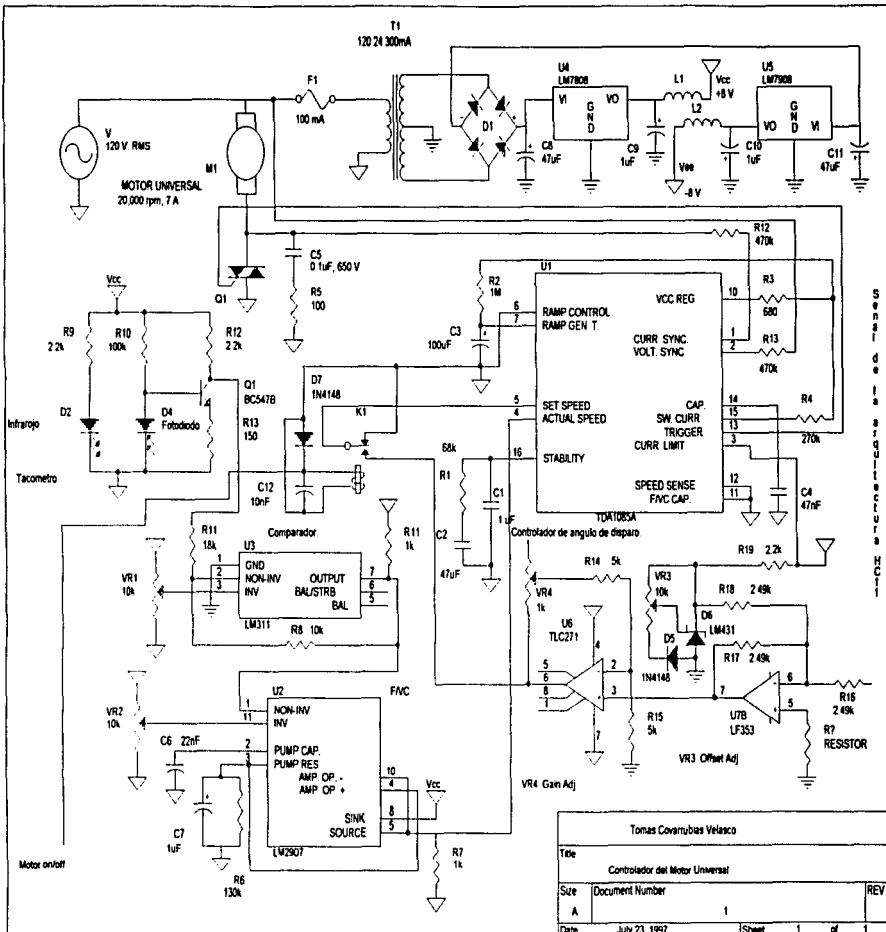
Para hacer lineal en su totalidad este subsistema (motor y extractor), debe existir una malla de realimentación que cuantifique la presión o gasto que entrega al motor, lo que es muy complejo para este aparato medidor³.

Se podrían evitar los dos primeros puntos al utilizar un control en lazo cerrado: un control lineal para las revoluciones del motor.

El control del motor en lazo cerrado se centró, entonces, en hacer de las revoluciones una función lineal de la señal de voltaje aplicada al circuito controlador de ángulo de disparo.

El circuito integrado TDA1085 tiene todos los elementos necesarios para acoplar directamente un tacómetro analógico o digital. Por las razones de construcción, se utilizó un tacómetro digital, lo que llevaría a utilizar la entrada digital del circuito integrado; sin embargo, la señal que debe alimentar al circuito debe ser bipolar. Esto es un problema para el diseño, pues la señal generada en el tacómetro y en el comparador es unipolar. Existen técnicas para generar señales bipolares a partir de una señal unipolar, sin embargo ello implicaba utilizar fuentes bipolares de voltaje considerable (± 16 VDC), lo que hace sofisticado el sistema de alimentación. Es por ello que se introdujo un circuito convertidor de frecuencia a voltaje que opera con señales unipolares. Esto tiene la ventaja de acceder a todos los elementos electrónicos del sistema controlador, permitiendo ajustes en sus distintas partes. El diagrama utilizado para el control del motor está en la página siguiente.

³Más aún si el método de orificio afecta considerablemente al sistema en su conjunto.



Este circuito está conformado por varias partes:

- 1.- El elemento controlador del ángulo de disparo (TDA1085A).
- 2.- Un tacómetro (conformado por un fotodiodo y un diodo infrarrojo).
- 3.- Un acondicionador de señales (el comparador LM311).
- 4.- Un convertidor de frecuencia a voltaje (LM2907).
- 5.- Acoplamiento con la arquitectura HC11.

Algo que se debe observar en la construcción de este controlador es que la alimentación de la electrónica de control y el motor deben tener una referencia común por varias razones:

1. El circuito TDA1085 monitorea señales de voltaje y corriente, por lo que deben estar referenciadas a la alimentación del motor.
2. El voltaje de compuerta del tiristor debe estar referenciado al voltaje en sus terminales (que son las del motor).

Es por estas razones que el circuito controlador está "vivo"; es decir, está directamente conectado a la línea de AC. Esto debe tomarse en cuenta al acoplar este controlador con la arquitectura HC11. El circuito controlador debe estar aislado de la arquitectura.

3.5.1 El tacómetro.

Un tacómetro es un dispositivo que permite generar una señal analógica o digital que es función del número de revoluciones. Existen dos procedimientos para medir revoluciones; uno de ellos utiliza un generador de CD y el otro modelo es a partir de una señal digital o analógica alterna, cuya frecuencia es la misma o múltiplo de las revoluciones del motor.

En este trabajo se utilizó un tacómetro digital. Las razones por las que se descartó un tacómetro analógico son la falta de recursos propios para este proyecto y la relativa complejidad de construir un tacómetro utilizando un generador (el trabajo mecánico sobre el motor, elaboración de piezas, caracterización, etc.).

El construir un tacómetro digital no es difícil, pues todo se resume a generar una señal cuadrada cuya frecuencia sea igual o múltiplo de las revoluciones. El circuito TDA1085 tiene ya todos los elementos necesarios para acoplar tanto un tacómetro analógico como un tacómetro digital. En las hojas de especificaciones del fabricante⁴ se incluyen ejemplos para ambas conexiones.

Para la generación de una señal cuadrada se contemplaron varias opciones. Una de ellas era la de colocar un disco ranurado sobre el eje del motor y detectar las revoluciones mediante interrupciones de luz; otra forma es la de generar una señal con inducción magnética, esto es, mediante un imán generar una señal alterna en un inductor. Estas dos formas presentan una dificultad: el trabajo mecánico sobre el motor es complejo. Se analizó una tercera opción: generar una señal cuadrada por reflejos de luz. Esta

⁴Motorola, "Linear/Interface IC's Device Data", tomo primero.

Este circuito está conformado por varias partes:

- 1.- El elemento controlador del ángulo de disparo (TDA1085A).
- 2.- Un tacómetro (conformado por un fotodiodo y un diodo infrarrojo).
- 3.- Un acondicionador de señales (el comparador LM311).
- 4.- Un convertidor de frecuencia a voltaje (LM2907).
- 5.- Acoplamiento con la arquitectura HC11.

Algo que se debe observar en la construcción de este controlador es que la alimentación de la electrónica de control y el motor deben tener una referencia común por varias razones:

1. El circuito TDA1085 monitorea señales de voltaje y corriente, por lo que deben estar referenciadas a la alimentación del motor.
2. El voltaje de compuerta del tiristor debe estar referenciado al voltaje en sus terminales (que son las del motor).

Es por estas razones que el circuito controlador está "vivo"; es decir, está directamente conectado a la línea de AC. Esto debe tomarse en cuenta al acoplar este controlador con la arquitectura HC11. El circuito controlador debe estar aislado de la arquitectura.

3.5.1 El tacómetro.

Un tacómetro es un dispositivo que permite generar una señal analógica o digital que es función del número de revoluciones. Existen dos procedimientos para medir revoluciones; uno de ellos utiliza un generador de CD y el otro modelo es a partir de una señal digital o analógica alterna, cuya frecuencia es la misma o múltiplo de las revoluciones del motor.

En este trabajo se utilizó un tacómetro digital. Las razones por las que se descartó un tacómetro analógico son la falta de recursos propios para este proyecto y la relativa complejidad de construir un tacómetro utilizando un generador (el trabajo mecánico sobre el motor, elaboración de piezas, caracterización, etc.).

El construir un tacómetro digital no es difícil, pues todo se resume a generar una señal cuadrada cuya frecuencia sea igual o múltiplo de las revoluciones. El circuito TDA1085 tiene ya todos los elementos necesarios para acoplar tanto un tacómetro analógico como un tacómetro digital. En las hojas de especificaciones del fabricante⁴ se incluyen ejemplos para ambas conexiones.

Para la generación de una señal cuadrada se contemplaron varias opciones. Una de ellas era la de colocar un disco ranurado sobre el eje del motor y detectar las revoluciones mediante interrupciones de luz; otra forma es la de generar una señal con inducción magnética, esto es, mediante un imán generar una señal alterna en un inductor. Estas dos formas presentan una dificultad: el trabajo mecánico sobre el motor es complejo. Se analizó una tercera opción: generar una señal cuadrada por reflejos de luz. Esta

⁴Motorola, "Linear/Interface IC's Device Data", tomo primero.

solución tendría la ventaja de evitar el trabajo mecánico sobre el motor, pues se colocaría un reflejante en alguna parte externa del motor; se escogió sobre el ventilador del motor. Tiene la desventaja de que un ambiente muy iluminado puede fallar la lectura si no se protege contra luz (hay que tener presente que está colocado en el ventilador de enfriamiento del motor, donde deben existir orificios que permitan el flujo de aire).

Para la detección de luz se utilizó una superficie reflejante, (un fragmento de un "Compact Disc") y una óptica utilizada en los lectores de CD; según la siguiente figura:

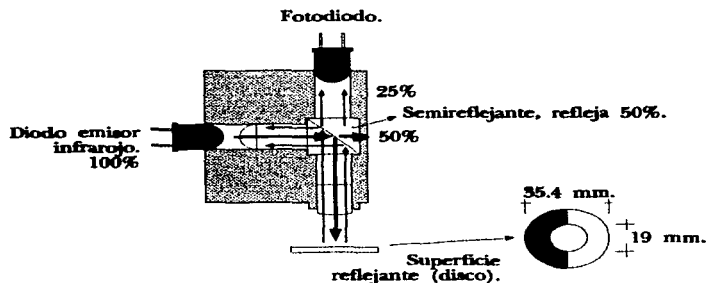


Fig. 3.13 Esquema del tacómetro digital.

El funcionamiento del sensor es (referirse a la figura superior): el diodo emite luz infrarroja que es rebotada en un 50% en el semiespcjo hacia el disco reflejante en el motor (indicada con el color rojo), el otro 50% de la luz se pierde; una vez que la luz rebota en el disco (en color verde), nuevamente pasa por el semiespejo que permite el paso del 50% de esa luz y el otro 50% se desvía hacia el led infrarrojo. El porcentaje de luz que pasa, que es menor al 25%⁵ del total emitido por el led, incide en el fotodiodo. De esta forma, y de acuerdo al disco, se genera un pulso luminoso que se transforma en una variación de impedancia en el fotodiodo. La impedancia de este medidor es muy alta, por lo que se requiere de un circuito acoplador y reforzador de la señal del fotodiodo; es por ello que se colocó un amplificador con un transistor bipolar de juntura y un comparador con histéresis. Además de dar el refuerzo necesario, hace la señal cuadrada, sin las perturbaciones provocadas por las impurezas e inclinaciones del disco⁶.

El diagrama del amplificador y del comparador es:

⁵Sería del 25%, suponiendo que la superficie es totalmente reflejante, es decir de un albedo del 100%, lo cual no es cierto del todo, por las impurezas del aluminio y del acabado. Se desconocen las pérdidas totales, pero operó satisfactoriamente en las pruebas.

⁶Al utilizar histéresis en el comparador se eliminan los ruidos y deformaciones de la señal del fotodiodo.

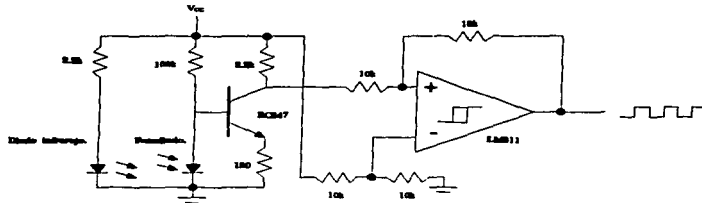


Fig. 3.14 Diagrama del amplificador y comparador.

La señal cuadrada generada tiene una frecuencia igual a las revoluciones del motor.

3.5.2 El convertidor de frecuencia a voltaje (F/VC).

La sección de conversión de frecuencia a voltaje se lleva a cabo en el circuito LM2907⁷. Este es un circuito convertidor de frecuencia a voltaje que opera a base de un bomba de carga. El circuito genera una señal cuadrada (con un comparador) a partir de la señal alterna que lo alimenta. A partir de los semiciclos positivos de la señal cuadrada se carga un capacitor (bomba de carga), el cual es descargado a través de una resistencia. El circuito diseñado a partir de esta parte es el sugerido por el fabricante. Los valores que se obtuvieron para el capacitor de carga y la resistencia de descarga fue en base a las ecuaciones que entrega el fabricante. El diagrama de bloques es:

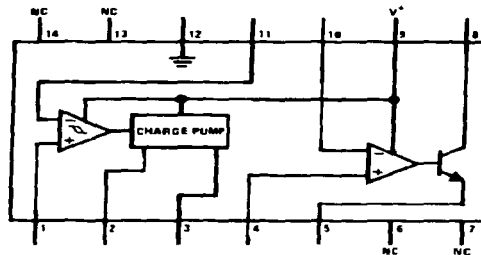


Fig. 3.15 Diagrama esquemático del LM2907.

El convertidor tiene como salida un amplificador operacional en colector y emisor abierto. Estos elementos están aislados de la bomba de carga, con el objeto de que el diseñador acondicione la salida a las necesidades de diseño. En este proyecto se utiliza como seguidor de la bomba de carga.

⁷National Semiconductor, "Híbrido products", pag.

3.5.3 Controlador de ángulo de disparo.

Este bloque del sistema controlador del motor está a cargo del circuito TDA1085. Es un controlador de ángulo de disparo diseñado para motores universales. El circuito puede operar en lazo cerrado, contiene los elementos necesarios para acoplar un tacómetro digital o analógico. En el TDA1085 se distinguen varias partes (ver figura 3.16):

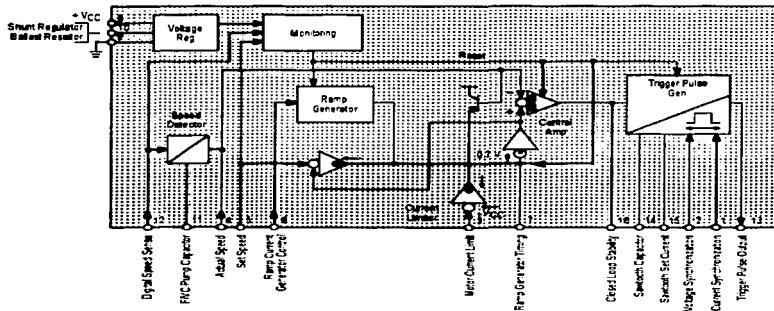


Fig. 3.16 Diagrama de bloques del TDA1085.

1. **Convertidor F/V:** éste convierte una señal bipolar digital en una señal de voltaje linealmente dependiente de la frecuencia de la señal digital. La entrada a este convertidor es a través de la terminal, 12. El método de conversión de frecuencia a voltaje es exactamente igual al del LM2907, es decir, a través de una bomba de carga. El capacitor de carga se coloca en la terminal 11. Esta parte del TDA1085 no se utiliza, por requerir señales bipolares, las cuales no se generan en el tacómetro digital de este circuito. La salida del convertidor está accesible en la terminal 4, en donde se puede conectar un tacómetro analógico. Es en esta terminal donde se conecta la señal de voltaje del convertidor LM2907.
2. **Generador de rampa:** esta parte se encarga de generar una rampa para la aceleración del motor. Si no se utiliza, el motor se acelera lo más rápido posible. Esta función está accesible a través de la terminal 6 del C. I.
3. **Protección contra sobrecorriente:** este subsistema monitorea el voltaje en la terminal 3 del C. I. Es en esta terminal donde se puede monitorear la corriente que circula por el motor, para ello se utiliza un resistencia muy baja. El fabricante sugiere el valor para un motor específico. Esta parte no es utilizada. Si se desea utilizar habrá que caracterizar y experimentar con el motor y obtener el valor óptimo de esta resistencia. La función de este subsistema es disminuir el voltaje promedio aplicado al motor para evitar el daño del tiristor o del motor.

4. **Comparador:** genera una señal de error a partir de la diferencia entre el voltaje en la terminal 5 y el voltaje en la terminal 4, que es el voltaje que entrega el F/Vc del mismo TDA1085 o del voltaje aplicado a la terminal 4. Es en función de esta señal de error que se determina el ángulo de disparo. También tiene un circuito generador de rampa para la aceleración del motor.
5. **Generador de disparo:** controla el disparo del tiristor. Este subsistema se encarga de monitorear a la señal de voltaje y la corriente en el tiristor. Es importante la sincronización con la corriente, pues en un circuito reactivo la corriente está desfasada respecto a la señal de voltaje⁸. En función de la frecuencia de operación se seleccionan los valores de capacitor y de resistencia para el temporizado de esta parte. Este subsistema entrega un pulso cuando el tiristor debe activarse. Este dispositivo funciona utilizando rampas (señal tipo diente de sierra) de la misma frecuencia de la línea y utilizando comparadores para generar el disparo.

Como consideraciones de diseño en este circuito hay que destacar varios puntos:

1. La señal de disparo del TRIAC es cuadrada y unipolar (0 [V] a Vcc). La resistencia que se interpone entre la terminal de disparo y el TRIAC tiene la finalidad de limitar la corriente entregada al TRIAC (y debe ser la corriente de disparo especificada por el fabricante). Esta resistencia debe dar la suficiente corriente para disparar al TRIAC.
2. El tiristor lleva una malla de protección⁹ (resistencia y capacitor) para evitar disparos no deseados. Esta malla es la sugerida por el fabricante. Tiene por finalidad evitar las variaciones de voltaje (dV/dt) abruptas cuando se apaga el TRIAC¹⁰.
3. La malla de estabilidad en lazo cerrado depende de las características del motor. En este trabajo se siguió la malla propuesta por el fabricante.
4. El convertidor F/Vc no se utiliza. El fabricante sugiere aterrizar la entrada y la malla de la bomba de carga. Se siguió esta sugerencia.
5. No se utiliza el regulador interno de voltaje. Queda conectado como un circuito de protección a sobretensión.

A pesar de que el circuito TDA1085 tiene un regulador lineal que permite su conexión directa a la línea de A. C., se prefirió utilizar un transformador y un regulador lineal de 15 [V]. Esto se debe fundamentalmente a dos razones:

1. Limitar la energía que pueda llegar al circuito, al entregar el transformador una corriente pequeña que puede ser absorbida por el regulador y el transformador en conjunto.

⁸Los TRIACs sólo se apagan cuando el voltaje en la compuerta es cero y la corriente que pasa por ellos es cero. Es por ello que el circuito debe monitorear la señal de voltaje y corriente, de esta manera se tiene información cierta del momento en el que se apaga y enciende el tiristor.

⁹En inglés se denomina "snubber network".

¹⁰Los tiristores se disparan cuando se alcanza una de las siguientes condiciones:

a) Señal en la compuerta.

b) Variaciones de voltaje grandes, es decir, dV/dt grandes, especificado por el fabricante.

c) Cuando se excede el voltaje de ruptura, especificado por el fabricante

2. Evitar la disipación de energía en forma de calor en resistencias y el regulador interno del TDA1085 cuando se utiliza directamente la línea de AC.

De esta forma, las variaciones en voltaje de la línea son limitadas en el transformador y absorbidas por el regulador lineal (LM7815).

El criterio para la selección del TRIAC fue:

1. El voltaje pico de la línea de AC es de ± 180 [V], lo que establece el valor mínimo de ruptura. Se prefirió utilizar uno de 600 [V]. Voltaje de ruptura del tiristor: 600 [V].
2. El motor demanda una corriente de 7 [A] bajo carga. Se propone el doble para evitar daños por sobrecorriente durante el arranque del motor. Corriente máxima: 15 [A].
3. El controlador entrega corriente al TRIAC, lo que puede dar dos casos: voltaje en compuerta positivo con voltaje en MT1 positivo (cuadrante I), y voltaje en compuerta positivo con voltaje en MT1 negativo (cuadrante III). Operación en cuadrantes¹¹ I, II, y III.

3.5.4. Acoplamiento del controlador con la arquitectura HC11.

Este acoplamiento merece ciertas consideraciones de diseño. El controlador TDA1085 necesita estar referenciado a la línea de AC para monitorear el voltaje y la corriente, y controlar el disparo del TRIAC. Es por esta razón que el polo negativo de la fuente que alimenta al controlador (15 [V]) es común a la referencia del circuito de AC. Esto es riesgoso en caso de falla para una arquitectura basada en un microprocesador por dos razones principalmente:

- 1.- Daño a la arquitectura por descargas en la línea (reposición total y costo de la misma).
- 2.- La seguridad del usuario.

Además hay que recordar que la arquitectura se conecta a una computadora personal (PC), lo que significa que la referencia para la PC y la arquitectura del HC11 deben ser la misma. Esto significa que ambas deben estar aisladas del vivo y del neutro de la línea de AC. Para la alimentación de la arquitectura del HC11 se utiliza una fuente de computadora aterrizada. En este caso no hay problema con la conexión de la PC, pues las computadoras también están aterrizadas; entonces la referencia es la misma. Pero para el acoplamiento del controlador del motor no se tiene esta misma referencia, ya que la referencia del controlador puede ser el neutro o la fase de la línea de alimentación (según la polaridad de la conexión); es ahí donde está el problema. Es por esta diferencia de comunes (voltajes) que se debe aislar al controlador. Para ello existen muchas formas de aislar un circuito. El problema es que se desea utilizar una forma de aislamiento que permita un relación lineal entre la entrada y la salida. Se proponen

¹¹El término cuadrante se refiere a los voltajes aplicados en las terminales de compuerta y terminal 1 del TRIAC. De esta definición se deducen cuatro posibles cuadrantes: MT1 positiva y G positiva (cuadrante I), MT1 positiva y G negativa (cuadrante II), MT1 negativa y G positiva (cuadrante III), y finalmente MT1 negativa y G negativa (cuadrante IV).

dos esquemas de acoplamiento. Ambos esquemas están basados en circuitos optoacoplados que permitan una relación lineal entre la entrada y la salida.

Uno de ellos puede ser con un convertidor de DC a DC que opera con un modulador de ancho de pulso (PWM). De esta forma se tiene una señal de voltaje promedio que es linealmente dependiente del ciclo de trabajo del tren de pulsos de entrada. Una de las razones que apoyan este esquema es el procesador HC11 tiene un periférico que puede realizar esta tarea de modulación.

El otro esquema es utilizando un convertidor digital-analógico optoacoplado a un puerto del HC11 o de la arquitectura.

Convertidor DC-DC.

El modelo de este tipo de convertidores de DC a DC está basado en el siguiente circuito:

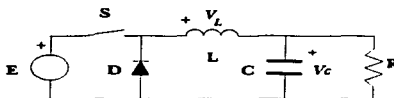


Fig. 3.17 Convertidor DC a DC.

En este circuito el capacitor y el inductor ofrecen almacenamiento de energía, suavizando la señal de salida (con un ligero rizo). Este circuito tiene dos formas de operación, una de ellas es que el modo de operación permite circular una corriente continua por el inductor; la otra forma de operación es que exista un intervalo de tiempo en el que el inductor se quede sin corriente. La diferencia entre uno u otro modo de operación radica en la cantidad de corriente que demande la carga, la frecuencia de operación y el ciclo de trabajo. Para el uso que se pretende dar a este circuito en esta tesis, se prefiere que el inductor no se quede sin corriente, pues de lo contrario la relación ciclo de trabajo-voltaje de salida se volvería no lineal. La frecuencia de operación y la cantidad de corriente que demande la carga determinarán las magnitudes de capacitancia e inductancia.

Hay que destacar que este circuito no se va a utilizar como un convertidor DC a DC de potencia, simplemente se desea utilizar este convertidor para obtener una relación lineal entre el ciclo de trabajo, que es lo que se va a variar en el controlador HC11 (entrada del sistema controlador del motor) y la salida de voltaje (salida del sistema), además de aislar a los dos circuitos¹².

Las consideraciones de análisis son: en el circuito el interruptor S es abierto y cerrado periódicamente; este período total es T, mientras que la fracción del período en el que el interruptor está cerrado es D; además se supone al capacitor lo suficientemente grande para que el voltaje V_C se mantenga constante.

Para iniciar con el análisis, obsérvese la siguiente figura, que es el caso en el que el interruptor S está cerrado:

¹²El aislamiento se logra al utilizar un optoacoplador como el interruptor S.

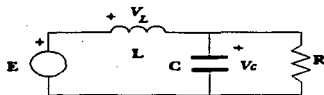


Fig. 3.18 Circuito con el interruptor cerrado.

La ecuación que describe el comportamiento del circuito para el intervalo en el que el interruptor permanece cerrado es:

$$E = v_L + V_c.$$

De la ecuación que define al inductor y despejando a la corriente en el inductor:

$$E = L \left(\frac{di_L}{dt} \right) + V_c$$

$$\frac{di_L}{dt} = \frac{E - V_c}{L} \quad (\text{ec. 3.1}).$$

Durante este intervalo de tiempo DT , la corriente del inductor se incrementa de manera lineal, con una pendiente como se muestra en las gráficas posteriores. La corriente del inductor comienza con un valor mínimo y termina con un valor máximo al final del intervalo DT .

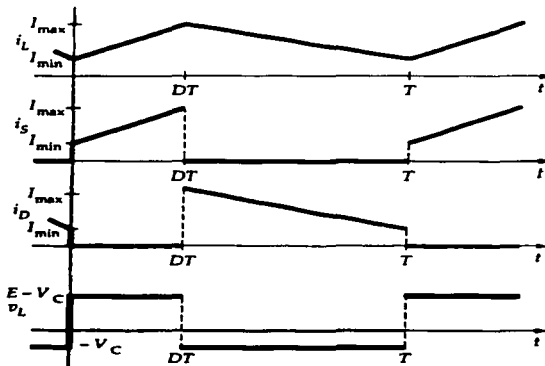


Fig. 3.19 Formas de onda en el inductor, interruptor y diodo.

Para el intervalo en el que el interruptor está abierto se tiene el siguiente circuito:

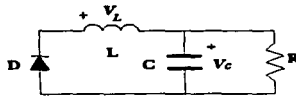


Fig. 3.20 Circuito con el interruptor S abierto.

La ecuación que rige este circuito es, despreciando la caída en el diodo:

$$0 = v_L + V_c.$$

De la misma manera como se trabajó a las ecuaciones anteriores, expandiendo y desarrollando:

$$0 = L \frac{di_L}{dt} + V_c.$$

$$\frac{di_L}{dt} = -\frac{V_c}{L} \quad (\text{ec. 3.2}).$$

Así, durante el intervalo $(1-D)T$ (que es el tiempo que está abierto el interruptor S), la corriente del inductor decrece a un ritmo constante de I_{max} a I_{min} . El valor final debe ser el mismo que el valor al principio del periodo. Hay que recalcar que la consideración para este análisis es que la corriente en el inductor es mayor que cero. Utilizando las expresiones anteriores (ec. 3.1 y 3.2) se puede obtener las diferencias de corriente:

$$I_{max} - I_{min} = \left(\frac{E - V_c}{L} \right) DT \quad (\text{ec. 3.3}).$$

$$I_{min} - I_{max} = \left(\frac{-V_c}{L} \right) (1-D)T \quad (\text{ec. 3.4}).$$

Ambas diferencias deben ser iguales, por lo que igualando la ecuación 3.3 y la 3.4:

$$\left(\frac{E - V_c}{L} \right) DT = \left(\frac{V_c}{L} \right) (1-D)T$$

$$(E - V_c)D = (V_c)(1-D)$$

$$V_c = DE$$

En esta igualdad final puede verse que el voltaje en el capacitor es exclusivamente función del intervalo en el que está el interruptor S cerrado. Las ecuaciones que definen las corrientes máximas y mínimas son:

$$I_{max} = (DE) \left[\frac{1}{R} + \frac{(1-D)T}{2L} \right] \quad (\text{ec. 3.5}).$$

$$I_{min} = (DE) \left[\frac{1}{R} - \frac{(1-D)T}{2L} \right] \quad (\text{ec. 3.6}).$$

Hay que hacer notar varios aspectos que no se consideraron en este análisis: el que ya se mencionó, que la corriente en el inductor es mayor que cero, el voltaje en el capacitor es invariante, es decir, no hay rizo, y por último no se tomaron en cuenta las pérdidas por resistencia en el inductor y el diodo. Aunque se pueden utilizar componentes que minimicen estas pérdidas, conviene realizar pruebas experimentales y ver si los resultados están alejados del cálculo. El paso siguiente es determinar los valores de los elementos que conforman este circuito.

La fuente (E) que se utiliza en este circuito es la del regulador lineal del controlador del motor, es decir, la fuente de 15 [V]. La frecuencia de operación puede ser un parámetro a variar, la literatura consultada¹³ sugiere frecuencias a partir de 8 [kHz]. Lo que determina la frecuencia en este convertidor es el microcontrolador HC11, por lo que se establece como valor inicial 7.8125 [kHz]¹⁴. Ahora bien, realmente no se desea transmisión de potencia por lo que la corriente en el circuito queda determinada por la corriente que alimenta al TDA1085. El TDA1085 recibe una señal de voltaje en la terminal 5 con una corriente de 1 [μA] como máximo. Con esta información se propone una resistencia tal que la corriente que circule por ella sea mayor a 10 veces la corriente que entra por la terminal 5 (pues lo que se desea es una señal de voltaje). La corriente que se propone es de 1 [mA] cuando el voltaje de salida (Vc) sea de 15 [V]. El valor de la resistencia es, entonces, de 15 [kΩ].

De la ecuación 3.4 puede determinarse el valor mínimo de la inductancia para que la corriente que circula por ella sea mayor que cero:

$$L = \left(\frac{TR}{2} \right) (1 - D)$$

El valor de inductancia más grande que arroja esta ecuación es cuando el interruptor se cierra por un intervalo de tiempo tal que el voltaje de salida sea 1 [V]¹⁵. Entonces el valor máximo y mínimo, que es cuando se quiere una salida de 13.5 [V]¹⁶, del ciclo de trabajo son:

$$D_{max} = \frac{V_c}{E} = \frac{13.5}{15} = 0.9 [V].$$

$$D_{min} = \frac{V_c}{E} = \frac{1}{15} = 0.0666 [V];$$

y la inductancia está determinada por el valor mínimo de D:

$$L_{min} = \left(\frac{TR}{2} \right) (1 - D) = \frac{(1/7812.5 [s])(15,000 [\Omega])}{2} (1 - 0.066) = 896.64 [mH].$$

¹³Marvin J. Fisher, "Power Electronics", PWS-KENT Pub. Co., capítulo 7: "DC to DC Converters".

¹⁴El generador de PWM del HC11 funciona en base al contador de 16 bits. El ciclo completo del contador es de 32.7 ms, que es una frecuencia de 30 Hz; sin embargo, si se toman sólo ocho bits para la frecuencia de salida, se tiene un periodo de 128 μs (7812.5 Hz). Esta es la frecuencia de la que se parte. Posteriormente se discute el incremento de esta frecuencia.

¹⁵Este es el voltaje en el que el motor empieza a girar.

¹⁶Para lograr máxima velocidad se deben aplicar 13.5 [V] al controlador TDA1085.

En el resultado de esta aproximación pueden observarse varias cosas importantes: el inductor es de un valor muy alto debido que una resistencia tan alta hace circular una corriente muy pequeña, por lo que se necesita un inductor muy grande para mantener una corriente durante el máximo intervalo que el interruptor está apagado, para una frecuencia dada. Un inductor de tamaño físico razonable tiene valores de inductancia del orden de cientos de micro Henries hasta algunos cientos de mili Henries, por lo que conviene disminuir el valor de la resistencia y subir la frecuencia de operación. El límite en la resistencia se determina a partir de cuánta corriente se consumirá en este circuito (esta corriente es sólo para generar una señal de voltaje, de desperdicio) y la frecuencia máxima a la que se puede operar la determina el modulador de ancho de pulso del HC11.

Analizando estas dos posibles soluciones: si se quiere aumentar la frecuencia del HC11 se sacrifican niveles de voltaje a la salida. El generador PWM del HC11 se genera a partir de un contador de 16 bits cuyo período de cuenta es de 500 [ns] para la velocidad más alta de reloj (8 [MHz]); es decir, que para una cuenta completa de 16 bits le toma 32.7 [ms] (30.51 [Hz]). Se propuso originalmente reducir la cuenta a períodos de 8 bits para tener 256 ciclos de 500 [ns] cada uno, lo que da un período de 128 [μs] (7812.5 [Hz]). Esta es la frecuencia de la que se partió para hacer el análisis. Lo que interesa es disminuir el tamaño del inductor en, por lo menos, 100 veces. La frecuencia de operación del PWM no se puede subir en 100 veces, ni siquiera en 10 veces, pues se tendrían que sacrificar bits de cuenta, lo que disminuye la resolución del convertidor DC-DC. La única salida es reducir la resistencia 100 veces, es decir, proponer una resistencia de 150 [Ω]. Esto lleva a varias consideraciones: una resistencia de 150 [Ω] que para una salida de voltaje de 13.5 [V] demanda una corriente de 90 [mA], lo que se vuelve significativo en el consumo de corriente del circuito, pues la fuente de alimentación es de 300 [mA].

Esta alternativa queda descartada. Sin embargo, sólo al hacer las pruebas del sistema en conjunto, se podrá volver a considerar esta opción si el método de convertidor digital-analógico presenta algún inconveniente.

Algo que deber quedar claro, es que no es una necesidad del circuito lograr los valores de inductancia obtenidos en los cálculos, pues están basados en la suposición de que el inductor mantiene una corriente para todo el período; esto es sólo necesario para obtener un convertidor DC-DC lineal. Si no se cumplen los requerimientos de inductancia arriba mencionados se obtiene un convertidor DC-DC que se comporta de la siguiente forma:

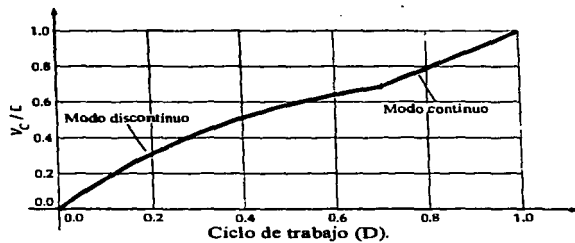


Fig. 3.21 Variación del voltaje en el capacitor para los dos modos de operación.

Este puede ser utilizado en el esquema de controlador del motor, pero se deben asumir este comportamiento no lineal en el esquema de control general, haciéndolo más sofisticado.

Convertidor D/A.

Este se basa en el uso de un convertidor digital-analógico de 8 bits, el cual permite 256 niveles de conversión unipolares. El diseño de este circuito se conforma de la siguiente manera:

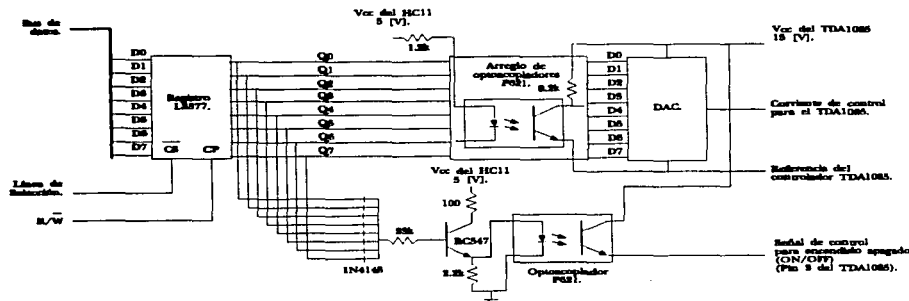


Fig. 3.22 Acoplamiento usando DAC.

El procesador HC11 escribe sobre un registro configurado en el mapa de memoria. Este registro contiene el valor numérico equivalente de la señal voltaje de control del motor. La salida del arreglo de optoacopladores es quien alimenta al convertidor D/A. Es en este punto donde se logra el aislamiento entre las partes; el DAC está alimentado por la fuente del controlador del motor cuya referencia es la línea de AC.

Existe un arreglo de diodos a la salida del registro LS377 que activan al transistor BC547. La intención de este arreglo es generar una señal que encienda y apague al motor. El objetivo de este circuito es asegurar el apagado al motor cuando se escribe un 0 en el registro, la salida del arreglo del optoacoplador es activar un relevador que active la señal de salida de la electrónica asociada al DAC cuando se escribe un número distinto de cero en el registro. Sin este circuito el motor nunca se detendría, pues la electrónica de control no alcanzaría el voltaje mínimo necesario para apagar al motor por la caída en la etapa de salida de los operacionales, que es de aproximadamente 80 [mV], sobre la referencia (-8 [V]); con esta señal se puede controlar el encendido y apagado del motor a través de la terminal 5 del TDA1085, que es la entrada de voltaje al controlador. Cuando se tiene un cero en el registro el optoacoplador no conduce; cuando se escribe algo distinto de cero, el optoacoplador conduce. De esta forma se puede utilizar un relevador para entregar al controlador cero (cuando se escribe un cero en el registro y se apaga el relevador) o el resultado de la conversión D/A.

Algunas observaciones respecto al circuito acoplador:

1. La configuración de los optoacopladores del DAC es: inversor a la entrada, inversor a la salida.
2. La corriente de los optoacopladores en su fotodiodo es de:

$$I = \frac{5 - (V_{OL} + V_d)}{1.2k} = 3[mA].$$

La cual entrega el registro LS377 (4 [mA], típico). La corriente que alimenta al transistor BC547 es despreciable, del orden de [μA].

3. La resistencia de 2.2 [k Ω] (en el circuito de apagado) en paralelo al fotodiodo tiene por objeto mejorar la respuesta en frecuencia del circuito al actuar como un elemento de descarga de la capacitancia parásita del fotodiodo y del transistor cuando éstos se apagan (uniones de semiconductores NP).

En el diagrama de la arquitectura del HC11 están detallados los elementos y conexiones de este circuito de acoplamiento, el diagrama en la siguiente página muestra el acoplamiento.

Los límites de operación de este circuito se podrían alcanzar si la velocidad de conversión D/A es tal que tanto el arreglo de optoacopladores como el DAC no puedan responder. Las velocidades de muestreo están definidas en el capítulo siguiente, que habla del modelo de control. Es hasta este capítulo donde se podría definir qué tipo de acoplamiento se requiere.

Puede verse que esta propuesta es más sencilla en cálculos que el PWM, que requiere de elementos inductivos, capacitivos, análisis de frecuencias, etc. La resolución de la respuesta de ambos métodos es la misma, 256 niveles de voltaje a la salida. La circuitería es más complicada para el DAC que para PWM.

Dos observaciones a estos circuitos:

1. Para el circuito de acoplamiento, tanto para PWM o para DAC, el consumo en potencia es considerable, uno por tener una resistencia baja para la modulación y el otro para la operación de los optoacopladores.
2. Tanto para el modelo de PWM y el de DAC existen las limitantes de velocidad de respuesta. Para el PWM se tiene como límite la frecuencia del PWM que es de 7812.5 [Hz] y para el DAC la respuesta en frecuencia de escritura del registro y los optoacopladores. Para ambos métodos existe un tercer factor que afecta su velocidad de respuesta: la velocidad de muestreo.

3.5.5 Descripción del circuito utilizando un convertidor D/A.

La salida del convertidor DAC0808 es en corriente y la entrada al TDA1085 es en voltaje. Hay que recalcar que el controlador de ángulo de disparo TDA1085 se debe alimentar con una fuente unipolar. Dentro del diseño del circuito controlador se tuvo la necesidad de utilizar una fuente bipolar debido a que el convertidor de frecuencia a voltaje, LM2905, la requiere. De esta forma, se diseñó una fuente que proporciona voltaje positivo de +8 [V] y voltaje negativo de -8 [V]. El controlador TDA1085 utiliza como referencia al voltaje de -8 [V] y como voltaje positivo (V_{cc}) al de 8 [V] (que dan una diferencia de 16 [V], que es el voltaje al que opera el TDA1085); así, el TDA1085 utiliza una polarización unipolar a partir de la fuente bipolar del circuito. Esta aclaración es necesaria, pues el convertidor DAC también utiliza alimentación bipolar y la salida es como un sumidero de corriente para la referencia de la fuente bipolar (tierra) a través de la terminal 4 del DAC. Este funcionamiento del DAC hace que la corriente que entra por la terminal salida (pin 4) tenga un intervalo desde 0 [A] hasta la corriente que se suministra por la terminal de voltaje de referencia; es decir, que la corriente que absorbe el circuito es la misma que entra por la terminal de voltaje de referencia positivo (pin 14). La corriente que entra por esta terminal está determinada por la resistencia colocada entre esa terminal y el voltaje de referencia, que es V_{cc} (8 [V]). El fabricante propone una corriente máxima de 4.2 [mA] para fuentes cuyo voltaje negativo sea menor a los -8 [V]. La corriente utilizada en este circuito es de 3.6 [mA]. Esta corriente es la que el circuito absorbe a la salida del convertidor (terminal 4). Si se utiliza el mismo valor de resistencia que se utiliza en la terminal de voltaje de referencia positivo se tendría un rango de 0 [V] a -8 [V]; sin embargo, la construcción del DAC no permite que se genere todo este intervalo de voltaje, pues existe un voltaje límite en la terminal de salida (terminal 4), por la construcción de las fuentes de corriente internas. Experimentalmente se encontró que el rango óptimo¹⁷ es a través de una resistencia de la mitad de la utilizada en la terminal de voltaje positivo de referencia, es decir, de 1.1 [k Ω]. De esta forma se tiene una variación de 0 [V] a -4 [V].

Esta explicación ha servido para exponer el punto: la corriente que entrega el convertidor DAC se convierte en un voltaje de 0 [V] a -4 [V]. (0 [V] para un valor hexadecimal de 00 en el DAC y -4 [V] para un valor hexadecimal FF); esto significa que este rango de voltaje debe convertirse en un rango de 0 [V] a 13.5 [V] del TDA1085 (que ve una fuente unipolar, donde la referencia es -8 [V] y V_{cc} es +8 [V]). Expresando este rango desde el punto de vista de la fuente bipolar, el intervalo es de -8 [V] hasta 5.5 [V] (una diferencia de 13.5 [V], que es lo que se utiliza en el TDA1085 como señal de entrada).

El proceso de acondicionamiento consiste en dos partes: invertir el voltaje que entrega el DAC (pues para un valor hexadecimal de 00 entrega un voltaje alto, mientras que para un valor hexadecimal mayor entrega menor voltaje); amplificar esta señal y hacer un corrimiento para que la variación sea de 0 [V] a 13.5 [V] (notar que para la fuente bipolar no es simétrico).

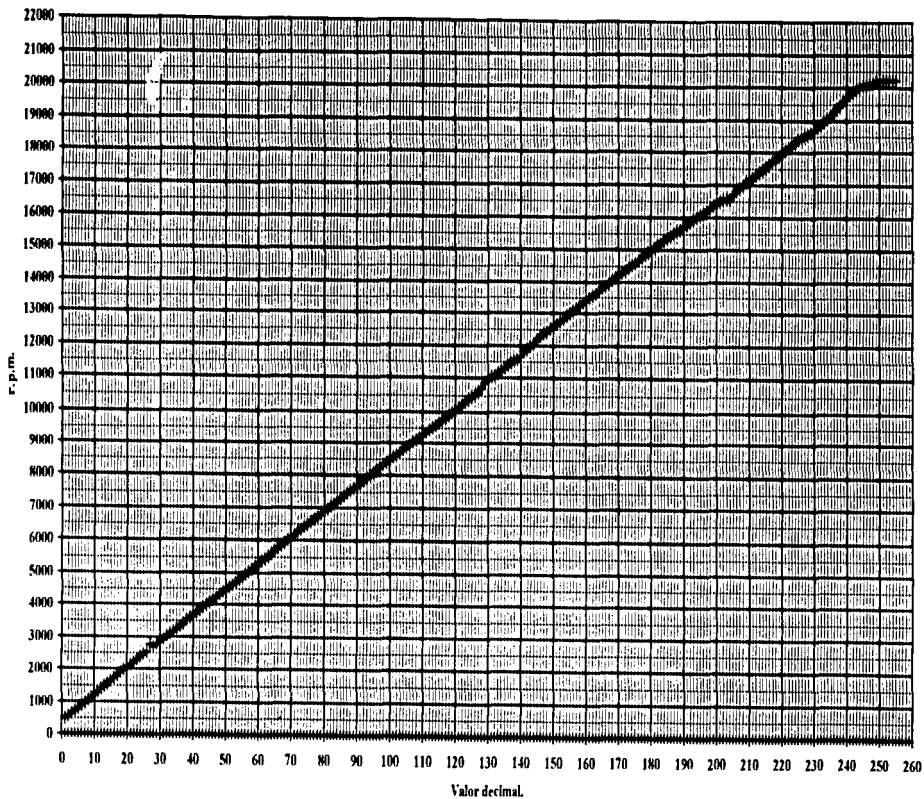
El potenciómetro VR1 hace el ajuste del corrimiento y el potenciómetro VR2 ajusta la ganancia. Todos los operacionales están alimentados con la fuente bipolar (± 8 [V]). De esta forma, cuando el DAC entrega una corriente de cero, el circuito entrega un voltaje de -7.5 [V]¹⁸ (que serían 0.5 para el controlador TDA1085) y para la corriente máxima del DAC entrega 5.5 [V] (que serían 13.5 [V] para el TDA1085).

Pruebas realizadas con el convertidor D/A

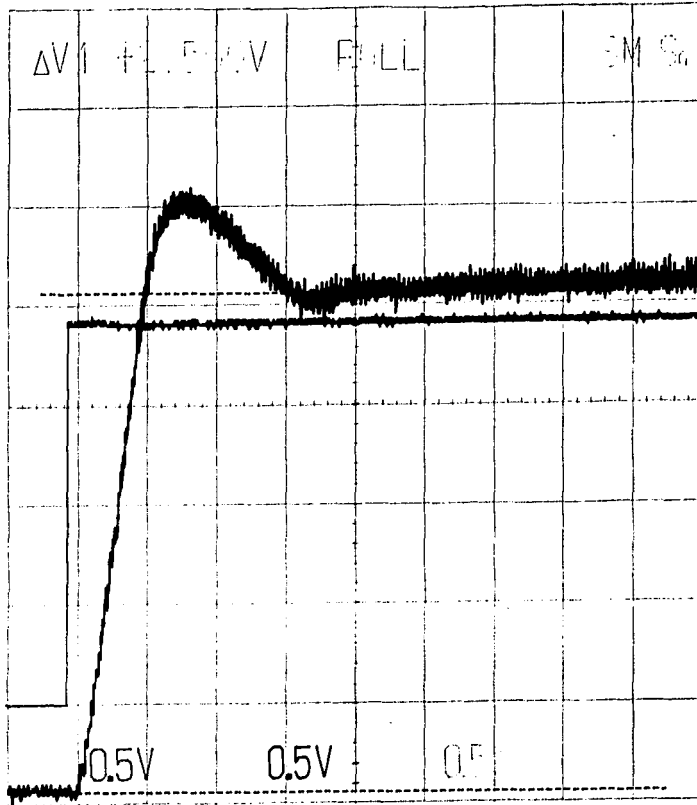
Se construyó el circuito utilizando el convertidor D/A DAC0808, la curva de valores decimales escritos en el registro vs. velocidad angular del motor y la respuesta a escalón del sistema se muestran en la siguientes hojas. Este circuito cumplió con las necesidades de la etapa de aislamiento y acoplamiento entre el controlador de ángulo de disparo y la arquitectura basada en el HC11. La respuesta del sistema es de segundo orden y las revoluciones a régimen permanente son directamente proporcionales al valor de la entrada, por lo tanto, es el circuito propuesto para este proyecto.

¹⁸La salida del TLC271 no es cero debido a las caídas de potencial en los transistores que forman la etapa de salida de este operacional.

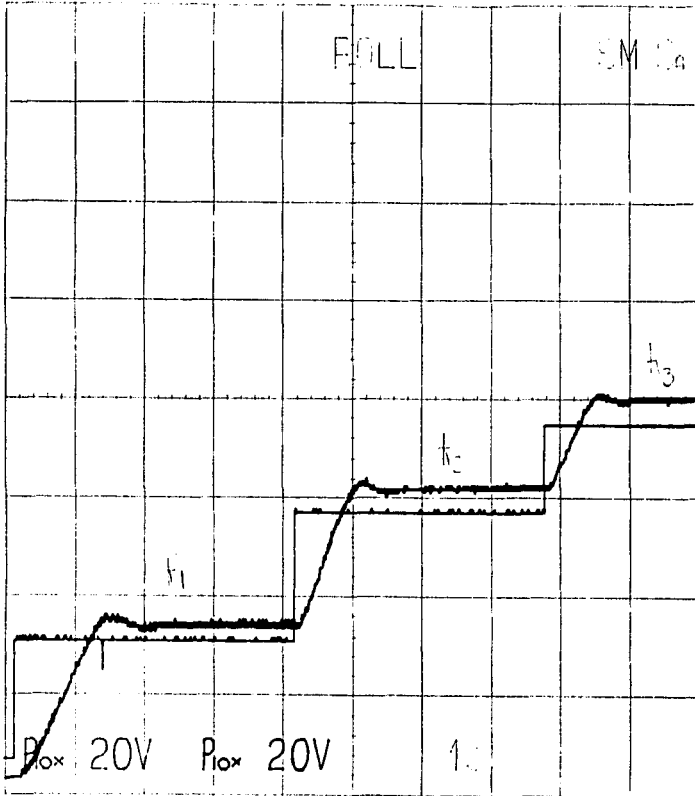
Fig. 3.25. Respuesta del controlador basado en DAC.



Respuesta a escalón.



Respuesta a incrementos de escalón. Salida medida a la salida del FVC.



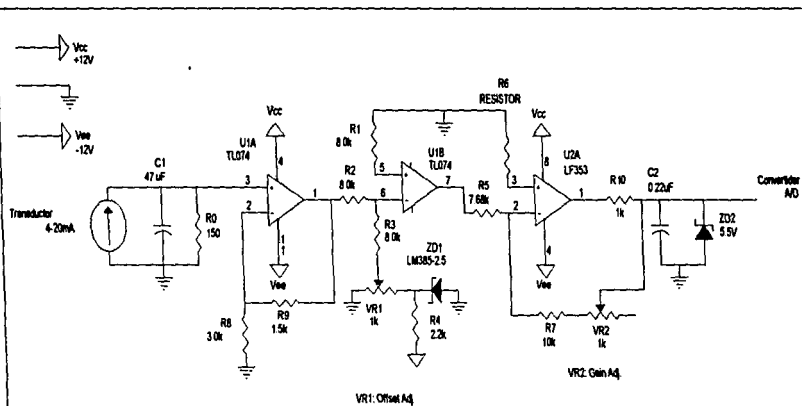
Parte II. Los transductores de presión.

Un transductor tiene la finalidad de hacer una conversión de una variable mecánica, físico o química a una variable eléctrica. En este caso, los transductores de presión hacen la conversión de movimiento mecánico en un diafragma, provocado por una diferencia de presión entre sus caras, a variaciones en corriente o voltaje. Los medidores de presión utilizados para este proyecto son transductores lineales de presión a corriente, con compensación a variación de temperatura. La marca de estos transductores es MAMAC y ofrecen las siguientes características:

1. Variación lineal entre la corriente y la presión.
2. Variación de corriente de 4 [mA], para la menor presión, y 20 [mA], para la mayor.
3. Alimentación de 12 a 24 [V].
4. Las variaciones por cambio en temperatura están compensadas.
5. Conexión en serie.
6. Unidireccionales diferenciales.

La información que entreguen los transductores de presión alimentará al convertidor A/D de la arquitectura basada en el HC11, para el control del aparato medidor. Es decir, el sistema microcontrolador, basado en el HC11, tomará la lectura del medidor de presión diferencial ubicado entre las caras del textil, y en base a esta lectura se cerrará el lazo de control para establecer la diferencia de presión establecida por la norma (0.5 pulgadas de agua) regulando las revoluciones del motor.

El convertidor A/D del HC11 utiliza una señal de voltaje que varía entre 0 [V] y 5 [V]. Es por esta razón que se necesita transformar la corriente que entrega el medidor de presión para que se obtenga una variación de 0 [V] a 5 [V]. Para ello se necesita hacer la transformación de corriente a voltaje. El circuito utilizado para lograr esta transformación y acondicionamiento está en la siguiente hoja.



Tomas Corvuelas Velasco		
Title		
Electronica para los Transductores		
Size	Document Number	REV
A		
Date:	September 21, 1997	Sheet 1 of 1

La conversión de corriente a voltaje se hace propiamente en la resistencia de 150 $[\Omega]$. Esta señal de voltaje tiene una variación de 0.6 [V] a 3 [V] (0.6 [V] corresponden a la corriente de 4 [mA] y los 3 [V] a una corriente de 20 [mA], que es la variación que entrega el medidor para una escala del 100%). Esta señal de voltaje se refuerza con un amplificador de ganancia de 1.5, conformado por el amplificador operacional TL074¹⁹; esta etapa entrega un rango de voltaje de 0.9 [V] a 4.5 [V]. Esta configuración asegura una alta impedancia a la entrada del amplificador; esto es de especial importancia, pues la etapa debe afectar en lo menor posible a la conversión de corriente a voltaje en la resistencia (al reflejarse como una carga muy grande paralela a la resistencia). La señal de salida de esta etapa amplificadora es la que se debe acondicionar propiamente. La señal pasa a través de un inversor (basado también en el TL074) en el que se hace la corrección de desplazamiento, pues se desea que para la lectura de menor presión se tenga un valor de voltaje de 0 [V]; es por esto que se utiliza una configuración de sumador-inversor. El voltaje que se alimenta al sumador es de -0.9 [V]; de esta forma, se logra que para una presión de 0 pulgadas de agua, en la que el transductor entrega 4 [mA] y el amplificador entrega 0.9 [V], se tenga un voltaje a la salida del sumador-inversor de 0 [V]. El voltaje de corrección se ajusta con el potenciómetro VR1, que permite una variación entre 0 [V] y 2.5 [V]²⁰. Esta etapa corrige el desplazamiento inherente a la conversión de corriente-voltaje de la resistencia; lo que falta ahora es una etapa que amplifique esta señal de manera que se tenga una variación de 0 [V] a 5 [V] cuando el transductor varía la corriente que pasa por él de 4 [mA] a 20 [mA] (hasta este sumador se tiene una variación de 0 [V] a 3.6 [V]). La etapa siguiente es un amplificador inversor, a la cual se le ajusta la ganancia a través del potenciómetro VR2 (la ganancia debe ser de aproximadamente 1.38). Con estas tres etapas se tiene un circuito acondicionador que hace la conversión de corriente a voltaje para el rango que utiliza el convertidor del microcontrolador HC11.

Ambos transductores operan de la misma forma, por lo que su circuitería es idéntica.

En la etapa de salida de este último operacional se puede observar una resistencia de 1k y un diodo zener. El conjunto de la resistencia y del diodo zener tienen como finalidad el limitar tanto la corriente que entrega el operacional como el voltaje²¹, ya que el convertidor A/D del HC11 está configurado para medir un voltaje máximo de 5 [V], además se impide voltajes negativos inferiores a 0.7 [V]. El arreglo de la resistencia y el capacitor también tienen el efecto de un filtro pasobajas cuya frecuencia de corte es:

$$f_o = \frac{1}{2\pi RC} = \frac{1}{2 \cdot 3.14159 \cdot 1[k\Omega] \cdot 0.22[\mu F]} = 723.43 [Hz].$$

¹⁹La característica de este amplificador operacional es su baja corriente de polarización, pues la etapa de entrada es de tipo FET.

²⁰El voltaje de 2.5 [V] se genera con un regulador paralelo de precisión, LM385-2.5.

La frecuencia natural del sistema²² es de aproximadamente 2 [Hz]. La frecuencia de corte depende de la frecuencia de muestreo, que para este sistema es de 1000 [Hz].

²¹El microcontrolador posee un circuito de protección. Sin embargo, Motorola sugiere colocar la resistencia de 1k para evitar el daño permanente a la configuración de entrada del convertidor A/D del HC11.

²²La frecuencia natural del sistema se determina experimentalmente en el capítulo 4, a partir de la respuesta a escalón del sistema.

Capítulo 4. Teoría de Control.

Este capítulo define los términos necesarios para desarrollar el modelo de control. Se hace una breve descripción de estos tipos de control y se presenta un método para encontrar el modelo matemático del sistema físico. Se selecciona un modelo de control y se desarrolla para la aplicación en el proyecto.

4.1 La necesidad del control en el medidor de permeabilidad.

La necesidad del control en el medidor de permeabilidad surge al momento de establecer una diferencia fija de presión para un textil dado. La norma establece que la permeabilidad debe ser determinada a una diferencia de presión fija, que es de 0.5 pulgadas de agua.

Es decir, en el sistema se debe controlar la presión aplicada por medio del extractor de aire de tal forma que se alcance y se establezca la diferencia de presión entre las caras del textil en media pulgada de agua.

4.2 Definiciones.

4.2.1 Definición de Sistema.

Puede ser entendido como un conjunto amplio de componentes o elementos en un arreglo tal que se comportan como una unidad.

4.2.2 Sistemas lineales y no lineales.

Un sistema es lineal si cumple con el principio de superposición¹.

4.2.3 Definición de control en lazo cerrado.

Un sistema de control de lazo cerrado es aquél en el que la señal de salida tiene efecto sobre la acción de control. Esto es, los sistemas de control de lazo cerrado son sistemas de control realimentado. La señal de error actuante, que es la diferencia entre la señal de salida y de realimentación, entra al detector o control para reducir el error y llevar la salida del sistema al valor deseado. Debe quedar claro que tanto la señal de referencia como la retroalimentada deben estar en el mismo tipo de unidades y magnitudes.

¹ El principio de superposición establece: $H\{\alpha x_a(t) + \beta x_b(t)\} = \alpha H\{x_a(t)\} + \beta H\{x_b(t)\}$

4.2.4 Definición de control en lazo abierto.

Los sistemas de control de lazo abierto son sistemas de control en los que la salida no tiene efecto sobre la acción de control. En un sistema de control de lazo abierto la salida ni se mide ni se realimenta para comparación con la entrada.

4.2.5 Estabilidad.

La característica más importante del comportamiento dinámico de un sistema de control es la estabilidad absoluta, es decir, si el sistema es estable o inestable en su conjunto. Un sistema es estable si, en ausencia de cualquier perturbación, la salida se mantiene en el mismo estado. Un sistema de control lineal invariante en el tiempo es estable si finalmente la salida retorna a su estado de equilibrio cuando el sistema es sometido a una perturbación. Un sistema de control lineal invariante en el tiempo es inestable si continúa indefinidamente una oscilación en la salida, o si la salida diverge sin límite de su estado de equilibrio cuando el sistema es sometido a una perturbación.

La ventaja del lazo cerrado sobre el lazo abierto es clara: el sistema de control de lazo cerrado permite una respuesta relativamente insensible a perturbaciones externas y a variaciones internas de parámetros del sistema. Aunque, desde el punto de vista de estabilidad, el control de lazo cerrado es más difícil, por la tendencia a corregir errores que puede generar oscilaciones de amplitud constante o variable.

4.3 Tipos de control.

Un control automático compara el valor efectivo de una salida en una planta contra un valor deseado, determina la variación y produce una señal de control que reduce la desviación a cero o a un valor muy pequeño.

Existen comúnmente seis tipos de control:

1. Control de dos posiciones o encendido/apagado.
2. Control proporcional.
3. Control integral.
4. Control proporcional e integral.
5. Control proporcional y derivativo.
6. Control proporcional, integral y derivativo.

Posteriormente se analiza el comportamiento del sistema medidor y se selecciona un tipo de control.

4.3.1 Control de dos posiciones o encendido/apagado.

El elemento accionador tiene exclusivamente dos posiciones, o encendido o apagado. Es el relativamente más simple y económico.

Si se tiene una señal de salida de control $m(t)$ y una señal de error actuante $e(t)$, en un sistema de dos posiciones, la señal $m(t)$ permanece en un valor máximo (M_1) o mínimo (M_2), según si la señal de error actuante sea positiva o negativa.

Existe una variación al control de dos posiciones en el que se permite un rango de variación en el que no se aplica ninguna excitación al sistema. La diferencia contra el otro tipo de control es que el control no actúa inmediatamente cuando se produce una señal de error, sino que permite un rango de variación del error. Este también es el que más se utiliza por su sencillez y su bajo costo. Es evidente que un sistema que requiera de señales de control precisas no puede ser controlado con este tipo de instrumentos.

Ambos sistemas de control tienen los siguientes diagramas de bloque:

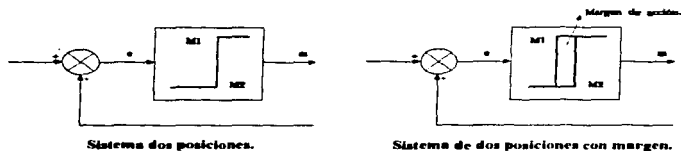


Fig. 4.1 Control ON/OFF.

Una de las desventajas de este tipo de control es que no hay ninguna forma de evitar o prevenir algunas oscilaciones en el sistema, que no hacen al sistema inestable.

4.3.2 Control proporcional (P).

En un control proporcional, la relación entre la salida del controlador $m(t)$ y la señal de error actuante $e(t)$ es:

$$m(t) = K_p e(t),$$

o en magnitudes de funciones en s:

$$\frac{M(s)}{E(s)} = K_p,$$

donde K_p se denomina sensibilidad proporcional o ganancia.

Cualquiera que sea el mecanismo en sí, y sea cual fuere la potencia que lo alimenta, el control proporcional es esencialmente un amplificador con ganancia ajustable. El diagrama de bloques es:

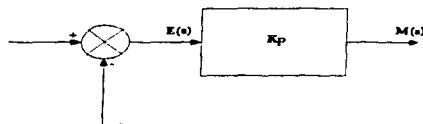


Fig. 4.2. Control proporcional.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

4.3.3 Control integral (I).

En el control integral la variación de la salida del controlador $m(t)$ varía proporcionalmente a la señal de error actuante $e(t)$. Es decir:

$$\frac{\delta m(t)}{\delta t} = K_i e(t),$$

o de manera integral:

$$m(t) = K_i \int_0^t e(t) dt,$$

donde K_i es una constante regulable. La función de transferencia de un control integral es:

$$\frac{M(s)}{E(s)} = \frac{K_i}{s}.$$

4.3.4 Control Proporcional e integral (PI).

La acción de control proporcional e integral queda definida de la siguiente manera:

$$M(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt.$$

Visto en funciones de transferencia:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right),$$

donde K_p representa la sensibilidad proporcional o ganancia y T_i el tiempo integral. Tanto K_p como T_i son regulables. El tiempo integral regula la acción de control integral, mientras que una modificación en K_p afecta tanto a la parte integral como a la proporcional de la acción de control. A la inversa del tiempo integral T_i se le llama frecuencia de reposición. La frecuencia de reposición es el número de veces por minuto (o por unidad de tiempo) que se duplica la parte proporcional de la acción de control. La frecuencia de repetición se mide en términos de repeticiones por minuto (o por unidad de tiempo). El diagrama de bloques de un control de este tipo es:

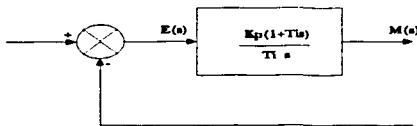


Fig. 4.3 Control PI.

Las siguientes gráficas ilustran el comportamiento del control integral cuando la señal de error es un escalón. Se compara con la respuesta de un proporcional en la línea discontinua:

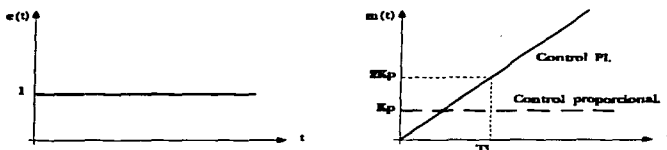


Fig. 4.4. Comportamiento del controlador PI si $e(t)$ es un escalón.

4.3.5 Control proporcional derivativo (PD).

La acción de control está definida por:

$$m(t) = K_p e(t) + K_p T_d \frac{\delta e(t)}{\delta t},$$

y la función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p (1 + T_d s),$$

donde K_p es la sensibilidad proporcional y T_d es el tiempo derivativo. Tanto K_p como T_d son regulables. La acción de control derivativa, a veces denominada control de velocidad, es cuando el valor de salida del control es proporcional a la velocidad de variación de la señal de error actuante. El tiempo derivativo T_d es el intervalo de tiempo en el que la acción de velocidad se adelanta al efecto de acción proporcional. El diagrama de bloques de este control es:

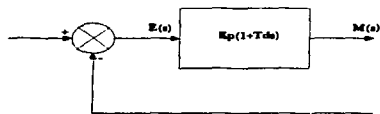


Fig. 4.5. Control PD.

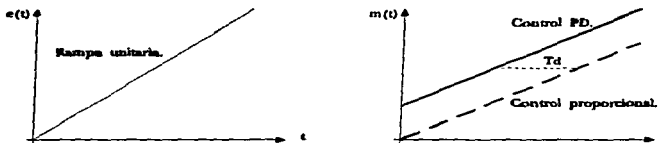


Fig. 4.6. Comparación de controlador proporcional y PD.

El control proporcional derivativo tiene la ventaja de ser anticipador, pero tiene la desventaja de ser extremadamente sensible al ruido. No existe un control derivativo puro, pues éste sólo responde a transitorios.

4.3.6 Control proporcional, integral y derivativo (PID).

A la combinación de los efectos de un control proporcional, derivativo e integral se le denomina control proporcional derivativo e integral. Esta combinación tiene las ventajas de cada una de las acciones de control individuales. La ecuación de control es:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^t e(t) dt.$$

La función de transferencia:

$$\frac{M(s)}{E(s)} = K_p \left(1 + T_d s + \frac{1}{T_i s} \right),$$

donde K_p representa la sensibilidad proporcional, T_d el tiempo derivativo y T_i el tiempo integral.

El diagrama de bloques de este control y el comportamiento para una rampa es:

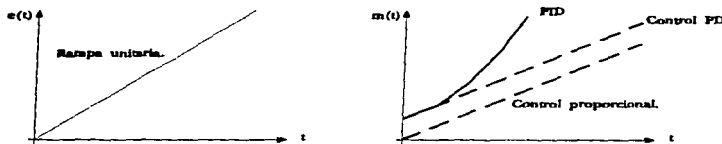


Fig. 4.7

4.4 Observaciones al sistema medidor de permeabilidad.

La descripción matemática de las características dinámicas de un sistema se denomina modelo matemático. El primer paso en el análisis de un sistema dinámico, consecuentemente, el primer paso para obtener un modelo de control adecuado, es elaborar su modelo matemático. Es la parte más importante y quizás la más difícil de todo el análisis.

Normalmente, conforme se avanza en el desarrollo de un modelo, se llega a un compromiso entre la simplicidad del mismo y la exactitud del resultado de su análisis. Si no es de vital importancia, es preferible obtener solamente un modelo razonablemente simplificado. La rapidez de los sistemas computacionales permiten, en muchos casos, utilizar tantas ecuaciones se necesiten para describir al sistema completo.

El medidor de permeabilidad puede ser visto como un conjunto de subsistemas:

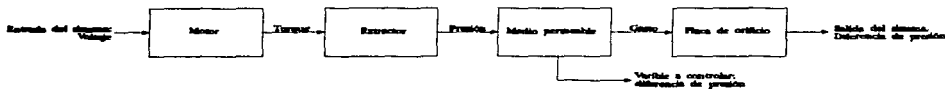


Fig. 4.8. Diagrama de bloques del sistema medidor.

- 1.- El sistema formado por el motor. **Entrada:** voltaje, **salida:** torque.
- 2.- El sistema torque-extractor. **Entrada:** torque, **salida:** presión neumática.
- 3.- La muestra de tela. **Entrada:** presión, **salida:** gasto.
- 4.- El orificio. **Entrada:** gasto, **salida:** diferencia de presión.

Notar que el sistema formado por el motor y su turbina puede almacenar energía (inercia), mientras que el textil y la placa de orificio son elementos que producen pérdidas por fricción.

La placa de orificio es un dispositivo medidor, sin embargo influye en el sistema completo de manera determinante (al provocar una caída de presión). Es un aspecto interesante en este sistema: el dispositivo medidor afecta considerablemente al sistema en conjunto².

El control utiliza como señal de realimentación la diferencia de presión en el medio permeable, es decir, la variable que se pretende controlar es la diferencia de presión en el medio permeable (el textil).

Dentro de estos subsistemas existen dificultades para determinar un modelo matemático que pueda ser utilizado para describir el conjunto y son:

- 1.- No se tiene la información técnica que describa al motor eléctrico.
- 2.- No se tiene la información técnica del extractor.
- 3.- No se sabe con exactitud cómo se comporta la tela.

² Su influencia es tal que es el que ha determinado el tipo de extractor y ventilador que se utiliza. Su influencia se debe a que es un método de medición extremadamente ineficiente, pues la pérdida de presión total está en el orden del 90% de la caída máxima de presión (vena contracta).

4.4.1 El motor.

Dadas las dificultades para encontrar un modelo matemático para el motor, se optó por utilizar un controlador de motores universales que tiene la característica principal de hacer al sistema lineal, es decir, el motor entrega de manera lineal revoluciones en función de una señal de voltaje. El fabricante asegura un comportamiento lineal para el control. Esto no significa que el motor muestre un comportamiento a transitorios de primer orden, de hecho puede esperarse que sea de primer o segundo orden.

Una vez definida la variable de entrada al sistema, se puede realizar experimentación para determinar el comportamiento del sistema en su totalidad

4.4.2 El textil.

Una aproximación matemática al comportamiento del textil la ofrece las ecuaciones de Darcy. Las ecuaciones que definen el comportamiento de un fluido a través de un medio poroso son:

$$q = -K \cdot \text{grad}(\varphi); \quad q = Q/A$$

donde K describe al medio poroso y al fluido. Y

$$q = -(k\sigma / \mu) \cdot \text{grad}(\varphi)$$

donde k describe solamente a la matriz porosa.

Como se mencionó en el primer capítulo, la dificultad de aplicar las ecuaciones de Darcy radica en definir al gradiente de presión. Se puede tratar de entender al gradiente como la caída de presión a través del espesor del textil, sin embargo, conviene más entenderlo como la caída entre ambas caras del textil, pues interesa la caída total y no el comportamiento de la presión a través del medio:

$$\text{grad}(\varphi) = \frac{\Delta P}{L}$$

El medio poroso del textil tiene un comportamiento de gasto no lineal al variar la presión. El gradiente no se comporta linealmente al variar la presión³, y el objetivo del control es estabilizar la presión hasta alcanzar estado estable (el gradiente se vuelve función de la posición únicamente). Dado que no existen elementos almacenadores de energía, se puede esperar que la reacción a transitorios no sea de importancia. El problema del control radica reaccionar de manera adecuada cuando se alcancen zonas de la curva que hagan al control extremadamente sensible a pequeñas variaciones de presión. Solamente determinando los parámetros de operación se puede analizar este problema.

Es por esto que se requiere obtener información a través de la experimentación para poder encontrar un patrón de comportamiento del gasto cuando se varía la presión. El resultado reportado en el experimento (relación presión-gasto) es:

³ Ver teoría y prueba realizada en el capítulo 1, pag. 14.

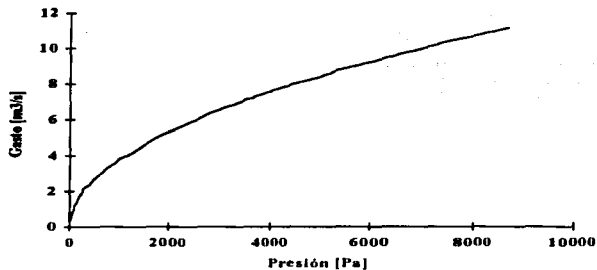


Fig. 4.9. Gasto en una franela.

Esta curva es una prueba realizada a un textil de algodón (franela) y muestra un comportamiento exponencial.

4.4.3 La placa de orificio.

Es el único subsistema del cual se tiene la información matemática aproximada, la ecuación que describe a la placa de orificio es:

$$Q = K_f A_2 \sqrt{\frac{2(p_1 - p_2)}{\rho}}$$

4.5 Experimentación como método para encontrar un modelo de control.

Por lo anteriormente expuesto conviene caracterizar a todo el sistema medidor como una caja negra y no como un conjunto de subsistemas. Para ello se requiere realizar pruebas y tratar de encontrar una aproximación al modelo matemático para el sistema en conjunto.

El procedimiento para encontrar un modelo matemático debe definir cuál es la entrada del sistema en lazo abierto y qué variable es la que se va a utilizar para el control.

1. **Entrada al sistema;** como se mencionó anteriormente, la entrada al sistema es una señal de voltaje que se aplica al controlador del motor.
2. **Variable a controlar:** la diferencia de presión medida en el medio permeable, ésta es la variable a controlar. La norma establece media pulgada de agua para el medio permeable, por lo que el control debe alcanzar y mantener esta diferencia de presiones.

El sistema debe alcanzar la presión de media pulgada de agua en el medio permeable, a partir de entonces el sistema se comportará a régimen permanente. De esta consideración se deduce que lo que realmente importa en este sistema es la respuesta desde el tiempo cero hasta alcanzar su valor en régimen permanente. Una vez alcanzado este valor no hay nada que afecte o perturbe al sistema. Lo que interesa,

entonces, en el diseño del controlador es determinar los parámetros del modelo de control basado en un criterio de respuesta a un transitorio único: el proceso de arranque.

La función de un control retroalimentado es el asegurar que el sistema en lazo cerrado tiene una respuesta dinámica y en estado estable de acuerdo a las necesidades de la aplicación. Generalmente, en un sistema de control, se desean satisfacer idealmente los siguientes criterios:

1. El sistema en lazo cerrado debe ser estable.
2. Los efectos de los disturbios deben ser minimizados.
3. Respuestas rápidas y suaves a cambios de puntos de operación.
4. Eliminar desplazamientos, corrimientos ("offsets").
5. Evitar la acción excesiva del control (que se puede manifestar en oscilaciones).
6. Robustez, es decir, respuesta satisfactoria a cambios en las condiciones del proceso y absorberencia de errores en el modelado del proceso.

En el proceso o sistema que se pretende controlar, existen varias observaciones referentes a lo que se señaló anteriormente:

1. Se busca estabilidad para alcanzar la diferencia de presiones antes y después del tejido.
2. No existen elementos que provoquen disturbios, siempre que se realicen las pruebas bajo las condiciones que establece la norma.
3. La respuesta no necesariamente tiene que ser rápida, pero sí estable y efectiva.
4. Se desea eliminar los efectos de una aproximación matemática deficiente al modelo del proceso, además de que el control debe asumir variaciones en la carga (gasto), y llegar al mismo punto de operación ("set-point"): media pulgada de agua.

En base a estas observaciones y objetivos de control, se puede definir el tipo de control que se desea: un control que establezca una presión de media pulgada de agua entre ambas caras del textil, la acción del control debe llevar al sistema a alcanzar el régimen permanente y mantenerse en tal estado; no se esperan disturbios una vez iniciado el proceso de control y la carga no varía una vez iniciado el proceso; el controlador debe responder a los distintos tipos de telas, que se manifiestan como una variación en el gasto (carga).

De todo esto se deduce que el primer paso es determinar un modelo matemático en base a las observaciones del sistema. Se sugiere como primera aproximación a un sistema de primer orden con un tiempo muerto.

4.5.1 Modelado de procesos dinámicos usando las curvas de reacción de un proceso.

El modelo de un sistema puede ser mecanístico o simplemente correlativo. En el caso de un sistema mecanístico, está basado en las ecuaciones matemáticas que describen el sistema físico o químico que será operado en el proceso. Un modelo mecanístico correcto no sólo debe coincidir en la información o datos medidos, sino que también debe prever correctamente el comportamiento del proceso en condiciones más allá de las que el proceso fue calibrado (pues está regido por las ecuaciones exactas del proceso). En contraste, en un modelo correlativo los datos experimentales se ajustan al objetivo del proceso o son correctos a un nivel de ingeniería (el modelo matemático no lo conforman las ecuaciones exactas del proceso, sino una aproximación).

En el diseño de sistemas de control, se nos presenta una ligera diferencia en el modelado de un sistema. El diseño del control está basado más en el criterio de una operación fuera de estado estable que en estado estable; y el objetivo en el modelado es reproducir con el modelo obtenido la respuesta a transitorios, la respuesta del proceso mismo o la respuesta del sistema en lazo abierto. Cuando se logra un modelo que reproduce la respuesta a transitorios de manera satisfactoria, se está en condiciones de sustituir el modelo mecanístico por el nuevo modelo, el correlativo. Con este procedimiento se está en condiciones de reemplazar sofisticadas ecuaciones que describen un proceso, cuya solución matemática es extremadamente compleja o simplemente no se conoce (es el caso de muchos fenómenos físicos y químicos). Debe quedar claro que el modelo no es exacto, pues su función de transferencia real ha sido reemplazada; pero para fines de ingeniería puede ser lo suficientemente exacta.

Es muy común modelar un proceso o sistema de manera aislada o como un proceso en lazo abierto formado por conjunto de elementos en serie; en cualquier caso, la respuesta a transitorios medida de manera experimental se denomina Curva de Reacción. La función que se aplica a este proceso puede ser cualquier, siempre que se pueda generar; sin embargo, es muy común utilizar un escalón, pues es la función más fácil de reproducir. En procesos que involucran flujo es conveniente utilizar un rápido incremento en la presión en la entrada de la corriente (respuesta a impulso). Esto se debe a que normalmente la duración de este impulso es considerablemente menor que las constantes de tiempo del proceso. Para este trabajo existe la dificultad de generar este tipo de excitación de manera experimental⁴. Por ello se sugiere la respuesta a escalón, que es fácilmente aplicable al motor.

Este método de modelado puede ser en el dominio del tiempo o bien en el dominio de la frecuencia, aplicando funciones sinusoidales puras o bien por el método de pruebas de pulso.

4.5.2 Modelo de primer orden para un proceso auto-regulable.

Un sistema es no auto-regulable cuando la respuesta no tiende a un valor determinado, o bien, se satura. El sistema o proceso que tiende a un valor es un sistema o proceso auto-regulable, que es un valor a régimen permanente. Es de esperarse que el sistema medidor de permeabilidad tenga este último

⁴Además debe recordarse que se pretende incluir en todo el sistema al motor, quien es realmente el que actúa como elemento almacenador de energía (efectos inerciales).

comportamiento. Esto es precisamente debido al efecto del controlador de revoluciones del motor en lazo cerrado (mantiene las revoluciones). Si no existiera este control se observaría que el sistema tiende a un valor de manera inestable⁵.

Los modelos dinámicos que convenientemente se utilizan en el método de correlación a través de las curvas de reacción son los que se han encontrado mecánicamente para sistemas dinámicos elementales; éstos son: sistemas de primer orden, segundo y de mayor orden con parámetros concentrados. Estos sistemas pueden ser vistos como un conjunto discreto de capacitancias y resistencias.

Por su simplicidad, el sistema de primer orden es el más utilizado, especialmente cuando una correlación burda es adecuada y suficiente para una respuesta a transitorios.

Para la caracterización de un sistema visto como una caja negra y en lazo abierto, se sigue la técnica de curvas de reacción del proceso⁶. Este método está basado en un sólo experimento que es hecho de manera manual. Se alimenta al proceso con una ligera variación escalón y se mide la respuesta en el tiempo. Esta respuesta a escalón es comúnmente llamada curva de reacción del proceso. Está caracterizada por dos parámetros: S , que es la pendiente de la tangente en el punto de inflexión, y θ que es el tiempo en que la tangente interseca al eje del tiempo.

En la siguiente gráfica se muestran dos tipos de respuesta para variaciones tipo impulso y tipo escalón:

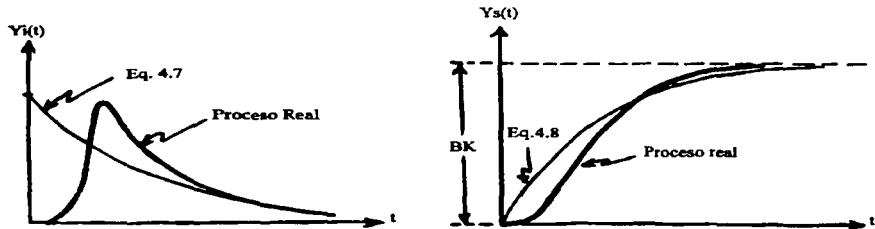


Fig. 4.10. Respuestas a) impulso, y b) escalón

La curva gruesa es la respuesta de un sistema real, mientras que la curva delgada es la respuesta de un sistema de primer orden. Las ecuaciones que describen a este sistema de primer orden se reescriben posteriormente. En esta gráfica puede observarse la importancia de tener un modelo correlativo de primer orden que se aproxime a la respuesta real del sistema.

⁵Fundamentalmente se debe a las características del motor universal.

⁶Ziegler, J. G., y N. B. Nichols, "Optimum Settings for Automatic Controllers", ASME 1942.

La ecuación diferencial de primer que describe a un sistema en lazo abierto es:

$$\left(\frac{dy}{dt}\right) + y = Kx.$$

Donde y es la variable de salida o respuesta del sistema y x es la función aplicada al sistema. La transformación de Laplace a este sistema de primer orden es:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K}{\tau s + 1}.$$

Substituyendo la función $X(s)$ por una función impulso y una función escalón:

$$X(s) = L[A\delta(t)] = A. \text{ Impulso de magnitud } A.$$

$$X(s) = L[Bu(t)] = \frac{B}{s} \text{ Escalón de magnitud } B.$$

Así, la respuesta a impulso y escalón en función del tiempo para un sistema de primer orden son:

$$y_1(t) = \left(\frac{AK}{\tau}\right)e^{-\frac{t}{\tau}}. \text{ Respuesta a impulso (ec. 4.1).}$$

$$y_2(t) = BK(1 - e^{-\frac{t}{\tau}}). \text{ Respuesta a escalón (ec. 4.2).}$$

Para la correlación de un sistema utilizando estas ecuaciones de primer orden se debe tener presente y perfectamente medido la presencia de tiempos muertos en el sistema. Las ecuaciones 4.1 y 4.2 tienen su representación gráfica en la figura 4.10.

Para aproximar la respuesta de un sistema de primer orden a la respuesta de un sistema real como el que se muestra en la figura 4.10, se introduce el tiempo muerto θ . Para la función $G(S)$:

$$G(s) = (Ke^{-\theta s}) / (\tau s + 1)$$

La respuesta a impulso y escaló con el tiempo muerto será:

$$y_1(t) = \left(\frac{AK}{\tau}\right)e^{-(t-\theta)/\tau}, \text{ respuesta impulso (ec. 4.3).}$$

$$y_2(t) = BK(1 - e^{-(t-\theta)/\tau}), \text{ respuesta escalón (ec. 4.4).}$$

Estas ecuaciones deben observar que $y(t) = 0$ si $t < \theta$.

Lo que resta ahora es determinar los parámetros de las ecuaciones 4.3 y 4.4 con valores experimentales del sistema, es decir, hacer la correlación. Se deben determinar experimentalmente los valores de K , θ y τ .

Determinación de K .

Este parámetro es la ganancia en estado estable del sistema y puede ser determinado de la curva de reacción del proceso. K se calcula como la razón del valor en estado estable de $y(t)$ para un valor de $x(t)$. La figura ilustra la forma de determinarlo:

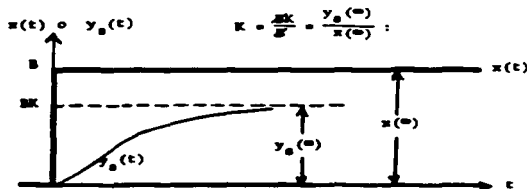


Fig. 4.11. Cálculo de K.

Esta curva es la reacción del proceso a un escalón.

Para la determinación de K utilizando un impulso se requieren seguir ciertas consideraciones. La ecuación 4.3 predice un valor máximo $y_{max} = AK/\tau$, que ocurre cuando $t = \theta$; de esta manera se determinaría experimentalmente los parámetros A, K y τ . Este procedimiento no es recomendable, pues el valor máximo de la respuesta experimental no corresponde al predicho por la ecuación 4.3, a menos, claro está, que el sistema sea de primer orden mecánico. Un método más efectivo del cálculo de los parámetros es la integración numérica de la ecuación 4.3:

$$A_c = \int_0^{\infty} (AK/\tau) e^{-(t-\theta)/\tau} dt.$$

De esta forma, si se tiene el área de la curva experimental puede determinarse el valor de K . La desventaja de este método es obvia: la magnitud A del impulso no siempre es experimentalmente accesible o difícil por la rapidez del experimento, y difícilmente controlable.

Determinación de los parámetros τ y θ .

El método clásico para la determinación de estos parámetros es a través de la evaluación de la respuesta a impulso o a escalón al 63.2% de la respuesta:

$$y_i(t + \theta) = \left(\frac{AK}{\tau} \right) e^{-1} = 0.368(AK/\tau), \text{ respuesta impulso (ec. 4.5).}$$

$$y_e(t + \theta) = BK(1 - e^{-1}) = 0.632BK, \text{ respuesta escalón (ec. 4.6).}$$

Estos puntos se muestran en la siguiente gráfica:

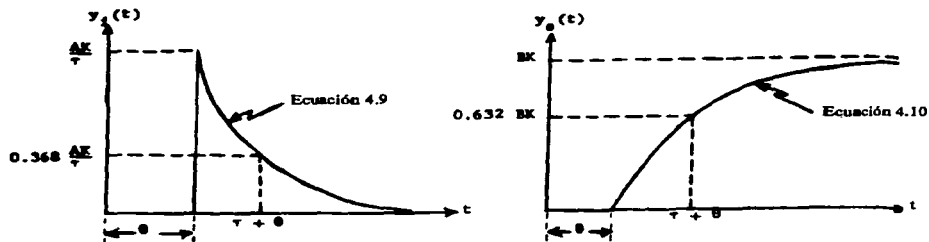


Fig. 4.12. Cálculo de τ y θ a partir del 63.2%.

El tiempo $(\tau + \theta)$ es medido fácilmente de las curvas de respuesta. Para obtener θ se traza una tangente al punto de inflexión, como lo muestra la siguiente figura:

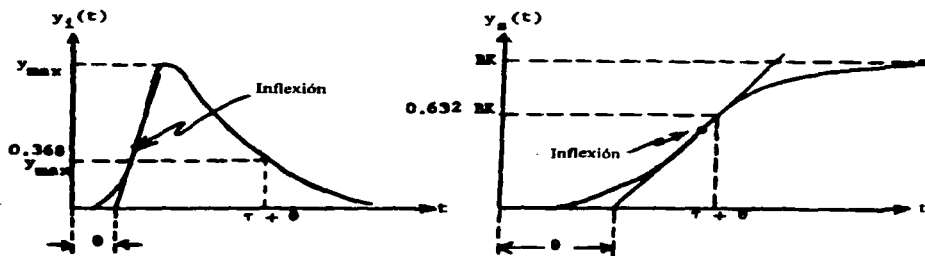


Fig. 4.13. Cálculo de τ y θ .

Una de las desventajas de esta forma de determinación de parámetros, es que se utiliza una porción muy pequeña de la curva. Existen otros métodos que involucran más información de las curvas de respuesta y hacen una linealización de las curvas. Estos métodos no se discuten aquí⁷.

4.5.3. Modelo de segundo orden para sistemas auto-regulables.

De la misma manera como se procedió con el sistema de primer orden, se describen las ecuaciones para el sistema de segundo orden y sus respuestas a excitación impulso y excitación escalón.

$${}^2y'' + 2\tau\zeta y' + y = Kx.$$

⁷Puede encontrarse el método de linealización en "Black-Box Dynamic Process Modeling Using Process Reaction Curves", W. C. Clemens, American Institute of Chemical Engineering.

es la ecuación que describe un sistema de segundo orden; el factor ζ se le denomina factor de amortiguamiento. Si este factor ζ es menor que 1 se dice que el sistema es subamortiguado o de respuesta oscilatoria y sus respuestas son:

$$y_i = \frac{AK}{\tau \cdot \sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\tau} t} \sin\left(\frac{\sqrt{1-\zeta^2}}{\tau} t\right) \quad (\text{respuesta a impulso}) \quad (\text{ec. 4.5}).$$

$$y_i = BK \left[1 - \left[\frac{1}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\tau} t} \sin\left(\frac{\sqrt{1-\zeta^2}}{\tau} t + \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}\right) \right] \right] \quad (\text{respuesta a escalón}) \quad (\text{ec. 4.6}).$$

Si $\zeta \geq 1$ el sistema puede ser convenientemente representado como dos sistemas de primer orden en serie. Si el factor $\zeta=1$ se dice que el sistema es críticamente amortiguado y tiene una simple constante de tiempo:

$${}^2y'' + 2\tau y' + y = Kx.$$

Las soluciones a este sistema son:

$$y_i = \frac{AK}{\tau^2} t e^{-t/\tau} \quad (\text{respuesta a impulso}) \quad (\text{ec. 4.7}).$$

$$y_s = BK \left[1 - (1 + t/\tau) e^{-t/\tau} \right] \quad (\text{respuesta a escalón}) \quad (\text{ec. 4.8}).$$

Si el factor $\zeta > 1$, el sistema es sobreamortiguado y posee dos constantes de tiempo distintas:

$${}_{1\tau_2}y'' + (\tau_1 + \tau_2)y' + y = Kx.$$

y las soluciones son:

$$y_i = \frac{AK}{\tau_1 + \tau_2} (e^{-t/\tau_2} - e^{-t/\tau_1}) \quad (\text{respuesta a impulso}) \quad (\text{ec. 4.9}).$$

$$y_s = BK \left[1 / (\tau_2 - \tau_1) \cdot (\tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}) \right] \quad (\text{respuesta a escalón}) \quad (\text{ec. 4.10}).$$

Los sistemas descritos por estas ecuaciones son mecánicos, y no se ha involucrado ningún tiempo de retardo θ para hacer la correlación con un sistema real. Si se quiere incluir el tiempo de retardo, entonces habrá que reemplazar t por $t-\theta$.

Curvas de respuesta para un sistema subamortiguado.

De la ecuación 4.6 se obtienen las siguientes expresiones:

$$y_s(\infty) = BK,$$

$$\text{sobrepaso} = \frac{C}{y_s(\infty)} = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (\text{ec. 4.11}),$$

$$\text{factor de atenuación} = \frac{C}{E} = \frac{D}{F} = (\text{sobrepaso})^2 \quad (\text{ec. 4.12}),$$

$$\text{período} = T = \frac{2\pi\tau}{\sqrt{1-\zeta}} \quad (\text{ec. 4.13}).$$

Como se hizo para el caso de primer orden, se puede obtener la ganancia en estado estable con la división experimental del valor de $y(\infty)$ por la magnitud del escalón B . Si se mide la distancia C y el período de oscilación T , se pueden substituir en las ecuaciones 4.12 y 4.13 y obtener dos ecuaciones con dos incógnitas, τ y ζ . Resolviendo sistemas simultáneos y midiendo el tiempo muerto, es fácilmente determinado el sistema real relativo. La siguiente figura muestra los puntos de medición, incluyendo el tiempo muerto:

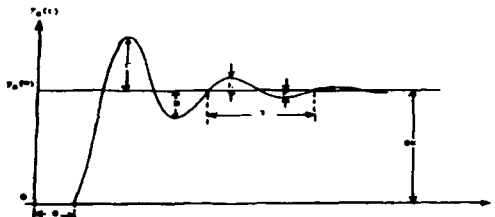


Fig. 4.14. Respuesta a escalón de un sistema subamortiguado.

Respuesta críticamente amortiguada.

Los parámetros de un sistema críticamente amortiguado pueden ser fácilmente determinados utilizando la curva de respuesta y el método de primer orden. La figura siguiente muestra la respuesta de un sistema críticamente amortiguado:

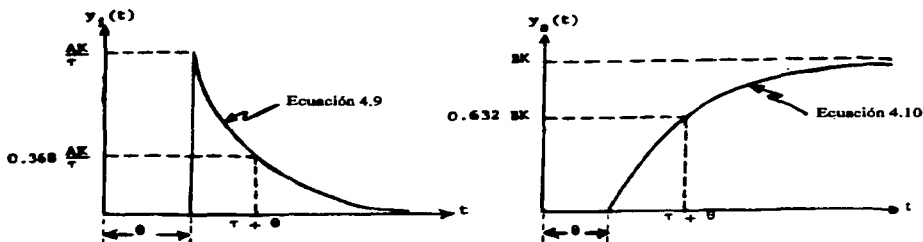


Fig. 4.15. Método para determinar parámetros de un sistema críticamente amortiguado.

Respuesta a impulso: el pico ocurre cuanto $t=\tau$; en ese punto $y_1(\max)=0.368AK/\tau$. El punto de inflexión ocurre en $t=2\tau$, donde $y_1(2\tau)=0.271AK/\tau$. El área por debajo de la curva es igual a AK .

Respuesta a escalón: el punto de inflexión ocurre cuando $t = \tau$, donde $y_1(\tau) = 0.264BK$. El valor en estado estable $y(\infty) = BK$.

El cálculo del tiempo muerto θ puede ser fácilmente determinado experimentalmente, como se muestra en la gráfica anterior.

Respuesta sobreamortiguada.

El método clásico para la determinación de parámetros de un sistema sobreamortiguado es el Oldenbourg-Sartorius. Con éste se pueden determinar τ_1 y τ_2 a partir de la respuesta a escalón. También permite la determinación del tiempo muerto θ . La siguiente gráfica muestra las mediciones:

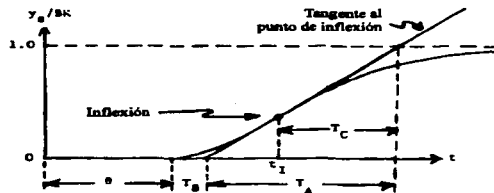


Fig. 4.16. Método Oldenbourg-Sartorius.

De la ecuación 4.12 se encuentra que:

$$T_A = \tau_1 \left(\frac{\tau_2}{\tau_1} \right)^{\tau_2 / (\tau_2 - \tau_1)} \quad (\text{ec. 4.14}), \text{ y } T_C = \tau_1 + \tau_2 \quad (\text{ec. 4.15}).$$

Trazando la tangente al punto de inflexión y siguiendo una solución de prueba y error para las ecuaciones 4.13 y 4.14 se obtienen los valores para τ_1 y τ_2 .

El tiempo muerto es calculado como $t_f + T_C - T_A - T_B$.

Puede determinarse la ganancia del sistema de la manera usual, la división del valor en estado estable y el valor del escalón.

Existen otros métodos para determinar los parámetros del sistema, éstos facilitan la tarea del cálculo reiterativo planteado⁸, basados en la medición en punto específicos de la curva de respuesta.

4.6 Curvas de respuesta y experimentación con el aparato medidor.

Una vez expuestos los métodos de aproximación de un sistema visto como una caja negra, el paso siguiente es observar la respuesta del aparato medidor para varios tipos de muestras textiles. A continuación se muestran varios resultados experimentales con distintos tipos de telas. El objeto de esta

⁸Estos son: método propuesto en "Digital Computer Process Control", Smith, C. L., editorial Intext, Scranton Pa., 1972; y método propuesto en "Process Control", Harriot, P., editorial McGraw-Hill, Nueva York, 1964.

experimentación es comparar la respuesta del aparato medidor con las respuestas de los sistemas de primer orden y segundo orden anteriormente expuestas. El primer intento de aproximación sugeriría un sistema de primer orden, por su simplicidad; pero, dependiendo de las curvas de respuesta que se muestran, determinará qué tipo de modelado se requiere.

Las pruebas se realizaron con un escalón, es decir, se están obteniendo las curvas de respuesta a escalón. Esta es la señal más fácil de reproducir en laboratorio y la que involucra a todos los elementos que conforman al aparato medidor.

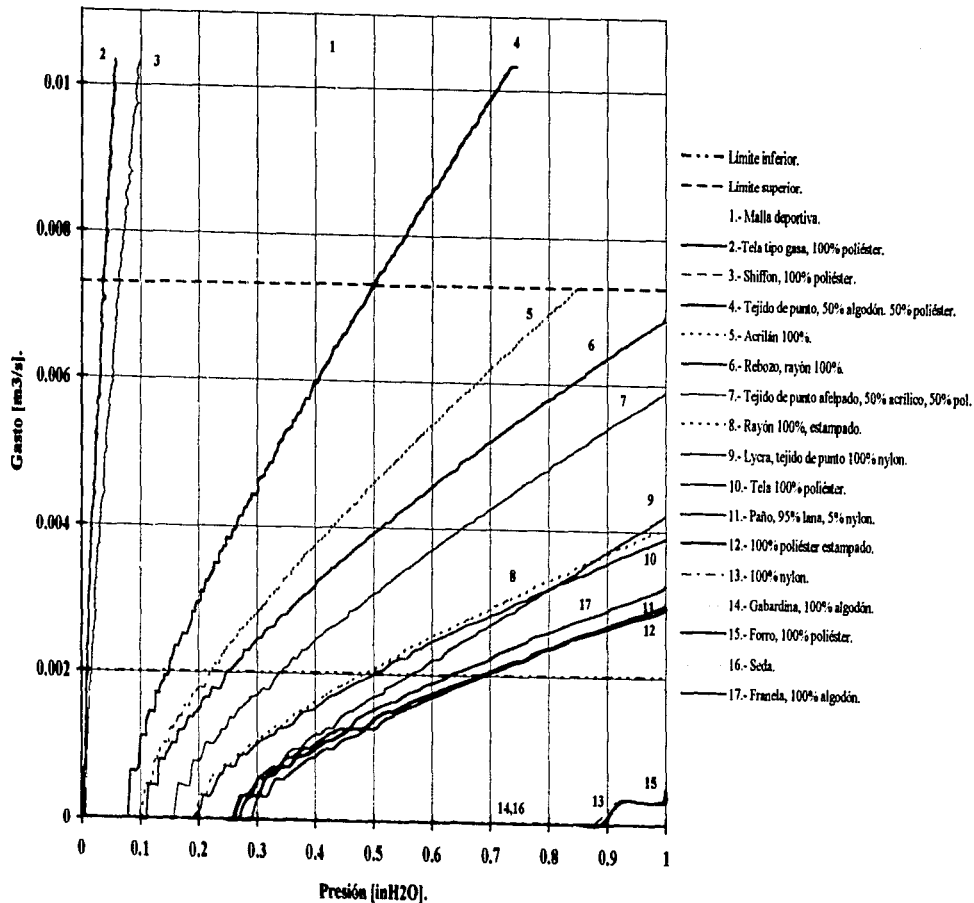
Debe recordarse que el aparato medidor sólo abarca un rango muy restringido del universo de telas que existen. En el capítulo 2 se determinaron los límites de operación de este aparato, los cuales fueron:

	Permeabilidad $K [m^3/s/m^2]$	Coefficiente de flujo K_F	N_{Re}	Gasto $[m^3/s]$ Q_c	P2-P3 [Pa]. Placa de orificio.	P1-P2 [Pa]. Textil.
Límite inferior.	0.4987	0.6094	2300	0.002	269.86	124.42
Límite superior teórico.	1.4961	0.6022	6899.63	0.006	2482.31	124.42
Nuevo límite.	1.8181	0.6020	8384.62	0.0073	3732.60	124.42

La permeabilidad mínima a medir es de 0.49 $m^3/s/m^2$, y la máxima es de 1.8181 $m^3/s/m^2$; a las que les corresponden gastos de 0.002 m^3/s y 0.0073 m^3/s . Con estas limitantes, se analizaron 17 tipos distintos de telas, con la finalidad de encontrar textiles aptos para este aparato y determinar un modelo de control adecuado a este conjunto de textiles. Los resultados numéricos y gráficas obtenidas en los experimentos se muestran en el apéndice D. En la siguiente hoja se muestra una gráfica con los resultados de estos datos. Las telas probadas y su composición fueron:

- 1.- Malla deportiva, 100% poliéster.
- 2.- Tela tipo gasa, 100% poliéster.
- 3.- Shiffon, 100% poliéster.
- 4.- Tejido de punto, 50% algodón y 50% poliéster.
- 5.- Acrilán 100%.
- 6.- Rayón 100%
- 7.- Tejido de punto afelpado, 95% acrílico y 50% poliéster.
- 8.- Tela estampada, 100% rayón.
- 9.- Lycra, tejido de punto 100% nylon.
- 10.- Tela 100% poliéster.
- 11.- Paño, 95% lana y 5% nylon.
- 12.- Tela 100% poliéster estampada.

Comparativo de valores de gasto vs. presión para cada tela.



- 13.- Tela 100% nylon.
 14.- Gabardina, 100% algodón.
 15.- Forro, 100% poliéster.
 16.- Seda 100%.
 17.- Francla, 100% algodón.

El intervalo de operación del aparato medidor está delimitado por las dos líneas discontinuas gruesas. En base a este intervalo, se descartan los siguientes tipos de telas por ser demasiado abiertas que no alcanzan a levantar la presión establecida por la norma, ½ pulgada de agua: malla deportiva, tela 100% poliéster tipo gasa, y shiffon (100% poliéster); por estar fuera del rango de medición del aparato: las anteriores y tejido de punto (50% algodón y 50% poliéster). Se descartan también, por no permitir el flujo suficiente para asegurar un flujo turbulento en el aparato medidor (consecuentemente las lecturas no son confiables): lycra (100% nylon), francla (100% algodón), paño (95% lana y 5% nylon), poliéster 100% estampado, forro (100% poliéster), gabardina (100% algodón), tela de 100% nylon tipo rompevientos, tela 100% poliéster estampada y seda.

Los tipos de tela que pueden ser probados en el aparato son: acrílan 100%, rebozo (100% rayón), tejido de punto afelpado, 100% poliéster, 100% rayón estampado.

La gráfica muestra el comportamiento de solamente estas 5 telas:

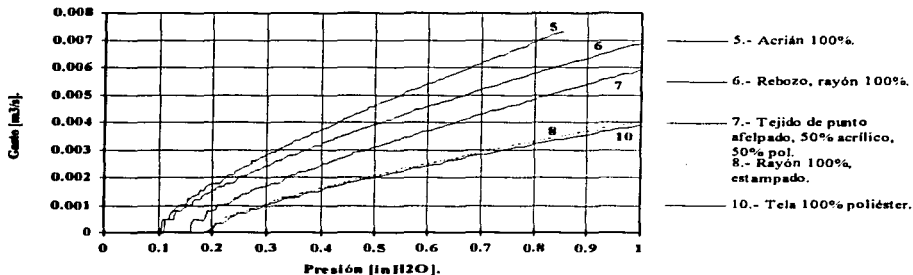


Fig. 4.18 Curvas para las 5 telas seleccionadas.

En las curvas de respuesta a escalón del aparato medidor del apéndice D se puede observar el comportamiento del aparato medidor. Las curvas tienen un marcado sobrepaso, lo que lleva a pensar en una aproximación de segundo orden. El efecto de segundo orden es debido al sistema de lazo cerrado del controlador del motor. Esto se debe fundamentalmente a la velocidad de respuesta del tacómetro. De estas curvas se deduce que el modelo matemático correlativo del aparato medidor debe ser aproximado a un

sistema de segundo orden.

4.7 El modelo matemático correlativo.

Una vez definido el subconjunto de telas con el que el aparato puede trabajar, se comienza con la correlación del modelo matemático. De acuerdo a lo apuntado en los anteriores apartados, la primera aproximación al modelo es un sistema de segundo orden. La ecuación de un sistema de segundo orden es (cc. 4.6):

$$y_i = BK \left[1 - \left[\frac{1}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\tau} t} \sin \left(\frac{\sqrt{1-\zeta^2}}{\tau} t + \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) \right] \right], \text{ respuesta escalón.}$$

Esta ecuación requiere de tres parámetros que deben ser determinados experimentalmente: K, que es la ganancia del sistema que responde a un escalón cuando $t \rightarrow \infty$ y las constantes del sistema de segundo orden ζ y τ .

Determinación de K.

Para la determinación de K, se aplicó un escalón y se esperó el tiempo necesario hasta que la presión en la placa de orificio de estableciera. Este procedimiento se hizo para varios valores de escalón, el resultado fue que los valores de K varían para una misma tela. Este resultado era de esperarse, pues al ser la permeabilidad una función no lineal de la presión aplicada, los valores de presión medidos en la tela no varían linealmente respecto al valor del escalón. Esta prueba se hizo para las 17 telas analizadas en el apartado anterior, los resultados de las pruebas aplicadas a las 17 telas se muestran en el apéndice D. Los resultados para las 5 telas susceptibles de analizarse en este aparato medidor son:

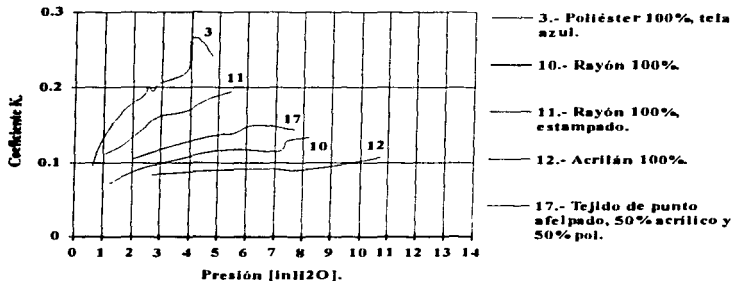


Fig. 4.19 Constante K del proceso.

De estas pruebas, puede verse que la constante K (sus unidades son $[\text{inH}_2\text{O}/\text{V}]$) del sistema de primer orden varía de 0.06 $[\text{inH}_2\text{O}/\text{V}]$ a 0.256 $[\text{inH}_2\text{O}/\text{V}]$. Esto puede repercutir en la estabilidad del sistema. Al confrontar con los distintos valores de K para las distintas telas surge la idea de realizar un control dinámico, es decir, que se seleccione la K adecuada de acuerdo a varias regiones de operación. El impacto de suponer una constante K invariante se desconoce hasta elaborar la experimentación con el modelo correlativo, al igual que la eficacia de suponer un modelo de segundo orden subamortiguado. En el apéndice D se muestran las mediciones de K para las cinco telas seleccionadas.

Determinación de ζ y τ .

De acuerdo a lo establecido por el método de curvas de respuesta, la constantes ζ y τ se determinan a partir de la curva de respuesta de un sistema de segundo orden:

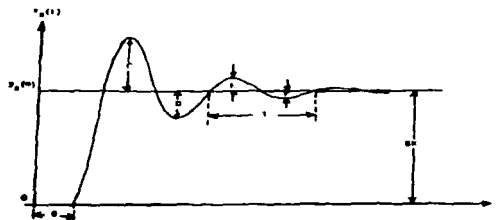


Fig. 4.20 Respuesta escalón de un sistema de segundo orden.

De acuerdo a este procedimiento, se encontraron los valores de θ (retardo), C (sobrepaso), D (distancia al primer valle), E (distancia a la segunda cresta), F (distancia al segundo valle), T (período) y t_l (tiempo de levantamiento (tiempo en que se cruza por primera vez el valor a régimen permanente) para los 5 tipos de telas.

Para la determinación de las características del modelo de segundo orden se analizaron tres casos para cada tela: una prueba con el escalón mínimo necesario para provocar flujo turbulento, una prueba con el escalón necesario para producir $1/2$ $[\text{inH}_2\text{O}]$ en el textil y finalmente una prueba para provocar la máxima caída de presión medible en el textil 1 $[\text{inH}_2\text{O}]$. Es decir, se elaboraron las pruebas para abarcar todo el rango de operación: inferior, medio y máximo. En casi todos los casos se realizaron estas pruebas para dos muestreos de tiempo, 5 segundos y 20 segundos, con la finalidad de observar el valor a régimen permanente. Estas gráficas están incluidas en el apéndice D en las pruebas de telas correspondientes.

En la siguiente tabla ("Información experimental") se muestran los valores experimentales medidos.

Información experimental.

Tipo de fertil.	n/dó	Excusión [V.]	Salida [mH2O]	Kp.	C [V.]	D [V.]	E [V.]	F [V.]	Periodo T [s.]	F [Hz.]	Retardo.	t [s.]	Tau [s.]
Rayón est. 1 (turbulento)	0.50	2.757575758	0.393939394	0.142857143	0.484848485	0.121212121	0.090909091	0.060606061	1.151515152	0.868421053	0.121212121	0.96969697	
Rayón est. 2 (turbulento)	2.00	2.772727273	0.409090909	0.147540984	0.545454545	0.121212121	0.045454545	0.03030303	1.212121212	0.825	0.121212121	0.606060606	
Poliéster 1 (turbulento)	0.50	2.924242424	0.460606061	0.157512953	0.424242424	0.090909091	0.060606061	0.015151515	0.818181818	1.222222222	0.242424242	1.606060606	
Poliéster 2 (turbulento)	2.00	2.909090909	0.484848485	0.166666667	0.53030303	0.151515152	0.075757576	0.060606061	1.090909091	0.916666667	0.363636364	0.848484848	
Rayón 1 (turbulento)	0.50	2.787878788	0.263636364	0.09465217	0.454545455	0.060606061	0.03030303	0.003030303	1.348484848	0.741573034	0.121212121	0.590909091	
Rayón 2 (turbulento)	2.00	2.96969697	0.266666667	0.089795918	0.212121212	0.060606061	0.03030303	0.015151515	0.909090909	1.1	0.545454545	1.939393939	
Tej. afelpado 1 (turbulento)	0.50	3.818181818	0.5	0.130952381	0.333333333	0.060606061	0.045454545	0.03030303	0.787878788	1.269230769	0.272727273	1.666666667	
Tej. afelpado 2 (turbulento)	2.00	2.954545455	0.327272727	0.110769231	0.303030303	0.090909091	0.045454545	0.03030303	0.96969697	1.03125	0.606060606	2.060606061	
Tej. afelpado 3 (turbulento)	0.50	2.96969697	0.312121212	0.105102041	0.303030303	0.090909091	0.03030303	0.015151515	1.03030303	0.970588235	0.303030303	1.696969697	
Acrílico 1 (turbulento)	0.50	2.954545455	0.224242424	0.075897436	0.303030303	0.03030303	0.257575758	0.021212121	1.090909091	0.916666667	0.227272727	1.242424242	
Acrílico 2 (turbulento)	2.00	2.954545455	0.23030303	0.077948718	0.181818182	0.060606061	0.024242424	0.003030303	0.96969697	1.03125	0.363636364	2.060606061	
Poliéster 1 (máxima P)	0.50	4.848484848	0.987878788	0.20375	0.515151515	0.242424242	0.121212121	0.060606061	0.696969697	1.434782609	0.121212121	1.515151515	
Poliéster 2 (máxima P)	2.00	4.909090909	1.036363636	0.211111111	0.363636364	0.242424242	0.090909091	0.03030303	0.606060606	1.65	0.363636364	2.787878788	
Rayón est. 1 (máxima P)	2.00	5.515151515	1.042424242	0.189019899	0.363636364	0.181818182	0.090909091	0.060606061	0.606060606	1.65	0.303030303	2.181818182	
Rayón est. 2 (máxima P)	0.50	5.454545455	0.981818182	0.18	0.363636364	0.181818182	0.060606061	0.042424242	0.575757576	1.736842105	0.46969697	2.848484848	
Tej. afelpado 1 (máxima P)	2.00	7.151515152	1.024242424	0.143220339	0.181818182	0.060606061	0.03030303	0.006060606	0.484848485	2.0625	0.545454545	3.878787879	
Tej. afelpado 2 (máxima P)	0.50	7.212121212	1.012121212	0.140336134	0.181818182	0.090909091	0.03030303	0.006060606	0.545454545	1.833333333	0.181818182	2.848484848	
Rayón 1 (máxima P)	1.00	8.181818182	1.006060606	0.122962693						0	0.363636364		1.545454545
Acrílico 1 (máxima P)	1.00	9.393939394	0.987878788	0.10516129						0	0.181818182		7.03030303
Acrílico 2 (máxima P)	2.00	5.515151515	0.515151515	0.093406593	0.181818182	0.060606061	0.045454545	0.03030303	0.606060606	1.65	0.424242424	2.424242424	
Rayón 1 (1/2 mH2O)	0.50	4.848484848	0.496969697	0.1025	0.272727273	0.090909091	0.045454545	0.015151515	0.696969697	1.434782609	0.151515152	1.484848485	
Rayón 2 (1/2 mH2O)	2.00	4.787878788	0.506060606	0.105692603	0.212121212	0.060606061	0.03030303	0.003030303	0.606060606	1.65	0.484848485	2.644444447	
Rayón est. 1 (1/2 mH2O)	0.50	3.287878788	0.5	0.152073733	0.348484848	0.121212121	0.03030303	0.021212121	0.909090909	1.1	0.363636364	1.818181818	
Rayón est. 2 (1/2 mH2O)	2.00	3.272727273	0.493939394	0.150925926	0.393939394	0.121212121	0.060606061	0.03030303	0.848484848	1.178571429	0.484848485	2.060606061	
Poliéster 1 (1/2 mH2O)	2.00	3.03030303	0.515151515	0.17	0.393939394	0.090909091	0.03030303	0.027272727	0.96969697	1.03125	0.448484848	1.696969697	
Poliéster 2 (1/2 mH2O)	0.50	3	0.484848485	0.161616162	0.424242424	0.154545455	0.03030303	0.024242424	0.090909091	0.916666667	0.090909091	0.787878788	
Tej. afelpado 1 (1/2 mH2O)	2.00	4.151515152	0.509090909	0.122627737	0.242424242	0.121212121	0.045454545	0.03030303	0.606060606	1.65	0.727272727	2.181818182	
Acrílico 1 (1/2 mH2O)	0.50	5.454545455	0.490909091	0.09	0.151515152	0.090909091	0.03030303	0.024242424	0.606060606	1.65	0.484848485	2.878787879	
Acrílico 2 (1/2 mH2O)	2.00	2.878787879	0.218181818	0.075789474	0.484848485	0.090909091	0.03030303	0.003030303	0.606060606	1.65	0.121212121	0.606060606	

En la tabla que se muestra en la hoja previa ("Información experimental") se pueden distinguir los tiempos de muestreo observando el apartado de tiempo por división: 0.5 [s/div] corresponde al muestreo de 5 segundos y 2 [s/div] para el muestreo de 20 segundos. Para el caso de máxima presión en las telas de rayón 100% y acrilán 100% se muestreó en 10 segundos. Para estas telas el sobrepaso, que caracteriza a un sistema de segundo orden, no se presentó. Se midió la constante τ que caracteriza a un sistema de primer orden.

De esta información experimental se obtienen los valores numéricos para un sistema de segundo orden mediante las siguientes expresiones:

$$1. \text{ Sobrepaso: } \text{Sobrepaso} = \frac{C}{K} = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (\text{ec. 4.16}).$$

$$2. \text{ Tasa de decaimiento: } C/E = D/F = (\text{sobrepaso})^2 \quad (\text{ec. 4.17}).$$

$$3. \text{ Factor de amortiguamiento: } \zeta = \frac{\sqrt{[\ln(\text{tasa})]^2}}{\sqrt{\pi^2 + [\ln(\text{tasa})]^2}} \quad (\text{ec. 4.18}).$$

$$4. \text{ Constante de tiempo: } = \frac{t_1 \sqrt{1-\zeta}}{\pi - \cos^{-1} \zeta} \quad (\text{ec. 4.19}).$$

Los resultados para la determinación de los parámetros del sistema de segundo orden para cada una de las telas se muestran en la siguientes dos hojas ("Modelo matemático"). Para cada tela se obtuvo un promedio de los parámetros para los distintos tiempos de muestreo⁹ (en negritas) y se obtuvieron valores promedio para las telas en conjunto y fueron éstos últimos los que se utilizaron para la determinación de los parámetros del controlador. La validez y corrección de esta suposición se comprobará hasta el momento de la experimentación y pruebas del control. Los parámetros del sistema de segundo orden se obtuvieron con las ecuaciones 4.16, 4.17, 4.18 y 4.19.

⁹ Estos promedios se utilizaron para el análisis de cada una de las telas.

Modelo matemático.

Tipo de tejido	K _p	K _{lela}	Sobrepeso (%)	Sobrepeso promedio (%)	Amortiguamiento	Amort. promedio	T [s]	T promedio [s]	Tasa dec.	Tasa promedio
Rayón est. 1 (turbulento)	0.143		24.615		0.407		1.152		0.188	
Rayón est. 2 (turbulento)	0.148	0.145	26.667	25.641	0.388	0.390	1.212	1.102	0.083	0.135
Poliéster 1 (turbulento)	0.158		18.421		0.474		0.818		0.143	
Poliéster 2 (turbulento)	0.167	0.162	21.875	20.140	0.435	0.455	1.091	0.955	0.143	0.143
Rayón 1 (turbulento)	0.095		34.483		0.321		1.348		0.067	
Rayón 2 (turbulento)	0.090	0.092	15.909	25.196	0.505	0.413	0.909	1.129	0.143	0.105
Tej. acilgado 1 (turbulento)	0.131		13.333		0.540		0.788		0.136	
Tej. acilgado 2 (turbulento)	0.111		18.519		0.473		0.970		0.150	
Tej. acilgado 3 (turbulento)	0.105	0.116	19.417	17.090	0.463	0.492	1.030	0.929	0.100	0.129
Acrílico 1 (turbulento)	0.076		27.027		0.384		1.091		0.850	
Acrílico 2 (turbulento)	0.078	0.077	15.789	21.400	0.507	0.446	0.970	1.030	0.133	0.092
Poliéster 1 (máxima F)	0.204		10.429		0.584		0.697		0.235	
Poliéster 2 (máxima F)	0.211	0.207	7.018	0.723	0.646	0.615	0.606	0.652	0.250	0.243
Rayón est. 1 (máxima F)	0.189		6.977		0.647		0.606		0.250	
Rayón est. 2 (máxima F)	0.180	0.185	7.407	7.192	0.638	0.642	0.576	0.591	0.167	0.200
Tej. acilgado 1 (máxima F)	0.143		3.550		0.728		0.485		0.167	
Tej. acilgado 2 (máxima F)	0.140	0.142	3.593	3.572	0.727	0.720	0.545	0.515	0.167	0.167
Rayón 1 (máxima F)	0.123	0.123	***	***	***	***	***	***	***	***
Acrílico 1 (máxima F)	0.105		***	***	***	***	***	***	***	***
Acrílico 2 (máxima F)	0.093	0.099	7.059	7.059	0.645	0.645	0.606	0.606	0.250	0.250
Rayón 1 (1/2 in120)	0.103		10.976		0.575		0.697		0.167	
Rayón 2 (1/2 in120)	0.106	0.104	8.383	9.679	0.619	0.597	0.606	0.652	0.143	0.143
Rayón est. 1 (1/2 in120)	0.152		13.939		0.531		0.909		0.087	
Rayón est. 2 (1/2 in120)	0.151	0.151	15.951	14.945	0.504	0.518	0.848	0.879	0.154	0.154
Poliéster 1 (1/2 in120)	0.170		15.294		0.513		0.970		0.077	
Poliéster 2 (1/2 in120)	0.162	0.166	17.500	16.397	0.485	0.499	1.091	1.030	0.071	0.071
Tej. acilgado 1 (1/2 in120)	0.123	0.123	9.524	9.524	0.599	0.599	0.606	0.606	0.188	0.100
Acrílico 1 (1/2 in120)	0.090		6.173		0.663		0.606		0.200	
Acrílico 3 (1/2 in120)	0.076	0.083	44.444	25.309	0.250	0.457	0.606	0.606	0.063	0.131
PROMEDIO =	0.132	0.132	15.714	15.135	0.528	0.536	0.831	0.812	0.177	0.183

Tipo de textil	θ [°]	θ promedio [°]	$\tan \alpha$ [°]	\tan promedio [°]	ϵ muerto [°]	ϵ muerto promedio [°]
Rayón est. 1 (turbulento)	0.970		0.445		0.121	
Rayón est. 2 (turbulento)	0.606	0.788	0.284	0.364	0.121	0.121
Poliéster 1 (turbulento)	1.606		0.685		0.242	
Poliéster 2 (turbulento)	0.848	1.227	0.378	0.531	0.364	0.303
Rayón 1 (turbulento)	0.591		0.295		0.121	
Rayón 2 (turbulento)	1.939	1.265	0.797	0.546	0.545	0.333
Tej. afelpado 1 (turbulento)	1.667		0.655		0.273	
Tej. afelpado 2 (turbulento)	2.061		0.880		0.606	
Tej. afelpado 3 (turbulento)	1.697	1.808	0.733	0.756	0.303	0.394
Acrílico 1 (turbulento)	1.242		0.584		0.227	
Acrílico 2 (turbulento)	2.061	1.652	0.845	0.714	0.364	0.295
Poliéster 1 (máxima P)	1.515		0.560		0.121	
Poliéster 2 (máxima P)	2.788	2.152	0.937	0.749	0.364	0.242
Rayón est. 1 (máxima P)	2.182		0.732		0.303	
Rayón est. 2 (máxima P)	2.848	2.515	0.969	0.851	0.470	0.386
Tej. afelpado 1 (máxima P)	3.879		1.114		0.545	
Tej. afelpado 2 (máxima P)	2.848	3.364	0.820	0.967	0.182	0.364
Rayón 1 (máxima P)	***		3.091	3.091	0.364	0.364
Acrílico 1 (máxima P)	***		3.515	3.515	0.182	
Acrílico 2 (máxima P)	2.424	2.424	0.816	0.816	0.424	0.303
Rayón 1 (1/2 in120)	1.485		0.556		0.152	
Rayón 2 (1/2 in120)	2.667	2.076	0.935	0.746	0.485	0.318
Rayón est. 1 (1/2 in120)	1.818		0.723		0.364	
Rayón est. 2 (1/2 in120)	2.061	1.939	0.847	0.785	0.485	0.424
Poliéster 1 (1/2 in120)	1.697		0.691		0.448	
Poliéster 2 (1/2 in120)	0.788	1.242	0.332	0.511	0.091	0.270
Tej. afelpado 1 (1/2 in120)	2.182	2.182	0.789	0.789	0.727	0.727
Acrílico 1 (1/2 in120)	2.879		0.938		0.485	
Acrílico 3 (1/2 in120)	0.606	1.742	0.322	0.630	0.121	0.303
PROMEDIO =	1.850	1.884	0.691	0.856	0.331	0.343

4.8 El modelo de control.

La asesoría consultada¹⁰ recomendó utilizar los valores que propone el método de Modelo Interno de Control (MIC) en un controlador proporcional (P) y uno proporcional integral (PI). A continuación se describe el método MIC.

El Modelo Interno de Control es un modelo de control que ataca implícitamente los problemas de las incertidumbres en el modelado de un sistema o proceso. La razón por la que se sugirió este tipo de control es la imprecisión en el modelado matemático del sistema. Como se explicó anteriormente, no se tiene la información matemática cierta del comportamiento del aparato medidor. Las incertidumbres no son tomadas en cuenta por la teoría clásica de control retroalimentado. En este trabajo no se utiliza el modelo de control MIC, simplemente se utiliza como reglas de ajuste a un control PI.

De manera similar a la técnica de Síntesis Directa, el método de diseño MIC está basado en un cierto modelo del proceso y su instrumentación (P^*), como puede ser el generado por las curvas de respuesta. La técnica MIC tiene dos ventajas importantes:

1. Considera de manera explícita la incertidumbre del modelo.
2. Permite al diseñador encontrar un balance entre el desempeño y la robustez a cambios en el proceso.

Considérese el siguiente diagrama:

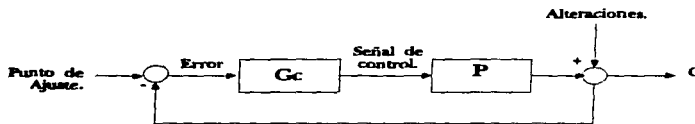


Fig. 4.21 a) Control por retroalimentación clásico.

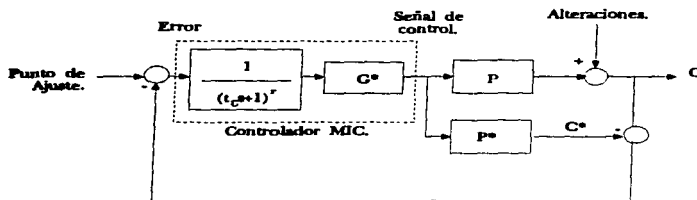


Fig. 4.21 b) Control basado en MIC.

La función de transferencia P representa al proceso y su instrumentación; la función P^* representa un modelo del proceso; la variable a controlar se representa con señal C y el valor de la predicción que

¹⁰Mto. Ernesto Ríos Patrón, consulta personal, School of Chemics, University of Illinois at Urbana.

entrega el modelo del proceso por C^* . Al sustraer la predicción del modelo a la respuesta del proceso real ($C-C^*$), se obtiene una señal que representa a las alteraciones, ruido e incertidumbres que el modelo matemático no puede representar con fidelidad. Dicha señal es alimentada a un controlador que se encuentra formado por dos funciones de transferencia en serie: un filtro de grado r y un función G^* , ambas funciones de transferencia forman el controlador MIC que se denominará Gc^* . El MIC está basado en el diagrama de bloques mostrado en la figura b). El controlador MIC se diseña en dos pasos:

Paso 1: el modelo del proceso P^* se factoriza como sigue:

$$P^* = P_+^* P_-^*$$

donde P_+^* incluye retrasos en el tiempo y ceros en el medio plano derecho, es especificado de tal manera que su ganancia sea igual a uno.

Paso 2: se selecciona el valor de r para determinar el orden del filtro, y se especifica al controlador de la siguiente manera:

$$Gc^* = \frac{1}{G_-^*} \left(\frac{1}{\tau_c (s+1)^r} \right)$$

Notar que en el caso de un modelo perfecto $C=C^*$ no existe retroalimentación, en este caso el controlador funcionará como un sistema de control por alimentación adelantada. Los sistemas de la figura b) tendrán un comportamiento dinámico idéntico si se cumple con la siguiente relación:

$$Gc = \frac{Gc^*}{1 - Gc^* P^*}$$

De esta forma se calcula un controlador por retroalimentación clásico utilizando el método MIC. Para determinar el valor de los parámetros PID de un controlador se iguala la expresión algebraica obtenida para Gc con la función de un controlador PID:

$$Gc = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s \right)$$

Resolviendo las igualdades, se encuentran los parámetros del controlador. Por ejemplo, para el caso de un modelo de segundo orden subamortiguado, en el que se selecciona un filtro de primer orden ($r=1$), se tiene:

$$K_c = \frac{2\tau_\zeta}{K_p \tau_c}$$

$$\tau_I = 2\zeta\tau$$

$$\tau_D = \frac{\tau}{2\zeta}$$

donde las constantes K_p , ζ y τ pueden ser determinadas a partir de la información experimental. Y como un primer intento se supone $\tau_c=1$.

En la siguiente hoja (“Parámetros del controlador”), se muestran los valores de los parámetros del controlador. Para la determinación de los parámetros del controlador se utilizaron los promedios globales. Se encontraron los valores para un controlador PID, pero solamente se desarrolló el proporcional (P) y el proporcional-integral (PI). Nuevamente se recalca que la validez de esta suposición será comprobada al realizar las pruebas de control.

Parámetros del controlador.

Tipo de textil	Kc	Kc tela	tau I	tau I tela	tau D	tau D tela
Rayón est. 1 (turbulento)	2.538		0.363		0.546	
Rayón est. 2 (turbulento)	1.491	2.015	0.220	0.291	0.366	0.456
Poliéster 1 (turbulento)	4.123		0.649		0.722	
Poliéster 2 (turbulento)	1.975	3.049	0.329	0.489	0.434	0.578
Rayón 1 (turbulento)	2.002		0.189		0.459	
Rayón 2 (turbulento)	8.965	5.484	0.805	0.497	0.789	0.624
Tej. afelpado 1 (turbulento)	5.403		0.707		0.607	
Tej. afelpado 2 (turbulento)	7.514		0.832		0.930	
Tej. afelpado 3 (turbulento)	6.455	6.457	0.678	0.739	0.793	0.777
Acrílico 1 (turbulento)	5.912		0.449		0.759	
Acrílico 2 (turbulento)	10.986	8.449	0.856	0.653	0.834	0.797
Poliéster 1 (máxima P)	3.213		0.655		0.480	
Poliéster 2 (máxima P)	5.730	4.471	1.210	0.932	0.725	0.603
Rayón est. 1 (máxima P)	5.008		0.947		0.566	
Rayón est. 2 (máxima P)	6.872	5.940	1.237	1.092	0.760	0.663
Tej. afelpado 1 (máxima P)	11.327		1.622		0.765	
Tej. afelpado 2 (máxima P)	8.498	9.912	1.193	1.407	0.564	0.664
Rayón 1 (máxima P)	16.800		3.273		0.133	
Acrílico 1 (máxima P)	20.681		3.606		0.077	
Acrílico 2 (máxima P)	11.262	15.971	1.052	2.329	0.632	0.355
Rayón 1 (1/2 lnH2O)	6.243		0.640		0.483	
Rayón 2 (1/2 lnH2O)	10.960	8.602	1.158	1.158	0.755	0.755
Rayón est. 1 (1/2 lnH2O)	5.051		0.768		0.680	
Rayón est. 2 (1/2 lnH2O)	5.665	5.358	0.855	0.855	0.840	0.840
Poliéster 1 (1/2 lnH2O)	4.168		0.709		0.673	
Poliéster 2 (1/2 lnH2O)	1.991	3.079	0.322	0.322	0.342	0.342
Tej. afelpado 1 (1/2 lnH2O)	7.713	7.713	0.946	0.946	0.659	0.659
Acrílico 1 (1/2 lnH2O)	13.830		1.245		0.707	
Acrílico 3 (1/2 lnH2O)	2.123	7.976	0.161	0.703	0.644	0.675
PROMEDIO =	6.186	6.748	0.770	0.887	0.649	0.628

4.9 El modelo de control digital.

De acuerdo a la tabla de parámetros del controlador, los valores de los parámetros del controlador son:

$$K_c = 6.186,$$

$$\tau_I = 0.77,$$

$$\tau_D = 0.649.$$

El efecto del tiempo muerto no será contemplado, pues este tiempo es considerablemente menor a las constantes de tiempo del sistema. La necesidad de incluir el tiempo muerto se establecerá hasta realizar las pruebas con el controlador.

Como se mencionó anteriormente, el primer intento de control será con los modelos proporcional (P) y proporcional-integral (PI). Las ecuaciones de un control proporcional son:

$$V = K_c e(n) + V(n-1),$$

donde K_c es la ganancia del controlador. A este controlador se le denomina controlador de posición.

La ecuación del control PI propuesto es:

$$V = K_c [e(n) - e(n-1)] + K_c t_m [e(n) - e(n-1)] / 2\tau,$$

donde K_c es la ganancia del controlador, t_m es el tiempo de muestreo y τ es la constante de integración. En esta ecuación el control proporcional es de rapidez de cambio (al obtener la diferencia entre el error actual y el anterior)¹¹. El tiempo de muestreo está ligado a la frecuencia natural del sistema. En las mediciones experimentales se encuentra que el periodo de menor duración es de 0.515 [ms]. Por criterio de Nyquist se debe muestrear a una frecuencia de por lo menos dos veces la frecuencia de la señal muestreada. La asesoría consultada¹² sugirió un periodo de muestreo menor a 1 segundo, el sistema HC11 fue programado para un muestreo de periodo de 124 [ms].

4.9.1 Programación del modelo de control digital.

Los programas para el control proporcional y el control proporcional-integral se encuentran en el apéndice H. Siguen el esquema de programas para el sistema operativo propuesto.

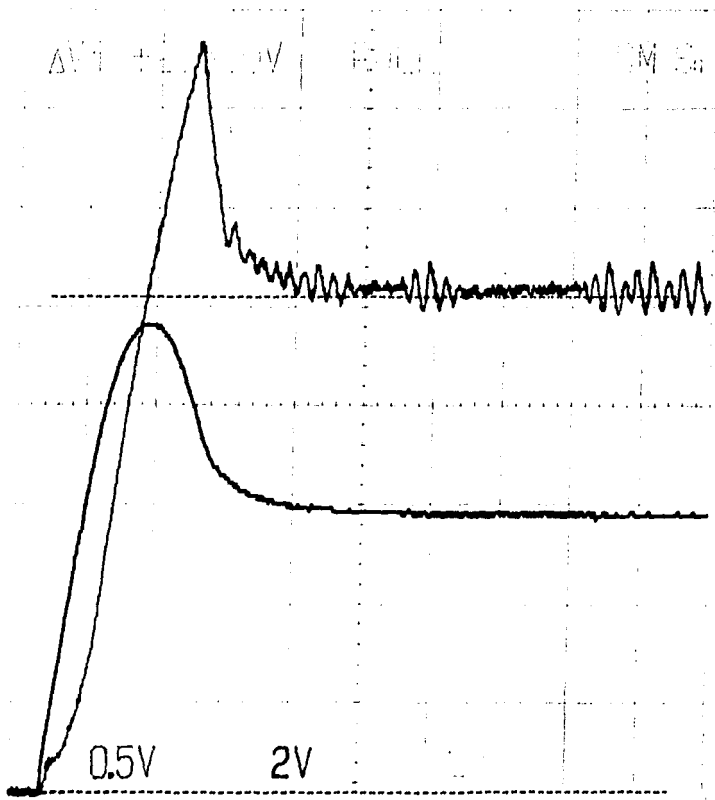
4.10 Experimentación y pruebas para mejorar la respuesta del sistema.

En las siguientes 7 gráficas se muestra el comportamiento de la acción de control modelando al textil como un sistema de segundo orden; las pruebas son para el textil más cerrado y para el textil más abierto. Se realizaron pruebas con el control proporcional (P) y con el proporcional-integral (PI). Las gráficas muestran la salida del sistema (presión en el textil) y la entrada (acción de control).

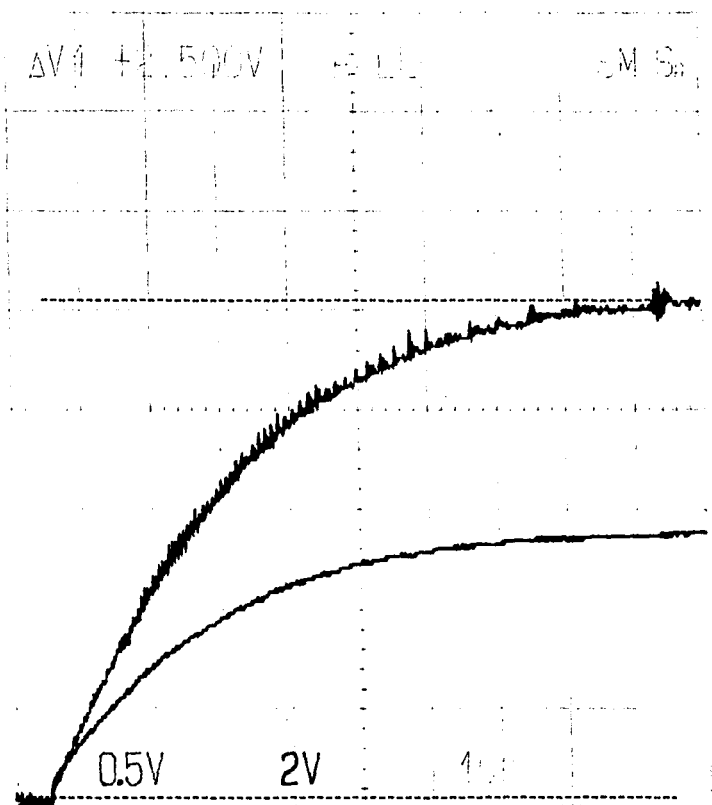
¹¹ La asesoría consultada recomendó el uso de un control proporcional de velocidad, pues la acción de control es más suave que uno de posición. Mto. Ernesto Ríos Patrón, School of Chemic, University of Illinois at Urbana C.

¹² Mto. Ernesto Ríos Patrón, School of Chemic, University of Illinois at Urbana C.

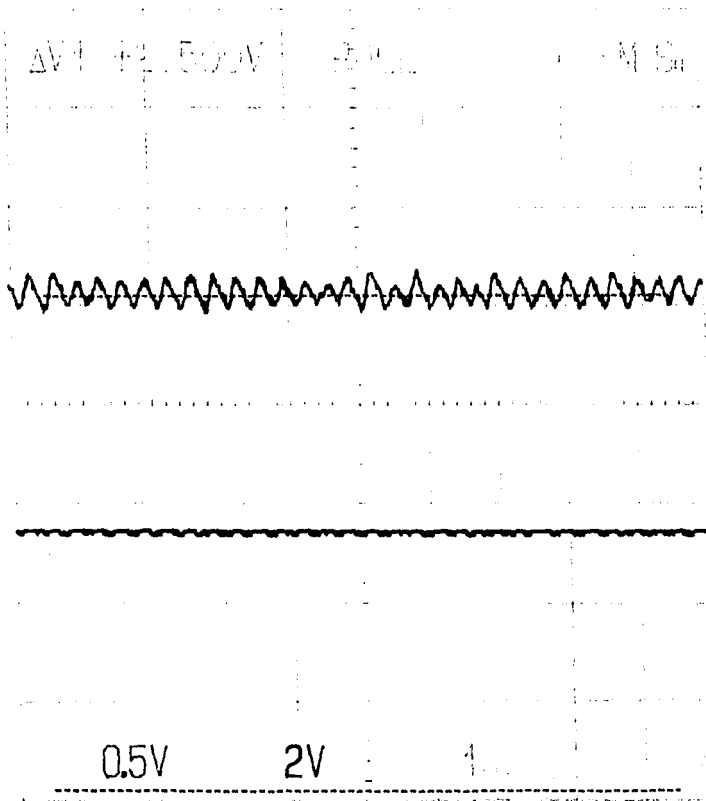
Salida del sistema y acción de control P, sistema de segundo orden, tejido abierto (acrián 100%).



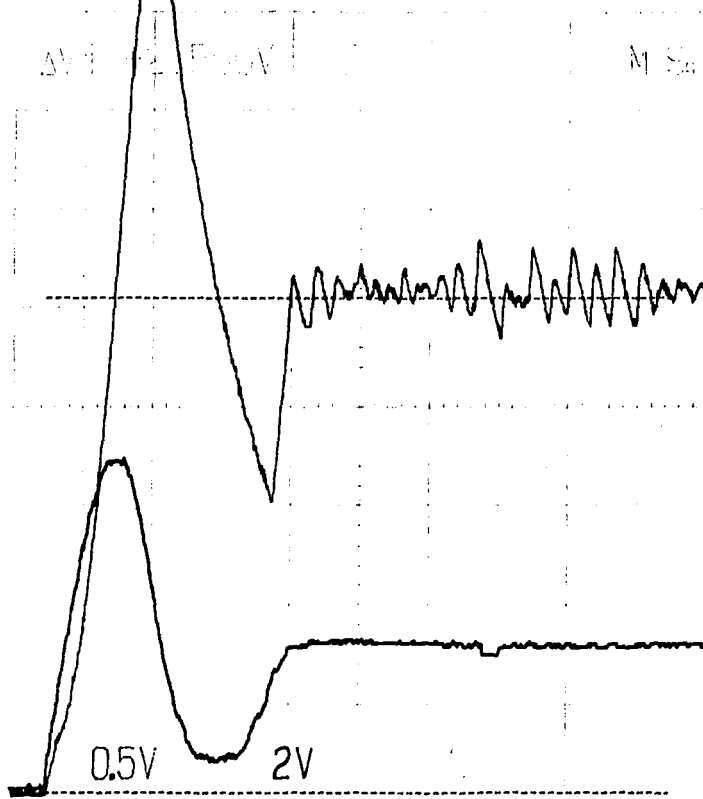
Salida del sistema y acción de control PI, sistema de segundo orden, tejido abierto (acrilán 100%).



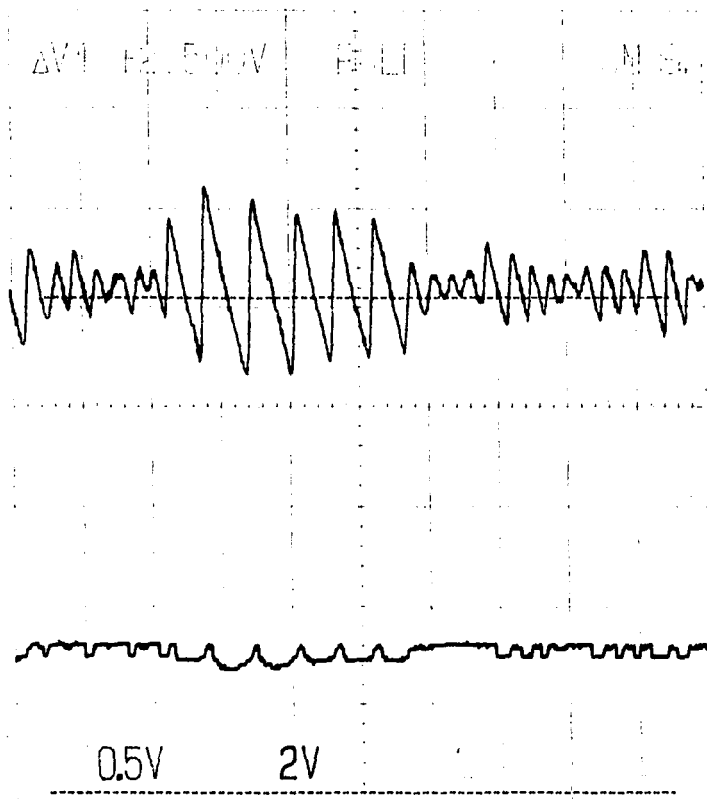
Salida del sistema a régimen permanente y acción de control PI, sistema de segundo orden, tejido abierto (acrilán 100%).



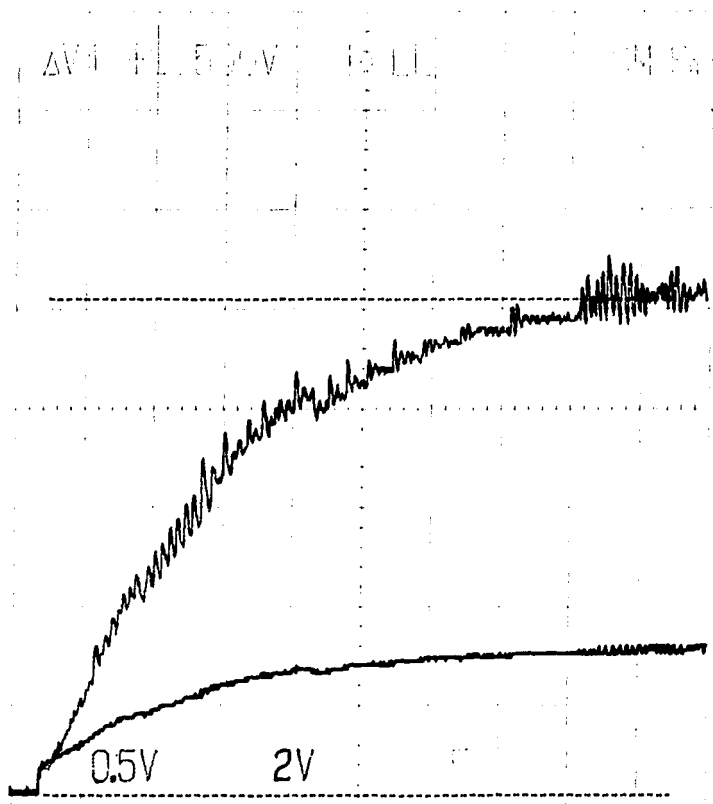
Salida del sistema y acción de control P, sistema de segundo orden, tejido cerrado (poliéster 100%).



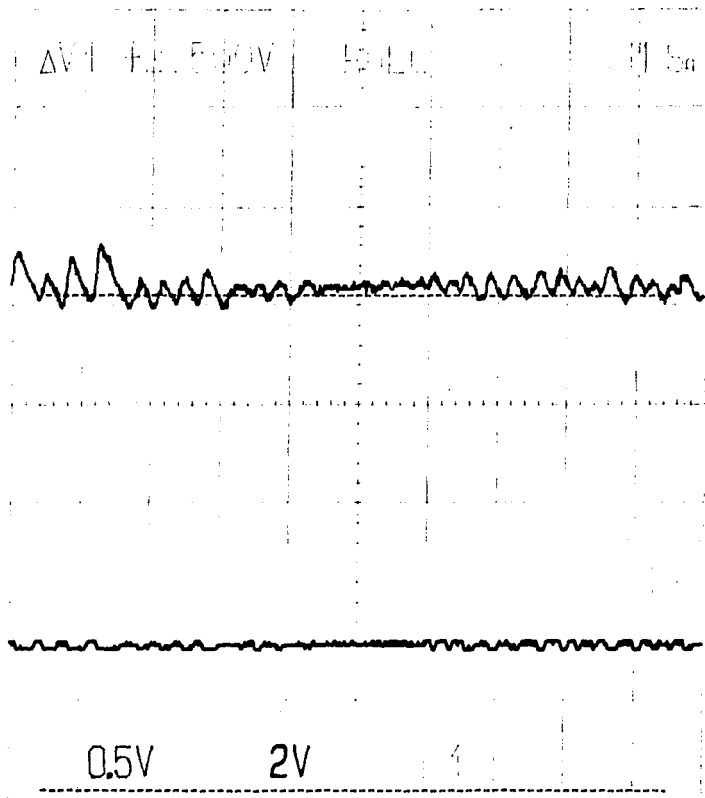
Salida del sistema a régimen permanente y acción de control P, sistema de segundo orden, tejido cerrado (poliéster 100%).



Salida del sistema y acción de control PI, sistema de segundo orden, tejido cerrado (poliéster 100%).



Salida del sistema a régimen permanente y acción de control PI, sistema de segundo orden, tejido cerrado (poliéster 100%).



Puede observarse claramente el comportamiento oscilatorio para el textil más cerrado con el modelo de control proporcional. El mismo tipo de control con el textil abierto tiene un comportamiento ligeramente más estable sin dejar de presentar oscilaciones esporádicamente. Estas oscilaciones disminuyeron al cambiar la constante de proporcionalidad del controlador a la mitad; sin embargo, la respuesta del sistema fue más lenta y aún oscilatoria para el textil más cerrado.

Al probar con el control integral, la respuesta del sistema fue considerablemente más lenta y las oscilaciones se presentaron en menor medida para ambas telas.

En un intento por mejorar el comportamiento del sistema, se analizó el comportamiento del tacómetro. Como se mencionó anteriormente, el controlador en lazo cerrado del motor provoca el sobrepaso en la respuesta a escalón del sistema. Esto se debe a la velocidad de respuesta del tacómetro. El tacómetro está conformado por una bomba de carga que opera con el disparo positivo de la señal cuadrada generada a partir de la señal de entrada. Para obtener un voltaje en función de la frecuencia de los disparos, se utiliza un capacitor, que se descarga durante el resto del tiempo que sigue al disparo. Este mecanismo de conversión produce un rizo en la señal de voltaje de salida de la bomba de carga. El fabricante sugiere un filtro pasobajas a la salida de la bomba de carga. Es precisamente este filtro el que provoca la lentitud en respuesta del convertidor F/V. Este filtro es necesario para disminuir el rizo provocado por la bomba de carga.

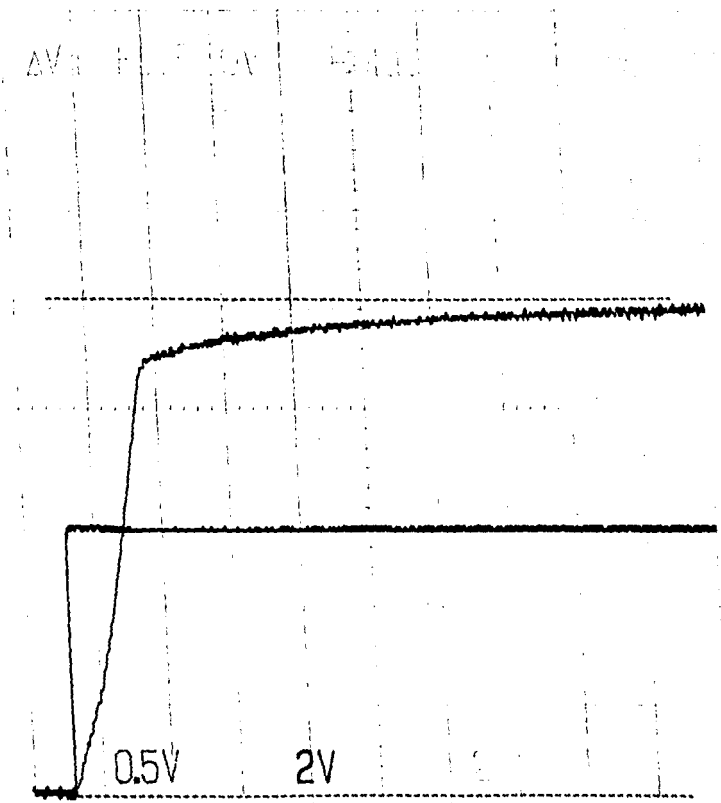
Dado que se desea disminuir el tiempo de respuesta del convertidor, debe disminuirse el valor del capacitor del filtro. Experimentalmente se disminuyó de 1 [μ F] a 0.22 [μ F]. La respuesta del convertidor aumentó de tal manera que es menor al tiempo de respuesta del motor. De esta forma se logró un comportamiento del sistema similar a un sistema de primer orden. El control proporcional diseñado para el sistema de segundo orden funcionó perfectamente para esta nueva modificación, de igual manera, el controlador PI operó de manera satisfactoria, salvo que la respuesta fue igualmente lenta que para el sistema de segundo orden. El sistema mostró mayor estabilidad que para el modelo de segundo orden.

Las repercusiones que tiene esta modificación es que se presentan ligeros golpeteos en el rotor del motor cuando se opera a bajas revoluciones, pues el controlador TDA1085 recibe la señal del tacómetro con un rizo considerable. Este problema no requirió ninguna consideración especial para este trabajo, pues las revoluciones se mantienen de manera estable. Si se desea eliminar este problema, se propone el aumentar la frecuencia del tacómetro óptico en un múltiplo de las revoluciones. De esta forma la bomba de carga operará a una frecuencia mayor y el valor del capacitor del filtro será menor, disminuyendo el tiempo de respuesta del tacómetro.

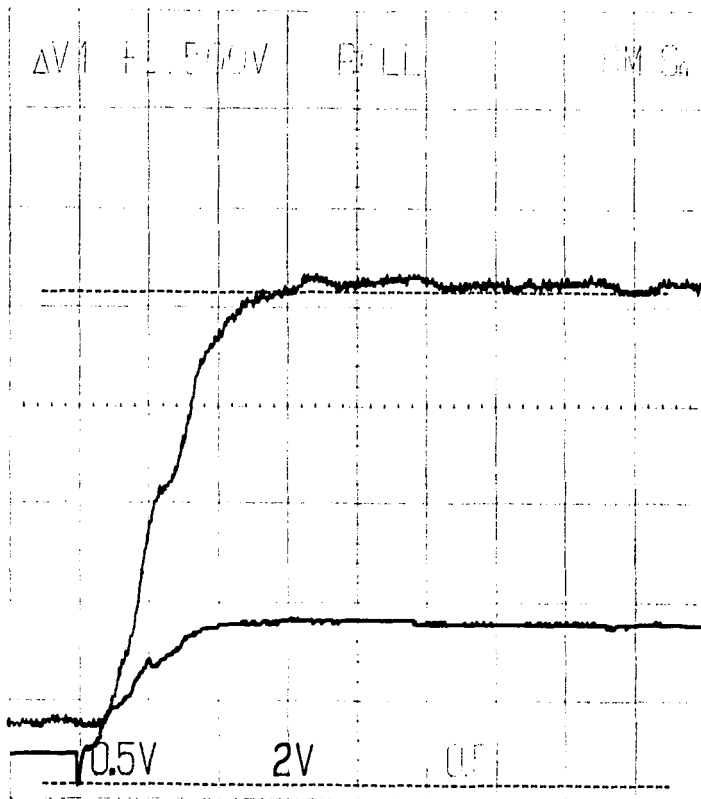
En las siguientes hojas se muestran las gráficas para la respuesta a escalón con el textil más cerrado y con el más abierto. También se muestran las gráficas de la acción del controlador proporcional y el proporcional integral para ambas telas. Podrá observarse que las oscilaciones desaparecieron considerablemente y el tiempo de levantamiento para el proporcional es considerablemente menor que el modelo de segundo orden.

Debe realizarse un análisis de ruido en los componentes del controlador de ángulo de disparo, pues durante las pruebas se encontraron distorsiones en los medidores de presión provocadas por la inducción producida por las escobillas del motor y la conmutación del controlador de ángulo de disparo. Las pruebas fueron realizadas sin tierra física para poder cuantificar la entrada y la salida del sistema con una referencia común.

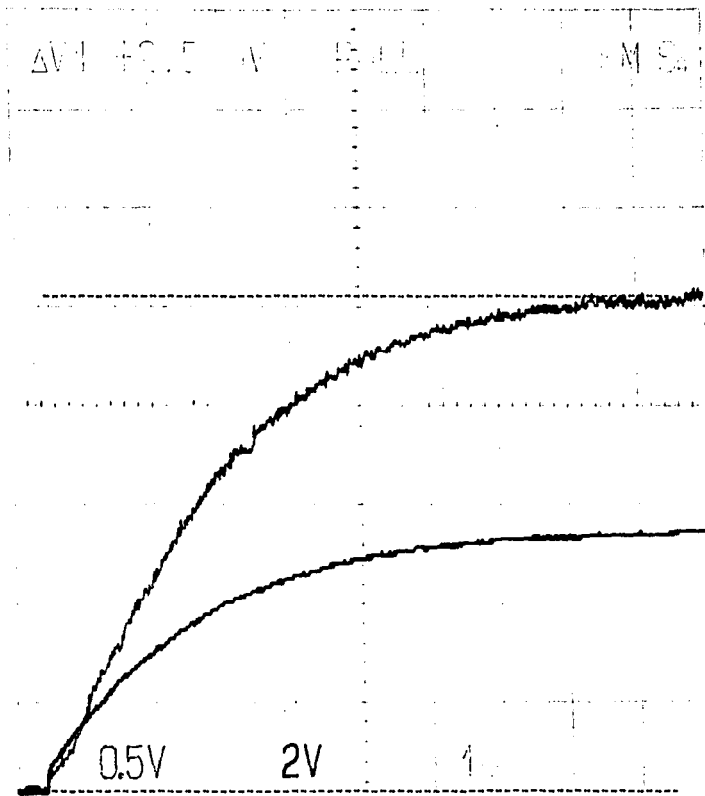
Respuesta escalón, tacómetro modificado, tejido abierto (acrilán 100%).



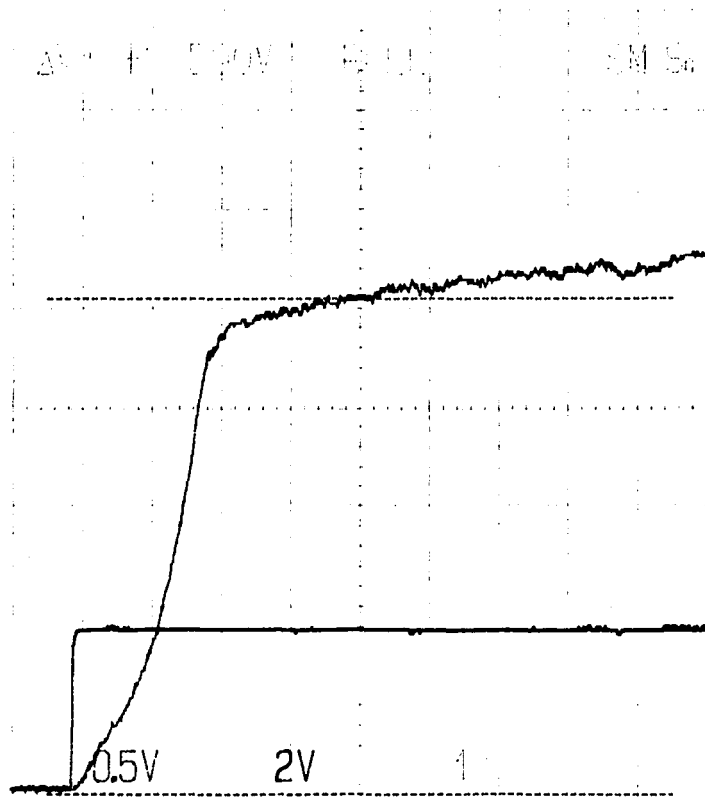
Salida del sistema y acción de control P, tacómetro modificado, tejido cerrado (poliéster 100%).



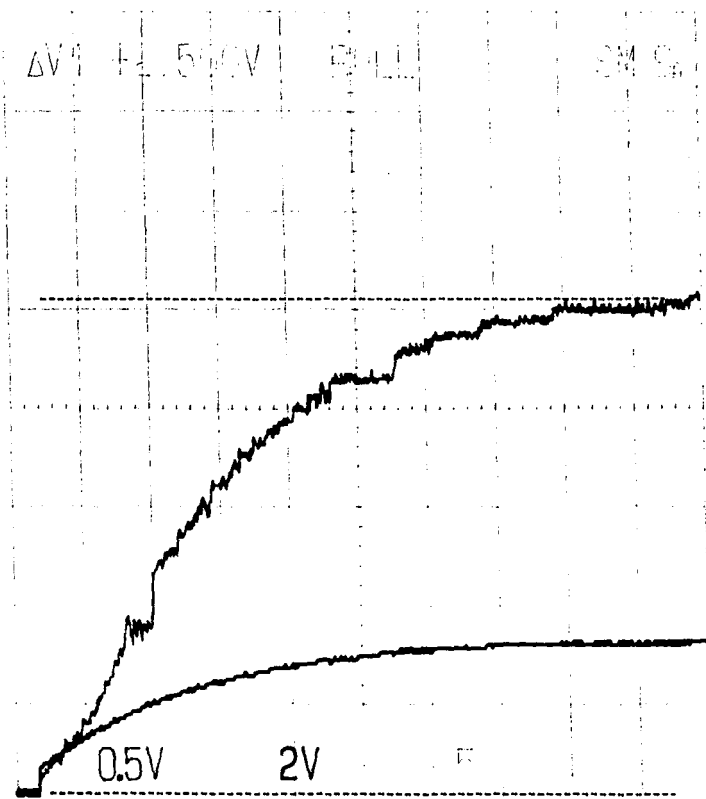
Salida del sistema y acción de control PI, tacómetro modificado, tejido abierto (acrilán 100%).



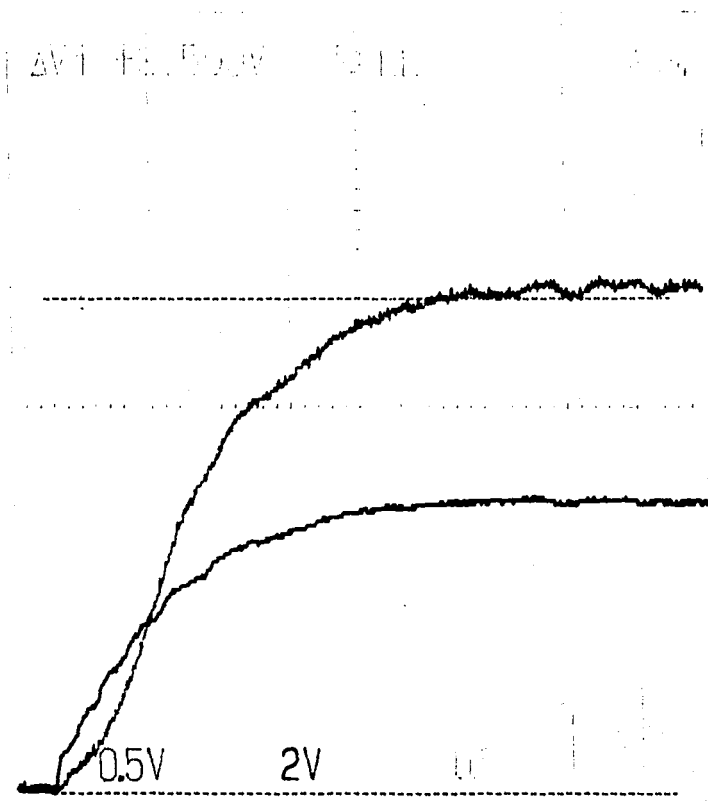
Respuesta escalón, tacómetro modificado, tejido cerrado (poliéster 100%).



Salida del sistema y acción de control PI, tacómetro modificado, tejido cerrado (poliéster 100%).



Salida del sistema y acción de control P, tacómetro modificado, tejido abierto (acrilán 100%).



Capítulo 5. Arquitectura del sistema.

En este capítulo se hace una descripción de la arquitectura utilizada con el procesador HC11. Se siguen, normalmente, las indicaciones de los fabricantes para acoplar todos los elementos de la arquitectura. Cada elemento de la arquitectura tiene una serie de funciones controladas por el microcontrolador. No se pretende proponer una arquitectura nueva.

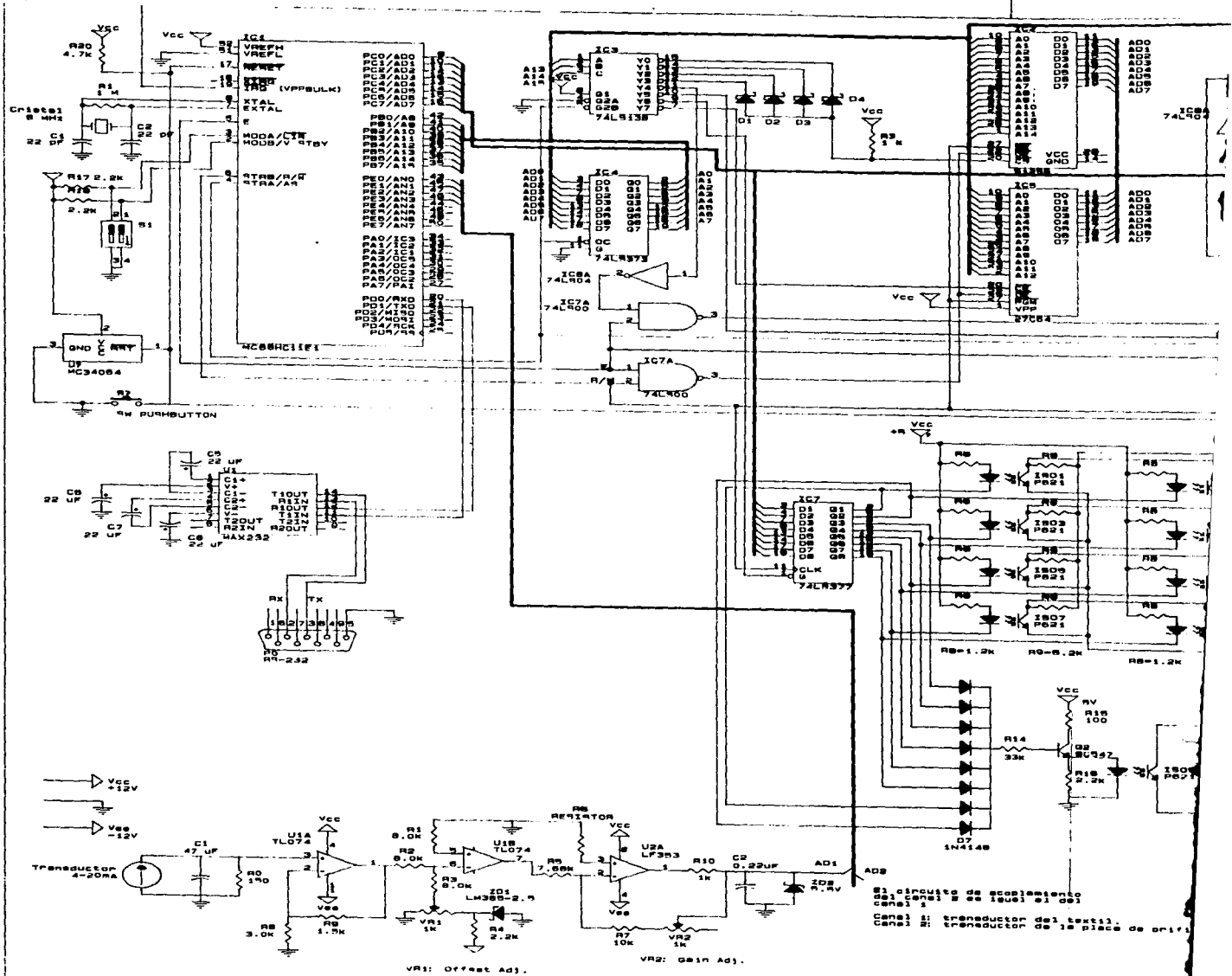
5.1 Los elementos de la arquitectura.

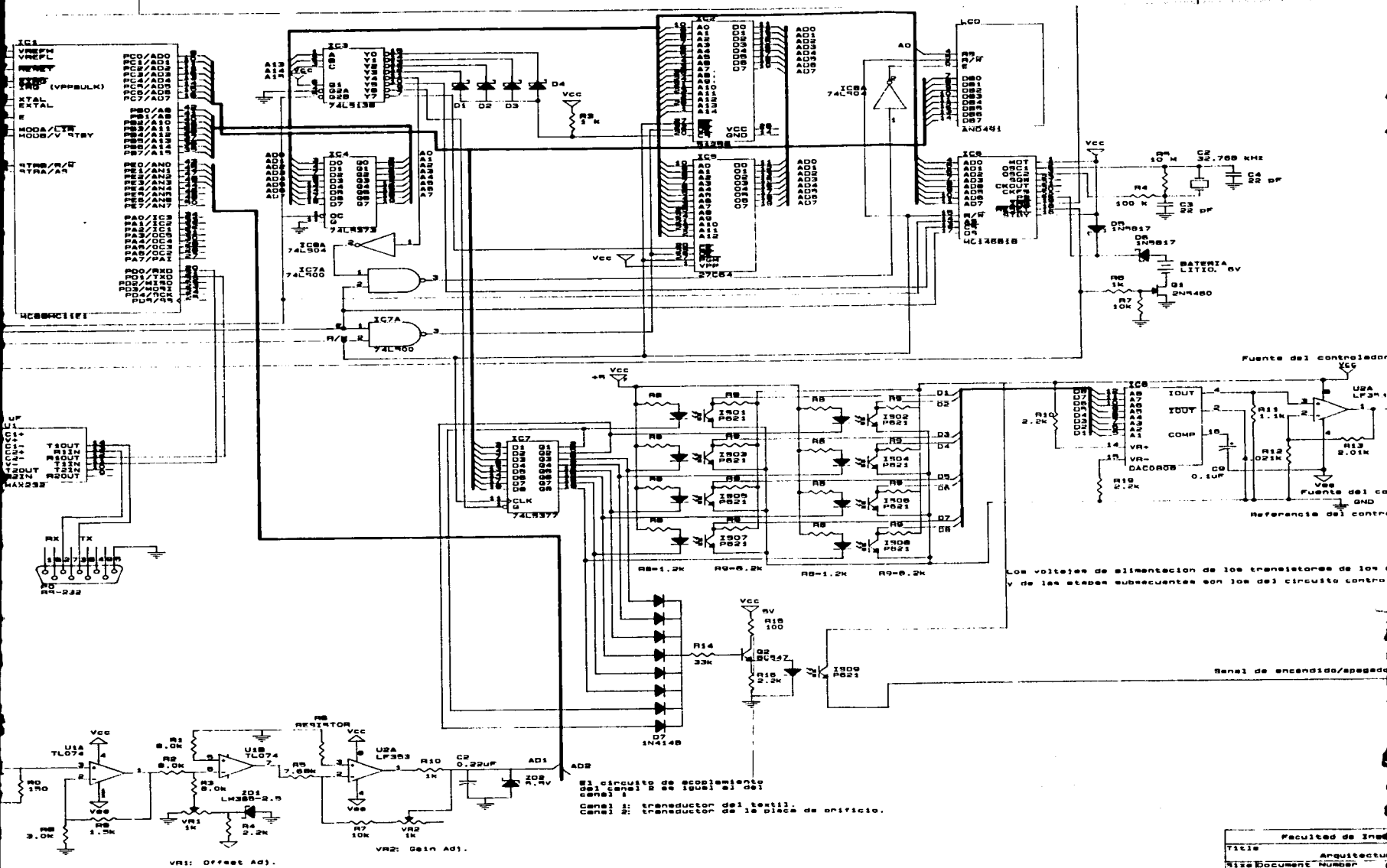
La arquitectura está conformada por los siguientes elementos:

- 1.- Un microcontrolador HC11.
- 2.- Una pantalla de cristal líquido(LCD).
- 3.- Un reloj de tiempo real(RTC).
- 4.- Circuitería para acoplar niveles RS-232.
- 5.- Memorias ROM y RAM.
- 6.- Circuitería para demultiplexión de direcciones.
- 7.- Circuitería para regulación de voltajes e inicialización.
- 8.- Circuitería de acoplamiento con transductores de presión.

Todos estos elementos se interconectaron de acuerdo a las sugerencias de los fabricantes. Sin embargo, hubo que hacer algunas modificaciones en la circuitería de decodificación de bloques de memoria para acoplar el reloj de tiempo real.

El diagrama de conexiones de los elementos se muestra en la siguiente página. En el diagrama se incluyen la primera etapa de acoplamiento con el controlador del motor, y la circuitería para el acoplamiento con los transductores de presión. No se incluye la circuitería para el canal 2 del convertidor A/D, pues es exactamente igual a la del canal 1, que sí se muestra.





Este circuito de acondicionamiento de señal es el que se muestra en el diagrama.
 Canal 1: transductor de la tensión.
 Canal 2: transductor de la pieza de orificio.

VR1: Offset Ad1.
 VR2: Gain Ad1.

5.2 El procesador HC11.

El microcontrolador HC11 de Motorola fue desarrollado en la década de los 70 con fines de control en la industria automotriz. Las principales características del HC11 son:

1. Convertidor A/D de 8 bits, método de distribución de carga.
2. Sistema de comunicación asíncrono (UART) y comunicación asíncrona periférica.
3. Contador de 16 bits para captura, comparación e interrupción en tipo real.
4. Acumulador de pulsos de 8 bits.
5. Circuitería para auto-monitoreo.
6. Controlador con un bus de 8 bits de datos y 16 de direcciones. Dos registros de propósito general de 8 bits (accesados también como un sólo registro de 16 bits). Dos registros para direccionamiento indexado de 16 bits, un registro de 16 bits para pila y un registro de 16 bits como contador de programa.
7. Tecnología HCMOS.

5.2.1 Modos de operación del HC11.

El controlador HC11 tiene dos modos de operación: modo extendido y "single chip" (circuito aislado). La diferencia básica entre estos dos mecanismos de operación es muy sencilla: el modo "single chip" está pensado para aquellas aplicaciones en las que el microcontrolador es el único elemento del sistema con procesador, comunicándose con el medio a través de puertos; mientras que en el modo extendido se utilizan dos de estos puertos como un bus de direcciones y un bus de datos, ello permite al microcontrolador aumentar su capacidad como sistema, es decir, permite la añadidura de más elementos característicos de una arquitectura con microprocesador.

El modo de operación para esta tesis es el modo extendido. Las razones son:

1. Los programas son muy extensos y la memoria interna del HC11 no es suficiente. De ahí la necesidad del uso de memorias adicionales.
2. El uso de un reloj, como parte de la arquitectura.
3. El uso de una pantalla de cristal líquido, para comunicación con el exterior.

La selección entre un modo y otro se hace a través de las terminales MODA y MODB, ambas a nivel lógico ¹.

5.3 Mapa de memoria y demultiplexión.

El mapa de memoria para esta arquitectura consiste en 8 bloques de 8 kbytes cada uno. Esta distribución permite utilizar memorias comerciales de 8 kbytes y mayores, haciendo algunas

¹También se puede cambiar el modo de operación escribiendo sobre el registro interno HPRIO, con el inconveniente de conservar los pseudo-vectores de interrupción.

modificaciones. La selección de cada uno de los bloques se hace utilizando un decodificador TTL 3 a 8. Así, las tres últimas líneas de dirección, las más significativas, A15, A14 y A13 alimentan al decodificador para segmentar el mapa de memoria en bloques de 8 192 Bytes, habilitando un bloque a la vez. Sin embargo, en el diagrama se puede observar que se utiliza una memoria RAM de 32 kbytes. Esto significa que la memoria utiliza cuatro bloques del mapa de memoria. En el circuito se puede observar el arreglo de diodos para poder habilitar la memoria para cuatro terminales del decodificador.

5.4 Pantalla de cristal líquido (LCD).

La pantalla de cristal líquido se encuentra ubicada en el bloque correspondiente a las direcciones A000H y A001H. Este tipo de pantalla permite conectarse directamente al bus de datos del micro. Las señales de control que requiere son: lectura/escritura (R/W), selección de registro (RS) y de habilitación (E). La señal de lectura/escritura es totalmente compatible con el HC11. La señal de habilitación requiere de lógica combinatorial para poder operar correctamente. Esta lógica combinatorial se define a partir de los diagramas de tiempo que el fabricante especifica:

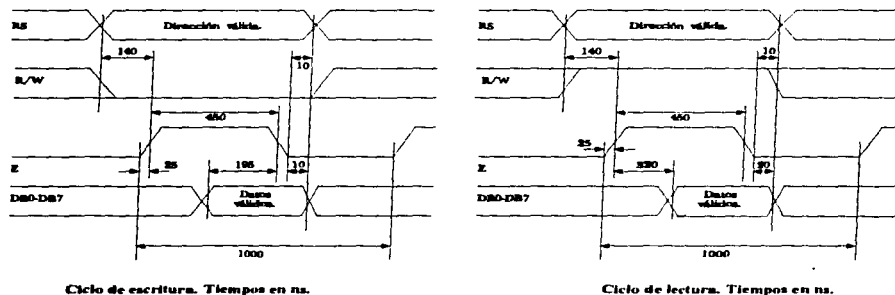


Fig. 5.1 Tiempos de acceso para la pantalla LCD.

En los diagramas de tiempo pueden verse los requerimientos de temporización: el ciclo total debe durar 1000 ns (1 μ s). El reloj E del controlador HC11 funcionando a 8 MHz es de 0.5 μ s, lo que hace suponer que no debería operar. Sin embargo, al consultar al fabricante se encontró que existen diferentes versiones de la pantalla de cristal líquido LCD, que tienen distintos tiempos de acceso. Al hacer las pruebas en ningún momento se detectaron problemas de comunicación, por lo que se intuye que esta versión de pantalla opera en ciclos de 0.5 μ s.

5.5 Memorias RAM y ROM.

La arquitectura utiliza dos memorias, una de tipo RAM y otra de tipo ROM. La memoria RAM es estática² y la ROM tipo EEPROM³. En la memoria ROM se almacena todo el sistema operativo y los programas del medidor.

En la memoria RAM se hace toda la descarga del sistema operativo y del medidor. Las memorias utilizadas fueron la 28C64 (EEPROM) y la 62C256 (RAM).

Estas memorias son totalmente compatibles con el procesador HC11. No se presentó ningún problema con su acoplamiento.

Al observar el diagrama del circuito, se puede ver que se utiliza una memoria de 32kB en un mapa de memoria dividido en bloques de 8kB. El circuito utilizado para habilitar los cuatro bloques que una memoria de 32kB consiste en un arreglo de diodos.

5.6 El reloj de tiempo real (RTC).

Como parte de la arquitectura se propuso el uso de un reloj. Este no representa una necesidad en el aparato medidor, pero se propuso como un elemento de un sistema con microprocesador.

El reloj utilizado es el MC146818⁴ fabricado por Motorola. Este es un reloj de uso común en arquitecturas basadas en microprocesadores. Sus características son:

1. Interface compatible a procesadores Intel y Motorola.
2. Reloj de tiempo real flexible, es decir, permite la programación del reloj de manera binaria o BCD, formato de 12 o 24 horas, alarmas, corrección por horario de verano, etc.
3. Memoria RAM de 50 bytes adicional.
4. Modo de operación en bajo consumo ("standby").

A pesar de que este reloj ha sido diseñado para ser usado con procesadores Motorola, existe un problema al intentar utilizarlo con el HC11 tratando de aprovechar el bus de direcciones multiplexado. Si se observa el diagrama de tiempos de lectura del reloj (fig. 5.2) se puede ver que la terminal CE debe estar habilitada antes de la habilitación de la señal DS (que es el reloj E del HC11).

² Existen dos tipos de memorias comerciales, dinámicas y estáticas. Las memorias dinámicas utilizan el principio de carga de capacitores para almacenar datos, evidentemente requieren de ciclos de refresco para conservar la información. Las memorias estáticas utilizan circuiteria lógica (flip-flop) para mantener su estado. La diferencia en velocidades favorecen a las estáticas, la diferencia de consumo eléctrico favorecen a las dinámicas.

³EEPROM: memoria que puede ser borrada eléctricamente desde sus terminales.

⁴Motorola Inc., "Motorola Microprocessor Data".

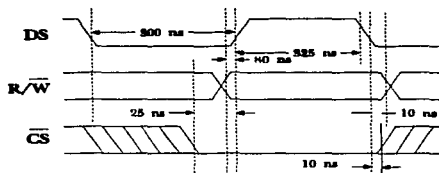


Fig. 5.2 Tiempos del RTC.

Esto no es posible con el esquema tradicional de habilitación del decodificador (que está habilitado con la señal E). Si el decodificador es habilitado por la señal E, entonces es de esperarse que la habilitación del reloj llegue después de la señal DS (que es E del HC11) por el tiempo de retraso en el decodificador. Esto se solucionó haciendo la habilitación del decodificador LS138 con la señal de aseguramiento de direcciones AS en lugar del reloj E, ya que el aseguramiento de las direcciones se da mucho antes que la habilitación del reloj E.

Se contemplaron dos esquemas de respaldo de energía; una de ellas es de la siguiente forma, que es la propuesta por el fabricante:

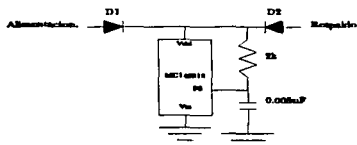


Fig. 5.3 Diagrama de respaldo con diodos y pila.

donde D1 es un diodo tipo Schottky (baja caída de voltaje) y D2 es un diodo convencional de silicio (caída de 0.5 a 0.7 V).

El tipo de pila de respaldo es de litio. Las pilas de litio son las que hasta el momento ofrecen mayor duración, comercialmente hablando. Se estudió la posibilidad de utilizar una pila recargable de Niquel-Cadmio, pero las pilas recargables tienen varios inconvenientes:

1. Duran mucho menos, sin recarga, que las de litio.
2. Requieren carga periódica.
3. Se requiere circuitería adicional para recarga. Esta circuitería debe entregar una corriente de carga constante (aproximadamente el 10% del valor nominal) y monitorear el voltaje de gasificación.
4. El costo del dispositivo sube al incluir la circuitería de recarga.

En general, las pilas recargables necesitan mantenimiento. Dado que se desconoce el tiempo de inactividad que se le pueda dar el medidor y el tiempo de uso, no se puede asegurar que la pila opere en condiciones óptimas. Las pilas recargables son especialmente sensibles a daños con tiempos de descarga prolongados y tiempos de carga incompletos. Es por estas razones que se decidió poner pila de litio y no recargable.

Se hicieron pruebas para determinar la duración de la pila. Si el cristal del reloj opera en los 4.19 MHz, la pila dura aproximadamente un mes (no se usó el sistema durante este mes). Si se utiliza la frecuencia más baja de oscilación del reloj (37.7 kHz), la pila puede durar hasta más de 10 veces (un año).

La pila es de un voltaje de 6 [V]. Con este voltaje y con el circuito de respaldo mostrado, la pila mantiene al reloj operando hasta que su voltaje baje a:

$$V' = 5.0 - (V_{\text{voltaje del diodo Schottky}}) + (V_{\text{voltaje del diodo de silicio}}) = 5.0 - 0.2 + 0.7 = 5.5 \text{ [V]}$$

Es decir, que hasta que la pila disminuya su voltaje a 5.5, por el uso, entonces se alternarán la fuente y la pila para alimentar el reloj. Mientras el voltaje de la pila se mantenga arriba de los 5.5 [V], el reloj operará solamente con la pila. Esto no presenta ningún inconveniente.

Si se quisiera alternar continuamente entre la fuente y la pila utilizando esta forma de conexión, se puede escoger una pila de un voltaje menor y obtener un voltaje regulado, a partir de la fuente, igual al de la pila.

La otra alternativa es diseñar un interruptor electrónico que conecte y desconecte la pila según la fuente se apague o encienda. Para evitar la disipación de energía en un circuito de este tipo se debe procurar utilizar circuitos de efecto de campo. El circuito utilizado para este sistema es:

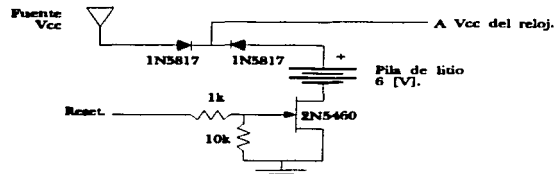


Fig. 5.4 Respaldo con pila e interruptor electrónico.

Este circuito cierra la conexión a la pila cuando la señal de "reset" permanece apagada (que es cuando el voltaje de alimentación están por debajo del voltaje de monitoreo del circuito MC34064 (típico de 4.6 [V]), es decir, cuando la fuente se apaga. El circuito conecta la terminal negativa de la pila; esto se debe a que el FET de canal N tiene un voltaje de apagado de aproximadamente 3.0 [V] entre las terminales

de compuerta ("gate") y fuente ("source"), de esta forma el voltaje de fuente siempre esta referenciado a la línea de tierra del circuito. Si en lugar de conectar y desconectar la terminal negativa de la pila se intentara con la positiva se encontrará que el circuito no opera; esto se debe a que la corriente que pasa por el FET es tan pequeña que la caída a través del FET es despreciable, lo que hace que el voltaje de fuente sea muy cercano al voltaje de drenaje ("drain"), lo que haría imposible encender y apagar al FET, pues no se alcanza el voltaje de apagado en la compuerta.

5.7 La interface RS-232.

El controlador HC11 tiene un puerto serial de comunicaciones que puede ser compatible con el estándar RS-232. Para hacerlo totalmente compatible se deben acoplar los niveles de voltaje del HC11 a los establecidos por el estándar RS-232, los cuales son:

9 [V] < estado alto < 12 [V].

-9 [V] < estado bajo < 12 [V].

Existen circuitos que realizan esta conversión de señales, sus diferencias radican básicamente en la obtención del voltaje especificado por la norma RS-232, capacidad de carga, y en el número de señales que pueden manejar ("fan-out"). Para el caso del HC11 solamente se necesitan las señales de transmisión (TX) y la de recepción (RX). El circuito que se utilizó es el MAX232⁵, que tiene su propio generador de voltajes RS-232. La conexión es la sugerida por el fabricante y se puede observar en el diagrama del circuito.

5.8 Regulación de voltaje y circuito de supervisión.

Para la regulación de voltaje se utilizó el circuito LM7805⁶, que es un regulador de 5 [V] a 1 [A]. Este circuito se utiliza comúnmente en circuitos digitales por su buena regulación en condiciones de operación no extremas. Sin embargo, a pesar de su buena regulación, se utilizan capacitores en cada uno de los componentes para evitar ruido por conmutación.

Se utiliza también un supervisor de voltaje, circuito MC34064⁷ de Motorola. Es el sugerido por Motorola para una arquitectura basada en el HC11. Su finalidad es reinicializar al procesador en caso de variaciones de voltaje y permitir un correcto arranque al encender la fuente. El supervisor de voltaje inicializa al procesador si la tensión de alimentación desciende por debajo de los 4.6 [V].

La conexión utilizada es la sugerida por el fabricante: se monitorea el voltaje de la fuente a través de la terminal 2. La terminal 3 esta configurada en colector abierto, de tal forma que al descender más allá del voltaje de supervisión, el circuito obliga a la terminal RST a un estado bajo. Mientras el voltaje sea superior al de supervisión, la terminal RST permanece en alta impedancia.

⁵Maxim Co., "RS232 Drivers Receivers", <http://www.maxim-ic.com>.

⁶Motorola Inc., "Linear and Interface Integrated Circuits", Motorola Inc.

⁷Motorola Inc., "Linear and Interface Integrated Circuits", Motorola Inc.

5.9 Conversión D/A.

La conversión D/A para entregar una señal de voltaje al controlador del motor, se hace en el circuito DAC 0808⁸. Es un convertidor D/A de ocho bits.

La importancia de este dispositivo radica, como se mencionó en el capítulo 3, en que además de ser el elemento que enlaza al mundo del control digital con el mundo de las señales analógicas, es un elemento aislante de circuitos. El aislamiento se hace mediante optoacopladores entre el registro y el convertidor DAC. Mientras el registro está alimentado por la arquitectura HC11, el convertidor DAC está alimentado por la fuente del controlador del motor.

El convertidor opera con corriente, por lo que utiliza un convertidor de corriente a voltaje y de ésta manera se entrega el rango de voltajes que necesita el motor, que es de 1 [V] a 13.5 [V]. Para una señal digital de 0h le corresponde una señal de voltaje de 1 [V], y para un valor de FFhex le corresponde un valor analógico de 13.5 [V]. La descripción de este circuito está en el capítulo 3.

El registro de escritura para controlar al motor esta en la dirección 8000hex. Es un registro comercial, 74LS373. En el diagrama de la arquitectura puede verse la conexión de este registro.

5.10 Conversión A/D.

Para la conversión A/D se utiliza el convertidor del HC11. La técnica de conversión que utiliza el procesador HC11 es la de distribución de carga.

Para la conexión del transductor de presión se requiere una conversión de voltaje a corriente, la descripción de este circuito esta detallada en la segunda parte del capítulo 3; hay que destacar el uso de un filtro paso baja en la entrada del convertidor A/D. La finalidad de este filtro es el de permitir el paso de la señal que se va cuantificar (evitando el "aliasing"). Esencialmente es un limitador de ancho de banda que evita se cuantifiquen señales no deseadas, que pueden ser generadas por la conmutación de los circuitos, inducción por campos eléctricos cercanos, ruido, etc. El criterio para el diseño de esta filtro está directamente relacionado con la frecuencia de muestro y con las señales que se deben rechazar. Se debe diseñar de tal manera que la frecuencia de muestreo sea, al menos, del doble de la frecuencia de la señal y que se supriman todas aquellas componentes que no se desean⁹.

La protección de la terminal del convertidor es con un diodo zener de 5.5 [V] de voltaje de conducción. Esta configuración protege contra voltajes superiores a los 5.5 [V] y contra voltajes inferiores a -0.7 [V].

⁸National Semiconductors, "Data Aquisition Devices".

⁹En el capítulo 3 se diseña este filtro.

Capítulo 6. Programación.

Como se mencionó al principio de este trabajo, uno de los objetivos personales es desarrollar un sistema operativo completo para el procesador seleccionado, el HC11 de Motorola. Una de las ideas al plantear este proyecto fue la de desarrollar un proyecto modular, es decir, que el cliente pudiera seleccionar los elementos que más le convengan. Por ello se trabajó con la finalidad de lograr un aparato independiente, que determine la permeabilidad y la muestra en una pantalla, y un conjunto de herramientas para PC, una que opera bajo Windows y la otra bajo DOS, para facilitar el manejo de la información mediante otros tipos de herramientas matemáticas o estadísticas que se encuentran disponibles para cualquier PC. Desafortunadamente la versión DOS no fue terminada, sólo se trazaron los elementos básicos.

La idea principal al elaborar estos programas fue la de utilizar algunos conceptos que se encuentran en los sistemas operativos modernos. Como referencia se tomó principalmente el modelo de programación utilizado por Windows y C++.

Es por ello que se decidió programar para plataformas Windows y DOS. Sin embargo, estos conceptos se trataron de llevar hasta la programación del controlador HC11. Sin embargo, dadas las herramientas con que se contaron para programar, no se logró lo esperado. Esto se debió básicamente a la dificultad de programación en lenguaje ensamblador, y lo que ello implica, como no disponer de las herramientas en tiempo de compilación como las que se tienen al utilizar lenguajes de alto nivel (Pascal, VBX, C, etc.) como son: nombres de variables sin restricción alguna, tipos definidos (enteros, flotantes, booleanas, estructuras, etc.), paso de parámetros y regresos de valores automáticos, mecanismos de herencia, creación y destrucción automáticos, depuración dinámica ligada al programa fuente, etc... Bastaría con revisar los programas realizados en ObjectWindows(C++) o C, y compararlos con el programa en ensamblador del HC11.

También hubo dificultades en la depuración del programa. Esto influyó en el tiempo de desarrollo del sistema. Existen depuradores y simuladores para el HC11, pero la tarea se volvió titánica sobre todo cuando se han alcanzado miles de líneas de programación y se utiliza memoria dinámica, como se realizó en este sistema.

Este capítulo está separado en varias partes: programación del sistema HC11 y programación para ambiente Windows. En la parte referente a la versión Windows no se presenta ninguna idea nueva.

Los programas pueden consultarse en los apéndices E y G.

6.1 Definición de sistema operativo.

Un sistema operativo puede verse como un conjunto de programas que tiene como finalidad ejercer un control sobre los elementos que componen un sistema de cómputo y que proporciona el enlace entre un programa de usuario y los recursos del procesador.

Los programas de usuario hacen llamadas o invocan al sistema para completar una tarea, a ésto se le denomina llamadas al sistema.

Desde el punto de vista de ejecución de procesos se pueden clasificar a los sistemas operativos en dos grandes bloques: los que permiten multiproceso y los que trabajan sobre una sólo tarea o multitarea¹.

Los conceptos de multiproceso, o proceso paralelo, y multitarea van muy ligados. Un sistema que permite el proceso paralelo debe tener la capacidad de atender a varios programas de usuario al mismo tiempo (ej. procesadores vectoriales). Mientras que un sistema que permite multitarea no necesariamente atiende varios procesos al mismo tiempo. Este es el caso de los programas diseñados para ambiente Windows en PC. Los procesos registrados comparten el procesador en lo que se llama tiempo compartido. Un proceso o tarea utiliza un cierto tiempo al procesador y después lo cede a otro proceso o tarea. En términos no muy rigurosos, las tradicionales PC pueden realizar tareas en proceso paralelo, pues mientras el procesador está ocupado en una tarea, el coprocesador numérico o el DMA² pueden estar realizando otras tareas; pero el usuario no tiene control sobre tales tareas y están más enfocadas al funcionamiento interno de la arquitectura PC.

El esquema de multitarea propuesto para el HC11, dadas sus limitaciones, está basado en el concepto de tiempo compartido.

6.2 Conceptos generales de sistemas operativos.

6.2.1 Multitarea.

Es el esquema de funcionamiento de una arquitectura basada en procesador que permite la ejecución de varios programas de usuario alternadamente. Existen varias técnicas para asignar tiempos a los programas. Algunos sistemas regulan el tiempo en que un programa está activo en intervalos fijos. Otros, como Windows, permiten al programa que utilice los recursos del sistema el tiempo que sea necesario; es el programa el que cede el control al sistema para efectuar el cambio de tarea. En estos sistemas los programas deben estar diseñados para responder al usuario de la manera más rápida y sencilla. El esquema de mensajes y eventos permite a los programadores cumplir con esta exigencia del sistema.

¹ El sentido de proceso o tarea es muy ambiguo, sin embargo se prefiere utilizar el término multitarea o tarea en aquellos sistemas que no realizan programas simultáneamente; el término multiproceso o proceso se emplea en los sistemas que permiten la ejecución de varios programas simultáneamente.

² DMA: Direct Memory Access; dispositivo que evita al procesador hacer movimiento de grandes cantidades de memoria entre los dispositivos periféricos.

6.2.2 Multiproceso.

Es el esquema de un sistema que tiene uno o varios procesadores que pueden ejecutar programas de usuario al mismo tiempo. Normalmente estas arquitecturas y sistemas operativos son sofisticados y se utilizan en sistemas administradores o servidores de cómputo. Este esquema es muy útil en programas donde los problemas matemáticos son característicos o en aquellos sistemas multiusuarios.

6.2.3 Administración de recursos.

Como se mencionó antes, es el sistema operativo el encargado de controlar el uso de los periféricos por parte de los programas de usuario. Es evidente que se simplifica el trabajo de un programador cuando la responsabilidad de el uso de los periféricos queda en un sólo elemento: el sistema operativo. Además esto permite que cuando se realizan modificaciones a los sistemas operativos el programa de usuario no debe ser modificado. Existen muchas formas de hacer las llamadas al sistema para el control de un recurso y básicamente dependen de las características del procesador y la arquitectura. En el caso de la PC, por ejemplo, se hacen llamadas al sistema a través de interrupciones de "software".

La importancia de una administración de recursos reside, principalmente, en optimar el uso de los mismos cuando varios programas requieren de su uso. Un ejemplo clásico son las unidades de memoria permanente o discos duros. Dejando al sistema operativo las tareas de escritura y lectura liberan al programador de diseñar rutinas exclusivas para el control del disco donde el uso ordenado de este recurso entre varios programas es clave. El sistema operativo proporciona un enlace seguro y confiable en la administración de periféricos o recursos.

6.2.4 Interrupciones.

Las interrupciones son un mecanismo de trabajo de los procesadores que permiten romper la secuencia normal de un programa. Existen dos tipos de interrupciones: de tipo "hardware" y de tipo "software". Las interrupciones de tipo "hardware" se producen, normalmente, activando alguna terminal o pin del procesador o internamente provocando una señal eléctrica al procesador. Las interrupciones de "software" se generan mediante una instrucción explícita del programa.

El mecanismo de atención a una interrupción de cualquiera de estos dos tipos es: al generarse la interrupción, por activación eléctrica o por una instrucción, el procesador detiene la ejecución del programa y verifica qué de tipo de interrupción se trata. Una vez identificada se toma el vector de esa interrupción. El vector no es otra cosa que la dirección en la que el procesador debe continuar y puede ser modificado por el usuario libremente.

Las interrupciones de "hardware" toman su vector correspondiente de una tabla localizada en una región específica de la memoria o dentro del procesador mismo, este procedimiento es automático y no depende del usuario.

Las interrupciones de "software" al igual que las de "hardware" toman su vector de una tabla localizada normalmente en una región de memoria o dentro del procesador. Si el procesador soporta más de una interrupción de software (como en la PC) el usuario identifica la interrupción enviando el tipo de interrupción que desea en algún registro del procesador. El proceso de identificación a través del registro es automático y no depende del usuario.

Existe otro mecanismo de interrupciones que es especialmente útil en sistemas multitarea o multiproceso, se denomina excepción. Las excepciones se generan cuando una instrucción no puede ejecutarse por alguna razón. Un ejemplo clásico es cuando en la PC se trata de acceder un segmento de memoria que no está presente o definido, o una simple división entre cero. En este mecanismo también existen varios tipos de excepciones que son automáticamente identificados por el procesador.

Las interrupciones proporcionan un mecanismo eficaz para responder a los periféricos o para comunicarse con el sistema operativo.

Una vez terminada la ejecución del programa que responde a una interrupción, el procesador permite resumir el programa interrumpido mediante una instrucción explícita. El regreso al programa interrumpido se hace al encontrar la instrucción de regreso de interrupción y se recupera, de algún lugar de almacenaje, normalmente la pila, la dirección donde se rompió el programa.

6.2.5 Mensajes y eventos.

Es un esquema de programación en el que la secuencia del programa se conduce por sucesos (eventos) o por mensajes (comunicación entre programas, procesos o tareas). Normalmente los eventos pasan cuando se genera una interrupción. Tal interrupción es atendida por un programa especial y se encarga de notificar al programa de usuario que posee tal recurso o interrupción mediante un mensaje debidamente identificado. El programa recibe el mensaje y lo interpreta como un evento y responde a tal. Los mensajes pueden ser enviados de un programa a otro sin que se haya generado un evento.

El esquema de mensajes y eventos es muy eficaz cuando el sistema depende en gran medida de la generación de interrupciones. También es un método muy ordenando y claro de programación.

6.2.6 Algunos conceptos de programación orientada a objetos.

La programación orientada a objetos es una de las técnicas o modelos de programación que comúnmente se utilizan más por varias razones:

1. Permite una abstracción del problema independientemente del lenguaje de programación.
2. Utiliza el manejo de memoria dinámica, lo que lleva a un ahorro en el uso de memoria además de administrarla de una manera más óptima.
3. Permite construir objetos complejos a partir de objetos elementales.

La programación orientada a objetos trata de entender al mundo como una colección de cosas, las que tienen características o propiedades y tienen métodos o procedimientos para modificarse (alterar tales propiedades) ³). También trata de entenderlo como objetos que están formados a partir de objetos más generales. Como ejemplo: todos entendemos el concepto de pelota, pero dentro de pelota existen varios tipos de pelotas. Así una pelota para fútbol engloba al objeto genérico pelota (porque es semiesférica y rueda). O bien, entender al objeto pelota como un caso particular de un objeto esfera o esferoide.

La programación orientada a objetos tiene 4 principales postulados:

1. Encapsulamiento: es la descripción de la estructura interna del objeto, de cómo se relaciona con otros objetos, y algunas formas de protección que aísla los detalles funcionales de objeto con respecto al exterior.
2. Jerarquía: se refiere a desarrollar una jerarquía de objetos donde exista un objeto raíz
3. superobjeto.
Herencia: se refiere al hecho de generar nuevos objetos a partir de un superobjeto u objeto raíz.
4. Polimorfismo: se refiere a la comunicación entre los superobjetos y los objetos derivados.

6.2.7 Código reubicable.

Se refiere a la posibilidad de poder cargar y ejecutar el código de un programa en cualquier dirección del mapa de memoria. Este esquema es muy útil y es obligado en cualquier arquitectura comercial. Existen muchos esquemas para trabajar código reubicable y dependen en gran medida de la manera de acceder la memoria por parte del procesador. En PC se utiliza el mecanismo de segmentos y desplazamientos, que es un puntero base y un puntero para desplazamientos. En el modo protegido de las máquinas 386 en adelante y otras arquitecturas el mecanismo es de paginación, que es también un esquema de una dirección base y una serie de desplazamientos sobre esa base. El responsable de la correcta ubicación de los programas es el sistema operativo.

6.3 Modelo de programación para el sistema basado en el HC11.

Los ejes directivos en la programación del HC11 fueron:

1. Utilizar algunos conceptos de programación orientada a objetos.
2. Multitarea.
3. Respuestas en función de eventos o sucesos.
4. Comunicación entre el sistema y los programas mediante eventos.

³ Borland define un objeto como a una región identificable de memoria que contiene un valor fijo o variable. Se utiliza un identificador para acceder al objeto. "Programmers Guide. Borland C++ for DOS, Windows and Windows NT", Borland Inc.

Como se dijo anteriormente, se tomó como modelo el sistema operativo Windows. Windows es un sistema programado con el modelo de objetos (aunque en un principio no lo fue), que reacciona en función de los eventos o mensajes que ocurren, y que tiene un mecanismo de operación en multitarea (tiempo compartido).

Las diferencias entre el HC11 y los procesadores que utilizan sistemas operativos como Windows, Windows NT, OS2, Unix, etc... son abismales. Hay que recordar que el HC11 es un procesador diseñado para control, es decir, de diseño específico, y no un procesador para ejecución de muchos programas y de distintos tipos (lo que sería un procesador de propósito general⁴, como la familia x86). Teniendo ésto en mente se desarrollaron las ideas anteriormente enumeradas, con la consiguiente limitación: no se puede tener un esquema que cubra totalmente estas ideas. Lo que limita principalmente al sistema en el procesador HC11 es:

1. Una arquitectura de bajo costo, lo que hace escasa la colección de periféricos.
2. Su mapa de memoria restringido a 64k (aunque existen mecanismos para poder acceder más memoria, implicando un incremento en tiempos de lectura/escritura).
3. El hecho de contar con solamente dos registros de propósito general (ocho bits) y dos para direccionamiento indexado.
4. El que es un procesador de 8 bits (limitando la manipulación de datos).
5. No existe el esquema de niveles de privilegios.
6. No existe mecanismo de excepciones.
7. No existe el concepto de protección de datos o memoria.
8. No existe un mecanismo que permita la generación de distintos tipos de interrupción a través de la terminal INT del microcontrolador⁵.

A continuación se describe el desarrollo de las ideas principales para el HC11.

6.3.1 Descripción general del sistema operativo e inicialización.

El sistema operativo y el programa de control para el medidor de permeabilidad están grabados en la ROM de la parte alta de memoria (a partir de la dirección E000H). En la parte de inicialización, que se

⁴ Hay que aclarar que la distinción entre procesadores de propósito general y de diseño específico es muy ambigua. Lo que se quiere decir es que los procesadores que se utilizan en máquinas que operan bajo Windows y OS/2 están diseñados para cubrir las exigencias del usuario que pueden variar desde cálculos matemáticos complejos hasta manipulación de datos en grandes cantidades (gráficos, procesadores de texto, etc...) haciéndolos procesadores más generales que uno que está dedicado exclusivamente a control.

⁵ En varios tipos de procesadores, se contempla el uso de circuitos PIC (controlador de interrupciones programable) lo que permite identificar, utilizando solamente la terminal o pin INT del procesador, de alguna forma (leyendo al PIC que entrega información) qué interrupción se generó en el PIC y asignarle distintos vectores para las distintas interrupciones.

ejecuta a partir de E000H, se descarga el sistema a partir de la dirección 200H. Estas rutinas que se descargan son propiamente del sistema operativo. En este bloque de datos pueden distinguirse varias partes:

1. Sección denominada SYS, que son rutinas para funcionamiento de la lógica del SO.
2. Sección con los métodos para manejar el puerto de comunicaciones serial, SCI.DRV.
3. Sección con métodos para manejar el reloj de tiempo real interno del HC11, RTI.DRV.
4. Sección con los métodos para manejar el convertidor A/D, ADC.DRV.
5. Sección para operar la zona de almacenaje en memoria ROM, STODEV.DRV.

Conforme se avanza en la ejecución de la inicialización del HC11 se encuentran varias llamadas a sistema para configurar los vectores de interrupción.

Para llamar a los diversos servicios del sistema, se utiliza el siguiente mecanismo:

```
ldaa    #SWI
ldab    #SWI_setvect
swi
```

donde: el registro A contiene el tipo de servicio, que puede ser:

```
SWI      equ    $01    ;servicios de SWI
SYS      equ    $02    ;servicios de sistema
STODEV   equ    $03    ;servicios de almacenaje ROM
SCI      equ    $04    ;servicios de comunicación
RTI      equ    $05    ;servicios de RTI
ADC      equ    $06    ;servicios de conversión
LCD      equ    $07    ;servicios de LCD
RTC      equ    $08    ;servicios de reloj
```

y el registro B contiene el tipo de función a llamar para cada uno de estos servicios. Es el sistema el que identifica el tipo de interrupción leyendo al registro A. Con este mecanismo se pueden generar hasta 255 interrupciones.

De esta manera, un programa de control solamente tendría que llamar a la función adecuada para obtener el uso de un recurso o configurar un recurso, sin necesidad de escribir directamente sobre los registros de control del recurso. El paso de parámetros es similar el mecanismo utilizado en C, a través de la pila, sólo que en este sistema se envía la dirección de la pila donde están los parámetros en la variable pública del sistema SYS.wparam. Esto se ilustrará un poco más adelante.

Continuando con el proceso de inicialización, el primer vector que se debe configurar es el vector SYS:

```
ldaa    #SYS
psha
ldx     #SYS.handler
pshx
tax
```

```

    stx          SYS.wparam
    ldaa        #SWI
    ldab        #SWI_setvect      ;setVect(uchar Num,lp Handler)
    swi
    pulx
    pula

```

La función `SWI_setvect` recibe como parámetros: el número de servicio y el vector de las rutinas que atenderán la petición. Puede verse el mecanismo de envío de parámetros, se colocan en la pila el número de servicio y el vector; después se envía la dirección donde quedaron los parámetros a través de la variable `SYS.wparam`. El contenido de esta función puede verse en el listado del programa, apéndice E.

Lo que realiza la rutina de servicio `SWI_setvect` es colocar la dirección enviada como parámetro en el vector correspondiente a `SYS`. De esta manera, los servicios de `SYS` quedan accesibles. Lo que se tendría que hacer es llamar a `SYS` con el número de función deseado. El programa de inicialización continúa colocando las direcciones de las rutinas de servicio para `STODEV.DRV` (rutinas de almacenaje en ROM), `RTI.DRV` (rutinas del reloj de tiempo real), `ADC.DRV` (rutinas del convertidor A/D), `SCI.DRV` (rutinas del puerto serial). Hay que hacer notar que las direcciones corresponden al intérprete de función de cada uno de estos servicios: `STODEV.handler`, `RTI.handler`, etc...

Una vez inicializados los vectores de servicio, el sistema verifica la existencia de la pantalla de cristal líquido con una simple rutina que escribe un "1" en la pantalla y espera la respuesta de ésta, previamente se configuró para 2 renglones, sin cursor y sin corrimientos. Este procedimiento es el sugerido por el fabricante.

Una vez detectada la pantalla de cristal líquido, se llama al servicio de instalación de controlador de recurso. Hay que remarcar varios puntos respecto a los controladores anteriormente mencionados:

1. La pantalla LCD es externa al procesador, por ello debe ser instalado el controlador de manera explícita.
2. Las rutinas LCD están guardadas en ROM, por lo que deben ser cargadas en la memoria RAM.
3. Se debe utilizar un método especial del sistema para colocar el controlador.

La manera de instalar un controlador ajeno al procesador es:

```

    ldd          #SYS_INSTALL      ;mensaje para instalar controladores
    std          $1,0 msg
    ldd          #rutaslcd
    std          SYS.wparam
    ldaa        #LCD              ;número de servicio que ocupara
    staa        SYS.lparam1
    ldaa        #SYS
    ldab        #MSG
    swi

```

En este llamado al sistema se utiliza un nuevo mecanismo de comunicación entre objetos: mensajes. Para ello se utiliza la estructura pública de SYS:

```

SYS.msg          fdb      0
SYS.wparam       fdb      0
SYS.lparam       fdb      0
SYS.lparam1      fdb      0

```

Tiene la misma conformación que la estructura para mensajes utilizada en Windows: msg es tipo INT (entero), wparam es tipo WORD (palabra), y lparam es tipo LONG INT⁶. El elemento lparam está dividido en dos, lparam1 y lparam, pues el HCl 1 no soporta tipos de 32 bits.

El controlador de SYS interpreta un mensaje cuando el registro B contiene cero. La respuesta a este mensaje consiste en localizar el programa en la ROM, obtener su PDT ("Program Descriptor Table"), buscar la memoria suficiente para tal programa y finalmente enviar al nuevo controlador el mensaje CM_INIT. La tabla de ubicación de programas está en la dirección F000H, y es sobre esta dirección donde el cargador busca el nombre del programa y obtiene su ubicación.

Cabe aquí hacer una breve descripción de la PDT, que se encuentra contenida en la zona descriptora del programa (FD), al inicio de cualquier programa que utilice este sistema:

```

LCD.DRV          fcc      "LCD.DRV"
                 fdb      $00aa
LCD.Descriptor   fcc      "reloc"
LCD.CompAddr     fdb      #LCD.DRV
LCD.EntryPoint   fdb      #LCD.handler-#LCD.Descriptor
LCD.StackSize    fdb      $0000
LCD.ProgramSize  fdb      #LCD.fin-#LCD.DRV
LCD.RelocAddr    fdb      #LCDa1+1
                 fdb      #LCDa2+2
                 fdb      #LCDa3+1
                 fdb      #LCDa4+1
                 fdb      #LCDa5+1
                 fdb      #LCDa6+1
                 fdb      #LCDa7+2
                 fdb      #LCDa8+2
                 fdb      #LCDa9+1
                 fdb      #outtext.contl+1
                 fdb      #outtext.finl+1
                 fdb      #LCD.espera+1
                 fdb      #LCD.default+1
                 fdb      #LCD.down+1
                 fdb      #LCD.minimize+1
                 fdb      #LCD.maximize+1
                 fdb      #LCD.dispatch+1
                 fdb      #LCD.msgdefault+1
                 fdb      #0000

```

⁶ INT: entero con signo, 16 bits; WORD: entero sin signo, 16 bits; LONG INT, doble entero, 32 bits.

```
LCD.FDEM          fdb          $55aa          ;Marca de final de archivo(FDEM)
```

El descriptor de programa contiene información sobre el programa:

- a) **Nombre del programa.**
- b) **"Descriptor":** que puede ser "reloc" si es reubicable en memoria. Es el único tipo que se desarrolló, con la idea de definir posteriormente programas no reubicables.
- c) **"CompAddr":** dirección de compilación. Este parámetro se tuvo que incluir debido a que el compilador utilizado compila en el formato S19 de Motorola⁷, y no se puede compilar sobre la dirección 0H y cargar en la zona de almacenaje de la ROM con las herramientas de programación y depuración utilizadas.
- d) **"EntryPoint":** desplazamiento sobre el origen del programa donde se encuentra el punto de entrada a este programa.
- e) **"StackSize":** tamaño de la pila necesaria para ejecutar este programa. No fue desarrollado este aspecto del programa. El cargador reserva memoria externamente al programa para la pila.
- f) **"ProgramSize":** contiene el tamaño del programa. Esta información es necesaria para el cargador.
- g) Hasta este punto está conformada la PDT. El siguiente grupo de datos esta formado por:
- h) **"RelocAddr":** direcciones que deben ser modificadas de acuerdo a la región de memoria donde fue cargado el programa. Estas direcciones son modificadas por el propio programa en la función de respuesta al mensaje CM_INIT. Como indicador de fin de lista de direcciones esta en valor 0hex.
- i) Para indicar fin de la tabla descriptora de archivo se emplea la secuencia 55AAH

En el listado del programa para la pantalla de cristal líquido (LCD) puede observarse la función de respuesta a mensaje CM_INIT, que es la responsable de modificar los vectores de los métodos de este controlador y de modificar las direcciones indicadas en la lista.

El mecanismo de carga de un programa que no sea un controlador difiere un poco. Puede verse esto en la función de respuesta al evento CM_STARTUP del intérprete del sistema operativo:

```
ldx          #tesisfile
pshx
tsx
stx          SYS.Wparam
ldaa        #SYS
ldab        #SYS_load
swi
pulx
```

⁷ Este formato genera archivos con la dirección de compilación incluida.

LCD.FDEM

fdb

\$55aa

;Marca de final de archivo(FDEM)

El descriptor de programa contiene información sobre el programa:

- a) Nombre del programa.
- b) "Descriptor": que puede ser "reloc" si es reubicable en memoria. Es el único tipo que se desarrolló, con la idea de definir posteriormente programas no reubicables.
- c) "CompAddr": dirección de compilación. Este parámetro se tuvo que incluir debido a que el compilador utilizado compila en el formato S19 de Motorola⁷, y no se puede compilar sobre la dirección 0H y cargar en la zona de almacenaje de la ROM con las herramientas de programación y depuración utilizadas.
- d) "EntryPoint": desplazamiento sobre el origen del programa donde se encuentra el punto de entrada a este programa.
- e) "StackSize": tamaño de la pila necesaria para ejecutar este programa. No fue desarrollado este aspecto del programa. El cargador reserva memoria externamente al programa para la pila.
- f) "ProgramSize": contiene el tamaño del programa. Esta información es necesaria para el cargador.
- g) Hasta este punto está conformada la PDT. El siguiente grupo de datos esta formado por:
- h) "RelocAddr": direcciones que deben ser modificadas de acuerdo a la región de memoria donde fue cargado el programa. Estas direcciones son modificadas por el propio programa en la función de respuesta al mensaje CM_INIT. Como indicador de fin de lista de direcciones esta en valor 0hex.
- i) Para indicar fin de la tabla descriptora de archivo se emplea la secuencia 55AAH

En el listado del programa para la pantalla de cristal líquido (LCD) puede observarse la función de respuesta a mensaje CM_INIT, que es la responsable de modificar los vectores de los métodos de este controlador y de modificar las direcciones indicadas en la lista.

El mecanismo de carga de un programa que no sea un controlador difiere un poco. Puede verse esto en la función de respuesta al evento CM_STARTUP del intérprete del sistema operativo:

```
ldx          #tesisfile
pshx
tsx
stx          SYS.wparam
ldaa        #SYS
ldab        #SYS_load
swi
pulx
```

⁷ Este formato genera archivos con la dirección de compilación incluida.

```

ldd          intRegreso
cpd          #$ffff
bne         startup.cont1
ldx         #Idleerror
jsr         display
stop

tsx
std         CREATE.Entry,x          ;Se obtiene puntero a estructura "create".
ldd         #tesisfile             ;Se manda el "entry point".
std         CREATE.Nombre,x        ;Se manda puntero a cadena nombre.
ldy         #128
pshy
tsy
sty
ldaa        SYS.wparam
#SYS        ;"Malloc" regresa puntero al principio del bloque;
ldab        #SYS_malloc            ;hay que ajustarlo al
swi         ;final para poder usarlo de pila.
puly
ldd          intRegreso
cpd          #$ffff
bne         startup.cont2
ldx         #Idleerror
jsr         display
stop

```

En este caso se utilizan las funciones STODEV_load, para colocarlo en memoria, SYS_malloc para reservar la pila, el mensaje al sistema CM_CREATE para dar de alta ese proceso, y finalmente el mensaje al sistema CM_RUN. Las funciones "display" son exclusivamente para depuración, no deberían estar incluidas en el sistema una vez completado. Los aspectos para dar de alta un proceso y la asignación de memoria dinámica serán explicados más adelante.

Una vez analizada la estructura del sistema, se pueden concluir varias reglas de programación bajo este sistema operativo:

1. Para utilizar un recurso se deben utilizar los métodos de tal recurso a través del sistema de interrupciones, indicando en el registro A el tipo de recurso y en el registro B el método llamado, o bien, indicar un cero para interpretación de mensajes.
2. El mecanismo de envío de mensajes es a través de las variables públicas del sistema SYS.msg, SYS.wparam y SYS.lparam.
3. El mecanismo para envío de parámetros es a través de la pila e indicando la dirección donde se encuentran los parámetros en la variable pública del sistema SYS.wparam⁸.
4. El regreso de valores se hace a través de la variable pública intRegreso. En ella se envía el valor devuelto o la referencia a la ubicación del valor devuelto.

⁸ Esto incluso permitiría enviar parámetros no necesariamente a través de la pila, pues sólo se indica en SYS.wparam una dirección.

5. Los controladores de dispositivos externos o recursos deben ser cargados a través del mensaje al sistema CM_INSTALL, indicando el número de servicio o recurso que utilizan y el nombre del dispositivo.
6. Los programas de usuario deben ser cargados primero en memoria y después dados de alta. Se inicia su ejecución a través del mensaje al sistema CM_RUN, enviando el identificador del proceso que en este sistema coincide con el "EntryPoint" del programa⁹.
7. Los programas tienen un sólo punto de entrada. Los programas deben terminar con la llamada al método del sistema SYS_terminate.
8. El intérprete de mensajes del programa debe contener todas las posibles acciones de respuesta a un mensaje. El mensaje se envía a través de las variables públicas SYS.msg.
9. Los programas deben contener una zona descriptora del programa al principio del archivo, con los puntos indicados anteriormente.
10. El programa es responsable de la modificación de sus direcciones en caso de no cargarse en la zona de memoria para la que fue diseñada.
11. Si el programa quiere recibir mensajes de los distintos recursos, éste deberá llamar a los métodos de habilitación de notificaciones de tales recursos, y cuando ya no los necesite más deberá deshabilitar la notificación de tal recurso, de lo contrario se provocará inestabilidad en el sistema.
12. Un recurso no puede ser compartido, su dueño es único y exclusivo.
13. La pila del programa debe contemplar a lo mucho 3 niveles de interrupción al generarse un evento para él o para otro programa.
14. No se permite la activación de interrupciones hasta que se haya cedido el control del programa al sistema. Es decir, no se deberán activar interrupciones hasta que se haya completado la atención al mensaje previamente recibido. Esto significa que no existen mensajes anidados, en caso de activar interrupciones durante la respuesta a una interrupción, generará error y el sistema se detendrá.
15. El programa deberá enviar el control a la dirección que le envíe el sistema cuando reciba un mensaje CM_RESUME.
16. Un programa ejecutándose en "background" o segundo plano no tendrá acceso a servicios exclusivos a programas en "foreground" o primer plano, para poder determinar este estado se puede utilizar la función del sistema SYS_IsFore.
17. La mínima cantidad de memoria que entrega el sistema es 256 bytes.
18. Se dispone de tanta memoria como el sistema sea capaz de entregar.

⁹ La función SYS_load regresa el punto de entrada al programa "EntryPoint".

19. La memoria se reserva consecutivamente. Existe la posibilidad de fragmentar la memoria del sistema de tal manera que una aplicación no pueda ejecutarse porque consecutivamente no existe la cantidad solicitada, aunque físicamente exista.
20. Si un programa reemplazará algún método, debe hacerlo a través de los métodos del objeto afectado, nunca deberá hacerlo directamente sin notificar al objeto afectado.

6.3.2 Mecanismo de generación de eventos.

Para generar un evento, el controlador de recursos debe pedir al sistema que habilite la notificación de tal evento. Existen 16 posibles eventos en la arquitectura HC11:

Evento	Captura
EV_XIRQ	SYS.XIRQhandler
EV_IRQ	SYS.IRQhandler
EV_RTI	SYS.RTIhandler
EV_TOI	SYS.TOIhandler
EV_ILL	SYS.ILLhandler
EV_COP	SYS.COPhandler
EV_SCI	SYS.SCIhandler
EV_SPI	SYS.SPIhandler
EV_TIC1	SYS.TIC1handler
EV_TIC2	SYS.TIC2handler
EV_TIC3	SYS.TIC3handler
EV_TOC1	SYS.TOC1handler
EV_TOC2	SYS.TOC2handler
EV_TOC3	SYS.TOC3handler
EV_TOC4	SYS.TOC4handler
EV_TOC5	SYS.TOC5handler
EV_PAO	SYS.PAOhandler
EV_PAI	SYS.PAIhandler

La generación de eventos fue desarrollada únicamente para: SCI, IRQ y RTI. El resto de las funciones que responden a la generación de interrupciones están exclusivamente para depuración.

Se debe habilitar la notificación de evento mediante la función SYS_externoif o modificando la variable del sistema SYS.EV_FLAGS y colocando el vector usando al función SYS.setvector, enviando como parámetro un entero, en el que los bits en 1 habilitarán eventos de acuerdo a:

Evento	SCI	SPI	PAI	PAO	TOI	TOC5	TOC4	TOC3	TOC2	TOC1	TIC3	TIC2	TIC1	RTI	IRQ	XIRQ
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

La secuencia de generación de eventos es:

1. Se genera la interrupción.
2. El sistema verifica que tal evento esté habilitado verificando la variable SYS.EV_FLAG que tiene la estructura mostrada arriba.

3. Si la bandera está habilitada, verifica la URT¹⁰ de tal interrupción. Existen 16 URTs para cada una de las interrupciones posibles.
4. Si la URT, que previamente fue inicializada al llamar SYS_setcvect, es distinta de cero se hace un llamado al controlador indicado en la URT, notificando en la estructura SYS.msg el tipo de evento que se generó. El controlador tomará la acción adecuada. Si debe enviar el mensaje a un programa tendrá que hacer uso de la función SYS_dispatch. Esta función se encargará de enviar el mensaje al programa indicado por el controlador utilizando el mensaje al sistema CM_XGTASK. Esta función de respuesta obligará a activar el programa cuyo punto de entrada ("EntryPoint") coincida con el enviado por SYS_dispatch.
5. El programa responde al evento que le envíe el controlador y una vez terminada la acción cederá al control al sistema enviando un mensaje CM_XGTASK.
6. El sistema decide concluir la atención al evento al programa que lo generó, enviándole un mensaje CM_RESUME. El programa deberá enviar el control a la dirección indicada en SYS.wparam.
7. Se regresa el control al punto de ruptura.
8. Continúa la ejecución del programa en turno.

Esto quizás quede un poco más claro con el ejemplo de RTI en TES.EXE. TES.EXE en su función respuesta al evento CM_INIT hace el llamado al método RTI_enablenotif:

```

TESa5  ldx    #TES.handler ← Dirección del receptor de mensaje.
        pshx
        tsx
        stx    SYS.wparam
        ldaa  #RTI
        ldab  #RTI_enablenotif
        swi

```

RTI_enablenotif activa la bandera correspondiente en la variable SYS.EV_FLAGS, y pide a SYS que coloque el vector de RTI en la RTIUURT de SYS:

¹⁰ URT: User Response Table.

```

RTI.enablenotif      ldx    SYS.wparam      ;enable(intURT).
                    ldx    0,x          ;Obtiene URT.
                    cmpx   #S0
                    beq    RTIenable.dis ;Si URT es cero, deshabilita.
                    ldd    RTI.URT
                    cpd    #S0
                    bne    RTIenable.err1
                    stx    RTI.URT
                    ldd    SYS.EV_FLAG
                    orab   #RTIFLAG
                    std    SYS.EV_FLAG
                    anda   #S0
                    andb   #RTIFLAG
                    xgdx
                    pshx
                    ldx    #RTI.handler
                    pshx
                    tsx
                    stx    SYS.wparam
                    ldaa   #SYS
                    ldab   #SYS_setevvector
                    swi
                    pulx
                    pulx
RTIenable.dis       ldaa   TMSK2
                    oraa   #S40
                    staa   TMSK2
                    rts
                    ldaa   TMSK2
                    anda   #Sbf
                    staa   TMSK2
                    ldaa   TFLG2
                    oraa   #S40
                    staa   TFLG2
                    ldd    #S0
                    std    RTI.URT
                    rts

```

Suponiendo que se genere el evento RTI, el primer método llamado es RTI.handler que pertenece a SYS:

```

SYS.RTIhandler      ldx    intRegreso      ← Salva el regreso de funciones, para
                    stx    subintRegreso  ← restituirlo posteriormente.
                    jsr    SWI.savemsg    ← Salva la estructura MSG.
                    ldaa   TFLG2         ← Limpia la interrupción RTI.
                    oraa   #S40
                    staa   TFLG2
                    ldd    SYS.EV_FLAG    ← Verifica si está activada la
                                                habilitación del evento.

                    anda   #S0
                    andb   #RTIFLAG
                    beq    SYS.RTIifin
                    ldd    #EV_RTI       ← Coloca el mensaje EV_RTI.

```

```

std      SYS.msg
ldx      RTIURT          ← Obtiene URT y verifica su validez.
cmpx     #0
beq      SYS.RTIfin
ldab     #MSG
jsr      0,x            ← Llama al controlador del evento.
ldx      subintRegreso ← Restituye la variable intRegreso .
stx      intRegreso
SYS.RTIfin jmp      SWI.iRet      ← Recupera la estructura MSG y hace rti.

```

El regreso de esta interrupción no se hace hasta que el mensaje fue recibido por el controlador y por el programa receptor. El controlador recibe el mensaje EV_RTI en su punto de entrada:

```

RTI.handler      psha
                  tsy
                  ldx      #RTI.tabla
                  clr       0,y
                  cmpb     0,y
                  beq      RTI.casemsg
RTI.case      inc  0,y
                  cmpb     0,y
                  bne      RTI.Isdefault
                  pula
                  ldx      0,x
                  jsr      0,x
                  rts
RTI.Isdefault    cmpb     #RTI_enablenoti
                  bhi      RTI.default
                  inx
                  inx
                  bra      RTI.case
RTI.default      ldx      #RTI.fundefault
                  jsr      display
RTI.stop1        bra      RTI.stop1
RTI.casemsg      pula
                  ldd      SYS.msg
                  cpd      #CM_INIT
                  bne      RTI.msg1
                  bsr      RTI.init
                  rts
RTI.msg1         cpd      #EV_RTI
                  bne      RTI.defaultmsg
                  bsr      RTI.ev_rti
                  rts
RTI.defaultmsg   ldx      #RTI.msgdefault
                  jsr      display
RTI.stop2        stop
                  bra      RTI.stop2

```

Este bloque lo único que hace es identificar si se trata de mensaje o llamada a algún método del recurso. Si es mensaje entonces lo identifica y verifica su propia URT para decidir si debe enviarlo a algún destinatario:

```

RTI.ev_rti      ldx      RTI.URT
                cmpx      #0
                beq      ev_rti.fin
                pshx
                tsx
                stx      SYS.wparam
                ldd      #EV_RTI
                std      SYS.msg
                ldaa     #SYS
                ldab     #SYS_dispatch
                swi
                pulx
                rts
ev_rti.fin     ldx      #ev_rti.mens
                jsr      display
                stop
                bra      RTI.stop1
ev_rti.mens    fcc      "URT cero"
                fdb      #EOB

```

Finalmente, la función SYS_dispatch obliga a cambiar de proceso para encontrar al destinatario del evento. Una vez encontrado se transfiere el control al proceso a través de su punto de entrada, dejando en la estructura MSG el mensaje y el registro B con ceros para que el proceso identifique que se trata de un mensaje y encuentre su función de respuesta. Una vez terminada la función de respuesta el proceso llama al sistema a través del mensaje CM_XGTASK:

```

TES.handler    ldx      #TES.tabla
                cmpb     #MSG
                beq      TES.msgs
TESa1          ldx      #TES.msgdefault
                jsr      display
                stop
                rts
TES.msgs       ldd      SYS.msg
                cpd      #CM_INIT
                bne     TES.msg2
                bsr     TES.init
                cli
                bra     TES.Idle
TES.msg2       cmpd     #EV_RTI
                bne     TES.msg3
                ldx     0,x
                jsr     0,x
                bra     TES.Idle

```

```

TES.msg3      cmpd      #CM_INICIAR
              bne       TES.msg4
              ldx       2,x
              jsr       0,x
              bra       TES.Idle
TES.msg4      cmpd      #CM_PAUSA
              bne       TES.msg5
              ldx       4,x
              jsr       0,x
              bra       TES.Idle
TES.msg5      cmpd      #CM_EXIT
              bne       TES.msg6
              ldx       6,x
              jsr       0,x
              bra       TES.Idle
TES.msg6      cmpd      #CM_STARTUP
              bne       TES.msg7
              ldx       2,x
              jsr       0,x
              bra       TES.Idle
TES.msg7      cmpd      #CM_RESUME
              bne       TES.msgdefault
              ldx       SYS.wparam
              ldd       SYS.lparam1
              jmp       0,x
TES.msgdefault ldx       #TES.defaultmsg
              jsr       display
              ldx       SYS.msg
              jsr       showIX
              stop

TES.Idle      ldy       #tesis.hwnd
              clr       .posic,y
              ldd       #CM_XGTASK
              std       SYS.msg
              ldd       #50      ;Cambio no forzado.
              std       SYS.wparam
              ldaa      #SYS
              ldab      #MSG
              swi
              cli
              wai
              bra       TES.Idle

```

Puede verse, en primera instancia, cómo se resuelve si se trata de mensajes o llamadas a funciones, a través del registro B. Una vez detectado el mensaje se envía el control a la función de respuesta adecuada. Terminada la ejecución de la función se pasa el control a la función TES.Idle en la que se llama al sistema a través del mensaje CM_XGTASK. Esto permite al sistema terminar la operación del evento resumiendo al proceso en el que se generó tal evento. Una vez terminada la atención al evento, el sistema encontrará en otro contexto un mensaje CM_XGTASK que en algún momento volverá a activar a este proceso TESIS y

resumirá la operación en la instrucción *cli*, habilitando nuevamente las interrupciones. Finalmente esperará la aparición de alguna interrupción.

6.3.3 Uso de objetos en el HC11.

Dado que no se pretende realizar un sistema que cumpla con estos aspectos de la programación orientada a objetos, sólo se mencionarán aquellos que se asemejan con el sistema programado para el HC11. Teniendo ésto en cuenta, se intentó llevar estos conceptos en el programa del HC11. Como ejemplo, se puede observar la parte programada para el puerto serial. Este objeto contiene los métodos necesarios para modificarse y constituirse como un elemento del sistema. Existen variables que son de uso exclusivo de este objeto, aunque algunas de ellas sean los registros propios del puerto. También se permite alterar las funciones que la definen modificando su correspondiente vector en la tabla¹¹:

SCI.tabla	fcb	#SCI.CloseComm	
SCI.f2	fcb	#SCI.EnableNotif	
SCI.f3	fcb	#SCI.GetError	
SCI.f4	fcb	#SCI.GetEvMask	
SCI.f5	fcb	#SCI.GetState	
SCI.f6	fcb	#SCI.OpenComm	
SCI.f7	fcb	#SCI.ReadComm	
SCI.f8	fcb	#SCI.SetEvMask	
SCI.f9	fcb	#SCI.SetState	
SCI.fa	fcb	#SCI.XmitChar	
SCI.fb	fcb	#SCI.WriteComm	
SCI.Id	fcv	0	
SCI.Baud	fcv	0	
SCI.Size	fcv	0	
SCI.EvtChar	fcv	0	
SCI.EofChar	fcv	0	
SCI.Verify	fcv	0	
SCI.TimeOut	fcv	0	
SCI.EV_MASK	fcv	\$0	;TDRE TCI RDRF IDLE ORIERR BRC BTCI
SCI.FLAGS	fcv	\$0	
SCI.STATUS	fcv	\$0	
SCI.RDR	fcv	\$0	
SCI.URT	fcv	\$0	
TX.handler	fcb	\$0	
RX.handler	fcb	#Rx.Handler	
SCI.lasterror	fcb	\$0	

¹¹ Cumpliendo con el mecanismo de redifinición de funciones.

Quizás no es la manera más óptima de substituir funciones de un objeto, pero ofrece un mecanismo claro. Cada objeto debe contener su propia función reemplazadora de funciones, lo que cumple con el postulado de encapsulamiento. Aunque para este objeto no esté definida. Incluso el objeto SCI puede ser reemplazado por otro objeto si se substituye su vector en la tabla del sistema, en el vector correspondiente a SCI. También podría darse el caso de que un nuevo objeto SCI substituyera a otro definido previamente, pero a la vez utilice al viejo objeto SCI (pues el nuevo objeto podría obtener el puntero al viejo objeto y manipularlo¹²).

Se puede ver, también, que se trató de manejar todo por regiones de memoria (como también se puede entender a los objetos desde un punto de vista físico).

6.3.4 Memoria dinámica.

Para el manejo de la memoria se utilizó la técnica de segmentación de memoria en bloques. La manera más óptima de dividir el mapa de memoria fue en bloques de 256 bytes. De esta forma, dado el mapa de memoria del HC11 (65536 bytes) se tendría un total de 256 bloques de 256 bytes cada uno. Para el control de la memoria se destinó una región de memoria donde se tiene el estado de cada uno de esos bloques. El proceso para determinar la conformación de este sistema de administración de memoria se aclara a continuación.

Una necesidad del manejo de memoria es la capacidad, en un momento dado, de liberar la memoria requerida por un programa o por el sistema. Esto implica que de alguna manera se debe tener la posibilidad de saber la ubicación de los bloques que pertenecen a un mismo dueño. Así surgió la idea de hacer un mapa de estos bloques de memoria, denominado Tabla de asignación de memoria o MAT("Memory Allocation Table"). Para poder definir el tamaño de esta tabla se tomó en cuenta:

1. Debe expresar el estado de un bloque de memoria: ocupado, libre o inexistente.
2. El sistema debe reconocer una cadena de memoria utilizada para poder liberarla.
3. No se contempla el sistema de programas fragmentados.

En base a estos tres puntos se decidió diseñar la tabla con 256 bytes, es decir un bloque. Cada byte de esta tabla refleja el estado del bloque que le corresponda. Es necesario dedicarle 1 byte a cada bloque porque de esta forma se pueden construir cadenas de bloques de hasta un tamaño de 65536 bytes o 256 bloques, que sería el máximo de memoria en el mapa del HC11. La cadena contiene el número de bloque dentro del total que requiera un proceso o programa y se incrementa hasta llegar al último, que se marca con FFH.

De esta forma, la MAT está localizada a partir de la dirección 100H hasta la 1FFH, de acuerdo al siguiente mapa:

¹² Lo que en términos de programación orientada a objetos se denomina herencia.

<u>Dirección de la tabla (hexadecimal).</u>	<u>Direcciones abarcadas (hexadecimal).</u>
100	0000-00FF
101	0100-01FF
102	0200-02FF
.	.
.	.
1FE	FE00-FEFF
1FF	FF00-FFFF

La forma de marcar los bloques es:

00 : Bloque disponible.

02 - FDH : Bloque en uso.

FEH : Bloque inexistente.

FFH : Fin de bloque.

Así se pueden construir cadenas desde 1 bloque (que en la tabla estaría marcada como FFH) hasta cadenas de 255 bloques (que en la tabla estarían marcadas desde 01 hasta FFH). El último bloque siempre está marcado como FFH. De esta forma si se pide al sistema liberar un bloque o cadena, el sistema puede localizar el inicio y fin de la cadena para liberarla.

De estas funciones se encarga SYS a través de las funciones SYS_malloc, que recibe como parámetro el número de bytes a reservar, y SYS_unload, que recibe como parámetro la una dirección de la cadena o bloque a eliminar. Cabe resaltar que si se da como parámetro una dirección intermedia, es decir que no es principio de bloque o de cadena, SYS_unload automáticamente localizará el primer bloque de tal cadena y liberará toda la cadena.

Este mecanismo permitió resolver el problema del manejo de memoria dinámica de manera óptima, pues para manejar 256 bloques de memoria se requiere de 1 sólo bloque, es decir 1/256 del total del mapa de memoria.

El sistema en su inicialización, a través del mensaje CM_INIT, verifica todo el mapa de memoria y se encarga de marcar los bloques inexistentes como tales, los disponibles como libres y los reservados como ocupados. Evidentemente los bloques correspondientes al BIOS estarán marcados como ocupados al igual que los bloques utilizados por los registros internos del HC11.

6.3.5 Multitarea.

El esquema que se utilizó, y que debe implementarse dadas las características del procesador y de la arquitectura, fue el de tiempo compartido¹³.

¹³ Una descripción detallada de este sistema está en: Milenkovic, Milan, "Sistemas Operativos. Conceptos y Diseño", McGraw-Hill.

Como se explicó anteriormente, un proceso tiene varias características:

1. Un sólo punto de entrada.
2. Una pila.
3. Un procedimiento de respuesta a eventos o mensajes.

Para que el sistema de multitarea funcione debe tener la capacidad de poder restaurar la pila y enviar al proceso información a través del único punto de entrada. La información que debe tener el sistema para su propio funcionamiento con varios procesos es una zona de memoria que contenga la información: estado actual del proceso, puntos de entrada, nivel de pilas, plano de ejecución (primer o segundo plano) y el punto de ruptura del proceso. Esta información está contenida en la tabla descriptora de procesos:

```

SYS.task1      fcc      "Nombre7890123456"
               fdb      $0      ;"Entry point"
               fdb      $0      ;"Stack"
               fdb      $0      ;URT
               fdb      $0      ;Ruptura
               fcb      $0      ;Modo

SYS.task2      fcc      "0123456789abcdef"
               fdb      $0      ;"Entry point"
               fdb      $0      ;"Stack"
               fdb      $0      ;URT
               fdb      $0      ;Ruptura
               fcb      $0      ;Modo

SYS.task3      fcc      "1234567890123456"
               fdb      $0      ;"Entry"
               fdb      $0      ;"Stack"
               fdb      $0      ;URT
               fdb      $0      ;Ruptura
               fcb      $0      ;Modo

SYS.task4      fcc      "0123456789abcdef"
               fdb      $0      ;"Entry"
               fdb      $0      ;"Stack"
               fdb      $0      ;URT
               fdb      $0      ;Ruptura
               fcb      $0      ;Modo
  
```

Las partes de la tabla son:

- a) Nombre: permite una cadena de caracteres alfanuméricos de hasta 16 letras.
- b) "Entry": punto de entrada al proceso.
- c) "Stack": dirección del la pila del proceso al llamarse a esta función.
- d) URT: tabla de respuestas del proceso. No se utiliza.
- e) Ruptura: dirección donde se encontraba el proceso al ser interrumpido accidentalmente o no.

f) **Modo: estado del proceso: ACTIVO, INACTIVO, SUSPENDIDO.** Sólo puede haber un proceso **ACTIVO** o **SUSPENDIDO**, no puede haber simultáneamente procesos **ACTIVO** y **SUSPENDIDO**.

En la tabla se pueden tener hasta cuatro procesos funcionando en tiempo compartido. Para dar de alta un proceso, que previamente fue cargado en memoria, se utiliza la función a respuesta de mensaje **CM_RUN**, que recibe como parámetro un puntero a una estructura **Create**, que contiene la información necesaria para ejecutar un programa:

```
struct_create   fdb      $0000   ;Entry point
                fdb      $0000   ;Stack
                fdb      $0000   ;URT
                fdb      $0000   ;Puntero a parámetros
                fdb      $0000   ;Puntero a nombre
```

El punto de entrada o "Entry point" se obtiene a partir de la función **SYS_load**. La pila o "Stack" se debería obtener de la misma función, pero no fue desarrollado. Se llama externamente a **SYS_malloc** para proporcionar un bloque de memoria para la pila. Los punteros a parámetros y nombre se utilizan como información adicional para el proceso y para el proceso mismo respectivamente. Así el sistema tiene la información del proceso en turno y el punto de ruptura se obtiene cuando se hace llamada a la función respuesta de mensaje **CM_XGTASK**. La función de respuesta al mensaje **CM_RUN** también hacer al proceso que se quiere correr como dueño del privilegio de primer plano, pasando a los demás a segundo plano. La función tiene la forma:

```
SYS.run         ldy      SYS.wparam   ;Se lee puntero a estructura "Create".
                idx      #SYS.task1
SYSrun.c1       epx      #SYS.task4   ;Busca un proceso libre en la tabla.
                bhl      SYSrun.er1   ;No hay proceso libre.
                ldd      TASK.Entry,x ;Obtiene el "entry point" de la tabla, si está
                cmpd     #0           ;ocupado (distinto de cero) busca otro en tabla.
                beq      SYSrun.cont1
                xgdx
                addd     #SYS.task2-#SYS.task1 ;Calcula la direccion del siguiente
                                                ;proceso.
                xgdx
                bra      SYSrun.c1
SYSrun.er1      idx      #SYSrunmsg
                jsr      display
                stop

SYSrun.cont1    pshx          ;Guarda puntero al proceso libre de la tabla de tareas.
                pshy          ;Guarda puntero a "struct_create"
                ldy      CREATE.Nombre,y ;Obtiene el puntero al nombre del proceso.
                ldaa     #S10        ;16 caracteres a transferir.
SYSrun.c2       cmpa      #0
```

```

    beq     SYSrun.cont2
    ldab   0,y
    stab   0,x
    iny
    inx
    deca
    bra     SYSrun.c2
SYSrun.cont2  ldx     #SYS.task1
SYSrun.c3    ldaa   TASK.Modo,x
            anda   #0f
            cmpa  #ACTIVO
            beq   SYSrun.cont3
            cmpa  #SUSPENDIDO
            beq   SYSrun.cont4
            cmpx  #SYS.task4
            bhs   SYSrun.cont5
            xgdx
            addd  #SYS.task2-#SYS.task1
            xgdx
            bra   SYSrun.c3
SYSrun.cont3  tsy
            iny   ;Decrementa IY para evitar las variables temporales
            iny   ;dejándola a nivel de entrada de esta función.
            iny
            iny
            sty   TASK.Stack,x
            ldd   intURT
            std   TASK.URT,x
            ldaa  #BACKGNDINACTIVO
            staa  TASK.Modo,x
            ldd   $13,y
            std   TASK.Ruptura,x
SYSrun.cont4  ldaa  #BACKGNDINACTIVO
            staa  TASK.Modo,x
SYSrun.cont5  pulx
            pulx  ;Obtiene puntero a "struct".
            clr   ;Obtiene puntero al "task" libre.
            ldd   TASK.Ruptura,x
            ldd   CREATE.Entry,y
            std   TASK.Entry,x
            std   intRegreso
            ldd   CREATE.URT,y
            std   TASK.URT,x
            ldd   CREATE.Stack,y
            std   TASK.Stack,x
            ldaa  #FOREGNDINACTIVO
            staa  TASK.Modo,x
            pulx  ;Obtiene regreso de esta subrutina.
            pulx  ;Regreso de esta subrutina.
            stx   SYSrun.intReg
            stx   SYSrun.intReg2
            pula  ;Regreso del servicio.
            oraa  ;Obtiene CCR.
            ldx   #10
            ldx   CREATE.Entry,y
            ;Desactiva interrupciones.

```

```

ldy      CREATE.Stack,y
dey
tys
pshx
puly      ;Pasa a IY el contenido de IX(Entry).
pshx      ;Empuja regreso para rti.
ldx      #S0
pshx      ;Empuja IY.
pshx      ;Empuja IX.
pshx      ;Empuja reg D.
psha      ;Empuja CCR.
ldx      #CM_INIT
pshx      ;Empuja SYS.msg.
ldx      #S0
pshx      ;Empuja SYS.wparam.
pshy      ;Empuja SYS.lparam1, que es el "Entry".
pshx      ;Empuja SYS.lparam.
ldx      SYSrun.intReg2
pshx      ;Empuja regreso de servicio.
ldx      SYSrun.intReg
pshx      ;Empuja regreso de subrutina.
rts

```

Nótese que esta función crea una pila ficticia para poder hacer el regreso, en donde, además de los registros del CPU, se encuentra la estructura MSG. También salva la información del proceso en turno que ejecutó el mensaje CM_RUN.

La función respuesta a mensaje CM_XGTASK tiene la finalidad de controlar la repartición de tiempo de procesador entre los procesos declarados:

```

SYS.xgtask    pshx
              tsy
              ldx      #SYS.task1
xgtask.c1    ldaa     TASK.Modo,x      ;Ciclo: busca proceso activo.
              anda     #S0f
              cmpa     #ACTIVO        ;Pregunta por proceso activo.
              beq      xgtask.cont1    ;Busca en el siguiente proceso.
              cmpa     #SUSPENDIDO     ;Un proceso es suspendido cuando se genera
              beq      xgtask.cont2    ;un mensaje para ese proceso
              beq      xgtask.cont2    ;ese estando activo, tiene que resumir
              beq      xgtask.cont2    ;en el punto donde se suspendió.
              cmpx     #SYS.task4
              bhs      xgtask.noXg     ;No encontré proceso activo.
              xgdx
              addd     #SYS.task2-#SYS.task1
              xgdx
xgtask.cont2 bra      xgtask.c1
              ldd      SYS.wparam
              cmpd     #S0
              bne      xgtask.err3
              bra      xgtask.cont5

```

```

xgtask.cont1  stx      0,y          ;Guarda el puntero al "TASK" activo en variable
                ;temporal.
                iny              ;Se deja la pila a nivel de entrada
                iny              ;de esta funcion.POR DEFINITION.
                sty      TASK.Stack,x ;Guarda la pila del proceso.
                ldd      $13,y      ;Obtiene punto de ruptura.
                std      TASK.Ruptura,x ;Guarda punto de ruptura.
                dey              ;Ajusto otra vez el apuntador a
                dey              ;la pila.
                ldd      intURT      ;Obtiene URT del proceso.
                std      TASK.URT,x  ;Lo guarda.
                ldd      SYS.wparam
                ldaa     TASK.Modo,x  ;Obtiene estado.
                anda     #sfo
                oraa     #INACTIVO    ;Lo hace inactivo.
                staa     TASK.Modo,x

xgtask.c2      xgdx          ;Ciclo: busca proceso por activar.
                addd     #SYS.task2-#SYS.task1
                xgdx

xgtask.cont4   ldaa     TASK.Modo,x  ;Busca siguiente proceso para habilitarlo.
                anda     #s0f
                cmpa     #INACTIVO    ;Pregunta si está inactivo.
                beq      xgtask.activa
                cmpx     0,y          ;Verifica si llegó al task que fue activo.
                ;(ya revisó todos los demás "entries"
                ;de la tabla).
                beq      xgtask.noproc
                cmpx     #SYS.task4
                blo      xgtask.c2
                ldx      #SYS.task1
                bra      xgtask.cont4

xgtask.activa ldd      SYS.wparam    ;Ve si se trata de un cambio obli-
                cpd      #s0          ;gado (por algún evento).
                beq      xgtask.cont5 ;Si wparam es cero, es cambio conse-
                ldd      TASK.Entry,x ;cutivo.
                cmpd     SYS.wparam    ;Verifica si es el "entry" que se pide.
                bne      xgtask.c2

xgtask.cont5   puly          ;Destruye variable temporal y pasa a lY el
                ;"entry" de la tabla que pidió el cambio.
                ldaa     TASK.Modo,x
                psha
                ldd      SYS.wparam    ;Verifica cambio obligatorio.
                cpd      #s0
                beq      xgtask.cont7
                cpd      TASK.Entry,y  ;Verifica si el proceso pedido es el mismo que
                ;el que estaba activo.
                bne      xgtask.cont7
                pula
                anda     #sfo
                oraa     #SUSPENDIDO
                bra      xgtask.cont9

xgtask.cont7   pula
                anda     #sfo
                oraa     #ACTIVO      ;Lo hace activo, respeta plano de ejecución.
                staa     TASK.Modo,x

```

```

ldy      TASK.Stack,x
tys
xgtask.cont11 ldd      TASK.Ruptura,x      ;Se obtiene el pc antes de ruptura.
std      $8,y      ;Se envia como parametro.
ldd      $d,y      ;Se obtiene el registro D antes de la ruptura.
std      $4,y      ;Se envia como parametro.
ldd      TASK.Entry,x      ;se hace el regreso a "Entry" al encontrar rti.
std      $13,y
ldd      #0      ;Registro D.
std      $d,y      ;Se deja en el registro D zeros(mensaje).
ldd      #CM_RESUME      ;Mensaje "default": CM_RESUME
ldx      SYS.wparam      ;Verifica cambio forzado
cpx      #50
beq      xgtask.cont6
ldab     #510
orab     $c,y
stab     $c,y
ldd      SYS.lparam1      ;Lee el mensaje a desplegar.
xgtask.cont6 std      $a,y      ;Deja mensaje en SYS.msg.
rts
xgtask.cont9 staa     TASK.Modo,x
ldy      TASK.Stack,x
tys
xgdy
addd     #514      ;Se apunta al principio de esta interrupción
xgdy
ldab     #515      ;para duplicar la pila y hacer el regreso
;sin dañar al antiguo regreso. Esto es sólo
xgtask.c3 ldaa     0,y      ;para programas suspendidos.
psha
dey
decb
beq      xgtask.cont10
bra      xgtask.c3
xgtask.cont10 tsy
bra      xgtask.cont11
xgtask.err4 ldx      #xgtask.error4
jsr      display
stop
xgtask.error1 fcc      "No hay p. activo"
fdb      #EOB
xgtask.error2 fcc      "No hay procesos"
fdb      #EOB
xgtask.error3 fcc      "Mensaje anidado"
fdb      #EOB
xgtask.error4 fcc      "Destinatario incorrecto"
fdb      #EOB

```

La forma de operar de esta función es:

1. Obtiene el proceso que está actualmente activo o suspendido. Si encontró un proceso marcado como **SUSPENDIDO** regresa el control a ese proceso. Si encontró un proceso marcado como **ACTIVO** lo hace **INACTIVO**.

2. **Localiza el siguiente proceso marcado como INACTIVO.** Si no existe otro proceso, regresa al primer proceso y lo hace activo nuevamente.
3. **Una vez localizado el siguiente proceso a activar, verifica que el cambio sea libre u obligatorio.** Esto se hace colocando a ceros la variable SYS.wparam para cambio libre y a un valor cuando se desea un cambio obligatorio (por ejemplo, si se necesita enviar un mensaje a un proceso). El valor indicado en SYS.wparam corresponde al punto de entrada del proceso que debe ser activado. Si no coinciden el valor del proceso inactivo localizado con el pedido, se busca el siguiente hasta que coincidan. Si no existe tal proceso, genera error el sistema y se detiene.
4. **Si el proceso pedido resultó ser el mismo que el que estaba activo antes de generar el mensaje CM_XGTASK, entonces el proceso se marca como suspendido en lugar de inactivo y se le envía el control.** Esto sucedería si ocurrió un evento dentro de un proceso que es el destinatario de tal evento.
5. **Notar que si existe un proceso suspendido y se entra a esta función de respuesta, el sistema generará un error de mensajes anidados.** El sistema no puede conservar dos puntos de ruptura para dos eventos anidados. Se debe resumir al proceso suspendido antes de atender otro evento.

6.3.6 Notas para mejorar el sistema con procesador.

Como se puede ver en el listado del programa, corren dos procesos. Uno de ellos es el sistema operativo propiamente, que lleva la hora, y el programa desarrollado para esta tesis. El procesador desperdicia tiempo al entrar al programa SO.EXE una vez que TESIS libera su ejecución. Puede rediseñarse el control del cambio de tareas para evitar que se entre inútilmente a SO.

La programación y diseño del sistema operativo puede realizarse de manera óptima si se tiene: un compilador de alto nivel que permita la compilación en función de una dirección base y desplazamientos, tanto para datos como para código ejecutable; depuradores dinámicos que ligan al programa fuente para manejar variables por nombre, creación y destrucción de objetos visible, árboles con objetos creados.

Un problema no contemplado es el del crecimiento de la pila. En el peor caso, generación de eventos y traslado de control, las pilas pueden crecer hasta 5 niveles de interrupción, además un crecimiento indeterminado debido al paso de parámetros. No se presentan problemas de sobreflujo en las pilas debido a que se utiliza un bloque completo para pila en SO y TESIS. Queda pendiente el desarrollo de rutinas para verificación del tamaño de la pila y algún mecanismo e protección.

En el mecanismo de interrupciones de software es el sistema el que identifica el tipo de interrupción leyendo al registro A, lo que toma tiempo. Esto podría ser desarrollado de mejor manera si el procesador soportara distintos tipos de interrupción de software. Así el procesador mismo es el que identifica el tipo de interrupción y obteniendo el vector correspondiente, ahorrando tiempo.

El intentar desarrollar un sistema operativo que cumpla cabalmente con la programación orientada objetos, con los esquemas de multiprocesos, generación de eventos, y con arquitecturas flexibles, implica:

1. Utilizar un procesador de por lo menos 32 bits.

2. **Herramientas diseñandas para el procesador que incluyan:** un compilador orientado a objetos, depuradores o simuladores para objetos, depuradores dinámicos, etc...
3. **Esquemas de ejecución en modos protegidos.**
4. **Diseñar un esquema de buses que permitan el acoplamiento de diversos periféricos y que éstos tengan comunicación con el proceador a través de interrupciones.**

6.4 Programación en Windows.

El esquema de funcionamiento de Windows permite una solución a la programación en base a eventos y mensajes. El programa en Windows fue solamente desarrollado con fines de experimentación. La descripción que a continuación se hace es lo que el programa deberá tener como partes fundamentales. El listado del programa se encuentra en el apéndice G.

El programa Windows consistiría fundamentalmente en una ventana principal con la opción de seleccionar tres posibles menús: *Archivo*, *Conectar* y *Ayuda*.

Las secciones se describen:

- a) *Archivo*: permitiría seleccionar el nombre del archivo en el que se colocarán los datos obtenidos y su ubicación. En la versión de este trabajo se almacena el conjunto de resultados en un archivo cuyo formato es texto plano; es decir, sin caracteres especiales. El nombre asignado a este archivo es "Results", y se almacena en el directorio de trabajo; no existe la posibilidad de alterar el nombre ni su ubicación.
- b) *Conectar*: este permite seleccionar tres opciones: *Configurar PC*, *Conectar* y *Sistema HC11*.
 - i) **Configurar PC**: Selecciona la velocidad, bits de stop, paridad, etc. Para el caso de esta tesis los parámetros en el sistema HC11 están fijos pero el programa en Windows permite modificar los parámetros de comunicación, con la idea de que a través del menú Sistema HC11 se puedan modificar los parámetros de comunicación del sistema HC11.
 - ii) **Iniciar Medición**: selecciona la caja de diálogo para conectar con el HC11.
 - iii) **Sistema HC11**: este menú tendría la función de acceder a las funciones del HC11 disponibles al usuario, como el reloj, tiempos de muestreo, configuración de las comunicaciones, etc. En este trabajo se desarrollo un pequeño programa con la finalidad de realizar las pruebas para obtener la curva de permeabilidad del textil. Este programa fue el utilizado para obtener las curvas de permeabilidad del apéndice D. El programa del HC11 para esta función se muestra en el apéndice F.
- c) *Ayuda*: abriría la base de datos para la ayuda del programa.

El cuadro de diálogo *Inicar Medición* permite:

1. *Conectar*: ejecuta el programa TESIS.EXE en el sistema HC11 si no se ha ejecutado. Empiezan a aparecer las lecturas en sus recuadros.

2. *Iniciar*: coloca el número de bloque a ceros y empieza el almacenaje de los datos en el disco.
3. *Pausa*: permite descartar algunos datos evitando que se almacenen en el archivo.
4. *Cerrar*: termina la adquisición de datos y salva el archivo en el disco.
5. *Unidades*: permite seleccionar entre unidades del S. I. y el sistema inglés.

En este programa cabe únicamente destacar el aspecto de la comunicación.

6.5 Comunicación.

La comunicación entre el sistema HC11 y la PC es a través del puerto serial RS-232. Windows ofrece toda una gama de funciones para establecer comunicación eficazmente. El funcionamiento de las comunicaciones en el sistema HC11 es:

1. El sistema interpreta la información en caracteres ASCII imprimibles, es decir, sólo caracteres alfanuméricos y numéricos. Si se envía un carácter no imprimible será descartado.
2. El sistema puede interpretar mensajes de la siguiente forma: se envía el carácter especial "Escape" (ESC), con valor hexadecimal 1B. El sistema espera recibir 16 bytes, que corresponden al mensaje a enviar: cuatro bytes para el mensaje (int), cuatro para wparam (uint) y ocho para lparam (dword). Estos bytes deben estar expresados en caracteres ASCII alfanuméricos imprimibles. El sistema automáticamente envía el mensaje al dueño actual del recurso. De esta manera, puede verse como TESIS.EXE termina en el sistema HC11 cuando se sale del programa en Windows. Se envía el mensaje CM_QUIT. El listado de las rutinas de comunicación del HC11 puede verse en los apéndices.
3. La adquisición de datos a través del puerto serial tiene el siguiente formato:

Significado	Bytes	Valor
Marca de inicios	1	03 hex
Número de bloque	3	30 - 39 hex
Separador	1	10 hex
Dato 1	3	30 - 39 hex
Separador	1	10 hex
Dato 2	3	30 - 39 hex
Separador	1	10 hex
Marca de fin	1	04 hex

Para tener un total de 14 bytes cada vez que se toma una lectura.

La transmisión de los datos del HC11 es a través de la función SCI_WriteComm. Esta función envía como parámetro la dirección del buffer a transmitir y la cantidad de bytes a transferir. Difiere del sistema utilizado en Windows en que el objeto de comunicación en Windows tiene un buffer receptor de los datos; así cuando se llama a "WriteComm" en Windows los datos se vacían en el buffer del objeto. En

el HC11 no es así. El objeto SCI va tomando los datos del buffer que destinó el usuario para ello. Habría que tener cuidado de no sobrescribir el buffer hasta que se haya terminado la transmisión del bloque. Esto se puede saber activando el mensaje EV_BTC("Block Transmission Completed").

No se contempló ningún mecanismo de corrección o detección errores. Solamente los mecanismos existentes en el HC11 y los errores que existen en la PC¹⁴. En versiones primitivas del sistema se utilizó el mecanismo de reenvío del dato recibido para confirmación.

¹⁴ Que serían, para el caso de la PC: sobreflujo, error de marco y error de paridad; para el HC11: error de marco, sobreflujo y ruido.

Capítulo 7. Conclusiones y recomendaciones.

En este último capítulo se hacen algunas observaciones y recomendaciones sobre el diseño y construcción del aparato medidor. En la parte final de este capítulo, página 169, se muestra el diseño final a bloques del sistema.

7.1 Sobre la Norma.

En el capítulo primero se hizo una descripción de la norma y de la física asociada al fenómeno de la permeabilidad. Se mostraron las incongruencias entre lo que los estudios de Darcy han hecho sobre la permeabilidad y lo que la norma establece. Es muy probable que la norma establezca su método de medición en base a un procedimiento industrial, sin embargo, resulta entendible su definición: gasto por unidad de área. Este método es perfectamente válido si satisface los requerimientos en la venta y manejo de telas. Sin embargo, parece más apropiado mostrar esta propiedad de los textiles como una curva y no como un valor a una presión preestablecida. Es decir, se arrojaría más información con una prueba realizada sobre todo un intervalo de presiones que sobre una sola presión.

Queda pues, como recomendación, el análisis de un nuevo método para expresar esta prueba de los textiles como una curva y no como un valor de gasto por unidad de área a una presión fija.

7.2 Sobre el diseño del aparato.

En el capítulo segundo se mostró el procedimiento para el diseño de un tubo medidor de permeabilidad. Para su diseño se tomaron los estándares ASME y se procuró exceder las tolerancias en cuanto al cálculo de valores. Producto de estas consideraciones es que el tamaño del aparato es considerable. Si se observa el tamaño del aparato medidor fabricado por "Frazier Instruments"¹ o por "United States Testing Company, Inc."² es considerablemente menor al de esta tesis. Para la disminución de las dimensiones de este aparato conviene estudiar otro método de medición de gasto, evitando las pérdidas y aspectos dinámicos y cinemáticos que implica la placa de orificio. Al utilizar algún método de obstrucción con bordes y reducciones suaves, las pérdidas y los efectos provocados por el chorro disminuyen. De esta forma podría construirse un aparato medidor de una longitud menor. La sección del tubo utilizado fue conforme a la norma, no necesariamente debe tener esta sección, pues la norma define a la permeabilidad como gasto por unidad de área.

La dimensión del extractor también queda sujeta a reducciones al utilizar algún otro método de medición. La placa de orificio de bordes rectangulares es la que provoca una pérdida por fricción considerable. En esta tesis es la placa de orificio la que determinó el tamaño del motor; mientras que la caída de presión en la muestra de la tela es de 0.5 [inH₂O], la caída en la placa de orificio es de

¹Frazier Precision Instruments Co. 210 Oakmont Ave., Gaithersburg, Md, USA.

²United States Testing Co. Inc. 1415 Park Ave., Hoboken, New Jersey 07030, USA.

aproximadamente 9 [inH₂O]. Este aspecto debe ser considerado en la elaboración de esta máquina para llevarse a punto de venta.

Las dimensiones del aparato también pueden verse disminuidas si se hace una caracterización del aparato medidor. Esto significa que deben olvidarse los estándares ASME y definir las características dinámicas, geométricas y cinemáticas del nuevo aparato.

7.3 Sobre la electrónica de control.

En este aspecto han quedado varios puntos por resolver.

Uno de ellos es el uso del controlador de ángulo de disparo TDA1085 sin la necesidad de un convertidor F/V externo. Para ello debe construirse una fuente que entregue los voltajes necesarios para operación del TDA1085. Evitando el uso de circuitería externa para la conversión de frecuencia a voltaje. También sería recomendable el análisis de control y la experimentación para evitar las oscilaciones del sistema. La información técnica del circuito TDA1085 que entrega motorola no es suficiente para encontrar un control óptimo teórico. Motorola sugiere pruebas experimentales para determinar el valor adecuado de los componentes. La ventaja del desarrollo de esta tesis con el convertidor de frecuencia a voltaje externo al TDA1085 es que se tiene control sobre los elementos del controlador, sin embargo, se convierte en un circuito grande.

No se pudo evaluar la posibilidad de un convertidor DC-DC basado en el PWM del HC11. El uso del DAC fue satisfactorio para este prototipo.

Durante las pruebas realizadas se pudo percibir el incremento en la temperatura del TRIAC, lo que lleva a considerar el uso de disipadores en el diseño del prototipo. Esto no representa ningún problema, pues la temperatura y potencia disipada no es de consideración. El uso de un disipador de tamaño razonable fue suficiente. El incremento en la temperatura del TRIAC sin disipador se manifestó en disparos no controlados (oscilaciones).

También durante las pruebas se percibieron problemas con la fuente. El disparo del TRIAC provoca caídas de voltaje en los reguladores. En el prototipo se incluyeron filtros con inductores y capacitancias que hicieron desaparecer casi en su totalidad este problema.

Durante las pruebas se presentó ruido. Los transductores de presión a voltaje operan con corriente, lo que los hace poco sensibles a inducciones provocadas por la conmutación de la electrónica y el ruido inducido en los embobinados del motor y en las escobillas; sin embargo en el cableado del circuito de conversión de corriente a voltaje sí se indujo ruido. En este prototipo no se desarrolló ningún tipo de protección contra inducciones. Se debe buscar la forma de aislar a la electrónica de la contaminación electromagnética provocada por el controlador de ángulo de disparo y el motor.

7.4 Sobre la arquitectura y programación del HC11.

Para un desarrollo completo de este aparato medidor habría que incluir un teclado para comunicación con el usuario.

Como se explicó en el la introducción de esta tesis, la idea base al diseñar el sistema operativo fue el de proponer un sistema que operase en función de eventos, objetos y el manejo de memoria dinámica. Algunas recomendaciones para el desarrollo de este sistema con estas ideas quedaron explicadas en el capítulo 6; aquí se hace una reflexión sobre éstas.

La programación y diseño de este sistema operativo fue una tarea titánica debido a que todo se hizo en lenguaje ensamblador. El tiempo de programación se puede disminuir considerablemente si dispone de un compilador de alto nivel que permita la compilación en función de una dirección base y desplazamientos, tanto para datos como para código ejecutable; depuradores dinámicos que ligen al programa fuente para manejar variables por nombre, creación y destrucción de objetos visible o árboles con objetos creados.

Un problema no contemplado es el del crecimiento de la pila. En el peor caso, generación de eventos y traslado de control, las pilas pueden crecer hasta 5 niveles de interrupción, además de un crecimiento indeterminado debido al paso de parámetros. No se presentan problemas de rebalse en las pilas debido a que se utilizan bloques completo de 256 bytes. Queda pendiente el desarrollo de rutinas para verificación del tamaño de la pila y algún mecanismo e protección. Esta revisión puede facilitarse si el depurador verifica la pila para rebalsos. En este sistema no se permiten mensajes anidados, lo que hace crítico el mecanismo de respuesta a eventos de alta prioridad.

En el mecanismo de interrupciones de software es el sistema el que identifica el tipo de interrupción leyendo al registro A, lo que toma tiempo. Esto podría ser desarrollado de mejor manera si el procesador soportara distintos tipos de interrupción de software. Así el procesador mismo es el que identifica el tipo de interrupción y obteniendo el vector correspondiente, ahorrando tiempo. Este problema está presente en el sistema de control. Para el muestreo y el control se utiliza el temporizador RTI. Este genera interrupciones a a un intervalo máximo de 32 [ms]. Durante las pruebas de control se observó que la atención del evento RTI es demasiado lenta, debido a todas las rutinas en punto flotante, y afecta a la transmisión de datos hacia la PC. Para la solución de este problema sería recomendable la medición de ciclos de reloj que toma atender a un evento RTI; hay que recordar que la atención al evento no tiene un longitud en tiempo siempre igual, pues si el proceso activo no es el dueño del evento RTI el sistema pierde tiempo en localización del propietario y encauzamiento del evento. El uso del temporizador de 16 bits TOF puede dar una solución al problema, pues el muestreo no es cada 32 [ms], sino aproximadamente cada 128 [ms]; de esta forma conviene utilizar TOF que RTI.

El intentar desarrollar un sistema operativo que cumpla cabalmente con la programación orientada objetos, con los esquemas de multiprocesos, generación de eventos, y con arquitecturas flexibles, implica:

1. Utilizar un procesador de por lo menos 16 bits.
2. Herramientas diseñadas para el procesador que incluyan: un compilador orientado a objetos, depuradores o simuladores para objetos, depuradores dinámicos, etc.
3. Esquemas de ejecución en modos protegidos.

4. **Diseñar un esquema de buses que permitan el acoplamiento de diversos periféricos y que éstos tengan comunicación con el procesador a través de interrupciones.**
5. **Generación de excepciones para proteger pilas, memoria y puertos.**

Como conclusión: las ideas planteadas para una arquitectura basada en el HC11 exceden las capacidades del procesador y del programador. Como una primera aproximación se han sugerido muchas ideas que pueden ser desarrolladas posteriormente.

7.5 Sobre el control.

En este aparato medidor se presentaron dos modelados para el sistema: uno de ellos, el primero, es de segundo orden, el otro, resultado de una modificación al circuito, fue un sistema de primer orden.

Al realizar las pruebas con el sistema de control del motor se trató de proporcionar la señal del tacómetro como una señal sin rizo. Esto lleva una desventaja: la respuesta del tacómetro se ve afectada. Sin embargo, al eliminar los filtros para el rizo, la respuesta del tacómetro mejoró de manera considerable, pero el rizo fue sustancial. Durante las pruebas de control, este hecho mejoró considerablemente el funcionamiento del sistema al punto de que un control proporcional es suficiente.

El efecto del rizo no se manifestó para el intervalo de revoluciones de las pruebas. Sin embargo, cuando las revoluciones son bajas se alcanza a percibir un golpeteo en el rotor del motor, el cual no se presentaba cuando se utilizó el primer filtro para el rizo.

La respuesta del control proporcional para el sistema de segundo orden puede mejorarse al disminuir la ganancia del controlador, lo que hace la respuesta del sistema más lenta. El control PI para el sistema de segundo orden presentó oscilaciones perceptibles.

La recomendación en este sentido es que el sistema tuvo un desempeño y mejorado adecuado al modelarse como un sistema de primer orden; esto se debió a la modificación del filtro de rizado del convertidor de frecuencia a voltaje. La respuesta del controlador de ángulo de disparo puede mejorarse si se utiliza un tacómetro óptico con un número de obstrucciones múltiplo de las revoluciones. La electrónica asociada a la bomba de carga debe modificarse para entregar un voltaje equivalente a las revoluciones del motor y no del número de obstrucciones. Al operar el convertidor F/V a una frecuencia mayor, el capacitor del filtro de rizado puede ser considerablemente menor, mejorando la respuesta del tacómetro.

El problema de las distintas ganancias del modelado del sistema desaparecen con cualquier tipo de control, pues al ser acumulativos la acción de control es la anterior con una nueva corrección función del nuevo error.

7.6 Sobre el programa en PC.

La programación en PC ofrece muchas posibilidades. En ambiente gráfico de los sistemas operativos actuales y los lenguajes de programación abren las puertas al desarrollo de un programa completo y que cubra todas las necesidades para la manipulación de la información adquirida a través del

HC11. En este sentido no hay ninguna recomendación; sin embargo, el programa presentado podría ser mejorado y completado en el sentido de comunicación con el usuario. Durante las pruebas realizadas se encontró un problema de comunicación que no pudo ser resuelto. La inicialización del objeto de comunicaciones bajo Windows es defectuosa para la transmisión. Las funciones para obtener errores y eliminarlos del puerto de comunicación no reportan error alguno; pero, la transmisión falla en ocasiones y para poder corregir el problema se debe salir del programa y entrar varias veces hasta que opere correctamente.

El programa fue desarrollado para realizar las pruebas del prototipo y no contempla otro uso. Se mencionaron algunas ideas en el capítulo 6 que deberían ser incluidas un programa para la máquina medidora y están básicamente enfocados en hacer del aparato medidor un dispositivo entendible y fácilmente manipulable además de ayudar a obtener toda la información necesaria sobre la permeabilidad de un textil.

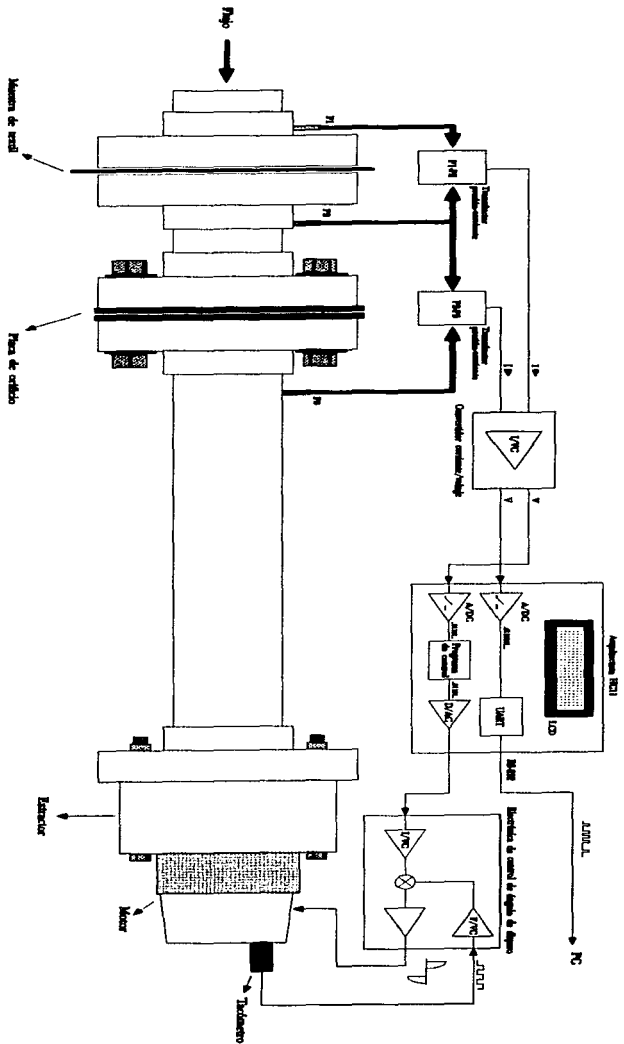


Diagrama por bloques del sistema medidor.

Bibliografía.

Medios porosos, mecánica de fluidos:

ANSI/ASTM Standards, Textile, parte 7 tomo 2, ASTM publications.

Jack P. Holman, "Métodos experimentales para ingenieros", segunda edición, México 1988.

Stephen Whitaker, "Introduction to Fluid Mechanics", R.E. Krieger Pub. 1981.

Jacob Bear, "Dynamics of Fluids in Porous Media", Dover Publications, New York 1988.

Robert H. Perry, "Perry's Chemical Engineer's Handbook", Mc. Graw-Hill 1989.

Kenneth Wark Jr., "Thermodynamics", quinta edición, McGraw-Hill Editions. EE.UU. 1989.

R. B. Bird, W. E. Stewart, "Fenómenos de Transporte", editorial Revertè, España 1982.

Leland K. Spink, "Principles and Practice of Flow Meter Engineering", The Foxboro Company, marzo 1978.

ASME, "Fluid Meters: their theory and application", ASME 1959.

ISO, "Measurement of fluid flow by means of orifice plates, nozzles and Venturi tubes inserted in circular cross-section conduits running full", ISO/DIS 5167-1, 1989.

Microcontrolador HC11:

Motorola, "M68HC11, Reference Manual", Motorola Inc., 1991

Motorola "AN974 MC68HC11 Floating Point Package", Motorola Inc., 1986.

Kevin Anderson, "8 bit Cross Assemblers, User's Manual", Motorola Inc., 1986.

Teoría de Control:

Francisco J. Rodríguez Ramírez, "Dinámica de sistemas", Editorial Trillas, México 1989.

Herbert P. Neff Jr., "Continuous and Discrete Linear Systems", Harper & Row Publishers, New York 1984.

Katsuhiko Ogata, "Ingeniería de Control Moderna", Prentice Hall, México 1980.

Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp, "Process Dynamics and Control", John Wiley and Sons, EE. UU. 1989.

W. C. Clements, Jr, "Black-Box Dynamic Process Modeling Using Process Reaction Curves", AIChE Modular Instruction, Series A: Process Control, Volume 1: Analysis of Dynamic Systems, 51-65, 1980.

Braatz, R. D., "Internal Model Control", The Control Handbook, CRC press. 1995.

Motores.

Siskind Charles, "Electrical Machines", McGraw-Hill, Singapore 1966.

Facultad de Ingeniería, "Generadores, motores y transformadores", UNAM 1992.

Motorola Inc, "Thyristor Data", Part 2: Theory and Applications, 1990.

Otros:

Arenas S. María del C., Santos J. Saúl, "Diseño y construcción de un sistema automático de registro de asistencias para el personal administrativo de la Facultad de Ingeniería", México D. F. julio 1995, director de tesis: M. en I. Lauro Santiago Cruz.

Marvin J. Fisher, "Power Electronics", PWS-KENT Pub. Co., EE.UU. 1991.

Robert F. Coughlin, Frederick F. Driscoll, "Operational Amplifiers and linear integrated circuits", Prentice Hall, New Jersey 1991.

Motorola Ltd.: "Fast and LS TTL Data",
 "Rectifiers and Zener Diodes Data",
 "Small-signal Transistors, FET's and Diodes Data",
 "Linear/Interface IC's Device Data",
 "Motorola Master Selection Guide",
 "Motorola Microprocessor Data".

National Semiconductor Corporation: "Linear Application Specific IC's Data Book", 1990.
 "Data Acquisition Devices Data Book", 1990.

Texas Instruments Inc., "Linear Circuits".

Programación:

Milan Milenkovic, "Sistemas Operativos, conceptos y diseño", McGraw-Hill, España 1988.

Chris H. Pappas, William H. Murray, "Manual de Borland C++", McGraw-Hill, México 1994.

Herbert Schildt, "Programación en Turbo C", McGraw-Hill, México 1989.

Herbert Schildt, "C para usuarios expertos", McGraw-Hill, México 1989.

Intel Corporation, "Intel386SX Microprocesor, programmer's referenc manual", EE. UU. 1991.

Borland International, "Borland C++ for Windows, Windows NT and DOS v. 4.02", EE. UU. 1993

Timothy Monk, "Windows: guide to serial communications", De. Sams, Indiana EE. UU. 1992.

Microsoft Corp., "Microsoft Win16/Win32 API Programmer's Reference", Microsoft Corp., 1993.

Programas utilizados:

Borland C++ v. 4.02, versión Windows.

Borland Turbo C v. 2.0, versión DOS.

Borland Turbo Pascal, como editor.

Avocet Systems Inc., AVSIM, depurador HC11.

Motorola Ltd., AS11, compilador HC11.

Motorola Ltd., PCBug v. 3.42, depurador en tiempo real HC11.

Internet:**- Programación en C/C++, ObjectWindows:**

Borland International, Inc.:

<http://www.borland.com>

"652 Interrupt Handlers as Member Functions", <http://loki.borland.com/cpp/cpptis/ti652.htm>

"396 Capturing Interrupts", <http://loki.borland.com/cpp/cpptis/ti396.htm>

"1782 Creating a Dialog as Main Window App.", <http://loki.borland.com/cpp/cpptis/ti1782.htm>

"1748 How to Use a Dialog Box as Main Window in a OWL 2 App.",

<http://loki.borland.com/cpp/cpptis/ti1784.htm>

"445 Interrupt Driven Serial Communications", <http://loki.borland.com/cpp/cpptis/ti445.htm>

- Compilador C y ensamblador para HC11

Motorola Corporation: <http://www.mot.com>

- Información sobre Windows:

Microsoft Corporation: <http://www.microsoft.com>

"The Windows 95 Resource Kit".

- ASTM

American Society of Testing Materials: <http://www.astm.org>

- Transductores de presión:

Dresser Instruments Inc.: <http://www.dresser.com>

Motorola Corporation: <http://www.mot.com>:

<http://design-net.com/senscon/press2.htm>

http://design-net.com/senscon/books/html/dl200_index.html

<http://desing-net.com/senscon/books/html/dl200/pdf/>

mpx906.pdf

mpx5010.pdf

mpx2010.pdf

mpx50.pdf

- Circuitos Electrónicos:

Maxim Co., 1410.pdf, "RS232 Drivers/Receivers", <http://www.maxim-ic.com>

Texas Instruments, s10s090b.pdf, "Programable Low-Power Operational Amplifiers", <http://www.ti.com>

Appendix A. Norma ASTM D-737.

Standard Test Method for AIR PERMEABILITY OF TEXTILE FABRICS¹

This Standard is issued under the fixed designation D 737; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reappraisal.

1. Scope

1.1 This method covers the direct determination of the air permeability of textile fabrics by the calibrated orifice method.

2. Applicable Documents

2.1 ASTM Standards:

D 123 Definitions of Terms Relating to Textiles²

D 1776 Recommended Practice for Conditioning Textiles and Textile Products for Testing²

3. Definitions

3.1 *air permeability, n*—the rate of air flow through a material under a differential pressure between the two fabric surfaces.

3.1.1 Air permeability is expressed in U.S. customary units as cubic feet of air per minute per square foot of fabric at a stated pressure differential between the two surfaces of the fabric.

3.2 For definitions of other textile terms used in this method refer to Definitions D 123.

4. Summary of Method

4.1 The rate of air flow through a known area of fabric is adjusted to secure a prescribed pressure differential between the two fabric surfaces in the test area and from this rate of flow the air permeability of the fabric is determined.

5. Uses and Significance

5.1 This method for testing textile fabrics for air permeability is not recommended for acceptance testing of commercial shipments since information on between-laboratory precision is incomplete. In some cases the pur-

chaser and the seller may have to test a commercial shipment of one or more specific materials by the best available method, even though the method has not been recommended for acceptance testing of commercial shipments. In such a case, if there is a disagreement arising from differences in values reported by the purchaser and the seller when using this method for acceptance testing, the statistical bias, if any, between the laboratory of the purchaser and the laboratory of the seller should be determined with each comparison being based on testing specimens randomly drawn from one sample of material of the type being evaluated.

5.2 Air permeability is an important factor in the performance of such textile materials as gas filters, and fabrics for clothing, tentage, mosquito netting, sails, and parachutes. In filtration, for example, air permeability is directly related to efficiency (pressure differential between the surfaces of a filter in use). Air permeability can also be used to provide an indication of the "breathability" of weather and rainproof fabrics or of coated fabrics in general.

5.3 Performance specifications, both industrial and military, have been set up on the basis of air permeability and are used in the purchase of materials where permeability is of interest.

5.4 Construction factors of finishing techniques can have an appreciable effect upon

¹ This method is under the jurisdiction of ASTM Committee D-13 on Textiles, and is the direct responsibility of Subcommittee D 13.59 on Fabric Test Methods, General. Current edition approved June 30, 1975. Published August 1975. Originally published as D 737 - 43 T. Last previous edition D 737 - 69.

² Annual Book of ASTM Standards, Part 32.

air permeability by causing a change in the length of air-flow paths through a fabric. Yarn twist is also important in that as twist increases, the circularity and density of the yarn increases, thus reducing the yarn diameter and the cover factor and increasing the air permeability. Yarn crimp and weave influence the shape and area of the interstices between yarns, and may permit yarns to extend easily. Such yarn extension would open up the fabric, increase the free area, and increase the air permeability. Hot calendering can be used to flatten yarns, thus reducing air permeability.

5.5 Since air permeability is not a linear function of pressure differential between fabric surfaces, all tests should be made at a prescribed pressure differential—0.5 in. (12.7 mm) of water. Certain fabrics, however, may be either so open or so dense that this pressure differential cannot be obtained on the apparatus. In these cases, when 0.5 in. (12.7 mm) of water is not used, the specific pressure differential used must be stated in the report.

6. Apparatus

6.1 *Permeability Testing Machine*³ including the following: a suction fan drawing air through a known area of fabric defined by a circular orifice having a diameter of 2.75 in. (70 mm), a vertical manometer for measuring the rate of air flow through the test area of the fabric, a clamp to hold the specimen over the orifice, means for adjusting the pressure differential between the fabric surfaces to 0.5 in. (12.7 mm), and a series of nozzles of varying diameters (Table 1). A calibrated test plate and a conversion chart to convert observed pressures to air flows are also needed. The clamp should effectively eliminate edge leakage (Note 3 under 9.4.2). The machine should have facilities for testing fabrics having different thicknesses and for testing sections of large pieces of cloth without cutting out specimens.

NOTE 1—A high-pressure machine³ is available for testing air-permeability ranges greater than those listed in Table 1.

7. Sampling

7.1 Take a lot sample as directed in the applicable material specification. In the ab-

sence of such specifications, take a sample comprising 10 percent of the rolls in the shipment rounded off to the next highest number.

7.2 Take test specimens that are believed to be representative of the sample to be tested and free of abnormal distortions.

7.3 Cutting test specimens is unnecessary. However, if this is desired, cut specimens at least 10 by 10 in. (255 by 255 mm).

8. Number of Specimens

8.1 Unless otherwise agreed upon, as when specified in applicable material specification, take a number of specimens such that the user may expect at the 95% probability level that the test result is no more than 5.0% of the average above or below the true average (that is, a theoretical average obtained from an infinite number of observations). Determine the number of specimens as follows:

8.1.1. *Reliable Estimate of v* —When there is a reliable estimate of v based upon extensive past records for similar materials tested in the user's laboratory as directed in the method, calculate the number of specimens using Eq 1:

$$n = (t^2 \times v^2) / A^2 = 0.154 \times v^2 \quad (1)$$

where:

- n = number of specimens (rounded upward to a whole number),
- v = reliable estimate of the coefficient of variation of individual observations on similar materials in the user's laboratory under conditions of single-operator precision
- t = 1.960, the value of Student's t for infinite degrees of freedom for two-sided limits, and a 95 % probability level ($t^2 = 3.842$),
- A = 5.0 % of the average, the value of the allowable variation, and
0.154 = value calculated from t^2/A^2 .

8.1.2 *No Reliable Estimate of v* —When there is no reliable estimate of v for the user's laboratory, Eq 1 should not be used directly. Instead specify the fixed number of seven specimens. This number of specimens is calculated using $v = 6.3$ % of the average, which

³ Apparatus is available from Frazier Precision Instrument Co., 210 Oakmont Ave., Gaithersburg, Md., and United States Testing Co., Inc., 1415 Park Ave., Hoboken, N. J.

is a somewhat larger value of v than is usually found in practice. When a reliable estimate of v for the user's laboratory becomes available, Eq 1 will usually require fewer than five specimens.

9. Conditioning

9.1 Bring the test specimens or test fabric to moisture equilibrium for testing in the standard atmosphere for testing textiles as directed in Recommended Practice D 1776.

10. Procedure

10.1 The following procedures for checking the equipment and the operating procedure are intended only for the Frazier apparatus.

10.2 Test the conditioned fabric in the standard atmosphere for testing.

10.3 *Procedure for Checking the Instrument:*

10.3.1 Adjust the inclined manometer so that it is level using the leveling bulb and the two screw supports. Adjust the oil (Meriam Red Oil, sp gr 0.827) in the inclined manometer to zero using the adjusting knob at the top right of the manometer.

10.3.2 Adjust the oil in the vertical manometer to zero by sliding the scale up or down as necessary.

10.3.3 Select a test plate for use in checking the operation of the apparatus.

NOTE 2—The apparatus is supplied with a calibrated test plate for use with the 4-mm (No. 5) nozzle. The test plate is used to detect malfunctions throughout the entire system of the apparatus due to such causes as air leakage. The test plate is calibrated by the apparatus manufacturer and is stamped with the corresponding value of air permeability that should be achieved. Other test plates are available from the manufacturer to check the calibration of every nozzle.

10.3.4 Insert the nozzle, as specified on the test plate, into the orifice through the door of the tank underneath the table. Check to be sure that the door is closed and fastened. Clamp the test plate across the orifice in the table of the apparatus and note the expected value of air permeability.

10.3.5 Start the motor and slowly adjust its speed by means of the rheostat to maintain the oil level in the inclined manometer at 0.5 in. (12.7 mm). Maintain the oil level steady and approach the level from a lower value.

10.3.6 Observe the level of oil in the vertical manometer and determine the corresponding permeability of air flow from the tables supplied with the testing machine. Check to be sure that the indicated permeability is within ± 2 percent of the value stamped on the calibrated test plate.

10.3.7 If the indicated air flow is not within the specified limits, stop testing until the cause of the variation has been eliminated. Malfunctioning could be caused by air leaks in the apparatus, physical damage to the nozzle or test plate, condition of the gasket, or oil overflow into the copper connecting tubes.

10.4 *Procedure for Testing Specimens:*

10.4.1 After proper functioning of the apparatus has been established, make a trial run, replacing the test plate with some of the fabric to be tested, to provide an indication of which nozzle will be required in performing the testing. Base the choice of the nozzle on the air flows listed in Table 1.

NOTE 3—Several nozzles are available with the apparatus for use in testing fabrics of various permeabilities. These nozzles are calibrated by the apparatus manufacturer and tables are supplied equating vertical manometer readings to air flow for each nozzle. The greater the permeability, the larger should be the opening of the nozzle. But at all times the nozzle should be small enough so the oil in the vertical manometer reaches at least the 3-in. line (the machine is slightly inaccurate below the 3-in. line).

10.4.2 Clamp the specimen, free from tension, across the test area. The clamp should effectively eliminate edge leakage.

NOTE 4—Since air leakage may affect test results, precautions must be taken especially with very heavy or lofty fabrics, to prevent leakage. The use of rubber gaskets, or greasing the portion of the fabrics that comes in contact with the clamp, may be possible solutions to the problem.

10.4.3 Start the motor and slowly adjust its speed by means of the rheostat to maintain the oil level in the inclined manometer at 0.5 in. Maintain the oil level steady approaching the level from a lower value.

10.4.4 Observe and record the level of oil in the vertical manometer.

11. Calculation

11.1 Determine the rate of flow of air passing through the test area of the fabric from the level of oil in the vertical manometer and the tables supplied by the manufacturer for each

individual nozzle. Record the observed rate of flow to two significant figures.

11.2 Multiply air-permeability values in U.S. customary units ($\text{ft}^3/\text{ft}^2\text{-min}$) by 0.508 to convert to metric units ($\text{cm}^3/\text{cm}^2\text{-s}$).

11.3 Calculate the average permeability of all specimens tested.

12. Report

12.1 State that the specimens were tested as directed in ASTM Method D 737.

12.2 Report the following information:

12.2.1 Average air permeability of the fabric in U.S. customary or metric units to three significant figures, and

12.2.2 Pressure differential between the fabric surfaces in inches or millimetres of water, if other than 0.5 in. (12.7 mm) of water.

13. Precision and Accuracy

13.1 *Test Data*—No recent interlaboratory test has been conducted using this method. Analysis of variance indicates that most of the variance in test results is variability in the fabric rather than single-operator or between-laboratory components of variance. Available information indicates that the single-operator component of variance for air permeability of fabric expressed as a coefficient of variation is typically about 4.5 % of the average, but may range from about 3.0 to 6.0 % of the average.

13.2 *Precision*—For the typical single-operator component of variance reported in 13.1, two averages of observed values should be considered significantly different at the 95 % probability level if the difference equals or exceeds the critical differences listed below:

Number of Observations in Each Average	Critical Differences* Percent of the Grand Average for Single-Operator Precision
1	12.5
4	6.2
5	4.4
16	3.1

* The critical differences were calculated using $t = 1.960$, which is based on infinite degrees of freedom.

NOTE 5—To convert the tabulated values of the critical differences to units of measure, multiply the average of the two specific sets of data being compared by the critical differences expressed as decimal fractions.

NOTE 6—The tabulated values of the critical differences should be considered to be a general statement, particularly with respect to between-laboratory precision. Before a meaningful statement can be made about two specific laboratories, the amount of statistical bias, if any, between them must be established, with each comparison being based on recent data obtained on randomized specimens from one sample of the material to be tested.

13.3 *Accuracy*—No justifiable statement can be made on the accuracy of Method D 737 for testing air permeability, since the true value of the property cannot be established.

TABLE 1 Nominal Sizes of Nozzles and Approximate Air Permeability Range of Fabrics

Nozzle Size, mm	Air Permeability Range	
	ft ³ /ft ² ·min	cm ³ /cm ² ·s
1.0	1 to 2.5	0.5 to 1.3
1.4	2.5 to 4.5	1.3 to 2.3
2.0	4.5 to 10	2.3 to 5.1
3.0	10 to 20	5.1 to 10
4.0	20 to 40	10 to 20
6.0	40 to 100	20 to 50
8.0	80 to 170	41 to 86
11.0	160 to 320	81 to 163
16.0	320 to 700	163 to 356

The American Society for Testing and Materials takes no position respecting the validity of any patent rights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of the validity of any such patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

This standard is subject to revision at any time by the responsible technical committee and must be reviewed every five years and if not revised, either reapproved or withdrawn. Your comments are invited either for revision of this standard or for additional standards and should be addressed to ASTM Headquarters. Your comments will receive careful consideration at a meeting of the responsible technical committee, which you may attend. If you feel that your comments have not received a fair hearing you should make your views known to the ASTM Committee on Standards, 1916 Race St., Philadelphia, Pa. 19103, which will schedule a further hearing regarding your comments. Failing satisfaction there, you may appeal to the ASTM Board of Directors.

Apéndice B. Coeficientes de descarga para Vena Contracta¹.

Tubo de 2 pulgadas, diámetro = 2.067"

βRe	500	1,000	2,000	3,000	4,000	5,000	6,000	8,000	10,000	15,000	20,000	25,000	50,000	100,000	10 ⁶
0.100	0.6179	0.6115	0.6071	0.6051	0.6039	0.6031	0.6025	0.6017	0.6011	0.6002	0.5997	0.5993	0.5984	0.5978	0.5967
0.150	0.6215	0.6136	0.6080	0.6056	0.6041	0.6031	0.6024	0.6013	0.6006	0.5995	0.5988	0.5984	0.5973	0.5965	0.5932
0.200	0.6265	0.6170	0.6103	0.6074	0.6056	0.6044	0.6035	0.6022	0.6014	0.6001	0.5993	0.5988	0.5974	0.5965	0.5949
0.250	0.6326	0.6215	0.6137	0.6102	0.6081	0.6067	0.6057	0.6048	0.6032	0.6017	0.6007	0.6001	0.5985	0.5974	0.5956
0.300	0.6401	0.6277	0.6182	0.6142	0.6118	0.6101	0.6089	0.6072	0.6061	0.6043	0.6032	0.6025	0.6007	0.5994	0.5973
0.325	0.6445	0.6307	0.6210	0.6167	0.6141	0.6123	0.6111	0.6092	0.6080	0.6061	0.6049	0.6041	0.6022	0.6008	0.5985
0.350	0.6494	0.6346	0.6242	0.6196	0.6168	0.6150	0.6136	0.6116	0.6103	0.6082	0.6070	0.6062	0.6041	0.6026	0.6002
0.375	0.6548	0.6390	0.6279	0.6230	0.6200	0.6180	0.6166	0.6145	0.6130	0.6108	0.6095	0.6086	0.6064	0.6048	0.6022
0.400	0.6441	0.6322	0.6209	0.6169	0.6138	0.6116	0.6100	0.6178	0.6163	0.6139	0.6125	0.6116	0.6092	0.6075	0.6017
0.425	0.6498	0.6371	0.6254	0.6214	0.6181	0.6158	0.6141	0.6117	0.6101	0.6176	0.6161	0.6151	0.6125	0.6107	0.6078
0.450	0.6636	0.6427	0.6307	0.6267	0.6231	0.6206	0.6188	0.6162	0.6142	0.6219	0.6203	0.6192	0.6165	0.6146	0.6114
0.475	0.6491	0.6371	0.6254	0.6214	0.6181	0.6158	0.6141	0.6117	0.6101	0.6176	0.6161	0.6151	0.6125	0.6107	0.6078
0.500	0.6564	0.6446	0.6329	0.6289	0.6255	0.6230	0.6212	0.6186	0.6167	0.6242	0.6225	0.6214	0.6187	0.6168	0.6136
0.520	0.6630	0.6513	0.6396	0.6356	0.6321	0.6296	0.6278	0.6252	0.6232	0.6307	0.6290	0.6279	0.6252	0.6233	0.6201
0.540	0.6703	0.6586	0.6469	0.6429	0.6394	0.6369	0.6351	0.6325	0.6305	0.6380	0.6363	0.6352	0.6325	0.6306	0.6274
0.560	0.6784	0.6667	0.6550	0.6510	0.6475	0.6450	0.6432	0.6406	0.6386	0.6461	0.6444	0.6433	0.6406	0.6387	0.6355
0.580	0.6874	0.6757	0.6640	0.6600	0.6565	0.6540	0.6522	0.6496	0.6476	0.6551	0.6534	0.6523	0.6496	0.6477	0.6445
0.600	0.6882	0.6765	0.6648	0.6608	0.6573	0.6548	0.6530	0.6504	0.6484	0.6559	0.6542	0.6531	0.6504	0.6485	0.6453
0.620	0.6987	0.6870	0.6753	0.6713	0.6678	0.6653	0.6635	0.6609	0.6589	0.6664	0.6647	0.6636	0.6609	0.6590	0.6558
0.640	0.7104	0.7042	0.7000	0.6969	0.6942	0.6925	0.6907	0.6881	0.6861	0.6936	0.6919	0.6908	0.6881	0.6862	0.6830
0.660	0.7235	0.7168	0.7123	0.7092	0.7065	0.7047	0.7029	0.7003	0.6983	0.7058	0.7041	0.7030	0.7003	0.6984	0.6952
0.680	0.7382	0.7310	0.7260	0.7224	0.7193	0.7173	0.7154	0.7128	0.7108	0.7183	0.7166	0.7155	0.7128	0.7109	0.7077
0.700	0.7469	0.7415	0.7326	0.7326	0.7326	0.7326	0.7283	0.7283	0.7283	0.7358	0.7341	0.7330	0.7303	0.7284	0.7252
0.720	0.7650	0.7591	0.7548	0.7548	0.7548	0.7548	0.7487	0.7487	0.7487	0.7562	0.7545	0.7534	0.7507	0.7488	0.7456
0.740	0.7857	0.7792	0.7744	0.7744	0.7744	0.7744	0.7676	0.7676	0.7676	0.7751	0.7734	0.7723	0.7696	0.7677	0.7645
0.750	0.7972	0.7903	0.7853	0.7853	0.7853	0.7853	0.7782	0.7782	0.7782	0.7857	0.7840	0.7829	0.7802	0.7783	0.7751
0.760	0.8024	0.7970	0.7920	0.7920	0.7920	0.7920	0.7849	0.7849	0.7849	0.7924	0.7907	0.7896	0.7869	0.7850	0.7818
0.770	0.8153	0.8096	0.8046	0.8046	0.8046	0.8046	0.7972	0.7972	0.7972	0.8047	0.8030	0.8019	0.7992	0.7973	0.7941
0.780	0.8294	0.8233	0.8183	0.8183	0.8183	0.8183	0.8108	0.8108	0.8108	0.8183	0.8166	0.8155	0.8128	0.8109	0.8077
0.790	0.8446	0.8382	0.8329	0.8329	0.8329	0.8329	0.8254	0.8254	0.8254	0.8329	0.8312	0.8301	0.8274	0.8255	0.8223
0.800	0.8612	0.8543	0.8485	0.8485	0.8485	0.8485	0.8410	0.8410	0.8410	0.8485	0.8468	0.8457	0.8430	0.8411	0.8379

Nota: todos los valores bajo la línea son extrapolaciones y están sujetos a tolerancias especiales.

¹ Obtenido de ASME. "Fluid Meters: their theory and application", ASME 1959.

Tubo de 3 pulgadas, diámetro = 3.068"

PiRe	500	1,000	2,000	3,000	4,000	5,000	6,000	8,000	10,000	15,000	20,000	25,000	50,000	100,000	500,000	10 ⁶
0.100	0.6159	0.6096	0.6051	0.6031	0.6019	0.6011	0.6005	0.5997	0.5991	0.5982	0.5977	0.5973	0.5964	0.5958	0.5950	0.5948
0.150	0.6203	0.6125	0.6069	0.6044	0.6030	0.6020	0.6012	0.6002	0.5995	0.5984	0.5977	0.5973	0.5962	0.5954	0.5943	0.5941
0.200	0.6258	0.6163	0.6096	0.6067	0.6049	0.6037	0.6028	0.6016	0.6007	0.5994	0.5986	0.5981	0.5967	0.5958	0.5945	0.5942
0.250	0.6322	0.6211	0.6132	0.6097	0.6077	0.6063	0.6052	0.6037	0.6027	0.6012	0.6003	0.5996	0.5981	0.5970	0.5955	0.5951
0.300	0.6398	0.6269	0.6179	0.6138	0.6114	0.6098	0.6086	0.6069	0.6057	0.6039	0.6029	0.6021	0.6003	0.5990	0.5973	0.5969
0.335	0.6442	0.6304	0.6207	0.6164	0.6138	0.6121	0.6108	0.6089	0.6077	0.6058	0.6046	0.6038	0.6019	0.6005	0.5987	0.5982
0.350	0.6471	0.6334	0.6239	0.6193	0.6166	0.6147	0.6133	0.6114	0.6100	0.6080	0.6068	0.6059	0.6038	0.6024	0.6004	0.5999
0.375	0.6498	0.6361	0.6267	0.6228	0.6198	0.6178	0.6163	0.6142	0.6128	0.6106	0.6093	0.6084	0.6062	0.6046	0.6025	0.6020
0.400		0.6439	0.6320	0.6267	0.6236	0.6214	0.6198	0.6176	0.6161	0.6137	0.6123	0.6114	0.6090	0.6073	0.6051	0.6045
0.425		0.6496	0.6369	0.6313	0.6279	0.6256	0.6239	0.6216	0.6199	0.6174	0.6159	0.6149	0.6124	0.6106	0.6082	0.6076
0.450		0.6561	0.6425	0.6365	0.6329	0.6305	0.6287	0.6261	0.6244	0.6217	0.6201	0.6190	0.6163	0.6144	0.6118	0.6112
0.475		0.6634	0.6490	0.6425	0.6388	0.6361	0.6343	0.6316	0.6296	0.6269	0.6250	0.6239	0.6210	0.6189	0.6162	0.6155
0.500			0.6563	0.6494	0.6453	0.6425	0.6405	0.6376	0.6356	0.6326	0.6307	0.6295	0.6264	0.6242	0.6213	0.6206
0.520			0.6629	0.6556	0.6513	0.6484	0.6462	0.6431	0.6411	0.6378	0.6359	0.6346	0.6313	0.6290	0.6259	0.6252
0.540			0.6702	0.6626	0.6580	0.6549	0.6526	0.6493	0.6471	0.6437	0.6417	0.6403	0.6368	0.6344	0.6311	0.6303
0.560			0.6783	0.6702	0.6654	0.6621	0.6597	0.6562	0.6539	0.6503	0.6481	0.6466	0.6430	0.6404	0.6369	0.6361
0.580			0.6873	0.6787	0.6736	0.6701	0.6675	0.6639	0.6613	0.6576	0.6553	0.6537	0.6498	0.6471	0.6434	0.6426
0.600			0.6882	0.6827	0.6790	0.6763	0.6725	0.6698	0.6658	0.6628	0.6603	0.6586	0.6546	0.6519	0.6481	0.6473
0.620			0.6987	0.6929	0.6889	0.6865	0.6826	0.6791	0.6747	0.6712	0.6694	0.6660	0.6629	0.6596	0.6557	0.6549
0.640			0.7104	0.7042	0.7000	0.6968	0.6924	0.6894	0.6848	0.6820	0.6801	0.6754	0.6721	0.6676	0.6636	0.6628
0.660			0.7235	0.7169	0.7123	0.7089	0.7042	0.7010	0.6960	0.6930	0.6909	0.6859	0.6823	0.6775	0.6735	0.6727
0.680				0.7383	0.7310	0.7261	0.7225	0.7173	0.7139	0.7084	0.7052	0.7030	0.6975	0.6936	0.6885	0.6877
0.700				0.7471	0.7417	0.7377	0.7321	0.7283	0.7224	0.7189	0.7165	0.7105	0.7063	0.7007	0.6953	0.6945
0.720				0.7653	0.7594	0.7550	0.7489	0.7447	0.7382	0.7343	0.7317	0.7251	0.7205	0.7143	0.7088	0.7080
0.740				0.7862	0.7797	0.7748	0.7680	0.7634	0.7562	0.7519	0.7489	0.7416	0.7365	0.7296	0.7240	0.7232
0.750				0.7929	0.7910	0.7858	0.7786	0.7737	0.7661	0.7615	0.7584	0.7507	0.7453	0.7380	0.7323	0.7315
0.760					0.8011	0.7977	0.7900	0.7848	0.7767	0.7719	0.7686	0.7604	0.7546	0.7469	0.7411	0.7403
0.770					0.8163	0.8105	0.8024	0.7968	0.7882	0.7830	0.7795	0.7708	0.7647	0.7565	0.7505	0.7497
0.780					0.8306	0.8244	0.8157	0.8098	0.8006	0.7951	0.7913	0.7820	0.7755	0.7667	0.7606	0.7598
0.790					0.8461	0.8395	0.8302	0.8238	0.8140	0.8081	0.8041	0.7941	0.7871	0.7777	0.7715	0.7707
0.800					0.8630	0.8560	0.8459	0.8391	0.8285	0.8222	0.8171	0.8072	0.7996	0.7895	0.7831	0.7823

Nota: todos los valores bajo la línea son extrapolaciones y están sujetos a tolerancias especiales.

Apendice C. Hojas de datos de los circuitos utilizados.

Motor Controllers (continued)

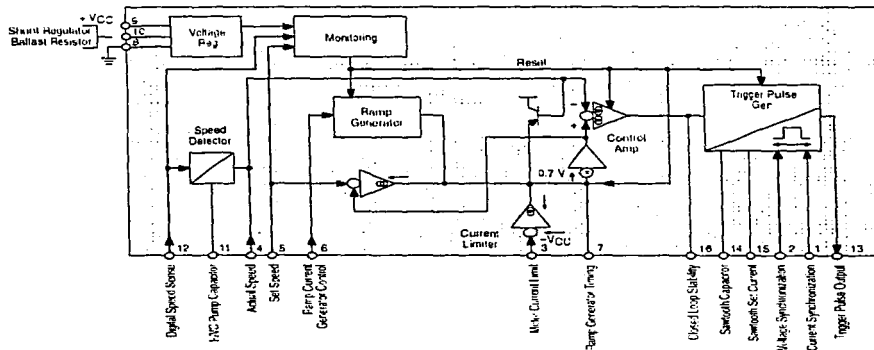
Universal Motor Speed Controller

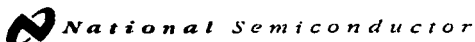
TDA1085C, CD

T_A = -10° to +120°C, Case 648, 751B

The TDA1085C is a phase angle triac controller having all the necessary functions for universal motor speed control in washing machines. It operates in closed loop configuration and provides two ramp possibilities.

- On-Chip Frequency to Voltage Converter
- On-Chip Ramps Generator
- Soft Start
- Load Current Limitation
- Tachogenerator Circuit Sensing
- Direct Supply from AC Line
- Security Functions Performed by Monitor





January 1995

DAC0808/DAC0807/DAC0806 8-Bit D/A Converters

General Description

The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of $255 I_{REF}/256$. Relative accuracies of better than $\pm 0.19\%$ assure 8-bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8-bit zero accuracy for $I_{REF} > 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

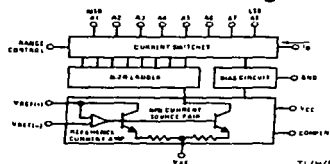
The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the

MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

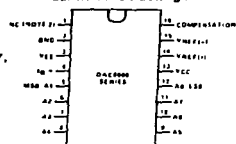
Features

- Relative accuracy: $\pm 0.19\%$ error maximum (DAC0808)
- Full scale current match: ± 1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ μs
- Power supply voltage range: $\pm 4.5V$ to $\pm 18V$
- Low power consumption: 33 mW @ $\pm 5V$

Block and Connection Diagrams



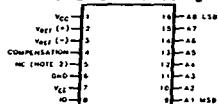
Dual-In-Line Package



Order Number
DAC0808, DAC0807,
or DAC0806
See NS Package
Number J16A,
M16A or N16A

TL/H/5687-1

Small-Outline Package



Top View

TL/H/5687-13

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS				
		J PACKAGE (J16A)*		N PACKAGE (N16A)*		SO PACKAGE (M16A)
7-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$	DAC0807LCJ	MC1408L7	DAC0808LCN	MC1408P8	DAC0808LCM
6-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$	DAC0806LCJ	MC1408L6	DAC0806LCN	MC1408P6	DAC0806LCM

*Note: Devices may be ordered by using either order number.

DAC0808/DAC0807/DAC0806 8-BIT D/A Converters

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage

V_{CC} +18 V_{DC}
 V_{EE} -18 V_{DC}

Digital Input Voltage, V_S-V_{I2} -10 V_{DC} to +18 V_{DC}

Applied Output Voltage, V_O -11 V_{DC} to +18 V_{DC}

Reference Current, I₁₄ 5 mA

Reference Amplifier Inputs, V_{I4}, V_{I5} V_{CC}, V_{EE}

Power Dissipation (Note 3) 1000 mW

ESD Susceptibility (Note 4) TBD

Storage Temperature Range -65°C to +150°C

Lead Temp. (Soldering, 10 seconds)

Dual-In-Line Package (Plastic) 260°C

Dual-In-Line Package (Ceramic) 300°C

Surface Mount Package

Vapor Phase (60 seconds) 215°C

Infrared (15 seconds) 220°C

Operating Ratings

Temperature Range

DAC0808LC Series

T_{MIN} ≤ T_A ≤ T_{MAX}

0 ≤ T_A ≤ +75°C

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = -15 V_{DC}, V_{REF}/R₁₄ = 2 mA, DAC0808: T_A = -55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E _r	Relative Accuracy (Error Relative to Full Scale I ₀) DAC0808LC (LM1408-B) DAC0807LC (LM1408-7), (Note 5) DAC0806LC (LM1408-6), (Note 5) Settling Time to Within 1/2 LSB (includes t _{PLH})	(Figure 4)				%
					±0.19	%
					±0.39	%
					±0.78	%
			T _A = 25°C (Note 6), (Figure 5)	150		
t _{PLH} , t _{PHL}	Propagation Delay Time	T _A = 25°C, (Figure 5)		30	100	ns
TC _{IO}	Output Full Scale Current Drift			-20		ppm/°C
MSB V _{IH} V _{IL}	Digital Input Logic Levels High Level, Logic "1" Low Level, Logic "0"	(Figure 3)	2		0.8	V _{DC} V _{DC}
MSB	Digital Input Current High Level Low Level	(Figure 3) V _{IH} = 5V V _{IL} = 0.8V		0 -0.003	0.040 -0.8	mA mA
I ₁₅	Reference Input Bias Current	(Figure 3)		-1	-3	μA
I _O	Output Current Range	(Figure 3)				
		V _{EE} = -5V	0	2.0	2.1	mA
		V _{EE} = -15V, T _A = 25°C	0	2.0	4.2	mA
I _O	Output Current	V _{REF} = 2.000V, R ₁₄ = 1000Ω, (Figure 3)	1.9	1.99	2.1	mA
		(Figure 3)		0	4	μA
		Output Current, All Bits Low				
I _O	Output Voltage Compliance (Note 2) V _{EE} = -5V, I _{REF} = 1 mA V _{EE} Below -10V	E _r ≤ 0.19%, T _A = 25°C			-0.55, +0.4	V _{DC}
					-5.0, +0.4	V _{DC}

Electrical Characteristics (Continued)

($V_{CC} = 5V$, $V_{EE} = -15V_{DC}$, $V_{REF}/R_{14} = 2mA$, DAC0808, DAC0807C, DAC0806C, $T_A = 0^{\circ}C$ to $+75^{\circ}C$, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
SR_{IREF}	Reference Current Slew Rate	(Figure 6)	4	8		$mA/\mu s$
	Output Current Power Supply Sensitivity	$-5V \leq V_{EE} \leq -16.5V$		0.05	2.7	$\mu A/V$
I_{CC} I_{EE}	Power Supply Current (All Bits Low)	(Figure 3)		2.3 -4.3	22 -13	mA mA
V_{CC} V_{EE}	Power Supply Voltage Range	$T_A = 25^{\circ}C$, (Figure 3)	4.5 -4.5	5.0 -15	5.5 -16.5	V_{DC} V_{DC}
	Power Dissipation All Bits Low	$V_{CC} = 5V$, $V_{EE} = -5V$ $V_{CC} = 5V$, $V_{EE} = -15V$ $V_{CC} = 15V$, $V_{EE} = -5V$ $V_{CC} = 15V$, $V_{EE} = -15V$		33 106 90 160	170 305	mW mW mW mW

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: Range control is not required.

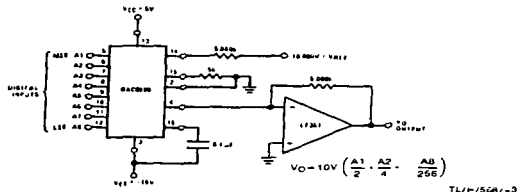
Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX} , θ_{JA} , and the ambient temperature, T_A . The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maximum Ratings, whichever is lower. For the device, $T_{JMAX} = 125^{\circ}C$, and the typical junction-to-ambient thermal resistance of the dual-in-line package when the board mounted is $100^{\circ}C/W$. For the dual-in-line N package, the number increases to $175^{\circ}C/W$ and for the small outline M package the number is $100^{\circ}C/W$.

Note 4: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Note 5: All current switches are tested to guarantee at least 50% of rated current.

Note 6: All bits switched.

Note 7: Pin-out numbers for the DAC080X represent the dual-in-line package. The small outline package pinout differs from the dual-in-line package.

Typical Application

LM2907/LM2917 Frequency to Voltage Converter

General Description

The LM2907, LM2917 series are monolithic frequency to voltage converters with a high gain op amp/comparator designed to operate a relay, lamp, or other load when the input frequency reaches or exceeds a selected rate. The tachometer uses a charge pump technique and offers frequency doubling for low ripple, full input protection in two versions (LM2907-8, LM2917-8) and its output swings to ground for a zero frequency input.

Advantages

- Output swings to ground for zero frequency input
- Easy to use: $V_{OUT} = f_{IN} \times V_{CC} \times R1 \times C1$
- Only one RC network provides frequency doubling
- Zener regulator on chip allows accurate and stable frequency to voltage or current conversion (LM2917)

Features

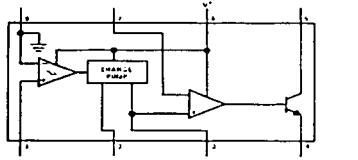
- Ground referenced tachometer input interfaces directly with variable reluctance magnetic pickups
- Op amp/comparator has floating transistor output
- 50 mA sink or source to operate relays, solenoids, meters, or LEDs

- Frequency doubling for low ripple
- Tachometer has built-in hysteresis with either differential input or ground referenced input
- Built-in zener on LM2917
- $\pm 0.3\%$ linearity typical
- Ground referenced tachometer is fully protected from damage due to swings above V_{CC} and below ground

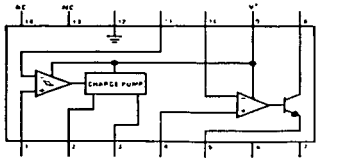
Applications

- Over/under speed sensing
- Frequency to voltage conversion (tachometer)
- Speedometers
- Breaker point dwell meters
- Hand-held tachometer
- Speed governors
- Cruise control
- Automotive door lock control
- Clutch control
- Horn control
- Touch or sound switches

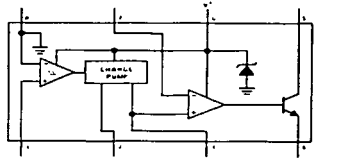
Block and Connection Diagrams Dual-In-Line and Small Outline Packages, Top Views



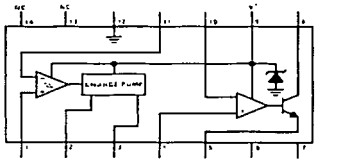
Order Number LM2907M-8 or LM2907N-8
See NS Package Number M08A or N08E



Order Number LM2907N
See NS Package Number N14A



Order Number LM2917M-8 or LM2917N-8
See NS Package Number M08A or N08E



Order Number LM2917M or LM2917N
See NS Package Number M14A or N14A

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	28V
Supply Current (Zener Options)	25 mA
Collector Voltage	28V
Differential Input Voltage	28V
Tachometer	28V
Op Amp/Comparator	28V
Input Voltage Range	
Tachometer LM2907-B, LM2917-B	= 28V
Tachometer LM2907, LM2917	0.0V to + 28V
Op Amp/Comparator	0.0V to + 28V

Power Dissipation		
LM2907-B, LM2917-B		1200 mW
LM2907-14, LM2917-14		1580 mW
(See Note 1)		
Operating Temperature Range		- 40°C to + 85°C
Storage Temperature Range		- 65°C to + 150°C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)		260°C
Small Outline Package		
Vapor Phase (60 seconds)		215°C
Infrared (15 seconds)		220°C
See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.		

Electrical Characteristics $V_{CC} = 12 V_{DC}$, $T_A = 25^\circ C$, see test circuit

Symbol	Parameter	Conditions	Min	Typ	Max	Units
TACHOMETER						
	Input Thresholds	$V_{IN} = 250 \text{ mVp-p} @ 1 \text{ kHz (Note 2)}$	≈ 10	≈ 25	≈ 40	mV
	Hysteresis	$V_{IN} = 250 \text{ mVp-p} @ 1 \text{ kHz (Note 2)}$		30		mV
	Offset Voltage	$V_{IN} = 250 \text{ mVp-p} @ 1 \text{ kHz (Note 2)}$				
	LM2907/LM2917			3.5	10	mV
	LM2907-B/LM2917-B			5	15	mV
	Input Bias Current	$V_{IN} = - 50 \text{ mV}_{DC}$		0.1	1	μA
V_{OH}	Pin 2	$V_{IN} = + 125 \text{ mV}_{DC}$ (Note 3)		8.3		V
V_{OL}	Pin 2	$V_{IN} = - 125 \text{ mV}_{DC}$ (Note 3)		2.3		V
$I_{2, I3}$	Output Current	$V_2 = V_3 = 6.0V$ (Note 4)	140	180	240	μA
I_3	Leakage Current	$I_2 = 0, V_3 = 0$			0.1	μA
K	Gain Constant	(Note 3)	0.9	1.0	1.1	
	Linearity	$f_{IN} = 1 \text{ kHz, } 5 \text{ kHz, } 10 \text{ kHz (Note 5)}$	- 1.0	0.3	+ 1.0	%
OP/AMP COMPARATOR						
V_{OS}		$V_{IN} = 6.0V$		3	10	mV
I_{BIAS}		$V_{IN} = 6.0V$		50	500	nA
	Input Common-Mode Voltage		0		$V_{CC} - 1.5V$	V
	Voltage Gain			200		V/mV
	Output Sink Current	$V_C = 1.0$	40	50		mA
	Output Source Current	$V_E = V_{CC} - 2.0$		10		mA
	Saturation Voltage	$I_{SINK} \approx 5 \text{ mA}$		0.1	0.5	V
		$I_{SINK} \approx 20 \text{ mA}$			1.0	V
		$I_{SINK} \approx 50 \text{ mA}$		1.0	1.5	V

Electrical Characteristics $V_{CC} = 12 V_{DC}$, $T_A = 25^\circ C$, see test circuit (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
ZENER REGULATOR						
	Regulator Voltage	$R_{DROP} = 470\Omega$		7.56		V
	Series Resistance			10.5	15	Ω
	Temperature Stability			+1		mV/ $^\circ C$
	TOTAL SUPPLY CURRENT			3.8	6	mA

Note 1: For operation in ambient temperatures above $25^\circ C$, the device must be derated based on a $150^\circ C$ maximum junction temperature and a thermal resistance of $10^\circ C/W$ junction to ambient for LM2007-B and LM2017-B, and $70^\circ C/W$ junction to ambient for LM2007-14 and LM2017-14.

Note 2: Hysteresis is the sum $-V_{IH} - (-V_{IL})$; offset voltage is their difference. See test circuit.

Note 3: V_{OH} is equal to $\frac{1}{2} \times V_{CC} - 1 V_{BE}$; V_{OL} is equal to $\frac{1}{2} \times V_{CC} - 1 V_{BE}$ therefore $V_{OH} - V_{OL} = V_{CC}/2$. The difference, $V_{OH} - V_{OL}$, and the mirror gain, I_2/I_3 , are the two factors that cause the tachometer gain constant to vary from 1.0.

Note 4: Be sure when choosing the time constant $R1 \times C1$ that $R1$ is such that the maximum anticipated output voltage at pin 3 can be reached with $I_2 \times R1$. The maximum value for $R1$ is limited by the output resistance of pin 3 which is greater than $10 M\Omega$ typically.

Note 5: Nonlinearity is defined as the deviation of V_{OUT} (at pin 3) for $I_N = 5$ kHz from a straight line defined by the V_{OUT} at 1 kHz and V_{OUT} at 10 kHz. $C1 = 1000$ pF, $R1 = 60k$ and $C2 = 0.22$ mfd.

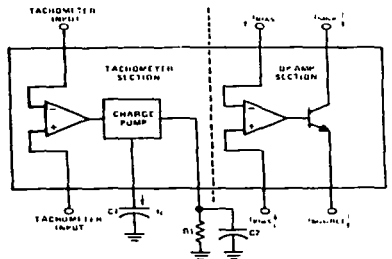
General Description (Continued)

The op amp/comparator is fully compatible with the tachometer and has a floating transistor as its output. This feature allows either a ground or supply referred load of up to 50 mA. The collector may be taken above V_{CC} up to a maximum V_{CC} of 28V.

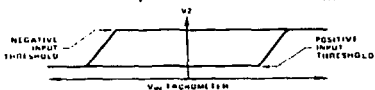
The two basic configurations offered include an 8-pin device with a *ground referenced tachometer* input and an internal connection between the tachometer output and the op amp non-inverting input. This version is well suited for single speed or frequency switching or fully buffered frequency to voltage conversion applications.

The more versatile configurations provide differential tachometer input and uncommitted op amp inputs. With this version the tachometer input may be floated and the op amp becomes suitable for active filter conditioning of the tachometer output.

Both of these configurations are available with an active shunt regulator connected across the power leads. The regulator clamps the supply such that stable frequency to voltage and frequency to current operations are possible with any supply voltage and a suitable resistor.

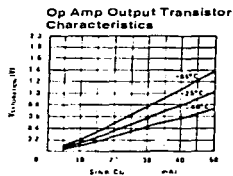
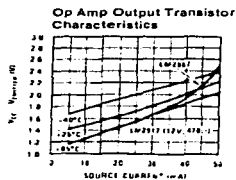
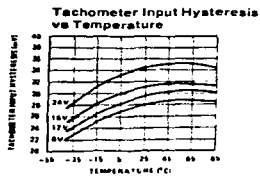
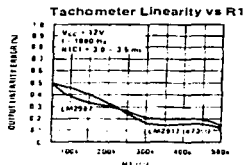
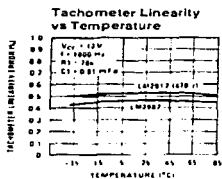
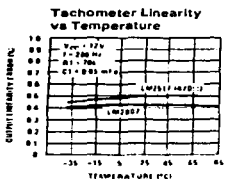
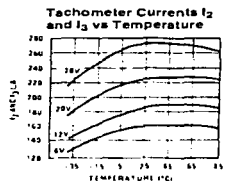
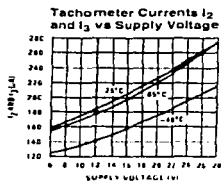
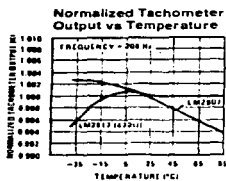
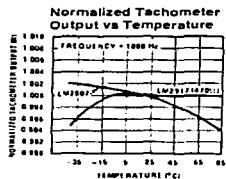
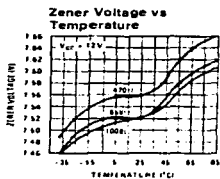
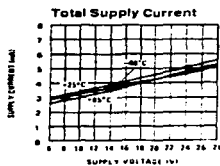
Test Circuit and Waveform


TLH/7942-0

Tachometer Input Threshold Measurement


- TLH/7942-1

Typical Performance Characteristics



TL/M7042-5

Applications Information

The LM2907 series of tachometer circuits is designed for minimum external part count applications and maximum versatility. In order to fully exploit its features and advantages let's examine its theory of operation. The first stage of operation is a differential amplifier driving a positive feedback flip-flop circuit. The input threshold voltage is the amount of differential input voltage at which the output of this stage changes state. Two options (LM2907-B, LM2917-B) have one input internally grounded so that an input signal must swing above and below ground and exceed the input thresholds to produce an output. This is offered specifically for magnetic variable reluctance pickups which typically provide a single-ended ac output. This single input is also fully protected against voltage swings to $\pm 28V$, which are easily attained with these types of pickups.

The differential input options (LM2907, LM2917) give the user the option of setting his own input switching level and still have the hysteresis around that level for excellent noise rejection in any application. Of course in order to allow the inputs to attain common-mode voltages above ground, input protection is removed and neither input should be taken outside the limits of the supply voltage being used. It is very important that an input not go below ground without some resistance in its lead to limit the current that will then flow in the ep-ii-substrate diode.

Following the input stage is the charge pump where the input frequency is converted to a dc voltage. To do this requires one timing capacitor, one output resistor, and an integrating or filter capacitor. When the input stage changes state (due to a suitable zero crossing or differential voltage on the input) the timing capacitor is either charged or discharged linearly between two voltages whose difference is $V_{CC}/2$. Then in one half cycle of the input frequency or a time equal to $1/2 f_{IN}$ the change in charge on the timing capacitor is equal to $V_{CC}/2 \times C_1$. The average amount of current pumped into or out of the capacitor then is:

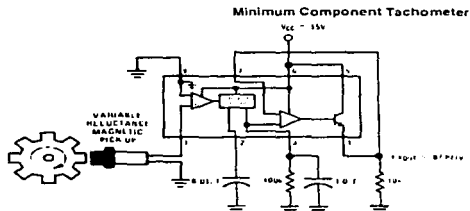
$$\frac{\Delta Q}{T} = I_C(AVG) = C_1 \times \frac{V_{CC}}{2} \times (2f_{IN}) = V_{CC} \times f_{IN} \times C_1$$

The output circuit mirrors this current very accurately into the load resistor R_1 , connected to ground, such that if the pulses of current are integrated with a filter capacitor, then $V_O = I_C \times R_1$, and the total conversion equation becomes:

$$V_O = V_{CC} \times f_{IN} \times C_1 \times R_1 \times K$$

Where K is the gain constant—typically 1.0.

Typical Applications



The size of C_2 is dependent only on the amount of ripple voltage allowable and the required response time.

CHOOSING R1 AND C1

There are some limitations on the choice of R_1 and C_1 which should be considered for optimum performance. The timing capacitor also provides internal compensation for the charge pump and should be kept larger than 500 pF for very accurate operation. Smaller values can cause an error current on R_1 , especially at low temperatures. Several considerations must be met when choosing R_1 . The output current at pin 3 is internally fixed and therefore V_O/R_1 must be less than or equal to this value. If R_1 is too large, it can become a significant fraction of the output impedance at pin 3 which degrades linearity. Also output ripple voltage must be considered and the size of C_2 is affected by R_1 . An expression that describes the ripple content on pin 3 for a single $R1C2$ combination is:

$$V_{RIPPLE} = \frac{V_{CC}}{2} \times \frac{C_1}{C_2} \times \left(1 - \frac{V_{CC} \times f_{IN} \times C_1}{I_2}\right) \text{ pk-pk}$$

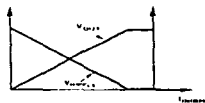
It appears R_1 can be chosen independent of ripple, however response time, or the time it takes V_{OUT} to stabilize at a new voltage increases as the size of C_2 increases, so a compromise between ripple, response time, and linearity must be chosen carefully.

As a final consideration, the maximum attainable input frequency is determined by V_{CC} , C_1 and I_2 :

$$f_{MAX} = \frac{I_2}{C_1 \times V_{CC}}$$

USING ZENER REGULATED OPTIONS (LM2917)

For those applications where an output voltage or current must be obtained independent of supply voltage variations, the LM2917 is offered. The most important consideration in choosing a dropping resistor from the unregulated supply to the device is that the tachometer and op amp circuitry alone require about 3 mA at the voltage level provided by the zener. At low supply voltages there must be some current flowing in the resistor above the 3 mA circuit current to operate the regulator. As an example, if the raw supply varies from 9V to 16V, a resistance of 4701 Ω will minimize the zener voltage variation to 160 mV. If the resistance goes under 4001 Ω or over 6001 Ω the zener variation quickly rises above 200 mV for the same input variation.



TL77942-B

LinCMOS™ TLC271, TLC271A, TLC271B PROGRAMMABLE LOW-POWER OPERATIONAL AMPLIFIERS

PROCESSING: NOVEMBER 1987 - REVISED AUGUST 1995

- **Input Offset Voltage Drift** . . . Typically 0.1 $\mu\text{V}/\text{Month}$, including the First 30 Days
- **Wide Range of Supply Voltages Over Specified Temperature Range:**
0°C to 70°C . . . 3 V to 16 V
-40°C to 85°C . . . 4 V to 16 V
-55°C to 125°C . . . 5 V to 16 V
- **Single-Supply Operation**
- **Common-Mode Input Voltage Range Extends Below the Negative Rail (C-Suffix and I-Suffix Types)**
- **Low Noise** . . . 25 nV/√Hz Typically at f = 1 kHz (High-Bias Mode)
- **Output Voltage Range includes Negative Rail**
- **High Input Impedance** . . . 10¹² Ω Typ
- **ESD-Protection Circuitry**
- **Small-Outline Package Option Also Available in Tape and Reel**
- **Designed-in Latch-Up Immunity**

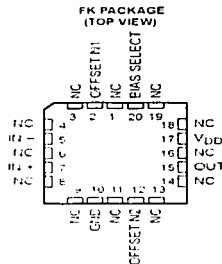
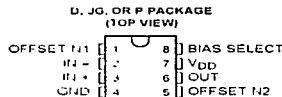
description

The TLC271 operational amplifier combines a wide range of input offset voltage grades with low offset voltage drift and high input impedance. In addition, the TLC271 offers a bias-select mode that allows the user to select the best combination of power dissipation and ac performance for a particular application. These devices use Texas Instruments silicon-gate LinCMOS™ technology, which provides offset voltage stability far exceeding the stability available with conventional metal-gate processes.

AVAILABLE OPTIONS

TA	V _{IO} max AT 25°C	PACKAGE			
		SMALL OUTLINE (D)	CHIP CARRIER (FK)	CERAMIC DIP (JG)	PLASTIC DIP (P)
0°C to 70°C	2 mV 5 mV 10 mV	TLC271BCD TLC271ACD TLC271CD	—	—	TLC271BCP TLC271ACP TLC271CP
-40°C to 85°C	2 mV 5 mV 10 mV	TLC271BDG TLC271ADG TLC271DG	—	—	TLC271BDP TLC271ADP TLC271DP
-55°C to 125°C	10 mV	TLC271MD	TLC271MFK	TLC271MJG	TLC271MP

The D package is available taped and reeled. Add R suffix to the device type (e.g., TLC271BCDR).



NC - No internal connection



LinCMOS is a trademark of Texas Instruments Incorporated.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor product™ and disclaimers thereto appears at the end of this data sheet.

PRODUCTS DATA INFORMATION IS CURRENT AS OF PUBLICATION DATE.
Products conform to specifications and the terms of Texas Instruments
standard warranty. Product processing does not necessarily include
testing of all parameters.

**TEXAS
INSTRUMENTS**
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1995, Texas Instruments Incorporated

TLC271, TLC271A, TLC271B
LinCMOS™ PROGRAMMABLE LOW-POWER
OPERATIONAL AMPLIFIERS

SLOS090B - NOVEMBER 1987 - REVISED AUGUST 1993

PARAMETER†	DEVICE FEATURES			UNIT
	BIAS-SELECT MODE			
	HIGH	MEDIUM	LOW	
PD	3375	525	50	μW
SR	3.6	0.4	0.03	$\text{V}/\mu\text{s}$
V_{II}	25	32	65	$\text{nV}/\sqrt{\text{Hz}}$
B_1	1.7	0.5	0.05	MHz
A_{VD}	23	170	460	V/mV

† Typical at $V_{DD} = 5\text{ V}$, $I_A = 25\text{ }\mu\text{A}$, $25\text{ }^\circ\text{C}$

description (continued)

Using the bias-select option, these cost-effective devices can be programmed to span a wide range of applications that previously required BIFET, NFET or bipolar technology. Three offset voltage grades are available (C-suffix and I-suffix types), ranging from the low-cost TLC271 (10 mV) to the TLC271B (2 mV) low-offset version. The extremely high input impedance and low bias currents, in conjunction with good common-mode rejection and supply voltage rejection, make these devices a good choice for new state-of-the-art designs as well as for upgrading existing designs.

In general, many features associated with bipolar technology are available in LinCMOS™ operational amplifiers, without the power penalties of bipolar technology. General applications such as transducer interfacing, analog calculations, amplifier blocks, active filters, and signal buffering are all easily designed with the TLC271. The devices also exhibit low-voltage single-supply operation, making them ideally suited for remote and inaccessible battery-powered applications. The common-mode input voltage range includes the negative rail.

A wide range of packaging options is available, including small-outline and chip-carrier versions for high-density system applications.

The device inputs and output are designed to withstand $\pm 100\text{-mA}$ surge currents without sustaining latch-up.

The TLC271 incorporates internal ESD-protection circuits that prevent functional failures at voltages up to 2000 V as tested under MIL-STD-883C, Method 3015.2; however, care should be exercised in handling these devices as exposure to ESD may result in the degradation of the device parametric performance.

The C-suffix devices are characterized for operation from $0\text{ }^\circ\text{C}$ to $70\text{ }^\circ\text{C}$. The I-suffix devices are characterized for operation from $-40\text{ }^\circ\text{C}$ to $85\text{ }^\circ\text{C}$. The M-suffix devices are characterized for operation over the full military temperature range of $-55\text{ }^\circ\text{C}$ to $125\text{ }^\circ\text{C}$.

bias-select feature

The TLC271 offers a bias-select feature that allows the user to select any one of three bias levels depending on the level of performance desired. The tradeoffs between bias levels involve ac performance and power dissipation (see Table 1).

TLC271, TLC271A, TLC271B
LinCMOS™ PROGRAMMABLE LOW-POWER
OPERATIONAL AMPLIFIERS
SLOS040B - NOVEMBER 1987 - REVISED AUGUST 1996

bias-select feature (continued)

Table 1. Effect of Bias Selection on Performance

TYPICAL PARAMETER VALUES T _A = 25 °C, V _{DD} = 5 V		MODE			UNIT
		HIGH BIAS R _L = 10 kΩ	MEDIUM BIAS R _L = 100 kΩ	LOW BIAS R _L = 1 MΩ	
P _D	Power dissipation	3.4	0.5	0.05	mW
SR	Slew rate	3.6	0.4	0.03	V/μs
V _n	Equivalent input noise voltage at f = 1 kHz	25	32	68	nV/√Hz
BT	Unity-gain bandwidth	1.7	0.5	0.09	MHz
φ _m	Phase margin	46	40	34	°
AvD	Large-signal differential voltage amplification	23	170	400	V/mV

bias selection

Bias selection is achieved by connecting the bias select pin to one of three voltage levels (see Figure 1). For medium-bias applications, it is recommended that the bias select pin be connected to the midpoint between the supply rails. This procedure is simple in split-supply applications, since this point is ground. In single-supply applications, the medium-bias mode necessitates using a voltage divider as indicated in Figure 1. The use of large-value resistors reduces the current drain of the divider from the supply line. However, large-value resistors used in conjunction with a large-value capacitor require significant time to charge up to the supply midpoint after the supply is switched on. A voltage other than the midpoint can be used if it is within the voltages specified in Figure 1.

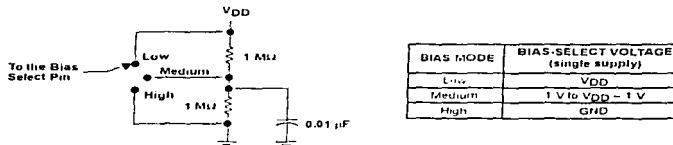
bias selection (continued)

Figure 1. Bias Selection for Single-Supply Applications

high-bias mode

In the high-bias mode, the TLC271 series features low offset voltage drift, high input impedance, and low noise. Speed in this mode approaches that of BiFET devices but at only a fraction of the power dissipation. Unity-gain bandwidth is typically greater than 1 MHz.

medium-bias mode

The TLC271 in the medium-bias mode features low offset voltage drift, high input impedance, and low noise. Speed in this mode is similar to general-purpose bipolar devices but power dissipation is only a fraction of that consumed by bipolar devices.

TLC271, TLC271A, TLC271B
linCMOS™ PROGRAMMABLE LOW-POWER
OPERATIONAL AMPLIFIERS
SLOS060H - NOVEMBER 1987 - REVISED AUGUST 1988

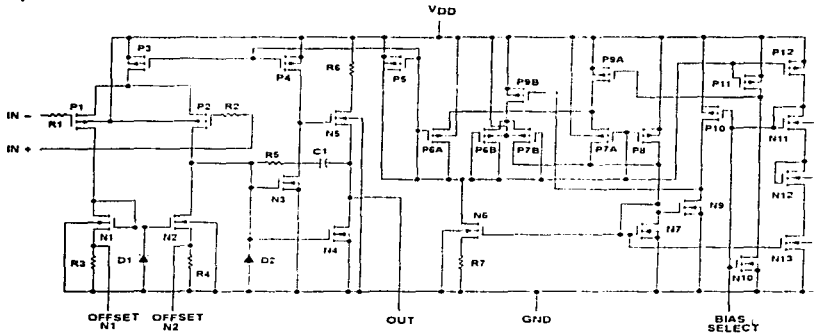
low-bias mode

In the low-bias mode, the TLC271 features low offset voltage drift, high input impedance, extremely low power consumption, and high differential voltage gain.

ORDER OF CONTENTS

TOPIC	BIAS MODE
schematic	all
absolute maximum ratings	all
recommended operating conditions	all
electrical characteristics operating characteristics typical characteristics	high (Figures 2 - 33)
electrical characteristics operating characteristics typical characteristics	medium (Figures 34 - 65)
electrical characteristics operating characteristics typical characteristics	low (Figures 66 - 97)
parameter measurement information	all
application information	all

equivalent schematic



TLC271, TLC271A, TLC271B
LinCMOS™ PROGRAMMABLE LOW-POWER
OPERATIONAL AMPLIFIERS
SLOS068B - NOVEMBER 1987 - REVISED AUGUST 1999

absolute maximum ratings over operating free-air temperature (unless otherwise noted)†

Supply voltage, V_{DD} (see Note 1)	18 V
Differential input voltage, V_{ID} (see Note 2)	± V_{DD}
Input voltage range, V_I (any input)	-0.3 V to V_{DD}
Input current, I_I	±5 mA
Output current, I_O	±130 mA
Duration of short-circuit current at (or below) 25°C (see Note 3)	Unlimited
Continuous total dissipation	See Dissipation Rating Table
Operating free-air temperature, T_A	C suffix
	I suffix
	M suffix
	0°C to 70°C
	-40°C to 85°C
	-55°C to 125°C
	-65°C to 150°C
Storage temperature range	260°C
Case temperature for 60 seconds FK package	260°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds D or P package	300°C
Lead temperature 1.6 mm (1/16 inch) from case for 60 seconds JG package	300°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. All voltage values, except differential voltages, are with respect to network ground.
 2. Differential voltages are at I_{I+} with respect to I_{I-} .
 3. The output may be shorted to either supply. Temperature and/or supply voltages must be limited to ensure that the maximum dissipation rating is not exceeded (see application section).

DISSIPATION RATING TABLE

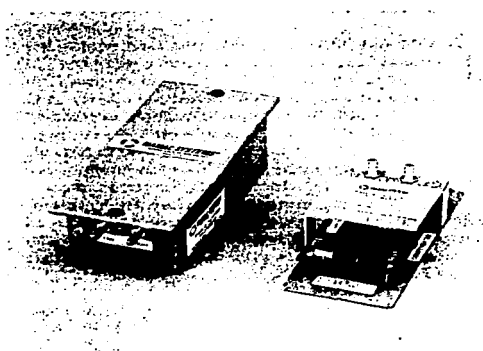
PACKAGE	$T_A = 25^\circ\text{C}$		$T_A = 70^\circ\text{C}$		$T_A = 85^\circ\text{C}$		$T_A = 125^\circ\text{C}$	
	POWER RATING	DERATING FACTOR ABOVE $T_A = 25^\circ\text{C}$	POWER RATING	POWER RATING	POWER RATING	POWER RATING	POWER RATING	
D	725 mW	5.8 mW/°C	464 mW	377 mW	145 mW			
FK	1375 mW	11.0 mW/°C	880 mW	715 mW	275 mW			
JG	1050 mW	8.4 mW/°C	672 mW	546 mW	210 mW			
P	1000 mW	8.0 mW/°C	640 mW	520 mW	200 mW			

recommended operating conditions

		C SUFFIX		I SUFFIX		M SUFFIX		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
Supply voltage, V_{DD}		3	16	4	16	5	16	V
Common-mode input voltage, V_{IC}	$V_{DD} = 5\text{ V}$	-0.2	3.5	-0.2	3.5	0	3.5	V
	$V_{DD} = 10\text{ V}$	-0.2	8.5	-0.2	8.5	0	8.5	V
Operating free-air temperature, T_A		0	70	-40	85	-55	125	°C

Low Pressure Transducer

Model PR-274/275



- 100% solid state, micro-machined, glass-on-silicon, ultra-stable capacitance sensor
- As low as $\pm 0.05''\text{wc}$ (± 12.5 pa)
- Can resolve less than $0.00001''\text{wc}$ (0.00025 pa)
- Up to 10 PSID overpressure without zero shift
- Up to 6 field selectable ranges in one unit
- Wide 12-40 VDC/12-35 VAC unregulated supply voltage
- Two temperature compensated output versions, 4-20 mA 2-wire or field selectable 0-5 VDC/0-10 VDC
- Non-interacting zero and span trimmers
- NIST traceable calibration
- Two rugged steel enclosure types NEMA 4 (IP-65) or panel mount for ease of installation
- Short circuit and reverse polarity protected
- Output impedance up to 1.6K ohms for mA units and 1K ohms minimum for VDC units

The PR-274/275 incorporates a new micro-machined glass-on-silicon (GI-Si) capacitance sensor. This technology revolutionizes very low pressure measurement. Temperature related zero drift, calibration shift due to overpressure, non-repeatability, non-linearity, and extremely low pressure sensitivity have been some of the problems which have plagued the controls industry. The PR-274/275 with the new GI-Si technology not only addresses all of the above shortcomings, but for the first time offers a reliable, accurate means to measure and control building/room pressure, air flow, duct pressure, filter pressure drop, or any other extremely low pressure application. Up to six field selectable direct or compound ranges, two enclosure types, field selectable outputs, fully temperature compensated NIST traceable accuracy, non-interacting zero and span adjustments, short circuit and reverse polarity protected output, and a liberal two year warranty are some of the features which make the PR-274/275 the industry's highest performance, ultra-stable, low pressure transducer.



7400 Flying Cloud Drive Minneapolis, MN 55344-3720
 USA • 800/843-5115 • 612/835-1626
 FAX 612/829-5331

Hamilton House • Dudley Road • Halesowen
 West Midlands • B63 9NH • United Kingdom
 TEL 0121/585-5335 • FAX 0121/585-7330

1st Floor • Esanda House • 104 Frame Street
 Adelaide • S. A. 5000 • Australia
 TEL (08) 232 4551 • FAX (08) 232 4715

The PR-274/275 incorporates sophisticated integrated circuits to not only provide a high level, fully conditioned and temperature compensated output, but also to offer field selectable flexibility which was unheard of in the industry. The PR-274/275 offers up to six field selectable pressure ranges in one unit. In this way, a customer does not need to know the exact pressure range prior to selection. By merely knowing the application, a unit may be selected and then later field configured for the desired pressure range. With fixed range units, in case of engineering error or incorrect selection, the only solution is expensive field recalibration or time consuming product exchange or replacement. Similarly, numerous units have to be kept in stock as spares to cover all ranges in case of field failure. The PR-274/275 with the field selectable pressure range feature, eliminates above mentioned costly inefficiencies. A single unit can be configured to cover all the pressure ranges in a particular application thereby eliminating any possibility of incorrect range selection. Additionally, one unit can be kept in stock and, in the event of a failure, it can be field configured thereby eliminating the need to stock numerous fixed range units. (For a complete listing of all the ranges available, please see the ordering information section on page three.)

On VDC output units, two additional field selectable options are available: dual outputs 0-5 or 0-10 VDC, and dual unregulated supply voltages 12-35 VAC or 12-40 VDC. By merely moving a shorting plug, one can select the desired output for the specific application. As far as supply voltage is concerned, the unit automatically configures for AC or DC and no field selection is necessary. Another feature is that the output is fully protected from short circuit to ground, or if the supply voltage is applied by mistake to the output. Past experience demonstrates that field related wiring problems do occur. Instead of denying this fact, the protection circuit is designed in to ensure trouble-free start-up. The VDC output unit is also designed to handle low impedance circuits. In fact, the unit can drive up to 1K ohms minimum. In this way, multiple controllers, indicators, or other devices can be paralleled to the output without performance degradation.

The mA output units can function over a wide unregulated supply voltage range: 12-40 VDC without any effect on calibration or performance. The unit has reverse polarity protection built in. As a result, it is next to impossible to damage the unit by miswiring. By using sophisticated low drop-out voltage regulators and CMOS integrated circuits, the mA output unit can drive very high output impedance. In fact, with only 12

VDC supply, the unit can drive 200 ohms. At 40 VDC, the unit is capable of handling up to 1600 ohms load. In this way, the output loop can be tied in series to multiple controllers, indicators, and other devices without degrading the performance.

Automated NIST traceable pressure controllers and precision Barocel* pressure sensors are utilized to calibrate and certify the PR-274/275 transducers. Calibration data on each unit is archived digitally for SPC and QC purposes. All automated calibration systems are networked and data is available on-line to numerous individuals at the same time. In this way, extremely high standards of quality and calibration integrity are maintained. Each unit is individually temperature compensated in an environmental chamber. The temperature compensation data is also digitized and archived for future reference purposes. Compensating each unit individually ensures that published specifications are adhered to.

Due to the low mass of the micro-machined capacitance GI-Si sensor, the mounting orientation error for ranges higher than 1.0"wc (250 pa) is negligible. For extremely low ranges, if the unit is installed as indicated on the label, there should be no orientation error. However, due to space limitation, if the unit cannot be installed in the indicated position, the error can be easily removed by merely adjusting the zero trimmer. Since the zero and span trimmers are non-interactive, adjustment to the zero should under no circumstance affect the calibration integrity of the unit including linearity and repeatability specifications across the range.

The PR-274/275 is available with two packaging options: a NEMA 4 (IP-65) fully gasketed, dust proof and splash proof enclosure, or a lightweight but rugged panel mount chassis for ease of installation with minimum space requirement in a control panel. The NEMA 4 (IP-65) enclosure has an external mounting bracket to facilitate field installation. A 1/2" (.875"/22.25mm dia.) knock-out for conduit connection is also provided. A liquid tight cable connector is also supplied if the unit is not being hard wired. Once installed, the enclosure maintains its environmental rating and protects the electronics and the sensing element from condensation, corrosive contaminants and other environmental pollutants. Both packaging options also have additional features for ease of installation, including unpluggable terminal blocks, rugged brass hose barsbs, easily accessible zero and span trimmers, and conveniently located shorting plugs for field selection.

SPECIFICATIONS:

Accuracy*: $\pm 1\%$ FS

Overpressure: 10 PSID

Supply Voltage: 12-40 VDC
12-35 VAC (VDC output units only)Supply Current: VDC Units - 10 mA max.
mA Units - 20 mA max.Enclosure: 18 Ga C. R. Steel NEMA 4 (IP-65)
or Panel Mount Chassis

Finish: Baked on enamel-PMS2GR88B

Compensated Temp Range: 25°F-150°F (-4°C-65°C)

T. C. Error: $\pm 0.0125\%/^{\circ}\text{F}$ ($0.02\%/^{\circ}\text{C}$)

Operating Temp Range: 0°F-175°F (-18°C-80°C)

Media Compatibility: Clean dry air or any inert gas

Environmental: 10-90%RH Non-Condensing

Termination: Unpluggable screw terminal block

Wire Size: 12 Ga max.

Load Impedance: 1.6K ohms max. at 40 VDC (mA output units)
1K ohms min. (VDC output units)

Weight: Enclosure 1.0 lbs. (.45 kg), Panel Mount: 0.5 lbs. (.25 kg)

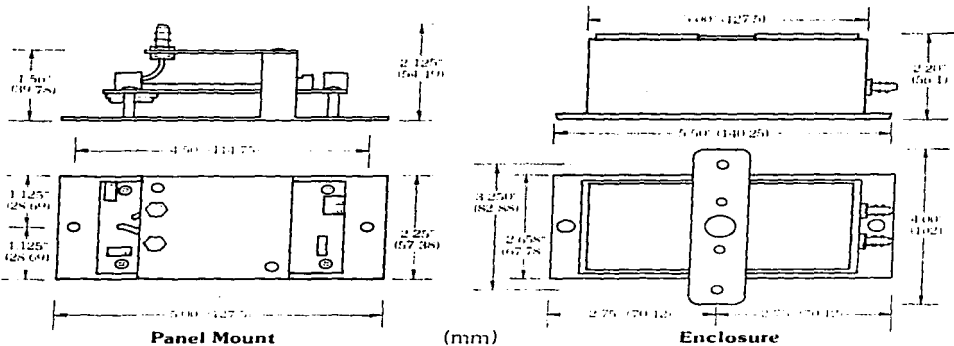
*Includes non-linearity, hysteresis and non-repeatability

ORDERING INFORMATION: PR-

PACKAGING		RANGE		OUTPUT
274 (enclosure)	R1 ("wc)	0 TO 0.10 / -0.05 TO + 0.05		mA (4-20 mA 2-wire)
275 (panel mount)	R2 ("wc)	0 TO 1.0 / 0 TO 0.5 / 0 TO 0.25 / -0.5 TO + 0.5 / -0.25 TO + 0.25 / -0.125 TO + 0.125		VDC (0-5 VDC or 0-10 VDC field selectable)
	R3 ("wc)	0 TO 5.0 / 0 TO 2.5 / 0 TO 1.25 / -2.5 TO + 2.5 / -1.25 TO + 1.25 / -0.625 TO + 0.625		
	R4 ("wc)	0 TO 30 / 0 TO 15 / 0 TO 7.5 / -15.0 TO + 15.0 / -7.5 TO + 7.5 / -3.75 TO + 3.75		
	R5*(pa)	0 TO 25 / -12.5 TO + 12.5		
	R6*(pa)	0 TO 250 / 0 TO 125 / 0 TO 62.5 / -125 TO + 125 / -62.5 TO + 62.5 / -31.25 TO + 31.25		
	R7*(pa)	0 TO 1250 / 0 TO 625 / 0 TO 312.5 / -625 TO + 625 / -312.5 TO + 312.5 / -156.25 TO + 156.25		
	R8*(pa)	0 TO 7500 / 0 TO 3750 / 0 TO 1875 / -3750 TO + 3750 / -1875 TO + 1875 / -937.5 TO + 937.5		

Example: PR-274-R2-mA: Enclosure unit with R2 Range which has six (6) field selectable range options and 4-20 mA output.

*Pascal ranges are available and stocked only in Australia and United Kingdom.

**WARRANTY:**

MAMAC Systems, Inc. warrants its products to be free of defects in material and workmanship for a period of two (2) years from date of shipment. If a unit is malfunctioning, it must be returned to the factory for evaluation. A return authorization number (RMA) will be issued by the customer service department and this number must be written or prominently displayed on the shipping boxes and all related documents. The defective part should be shipped freight pre-paid to the factory. Upon examination by MAMAC Systems, Inc. if the unit is found to be defective, it will be repaired or replaced at no charge to the customer. However, this warranty is void if the unit shows evidence of being tampered with, damaged during installation, misapplied, misused, or used in any other operating condition outside of the unit's published specifications.

MAMAC Systems, Inc. makes no other warranties or representations of any kind whatsoever, expressed or implied, except that of title. All implied warranties including any warranty of merchantability and fitness for a particular purpose are hereby disclaimed. User is responsible to determine suitability for intended use.

LIMITATIONS OF LIABILITY:

The remedies of buyer set forth herein are exclusive and the total liability of MAMAC Systems, Inc. with respect to an order, whether based on contract, warranty, negligence, indemnification, strict liability or otherwise, shall not exceed the purchase price of the product upon which liability is based. In no event shall MAMAC Systems, Inc. be liable for consequential, incidental or special damages.

MAMAC Systems Inc. reserves the right to change any specifications without notice to improve performance, reliability, or function of our products.

Every precaution for accuracy has been taken in the preparation of this manual, however, MAMAC Systems, Inc. neither assumes responsibility for any omissions or errors that may appear nor assumes liability for any damages that result from the use of the product in accordance with the information contained in the manual.



MAMAC SYSTEMS®
MONITOR • DECISION • CONTROL

7400 Flying Cloud Drive Minneapolis, MN 55344-3720
USA • 800/843-5115 • 612/835-1626
FAX 612/829-5331

Hamilton House • Dudley Road • Halesowen
West Midlands • B63 3NH • United Kingdom
TEL 0121/585-5335 • FAX 0121/585-7330

1st Floor • Esanda House • 104 Frome Street
Adelaide • S. A. 5000 • Australia
TEL (08) 232 4551 • FAX (08) 232 4715

Apéndice D. Curvas de permeabilidad y mediciones.

Este apéndice contiene los resultados de las mediciones y pruebas hechas para determinar el modelo de control y alcance de este aparato.

Se experimentó con 17 telas distintas de las cuales 5 pueden ser medidas en este aparato. Las pruebas realizadas a las 17 telas fueron dos: una de ellas consiste en observar la respuesta a escalón con incrementos y la otra consiste en determinar el gasto para distintos valores de presión en la tela. La primera de estas dos pruebas arroja información sobre la ganancia K del modelo matemático del sistema; la segunda prueba ubica a cada una de las telas en el rango del aparato medidor. Ambas pruebas están resumidas y comparadas en las gráficas tituladas "Valores de gasto vs. presión para cada tela" y "Valores de la ganancia K para cada una de las telas", en las siguientes hojas.

A las 5 telas factibles de ser medidas en este aparato se les determinó la respuesta a escalón para tres condiciones: el escalón mínimo necesario para provocar flujo turbulento en el aparato, un escalón para provocar la caída de $\frac{1}{2}$ [inH₂O] en el textil y, finalmente, el escalón necesario para provocar la máxima caída de presión medible en el textil (1 [inH₂O]). Estas tres pruebas a escalón se realizaron, según el caso, en dos intervalos de tiempo. Normalmente se eligió un intervalo de tiempo de 5 segundos para observar el transitorio y otra de 20 segundos para observar el valor a régimen permanente.

Las 17 telas analizadas, su composición y las pruebas incluidas son:

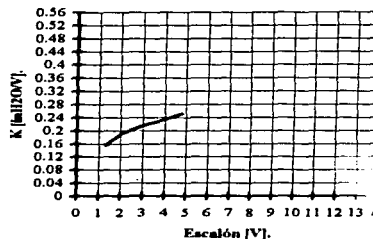
1. Malla deportiva, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; cuatro gráficas de respuesta a incrementos de escalón.
2. Tela tipo gasa, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; una gráfica de respuesta a incrementos de escalón.
3. Shiffon, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a incrementos de escalón.
4. Tejido de punto, 50% algodón y 50% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; una gráfica de respuesta a incrementos de escalón.
5. Tela de acrilán 100%. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a escalón que producen flujo turbulento, muestras de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón que producen una caída de $\frac{1}{2}$ [inH₂O] en el textil, muestras de 5 seg. y 20 seg.; una gráfica de respuesta a escalón para provocar la máxima presión medible en el textil (1 [inH₂O]) en muestra de 10 seg.; dos gráficas de respuesta a incrementos de escalón.
6. Rebozo, 100% rayón. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a escalón para provocar flujo turbulento, muestras de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón para provocar una caída de $\frac{1}{2}$ [inH₂O] en el textil, en muestras de 5 seg. y 20 seg.; una

- gráfica de respuesta a escalón para provocar la máxima caída de presión en el textil (1 [inH₂O]) en muestra de 10 seg.; dos gráficas de respuesta a incrementos de presión.
7. Tejido de punto afelpado, 50% acrílico y 50% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a escalón para provocar flujo turbulento, muestras de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón para provocar ½ [inH₂O] de caída de presión en el textil, muestra de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón para provocar la máxima caída de presión en el textil (1 [inH₂O]), muestra de 5 seg. y de 20 seg.; dos gráficas de respuesta a incrementos de escalón.
 8. Tela estampada de rayón. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a escalón para provocar flujo turbulento, muestra de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón para provocar una caída de ½ [inH₂O] en el textil, muestra de 5 seg. y 20 seg.; dos gráficas de respuesta a escalón para provocar la máxima caída de presión en el textil (1[inH₂O]), muestra de 5 seg. y 20 seg.; dos gráficas de respuesta a incrementos de presión.
 9. Lycra, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a incremento de escalón.
 10. Tela de poliéster, 100%. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a escalón para provocar flujo turbulento, muestra de 5 seg. y de 20 seg.; dos gráficas de respuesta a escalón para provocar una caída de ½ [inH₂O] en el textil, muestra de 5 seg. y de 20 seg.; dos gráficas de respuesta a escalón para provocar la caída máxima de presión en el textil (1 [inH₂O]), muestra de 5 seg. y de 20 seg.; tres gráficas de respuesta a incrementos de presión.
 11. Paño, 95% lana y 5% nylon. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a incrementos de escalón.
 12. Tela estampada, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a incrementos de presión.
 13. Tela de nylon 100%. Pruebas: tabla y gráfica de gasto vs. presión; dos gráficas de respuesta a incrementos de presión.
 14. Gabardina, 100% algodón. Pruebas: tabla de gasto vs. presión; dos gráficas de respuesta a incrementos de escalón.
 15. Forro, 100% poliéster. Pruebas: tabla y gráfica de gasto vs. presión; 1 gráfica de respuesta a incrementos de presión.
 16. Seda. Pruebas: tabla de gasto vs. presión.
 17. Franca, 100% algodón. Pruebas: tabla y gráfica de gasto vs. presión; 1 gráfica de respuesta a incrementos de presión.

Valores de ganancia (K) del sistema para cada muestra.

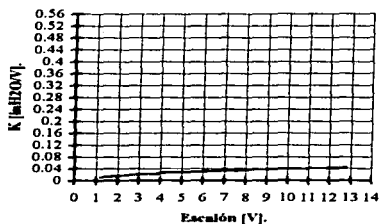
1.- Franela, 100% algodón.

Escalón [V]	Salida [mHzO]	Constante K.
1.25	0.19375	0.155
1.28125	0.196875	0.153658537
2.125	0.4125	0.194117647
2.1875	0.425	0.194285714
3.03125	0.65625	0.216494845
3.96875	0.925	0.233070866
4.8437	1.21875	0.251615501



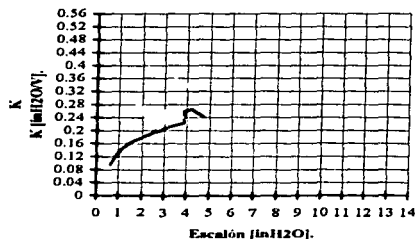
2.- Malla deportiva, 100% poliéster.

Escalón [V]	Salida [mHzO]	Constante K.
1.1875	0.0125	0.010526316
2.0625	0.034375	0.016666667
2.96875	0.0625	0.021052632
3.875	0.09375	0.024193548
3.875	0.096875	0.025
4.78125	0.140625	0.029411765
5.71875	0.171875	0.030054645
6.59375	0.21875	0.033175355
7.5	0.259375	0.034583333
8.3125	0.3125	0.037593985
9.1875	0.359375	0.039115646
10.125	0.40936	0.040430617
11	0.45312	0.041192727
11.9375	0.509375	0.042670157
12.81	0.559375	0.043667057



3.- Poliéster 100% tela azul.

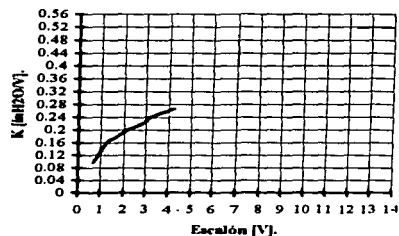
Escalón [V]	Salida [mHzO]	Constante K.
0.625	0.05936	0.094976
0.90625	0.1125	0.124137931
1.21875	0.18125	0.148717949
1.5	0.24375	0.1625
1.796875	0.3125	0.173913043
2.140625	0.39062	0.182479416
2.421875	0.459375	0.189677419
2.59375	0.51875	0.2
2.71875	0.53125	0.195402299
2.96875	0.6125	0.206315789
3.875	0.8625	0.222580645
3.90625	1.0125	0.2592
4.21875	1.11875	0.265185185
4.78125	1.15	0.240522876



Valores de ganancia (K) del sistema para cada muestra.

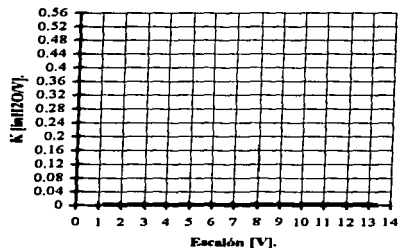
4.- Poliéster 100%, estampado.

Escalón [V]		Constante K.
0.6875	0.063625	0.095454545
1.296875	0.209375	0.161445783
1.578125	0.275	0.174257426
2.203125	0.44375	0.20141844
2.484375	0.51875	0.208805031
3.09375	0.70625	0.228282828
3.03125	0.70625	0.232989691
3.28125	0.79375	0.241904762
3.90625	1.0125	0.2592
4.21875	1.11875	0.265185185



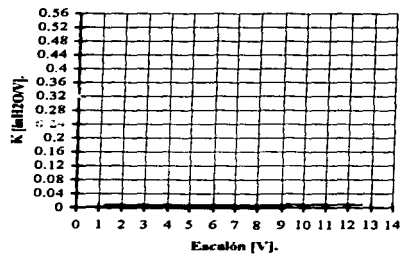
5.- Poliéster 100%, tipo gasa.

Escalón [V]		Constante K.
1.25	0.00625	0.005
4.75	0.01875	0.003947368
9.25	0.045	0.004864865
13.75	0.07125	0.005327103



6.- Shiffon, 100% poliéster.

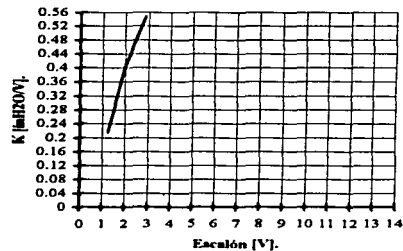
Escalón [V]		Constante K.
1.25	0.0075	0.006
4.8125	0.03125	0.006493506
9.3125	0.07125	0.007651007
9.25	0.075	0.008108108
12.625	0.12125	0.00960396



Valores de ganancia (K) del sistema para cada muestra.

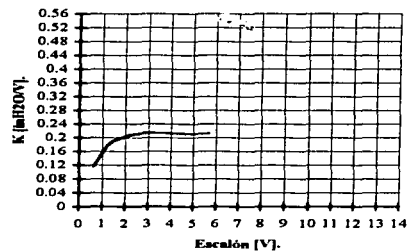
7.- Forro, 100% poliéster.

Escalon [V]		Constante K.
1.25	0.26875	0.215
2	0.80625	0.403125
2.875	1.575	0.547826087



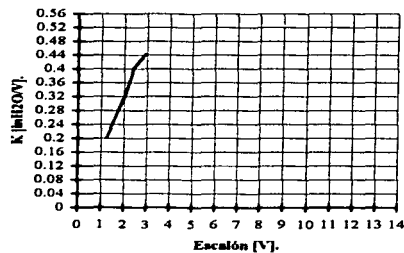
8.- Lycra, nylon 100% (tejido de punto).

Escalon [V]		Constante K.
0.625	0.07186	0.114976
1.2343	0.21874	0.177217856
1.5156	0.29062	0.191752441
2.1406	0.44062	0.205839484
2.4218	0.50936	0.2103229
3.0468	0.65312	0.21436261
3.875	0.825	0.212903226
4.78125	1.0125	0.211764706
5.65625	1.2	0.212154696



9.- Nylon 100%

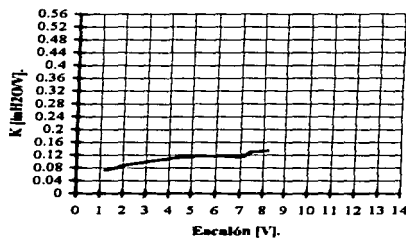
Escalon [V]		Constante K.
1.25	0.25	0.2
2.0625	0.675	0.327272727
2.4375	0.98125	0.402564103
2.9375	1.3	0.442553191



Valores de ganancia (K) del sistema para cada muestra.

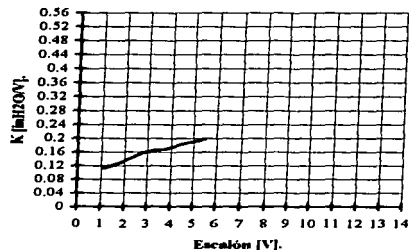
10.- Rayon 100%.

Escalón [V]		Constante K.
1.21875	0.0875	0.071794872
2.09375	0.184375	0.088059701
2.96875	0.29062	0.097893053
3.84375	0.4125	0.107317073
4.75	0.54375	0.114473684
5.625	0.6625	0.117777778
7.125	0.8125	0.114035088
7.4375	0.95625	0.128571429
8.25	1.1	0.133333333



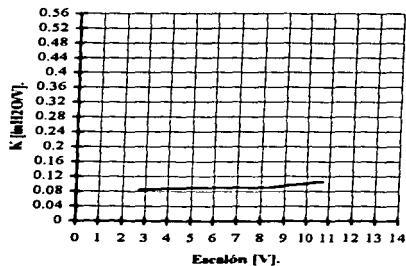
11.- Rayón 100%, estampado.

Escalón [V]		Constante K.
1.0909	0.121212	0.111111926
2	0.2666666	0.1333333
2.8125	0.44686	0.158883556
3.125	0.50936	0.1629952
3.75	0.63125	0.168333333
4.625	0.85625	0.185135135
5.5	1.08124	0.196389091



12.- Acrilón 100%.

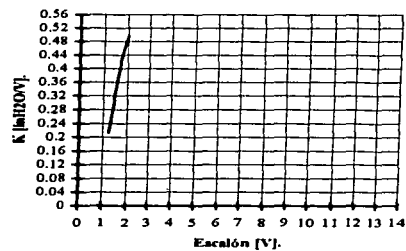
Escalón [V]		Constante K.
2.71875	0.228125	0.083908046
3.625	0.50936	0.090552889
8.3125	0.75	0.090225564
10.75	1.14375	0.106395349



Valores de ganancia (K) del sistema para cada muestra.

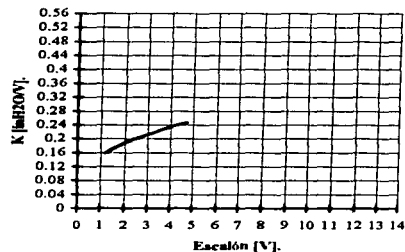
13.- Seda 100%.

Escalón [V]		Constante K.
1.25	0.26562	0.212496
1.6875	0.7	0.414814815
2.025	1.00625	0.49691358



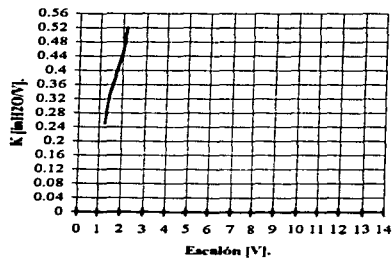
14.- Paño, 95% lana, 5% nylon.

Escalón [V]		Constante K.
1.151515	0.18125	0.157401337
2.0625	0.39375	0.190909091
3	0.6375	0.2125
3.0625	0.65	0.212244898
3.875	0.9	0.232258065
4.75	1.175	0.247368421



15.- Gabardina, 100% algodón.

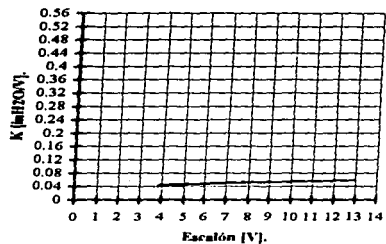
Escalón [V]		Constante K.
1.25	0.3125	0.25
1.4375	0.475	0.330434783
2	0.89375	0.446875
2.1875	1.1375	0.52



Valores de ganancia (K) del sistema para cada muestra.

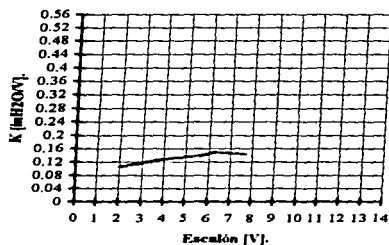
16.- Tejido de punto, 50% algodón, 50% poliéster.

Escalon [V]		Constante K.
3.875	0.16875	0.043548387
7.3125	0.375	0.051282051
10	0.5375	0.05375
12.9375	0.75312	0.058212174

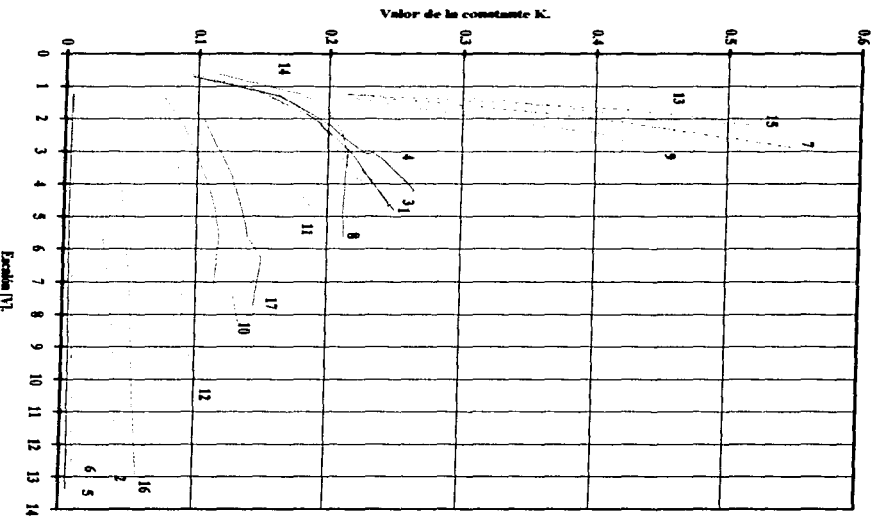


17.- Tejido de punto afelpado, 50% acrílico, 50% poliéster.

Escalon [V]		Constante K.
2	0.20936	0.10468
3.75	0.475	0.126666667
5	0.6812	0.13624
5.625	0.7812	0.13888
6.265	0.9375	0.149640862
7.735	1.11254	0.143831933



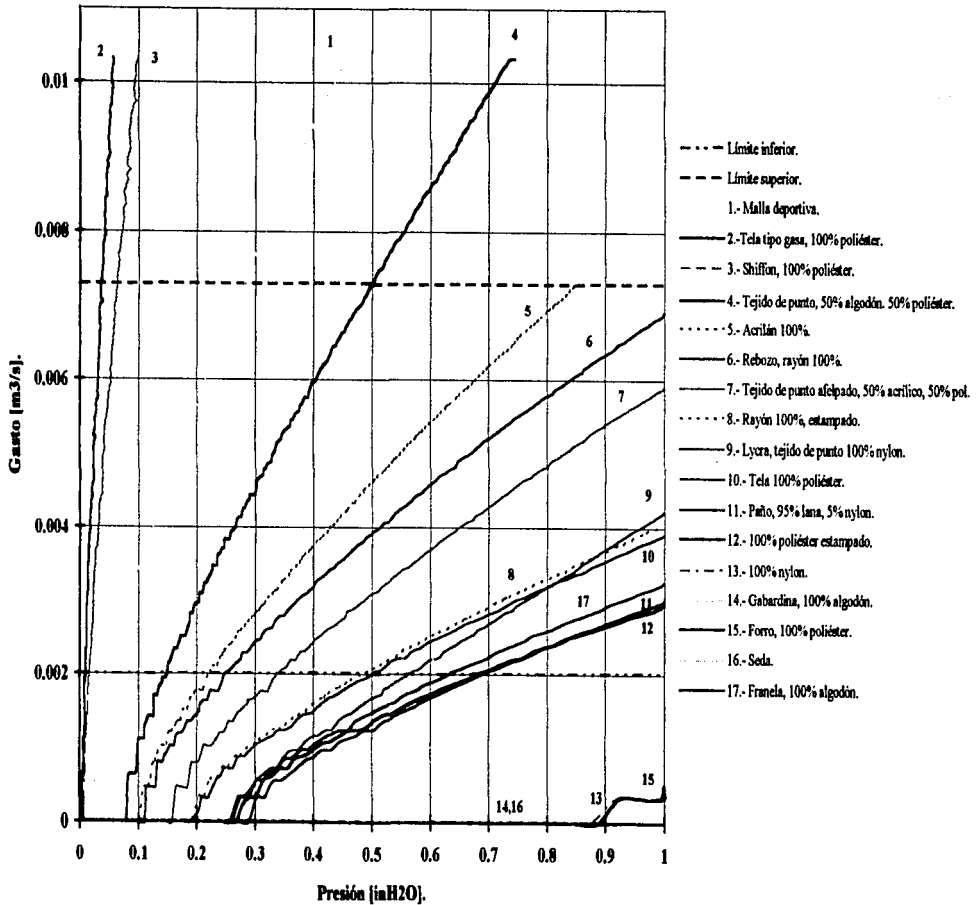
Comparativo de los valores de la ganancia K para cada una de las telas.



- 2. Malla tejedora, 100% poliéster.
- 3. Poliéster 100%, tela real.
- 6. Siliqua, 100% poliéster.
- 16. Telo de punto, 50% algodón y 50% pol.
- 12. Acetato 100%.
- 10. Rayón 100%.
- 11. Rayón 100%, estampado.
- 4. Poliéster 100%, estampado.
- 8. Liana, 100% nylon (telo de punto).
- 14. Ray, 50% lana y 50% nylon.
- 5. Poliéster 100%, 100% paja.
- 9. Nylon 100%.
- 15. Galdana, 100% algodón.
- 1. Franela, 100% algodón.
- 7. Franá, 100% poliéster.
- 13. Seda 100%.
- 17. Telo de punto tejedor, 50% algodón y 50% pol.

Excedido [1]

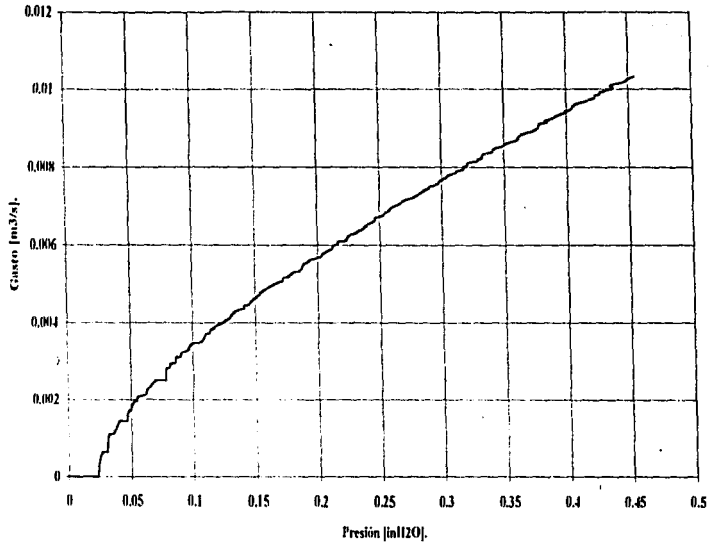
Comparativo de valores de gasto vs. presión para cada tela.



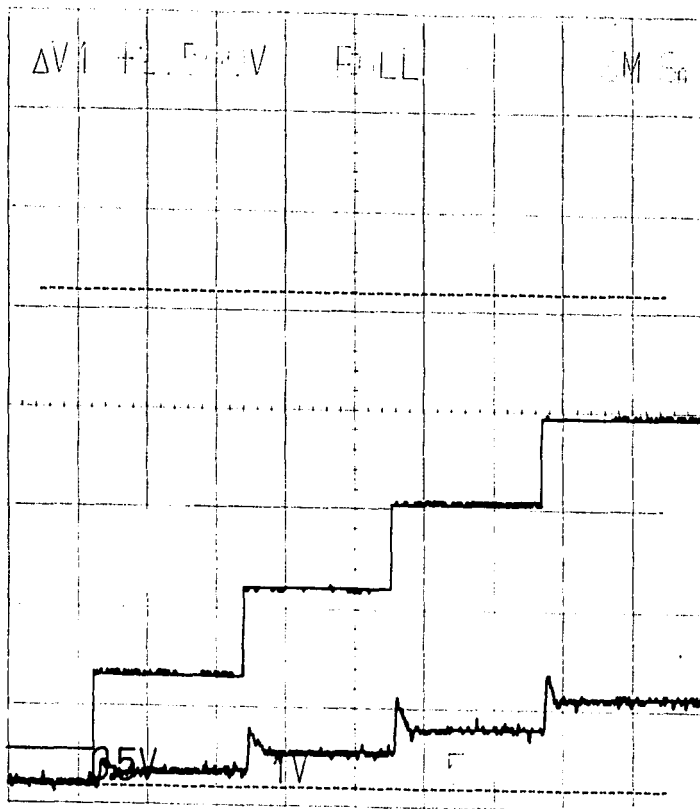
Malla deportiva.

	Presión [mfl20]	Gasto [m³/s]		Presión [mfl20]	Gasto [m³/s]		Presión [mfl20]	Gasto [m³/s]		Presión [mfl20]	Gasto [m³/s]		Presión [mfl20]	Gasto [m³/s]
0	0.00039	0	45	0.04862	0.001711152	90	0.13372	0.004260724	135	0.24705	0.006627048	180	0.36274	0.00880157
1	0	0	46	0.05019	0.001735426	91	0.13803	0.004328898	136	0.24705	0.00669057	181	0.36588	0.008851319
2	0.00196	0	47	0.05058	0.001782982	92	0.14117	0.004362606	137	0.25215	0.00673983	182	0.37215	0.00891456
3	0.00313	0	48	0.05215	0.001918508	93	0.14117	0.004429213	138	0.25451	0.006813807	183	0.3749	0.008954755
4	0.00352	0	49	0.05411	0.001940253	94	0.14156	0.004433947	139	0.25451	0.006810936	184	0.37725	0.009015101
5	0.00392	0	50	0.0549	0.001940253	95	0.1447	0.004471516	140	0.25686	0.006844524	185	0.37764	0.00906152
6	0.00431	0	51	0.05607	0.00206558	96	0.14902	0.004614234	141	0.25921	0.006917382	186	0.3796	0.009119017
7	0.0047	0	52	0.06235	0.00214504	97	0.1498	0.004618778	142	0.26117	0.00697777	187	0.38509	0.009260253
8	0.00705	0	53	0.06274	0.002174099	98	0.15333	0.004708436	143	0.26352	0.007010559	188	0.38352	0.009180608
9	0.00705	0	54	0.06274	0.002240385	99	0.15333	0.004730622	144	0.26666	0.007067133	189	0.38941	0.009255629
10	0.00784	0	55	0.06509	0.002322898	100	0.15568	0.004796455	145	0.26902	0.007114268	190	0.39137	0.009289372
11	0.00784	0	56	0.06588	0.002340814	101	0.16078	0.004882888	146	0.27215	0.007172751	191	0.39451	0.009341091
12	0.00823	0	57	0.06784	0.002437108	102	0.16078	0.004887812	147	0.27275	0.007225104	192	0.3996	0.009396877
13	0.00941	0	58	0.07019	0.00251488	103	0.1647	0.004967817	148	0.27882	0.007265589	193	0.39921	0.009428041
14	0.01098	0	59	0.07372	0.00250488	104	0.17058	0.005051319	149	0.28274	0.007325776	194	0.40392	0.009474501
15	0.01411	0	60	0.07803	0.00259488	105	0.17254	0.005067868	150	0.28431	0.007362811	195	0.40431	0.009505224
16	0.01568	0	61	0.07843	0.002538018	106	0.17254	0.005129409	151	0.28627	0.007405185	196	0.40627	0.009568788
17	0.01568	0	62	0.07843	0.002697804	107	0.17607	0.005161919	152	0.29019	0.007464246	197	0.40823	0.009612534
18	0.01568	0	63	0.07843	0.002811707	108	0.17843	0.005214312	153	0.29058	0.007514569	198	0.41294	0.009660315
19	0.01568	0	64	0.08078	0.002839338	109	0.18117	0.005293927	154	0.29725	0.007569996	199	0.41725	0.009701267
20	0.01568	0	65	0.08196	0.002921172	110	0.18549	0.005317582	155	0.29803	0.007624948	200	0.42313	0.009763555
21	0.01607	0	66	0.08313	0.00295674	111	0.18745	0.005372362	156	0.3	0.007679775	201	0.42352	0.009823458
22	0.01686	0	67	0.08627	0.002970836	112	0.18823	0.005490649	157	0.30117	0.007704273	202	0.42745	0.009870218
23	0.0196	0	68	0.08705	0.003101697	113	0.18902	0.005503134	158	0.30313	0.007733983	203	0.42745	0.009910302
24	0.02	0	69	0.0898	0.003101697	114	0.19176	0.005540994	159	0.30588	0.007804004	204	0.43137	0.009948259
25	0.02078	0	70	0.09098	0.003207807	115	0.19372	0.005589867	160	0.31098	0.007846714	205	0.43686	0.010006879
26	0.02313	0	71	0.09333	0.003233749	116	0.19607	0.0056227143	161	0.31254	0.007894598	206	0.43803	0.010044471
27	0.02352	0	72	0.09647	0.003291453	117	0.20274	0.005678037	162	0.31764	0.007947341	207	0.43647	0.010075573
28	0.02392	0.000204482	73	0.09725	0.003360617	118	0.20392	0.005748508	163	0.31764	0.007999958	208	0.44039	0.010127489
29	0.02349	0.000578455	74	0.09843	0.003403928	119	0.20392	0.005748508	164	0.32078	0.008052011	209	0.44705	0.010170755
30	0.02705	0.000646739	75	0.10117	0.003470851	120	0.20784	0.005831592	165	0.32117	0.008103729	210	0.44902	0.010230185
31	0.02784	0.000646739	76	0.10666	0.003482867	121	0.21176	0.005895976	166	0.32823	0.008147486	211	0.45098	0.010279073
32	0.02784	0.000646739	77	0.1098	0.003600969	122	0.21254	0.0059662817	167	0.32941	0.008196217	212	0.45449	0.010325836
33	0.03098	0.000678321	78	0.1098	0.003612552	123	0.21568	0.006007963	168	0.33215	0.008229335	213	0.45449	0.010327902
34	0.03137	0.00081808	79	0.11058	0.003692712	124	0.21686	0.006073383	169	0.33333	0.008302674			
35	0.03176	0.001101379	80	0.11333	0.00372096	125	0.22196	0.006101334	170	0.33372	0.008333715			
36	0.03529	0.001120216	81	0.11411	0.003815318	126	0.22431	0.006169427	171	0.3396	0.00837537			
37	0.03568	0.001150956	82	0.11607	0.003837155	127	0.22509	0.00621021	172	0.34156	0.008427635			
38	0.03803	0.001244059	83	0.11764	0.003875143	128	0.23176	0.006303687	173	0.34196	0.00846468			
39	0.03921	0.001323545	84	0.12039	0.003934059	129	0.23333	0.006343327	174	0.34627	0.008521406			
40	0.04	0.001402137	85	0.12509	0.004007794	130	0.23607	0.006389402	175	0.34941	0.00858252			
41	0.04078	0.001446189	86	0.12588	0.004038989	131	0.23803	0.006441498	176	0.35294	0.00862385			
42	0.04549	0.001446189	87	0.12902	0.004095548	132	0.24039	0.006496461	177	0.35882	0.008677077			
43	0.04705	0.001446189	88	0.13019	0.004141261	133	0.24235	0.006533492	178	0.36078	0.008713061			
44	0.04705	0.001557597	89	0.13019	0.004171458	134	0.2447	0.006563705	179	0.36156	0.008763515			

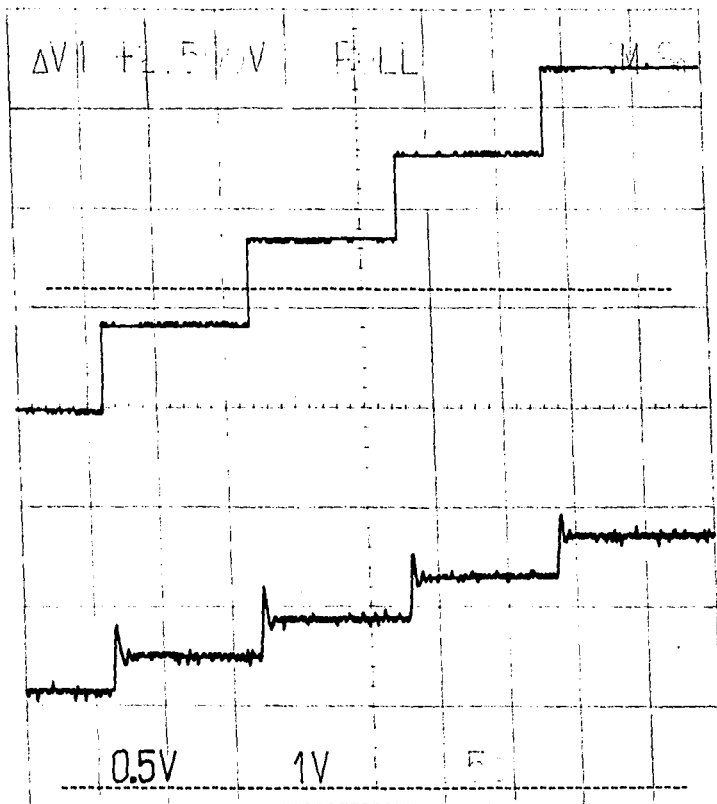
Malla deportiva.



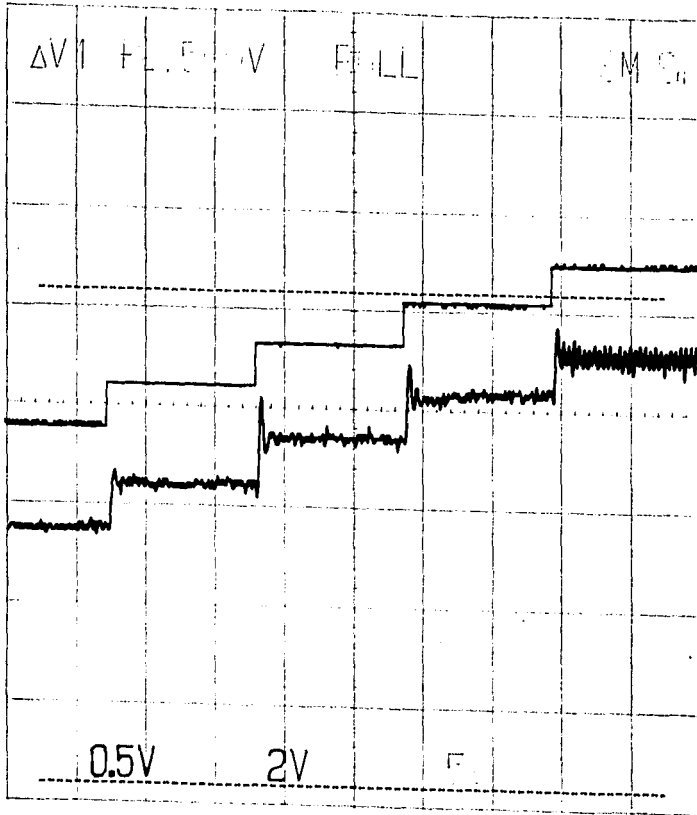
Malla deportiva, 100% poliéster.



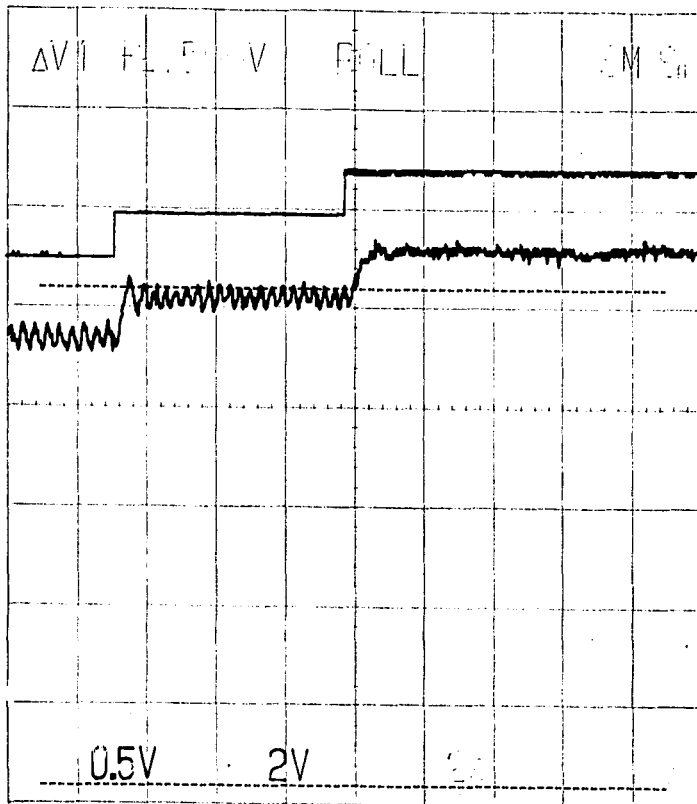
Malla deportiva, 100% poliéster.



Malla deportiva, 100% poliéster.



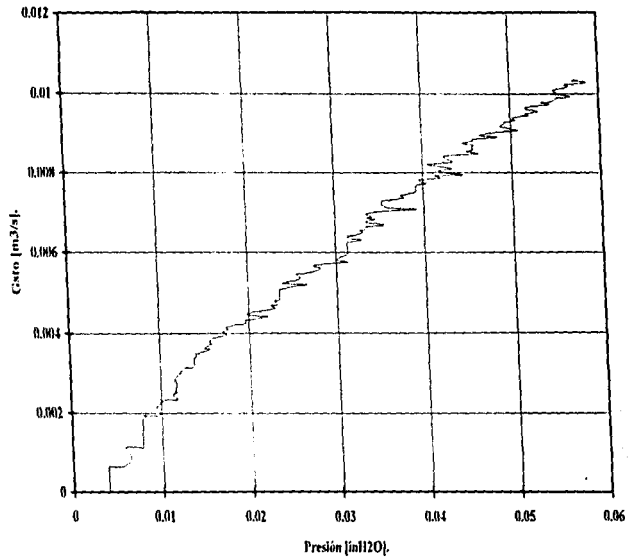
Malla deportiva, 100% poliéster.



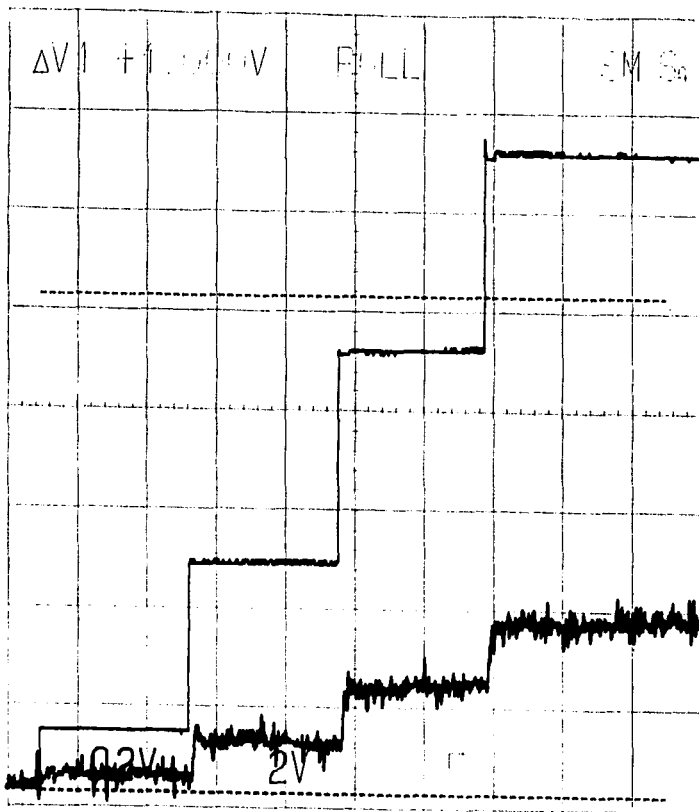
100% poliéster, tipo gasa.

	Presión [m120]	Gasto [m3/s]		Presión [m120]	Gasto [m3/s]		Presión [m120]	Gasto [m3/s]		Presión [m120]	Gasto [m3/s]		Presión [m120]	Gasto [m3/s]
0	0	0	45	0.00784	0.00147483	90	0.01705	0.004038989	135	0.03156	0.006333511	180	0.0449	0.008526411
1	0	0	46	0.00784	0.001570962	91	0.01764	0.004085335	136	0.03137	0.00639274	181	0.04568	0.008560537
2	0	0	47	0.00784	0.001711152	92	0.01745	0.004141261	137	0.03254	0.006438186	182	0.04549	0.008619107
3	0	0	48	0.00784	0.001711152	93	0.01843	0.004212137	138	0.03313	0.006499744	183	0.04568	0.008677077
4	0	0	49	0.00784	0.001771218	94	0.0196	0.004245969	139	0.03274	0.006538195	184	0.0445	0.008722849
5	0	0	50	0.00784	0.001852028	95	0.0196	0.004328998	140	0.03333	0.006571777	185	0.04568	0.008780134
6	0	0	51	0.00823	0.001940253	96	0.0196	0.004338579	141	0.03352	0.006627048	186	0.04568	0.008827587
7	0	0	52	0.00784	0.001940253	97	0.02215	0.004424474	142	0.03549	0.006690597	187	0.04843	0.008877191
8	0	0	53	0.00998	0.001940253	98	0.02117	0.004433947	143	0.03352	0.006749055	188	0.04647	0.00891476
9	0	0	54	0.0096	0.002125391	99	0.0198	0.004471516	144	0.03372	0.006799657	189	0.04705	0.008952371
10	0.00019	0	55	0.00941	0.00214504	100	0.02078	0.004536524	145	0.0345	0.006820665	190	0.04862	0.00903656
11	0.00039	0	56	0.0098	0.002193231	101	0.02294	0.004618798	146	0.03352	0.006858721	191	0.05078	0.009052294
12	0.00019	0	57	0.01	0.002277453	102	0.02254	0.004677259	147	0.03411	0.006886247	192	0.05	0.0090982292
13	0.00117	0	58	0.01058	0.002331911	103	0.02333	0.004708436	148	0.03352	0.006906785	193	0.04882	0.009150933
14	0.00156	0	59	0.01176	0.002331911	104	0.02294	0.004792117	149	0.03529	0.007034355	194	0.04901	0.009187577
15	0.00196	0	60	0.01137	0.002437108	105	0.02352	0.004813881	150	0.03921	0.007075932	195	0.04941	0.009251408
16	0.00196	0	61	0.01176	0.00250488	106	0.02352	0.00482888	151	0.03588	0.00711266	196	0.05058	0.009298362
17	0.00196	0	62	0.01176	0.00250488	107	0.02352	0.004921293	152	0.03529	0.00718142	197	0.05019	0.009332142
18	0.00196	0	63	0.01176	0.00250488	108	0.02352	0.004976256	153	0.03529	0.007230761	198	0.05078	0.009387981
19	0.00196	0	64	0.01156	0.00250488	109	0.02352	0.00503888	154	0.03529	0.007288533	199	0.05215	0.0094428041
20	0.00196	0	65	0.01176	0.002666633	110	0.02352	0.005059288	155	0.03666	0.007328445	200	0.05176	0.009465678
21	0.00215	0	66	0.01156	0.00277428	111	0.0249	0.005133462	156	0.03784	0.007379694	201	0.05313	0.009527268
22	0.00235	0	67	0.01156	0.002819095	112	0.02666	0.005198263	157	0.03745	0.007424844	202	0.05254	0.009571018
23	0.00333	0	68	0.01176	0.002877885	113	0.02392	0.005214312	158	0.03843	0.007472578	203	0.05176	0.009612534
24	0.00352	0	69	0.01196	0.00295674	114	0.02549	0.005293927	159	0.03901	0.007514569	204	0.05215	0.009658107
25	0.00392	0	70	0.01215	0.002963766	115	0.02588	0.005333306	160	0.03921	0.007572777	205	0.0545	0.009712072
26	0.00392	0	71	0.01215	0.003005816	116	0.02588	0.005372362	161	0.03921	0.007624948	206	0.05352	0.009761552
27	0.00372	0	72	0.01274	0.003101697	117	0.02549	0.005449669	162	0.03921	0.007679775	207	0.0547	0.009789195
28	0.00392	0	73	0.01352	0.003101697	118	0.02627	0.005453517	163	0.04039	0.007733983	208	0.05529	0.00987868
29	0.00392	0	74	0.01372	0.003201261	119	0.02764	0.005525994	164	0.0498	0.007761099	209	0.05607	0.009897558
30	0.00392	0.0013908	75	0.01372	0.003233749	120	0.02823	0.005593618	165	0.0396	0.007817433	210	0.05686	0.009935561
31	0.00392	0.000646739	76	0.01372	0.003263969	121	0.02745	0.005638284	166	0.04196	0.007865496	211	0.0549	0.009990165
32	0.00392	0.000646739	77	0.01372	0.003360617	122	0.02764	0.005675274	167	0.04137	0.007918432	212	0.05529	0.010017355
33	0.0047	0.000646739	78	0.01392	0.003397759	123	0.03058	0.005733924	168	0.0448	0.007942195	213	0.0549	0.010079806
34	0.00431	0.000646739	79	0.01392	0.00347689	124	0.03137	0.005748508	169	0.04235	0.007999598	214	0.05607	0.010137666
35	0.00549	0.000646739	80	0.01431	0.003482677	125	0.03	0.005820821	170	0.04176	0.008057087	215	0.05666	0.010178967
36	0.00647	0.00084327	81	0.01549	0.003589348	126	0.03078	0.005892247	171	0.04333	0.008103729	216	0.05627	0.010219927
37	0.00588	0.001120216	82	0.0149	0.00366679	127	0.03117	0.005924082	172	0.04176	0.008155119	217	0.05862	0.010275904
38	0.00784	0.001120216	83	0.01529	0.003681382	128	0.03137	0.005990777	173	0.04058	0.008206405	218	0.05705	0.010327902
39	0.00784	0.001120216	84	0.01568	0.003715318	129	0.03137	0.006039247	174	0.04333	0.008251987	219	0.05784	0.010372902
40	0.00784	0.00127724	85	0.01549	0.00376565	130	0.03137	0.006101334	175	0.04133	0.008306274			
41	0.00784	0.001293595	86	0.01568	0.003826252	131	0.03137	0.00614228	176	0.04274	0.008337715			
42	0.00784	0.001446189	87	0.01588	0.003875143	132	0.03137	0.006189852	177	0.04235	0.008390427			
43	0.00784	0.001446189	88	0.01725	0.003934059	133	0.03137	0.006237061	178	0.04333	0.008430166			
44	0.00784	0.001446189	89	0.01745	0.003955285	134	0.03294	0.006303687	179	0.04627	0.008477062			

100% poliéster, tipo gasa.



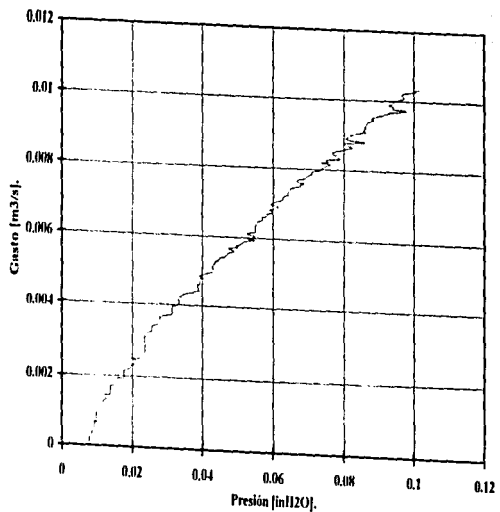
100% poliéster, tipo gasa.



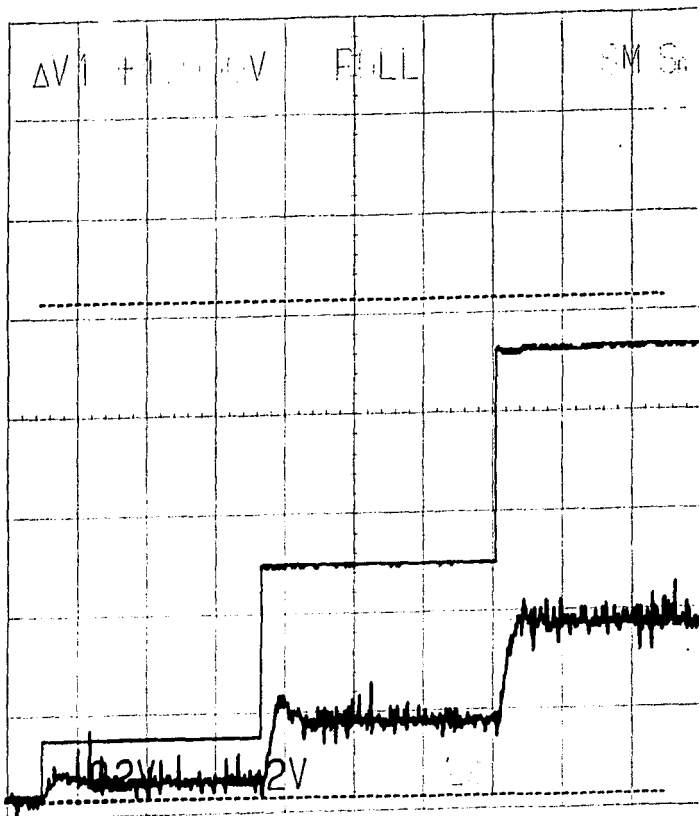
Shifton, pol. 100%

Presion [in]120	Gasto [m3/s]	Presion [in]120	Gasto [m3/s]	Presion [in]120	Gasto [m3/s]	Presion [in]120	Gasto [m3/s]	Presion [in]120	Gasto [m3/s]					
0	0	0	45	0.01372	0.001711152	90	0.03392	0.004231164	135	0.05803	0.006595587	180	0.08627	0.008815698
1	0	0	46	0.01392	0.001711152	91	0.03529	0.004333742	136	0.05843	0.006665032	181	0.08176	0.008855937
2	0	0	47	0.01568	0.001817838	92	0.03607	0.004348198	137	0.0598	0.006702535	182	0.08058	0.008893933
3	0	0	48	0.01568	0.001940253	93	0.03921	0.004424474	138	0.0596	0.006770883	183	0.08137	0.008940647
4	0.00156	0	49	0.01568	0.001940253	94	0.03901	0.004433947	139	0.05901	0.006795519	184	0.08196	0.008994174
5	0.00196	0	50	0.01764	0.001942553	95	0.03882	0.004499498	140	0.06058	0.006813807	185	0.0847	0.009049938
6	0.00196	0	51	0.01745	0.002115582	96	0.03862	0.004614234	141	0.06137	0.006881049	186	0.08627	0.009102821
7	0.00196	0	52	0.01764	0.00214504	97	0.03921	0.004618778	142	0.05941	0.006935605	187	0.08568	0.009133234
8	0.00196	0	53	0.01764	0.002164429	98	0.03921	0.004708436	143	0.06019	0.006995581	188	0.08627	0.009192026
9	0.00294	0	54	0.01941	0.002240385	99	0.03921	0.004708436	144	0.06196	0.0070553	189	0.08607	0.009244289
10	0.00372	0	55	0.01928	0.002331911	100	0.03921	0.004796455	145	0.06274	0.007114268	190	0.08647	0.009284969
11	0.00392	0	56	0.02078	0.002331911	101	0.04	0.004826899	146	0.06274	0.007161092	191	0.08666	0.00934775
12	0.00392	0	57	0.02098	0.002428558	102	0.03921	0.004882888	147	0.06333	0.007207613	192	0.08764	0.009396877
13	0.00392	0	58	0.0196	0.00250488	103	0.04117	0.004959401	148	0.06411	0.007242307	193	0.08862	0.009438971
14	0.00392	0	59	0.0196	0.00250488	104	0.04235	0.005009076	149	0.06411	0.007288553	194	0.08803	0.009487627
15	0.00392	0	60	0.0198	0.00250488	105	0.04313	0.005051319	150	0.0647	0.007365467	195	0.08941	0.009544792
16	0.00392	0	61	0.02156	0.00250488	106	0.04274	0.005133462	151	0.06509	0.00742717	196	0.09058	0.00958197
17	0.00568	0	62	0.02333	0.00266653	107	0.04254	0.005133462	152	0.06764	0.007492298	197	0.09254	0.009629903
18	0.00784	0	63	0.02352	0.002789298	108	0.04313	0.005214312	153	0.06764	0.007545025	198	0.09313	0.009673189
19	0.00784	0	64	0.02352	0.002819095	109	0.04333	0.005214312	154	0.06862	0.007583803	199	0.09784	0.009728266
20	0.00784	0	65	0.02352	0.002892366	110	0.04352	0.005292927	155	0.06666	0.007624948	200	0.09411	0.009763555
21	0.00784	0	66	0.02352	0.002963766	111	0.04445	0.00535287	156	0.06705	0.007679775	201	0.09411	0.009819114
22	0.00784	0	67	0.02352	0.002963766	112	0.04509	0.005422722	157	0.06784	0.007695962	202	0.09294	0.009870218
23	0.00784	0	68	0.02352	0.003081398	113	0.04627	0.00549669	158	0.06921	0.007739497	203	0.09372	0.009921454
24	0.00784	0	69	0.02352	0.003101697	114	0.04705	0.005525894	159	0.07019	0.007814703	204	0.09411	0.009965043
25	0.00784	0	70	0.02392	0.003161811	115	0.04764	0.005559859	160	0.07098	0.007846714			
26	0.00784	0	71	0.02549	0.003233749	116	0.04882	0.005608536	161	0.07098	0.007910571			
27	0.00784	0	72	0.02549	0.003233749	117	0.04725	0.005671576	162	0.07235	0.007950025			
28	0.00784	0.000204482	73	0.02549	0.003360617	118	0.0498	0.005719333	163	0.07352	0.007991954			
29	0.0096	0.000646739	74	0.02588	0.003385441	119	0.04921	0.005748508	164	0.07411	0.008049361			
30	0.00921	0.000646739	75	0.02745	0.003452723	120	0.05098	0.005820821	165	0.0749	0.008096882			
31	0.0098	0.000846739	76	0.02745	0.003482867	121	0.05215	0.005888685	166	0.07588	0.008157735			
32	0.0098	0.000646739	77	0.02725	0.003530572	122	0.0549	0.00593116	167	0.07333	0.008188404			
33	0.01	0.000646739	78	0.02784	0.003600969	123	0.05274	0.005962817	168	0.07529	0.00821658			
34	0.0098	0.000891476	79	0.02764	0.003652876	124	0.0549	0.006032473	169	0.0749	0.008274792			
35	0.01058	0.001120216	80	0.03098	0.003715318	125	0.05313	0.006073883	170	0.07882	0.008326888			
36	0.01176	0.001120216	81	0.03117	0.003726546	126	0.05254	0.00610483	171	0.07823	0.008370274			
37	0.01176	0.001120216	82	0.03137	0.003826252	127	0.0549	0.006169427	172	0.07784	0.008430166			
38	0.01176	0.00127724	83	0.03137	0.003875143	128	0.0549	0.006237061	173	0.07666	0.008474755			
39	0.01254	0.001309582	84	0.03137	0.003934059	129	0.0549	0.006303687	174	0.07745	0.008538496			
40	0.01196	0.001446189	85	0.03137	0.004038989	131	0.05529	0.006372965	176	0.07882	0.008558252			
41	0.01294	0.001446189	86	0.03333	0.004090423	132	0.05588	0.006438186	177	0.08098	0.008628796			
42	0.01372	0.001460586	87	0.03333	0.004141261	133	0.05705	0.006499744	178	0.08235	0.008672158			
43	0.01372	0.001446189	88	0.03313	0.004206386	134	0.05686	0.006563705	179	0.08176	0.008775273			

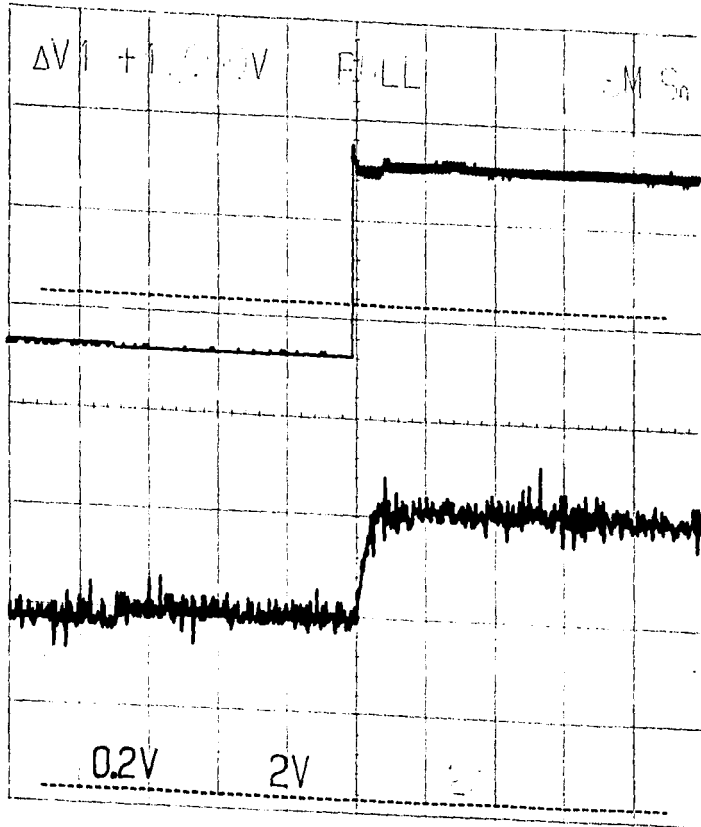
Shiffon, poliéster 100%



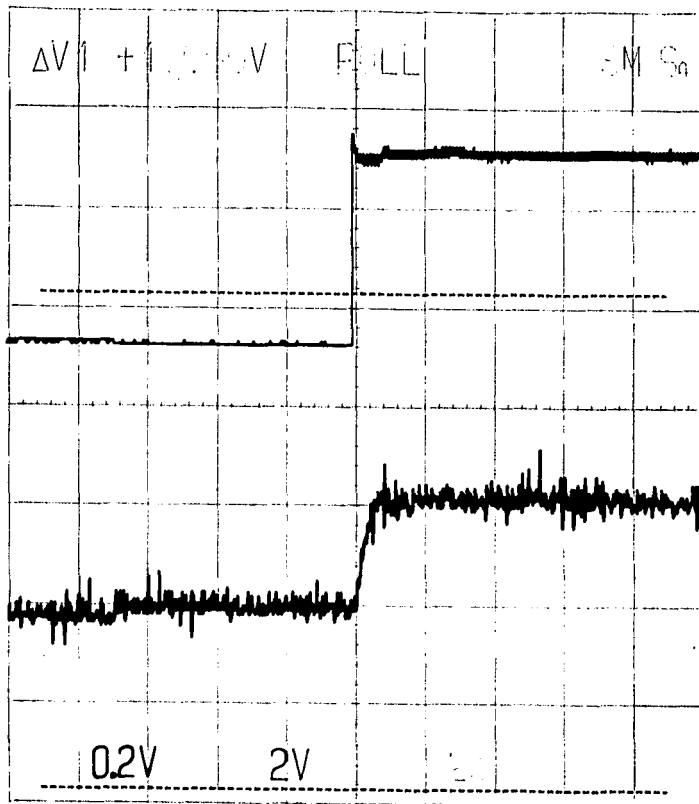
Shiffon, poliéster 100%



Shiffon, poliéster 100%



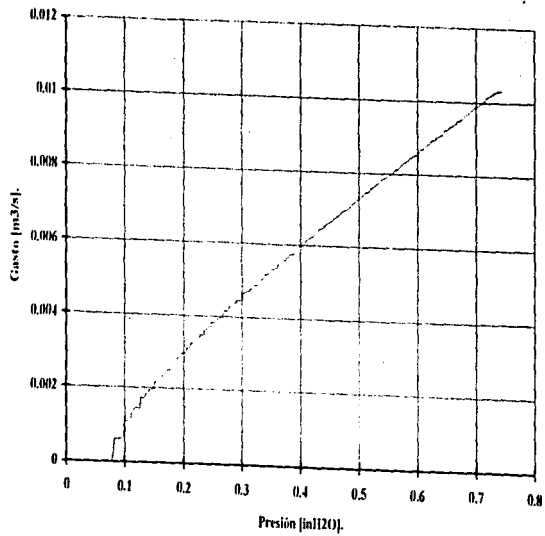
Shiffon, poliéster 100%



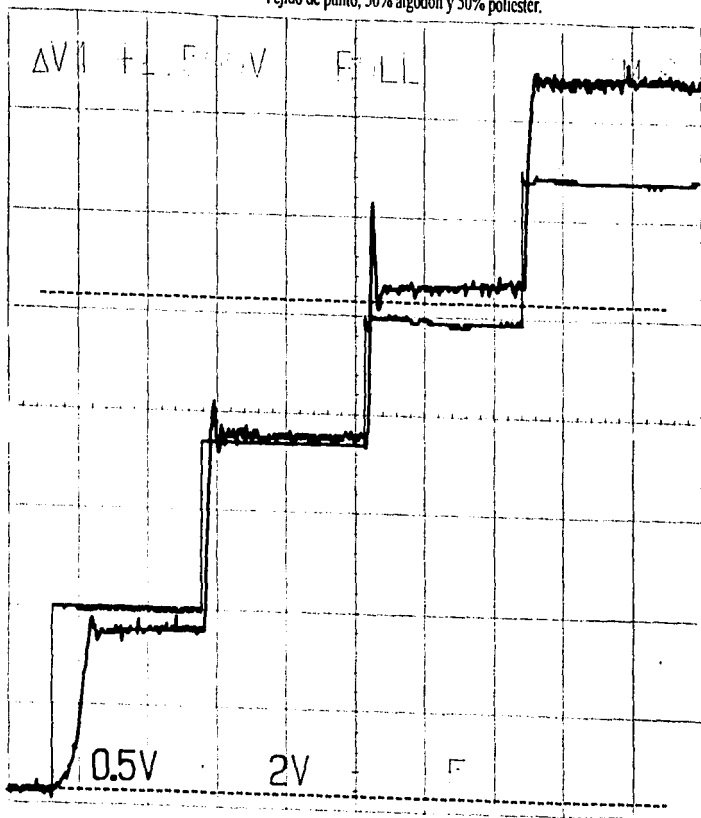
Tejido de punto, negra 50% alg. 50% pol.

	Presion [H2O]	Gasto [m3/s]		Presion [H2O]	Gasto [m3/s]		Presion [H2O]	Gasto [m3/s]		Presion [H2O]	Gasto [m3/s]		Presion [H2O]	Gasto [m3/s]
0	0	0	45	0.12627	0.001711152	90	0.27607	0.004146281	135	0.43921	0.006499744	180	0.61098	0.008705917
1	0.00784	0	46	0.12941	0.001711152	91	0.27843	0.004241068	136	0.44509	0.006579665	181	0.61372	0.008763515
2	0.01568	0	47	0.13254	0.001711152	92	0.28235	0.004265644	137	0.44784	0.0066264096	182	0.61696	0.008837048
3	0.01568	0	48	0.1349	0.001806291	93	0.28549	0.004338579	138	0.45254	0.006649276	183	0.62156	0.008881996
4	0.01568	0	49	0.14117	0.001885608	94	0.29019	0.004386502	139	0.45451	0.006693245	184	0.62666	0.008938261
5	0.01921	0	50	0.14117	0.001940253	95	0.29019	0.004433947	140	0.45843	0.006755374	185	0.63137	0.008973396
6	0.02235	0	51	0.14509	0.001940253	96	0.29803	0.004433947	141	0.46117	0.006798657	186	0.63137	0.009000296
7	0.02352	0	52	0.14902	0.00198285	97	0.30196	0.004527306	142	0.4647	0.006832306	187	0.63607	0.009052194
8	0.02745	0	53	0.15137	0.00214504	98	0.30196	0.004614214	143	0.46823	0.006896275	188	0.64274	0.009116677
9	0.02745	0	54	0.15294	0.00214504	99	0.30588	0.004659322	144	0.47451	0.006908072	189	0.64447	0.009162388
10	0.03137	0	55	0.15686	0.002231082	100	0.31294	0.004708436	145	0.47843	0.007010559	190	0.64745	0.009210188
11	0.03294	0	56	0.16078	0.002331911	101	0.31372	0.004752668	146	0.48274	0.0070553	191	0.65098	0.009248711
12	0.03529	0	57	0.16431	0.002331911	102	0.31764	0.004796455	147	0.48627	0.007114268	192	0.6549	0.009287075
13	0.03921	0	58	0.16666	0.002340814	103	0.32156	0.004857116	148	0.49019	0.007172751	193	0.6549	0.009309636
14	0.0396	0	59	0.17254	0.002494991	104	0.32745	0.004891473	149	0.49411	0.007230761	194	0.66156	0.009370164
15	0.04666	0	60	0.17254	0.00250488	105	0.32941	0.004967817	150	0.49803	0.007285625	195	0.66549	0.009408033
16	0.04705	0	61	0.17647	0.00250488	106	0.33294	0.005018092	151	0.50196	0.007322864	196	0.66745	0.009465678
17	0.04745	0	62	0.18039	0.00250488	107	0.3349	0.005051319	152	0.50509	0.007371257	197	0.67411	0.009515625
18	0.05058	0	63	0.18313	0.002573097	108	0.33803	0.005133462	153	0.5098	0.007444691	198	0.67843	0.009573246
19	0.05496	0	64	0.18823	0.002731245	109	0.3447	0.005178081	154	0.51176	0.0074696	199	0.6796	0.009614753
20	0.06078	0	65	0.18862	0.002819095	110	0.34509	0.005214312	155	0.51764	0.007511729	200	0.68274	0.009660315
21	0.06235	0	66	0.19254	0.002826526	111	0.34902	0.005274144	156	0.52078	0.007564556	201	0.68941	0.009728701
22	0.06274	0	67	0.19607	0.002935438	112	0.35529	0.005325433	157	0.52274	0.00760861	202	0.69098	0.009767924
23	0.06274	0	68	0.20078	0.002963766	113	0.36078	0.005372362	158	0.52666	0.007652411	203	0.69451	0.009808426
24	0.06705	0	69	0.20392	0.002977888	114	0.36117	0.005438141	159	0.5298	0.007706811	204	0.69843	0.009855257
25	0.07058	0	70	0.2047	0.003101697	115	0.3647	0.005449669	160	0.53333	0.007733983	205	0.70235	0.009895582
26	0.07098	0	71	0.2098	0.003101697	116	0.36862	0.005525894	161	0.53725	0.00780127	206	0.70666	0.009946114
27	0.07843	0	72	0.21254	0.00323749	117	0.3749	0.005574867	162	0.54156	0.007857355	207	0.71098	0.010004747
28	0.07843	0	73	0.21568	0.003292799	118	0.37647	0.005636087	163	0.54745	0.00789776	208	0.71372	0.010031897
29	0.07882	0	74	0.2196	0.003292799	119	0.38039	0.005675274	164	0.55058	0.007950025	209	0.71607	0.010081923
30	0.08235	0.000613562	75	0.22352	0.003360617	120	0.38431	0.005748508	165	0.55647	0.008005068	210	0.72549	0.010193278
31	0.08627	0.000646739	76	0.22745	0.003428386	121	0.39019	0.005784808	166	0.55764	0.008052011	211	0.73058	0.010252578
32	0.08991	0.000646739	77	0.22784	0.003482867	122	0.39215	0.005820821	167	0.56156	0.008106361	212	0.73647	0.010311741
33	0.09019	0.000646739	78	0.23529	0.003482867	123	0.39568	0.005892247	168	0.56666	0.008155119	213	0.74588	0.010327902
34	0.09411	0.000646739	79	0.23529	0.003600969	124	0.39725	0.005955777	169	0.56941	0.008206405	214	0.73568	0.010325836
35	0.09764	0.000708472	80	0.24039	0.003658615	125	0.40274	0.005996067	170	0.57254	0.008257156			
36	0.0996	0.001101379	81	0.24313	0.003709668	126	0.40666	0.006032473	171	0.57686	0.008305243			
37	0.10196	0.001120216	82	0.24705	0.003715318	127	0.41019	0.006101334	172	0.58039	0.008345388			
38	0.10823	0.001120216	83	0.24705	0.003826252	128	0.41568	0.006169427	173	0.58392	0.00838534			
39	0.1098	0.001138726	84	0.2549	0.003826252	129	0.4196	0.006220221	174	0.58549	0.008435015			
40	0.11019	0.0012774	85	0.25843	0.003934059	130	0.42352	0.006253855	175	0.58823	0.00846237			
41	0.11411	0.001387134	86	0.26	0.003934059	131	0.42666	0.006310452	176	0.59254	0.008514101			
42	0.11647	0.001446189	87	0.26666	0.004018204	132	0.4298	0.006369617	177	0.59843	0.008555759			
43	0.12235	0.001446189	88	0.26666	0.004038989	133	0.43137	0.00642215	178	0.60196	0.008603062			
44	0.12549	0.001446189	89	0.27058	0.004131161	134	0.43882	0.006470687	179	0.60509	0.008660426			

Tejido de punto, 50% algodón 50% poliéster.



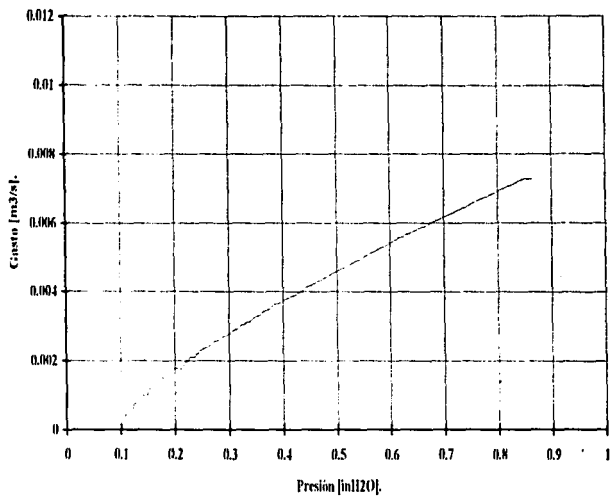
Tejido de punto, 50% algodón y 50% poliéster.



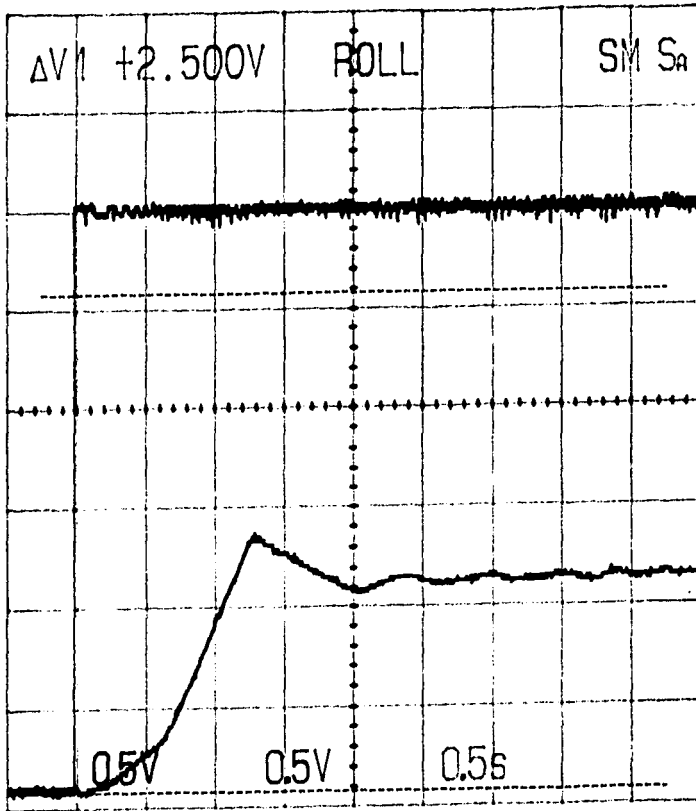
Azúlan 100%

	Presión [m(t20)]	Gasto [m3/s]		Presión [m(t20)]	Gasto [m3/s]		Presión [m(t20)]	Gasto [m3/s]		Presión [m(t20)]	Gasto [m3/s]
0	0.00039	0	45	0.20627	0.88235	90	0.46666	5.2352	133	0.75294	12.211
1	0.01568	0	46	0.21215	0.88235	91	0.47058	5.2588	136	0.76352	12.447
2	0.01568	0	47	0.21803	1.0588	92	0.47843	5.4705	137	0.77058	12.658
3	0.02	0	48	0.22352	1.1235	93	0.48627	5.5882	138	0.77647	12.805
4	0.02352	0	49	0.22745	1.2352	94	0.48745	5.6941	139	0.78274	12.988
5	0.02745	0	50	0.23529	1.2352	95	0.49411	5.8235	140	0.78862	13.182
6	0.03137	0	51	0.23921	1.3529	96	0.50235	5.9529	141	0.79843	13.447
7	0.03529	0	52	0.24392	1.4058	97	0.50745	6.1176	142	0.80313	13.617
8	0.0396	0	53	0.24705	1.4705	98	0.51725	6.2823	143	0.80902	13.794
9	0.04705	0	54	0.2549	1.5705	99	0.52196	6.4	144	0.81647	14.023
10	0.04705	0	55	0.25882	1.6	100	0.52627	6.5294	145	0.82431	14.188
11	0.05098	0	56	0.26666	1.7058	101	0.53333	6.6529	146	0.83098	14.423
12	0.0549	0	57	0.27058	1.8235	102	0.54117	6.8235	147	0.83764	14.635
13	0.06274	0	58	0.2749	1.8705	103	0.54666	6.9235	148	0.84313	14.823
14	0.06274	0	59	0.28235	1.9411	104	0.55333	7.1176	149	0.85058	14.994
15	0.06666	0	60	0.28745	2.0529	105	0.56039	7.2647	150	0.85725	15
16	0.07058	0	61	0.29411	2.1176	106	0.56509	7.3882	151	0.86509	15
17	0.07843	0	62	0.29803	2.1764	107	0.57254	7.5235			
18	0.0796	0	63	0.30549	2.2882	108	0.58039	7.7058			
19	0.08627	0	64	0.31372	2.4	109	0.58431	7.8176			
20	0.0898	0	65	0.31647	2.4294	110	0.59137	7.9588			
21	0.09294	0	66	0.32156	2.5647	111	0.59686	8.1117			
22	0.09803	0	67	0.32941	2.647	112	0.60392	8.2941			
23	0.10196	0	68	0.33333	2.7647	113	0.61176	8.4117			
24	0.1098	0.05882	69	0.33725	2.7647	114	0.61647	8.6			
25	0.11019	0.05882	70	0.34509	2.9411	115	0.62588	8.7705			
26	0.11764	0.05882	71	0.34902	3.0176	116	0.62784	8.9058			
27	0.12156	0.09411	72	0.35843	3.1235	117	0.63254	9			
28	0.12549	0.1647	73	0.36117	3.2352	118	0.64313	9.2294			
29	0.12941	0.17647	74	0.36662	3.3529	119	0.64902	9.3882			
30	0.13333	0.22352	75	0.37647	3.4705	120	0.65725	9.5882			
31	0.14117	0.29411	76	0.38039	3.5823	121	0.66274	9.7058			
32	0.14156	0.29411	77	0.38431	3.6941	122	0.67058	9.8941			
33	0.14902	0.29411	78	0.39215	3.7647	123	0.67686	10.058			
34	0.15294	0.41176	79	0.39607	3.8294	124	0.68196	10.241			
35	0.15686	0.41176	80	0.40392	4.0117	125	0.6898	10.411			
36	0.16196	0.47647	81	0.41098	4.1	126	0.69607	10.588			
37	0.16902	0.52941	82	0.41568	4.1941	127	0.70196	10.752			
38	0.17254	0.52941	83	0.42352	4.2941	128	0.71058	10.964			
39	0.647	0.64117	84	0.42745	4.447	129	0.71607	11.135			
40	0.8392	0.64705	85	0.43254	4.5352	130	0.72313	11.311			
41	0.18823	0.75882	86	0.4396	4.647	131	0.72941	11.47			
42	0.19607	0.82941	87	0.44509	4.7882	132	0.73725	11.641			
43	0.19803	0.8647	88	0.4549	4.9529	133	0.74313	11.815			
44	0.20392	0.88235	89	0.45882	5.0352	134	0.74902	12.029			

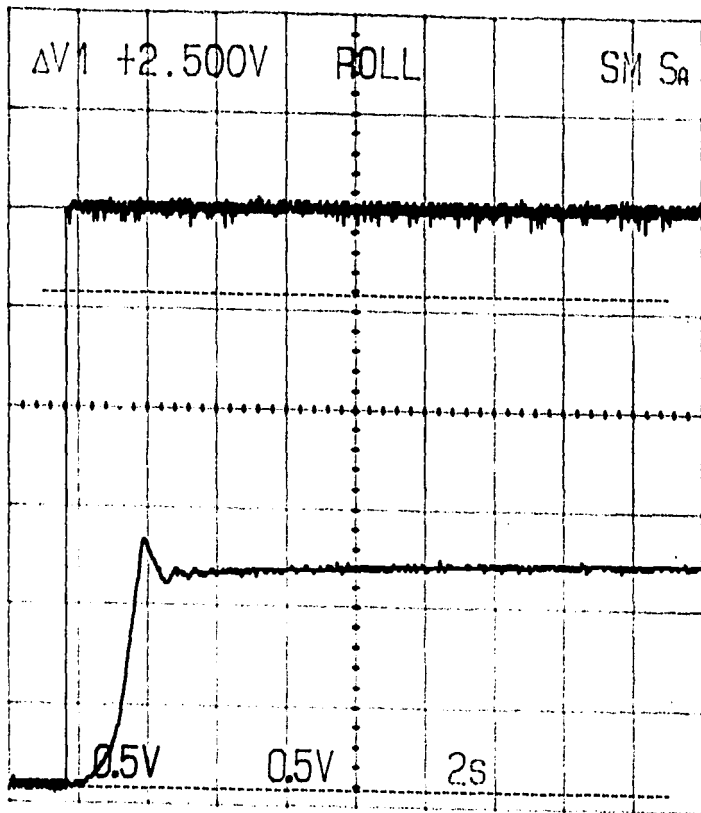
Acrilán 100%



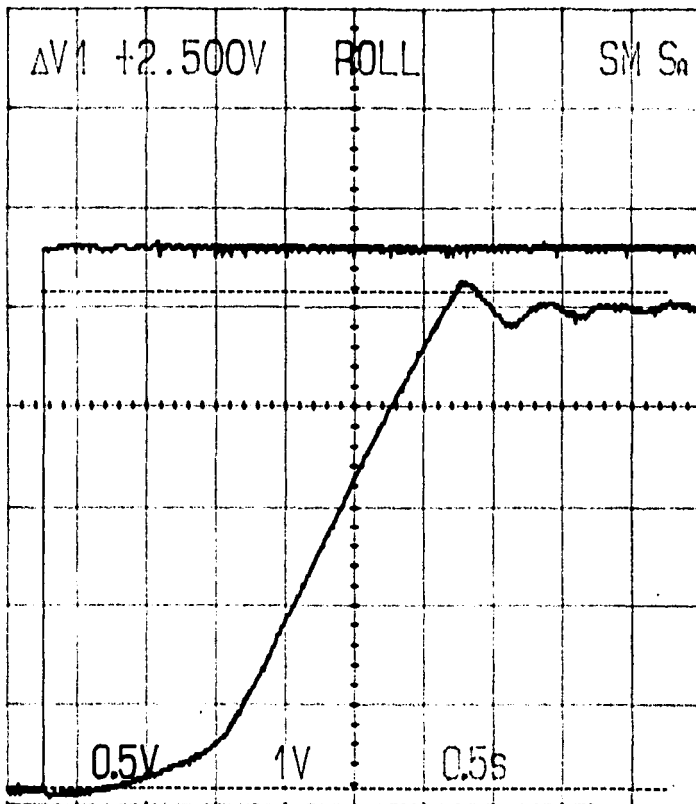
Acrilan 100%



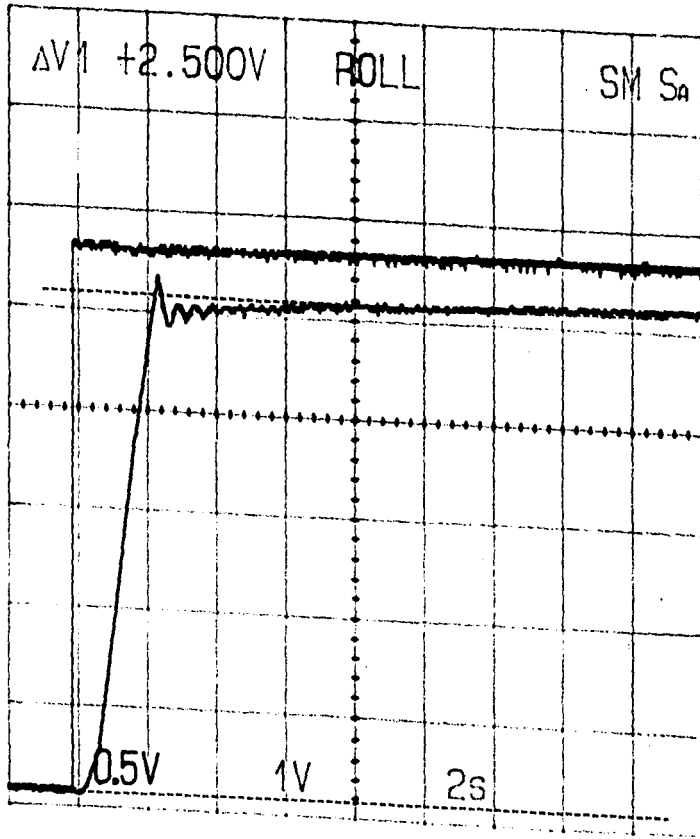
Acilán 100%



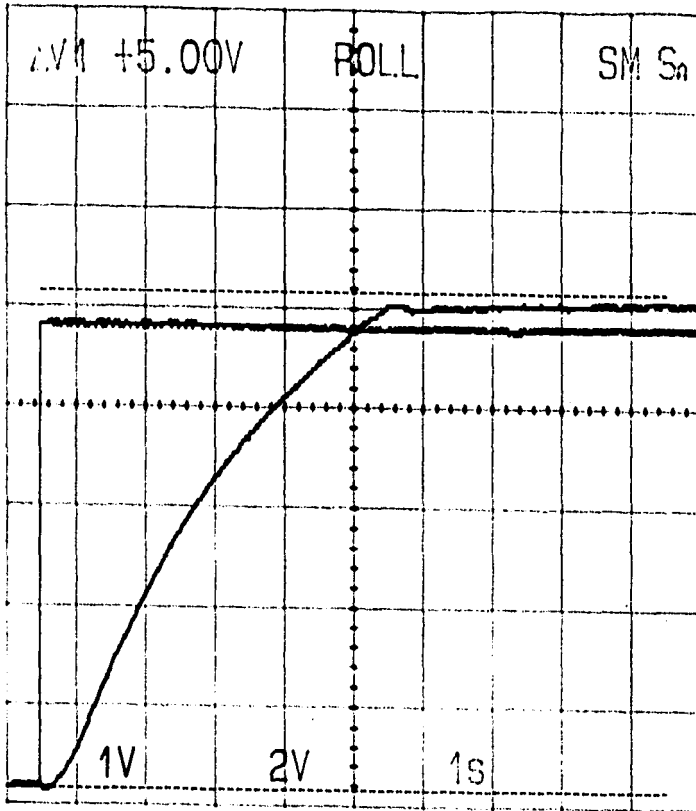
Acrlán 100%



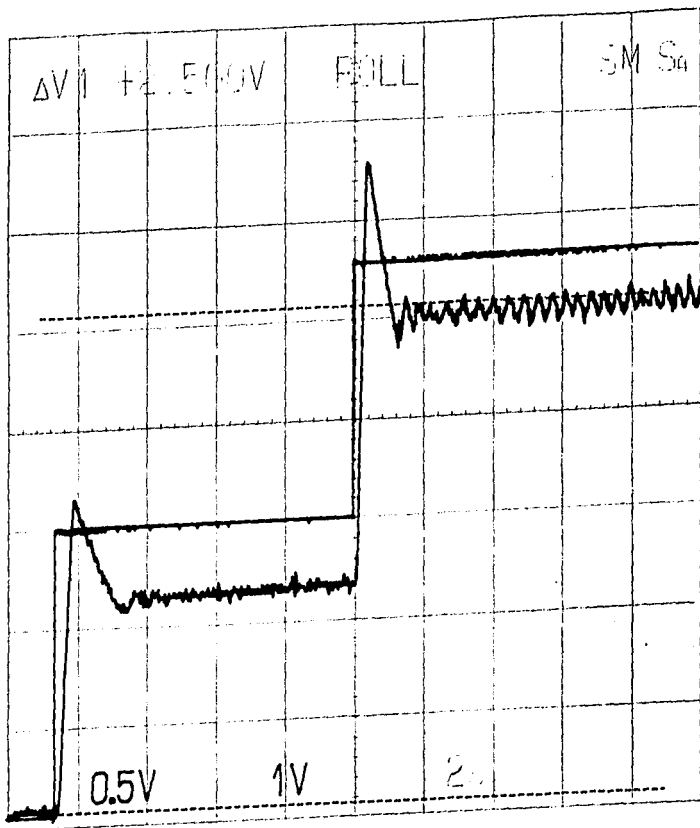
Acritán 100%



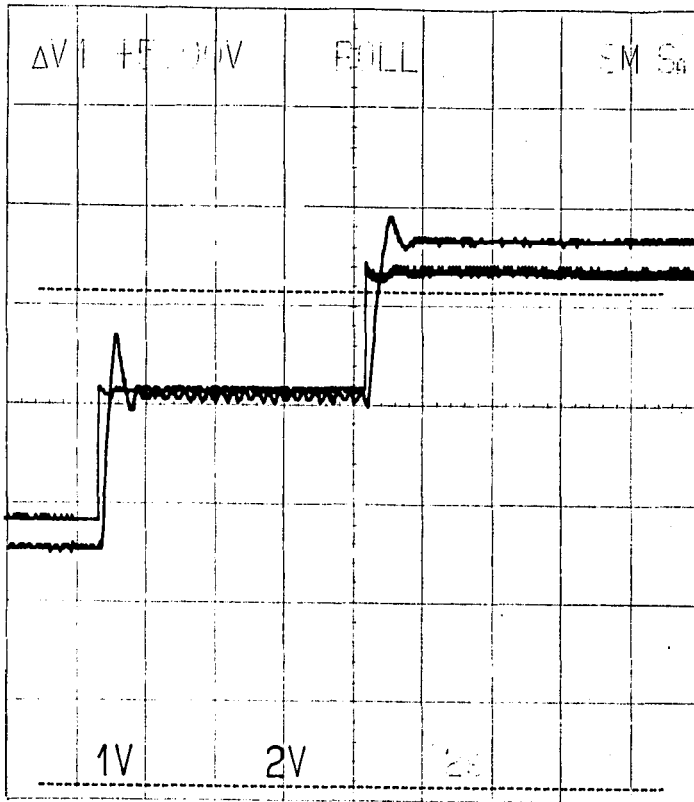
Acrilan 100%



Acrlán 100%



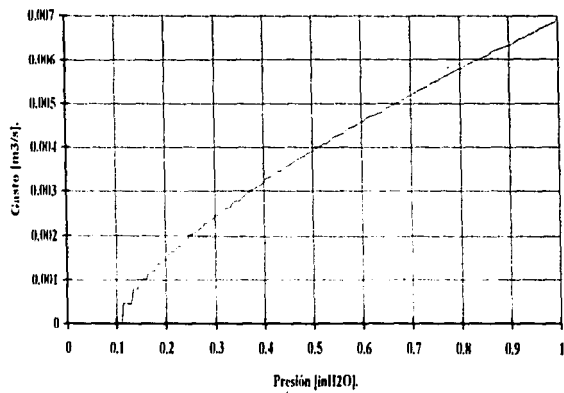
Acrilán 100%



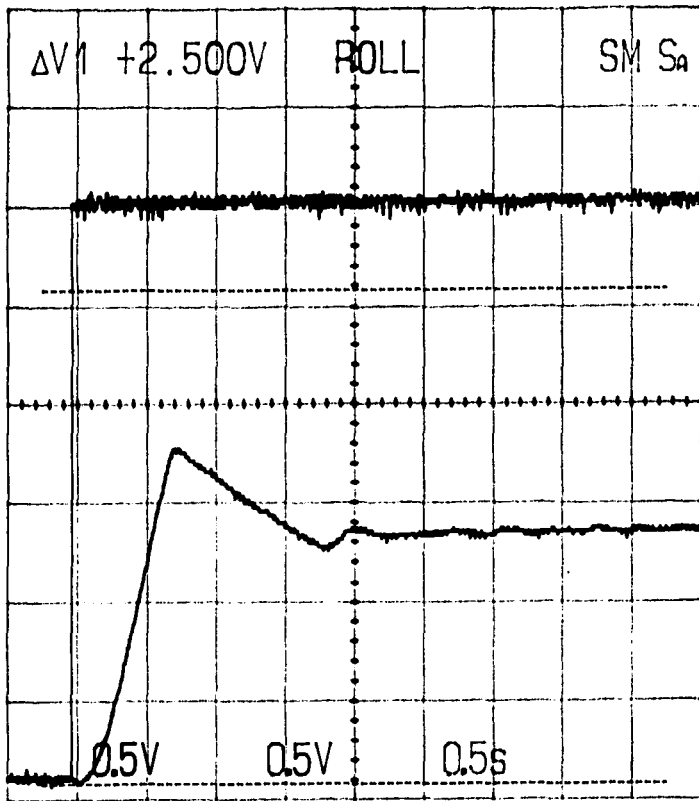
Rebozo, rayón 100%

	Presión [mH2O]	Gasto [m3/s]		Presión [mH2O]	Gasto [m3/s]		Presión [mH2O]	Gasto [m3/s]		Presión [mH2O]	Gasto [m3/s]
0	0.0039	0	45	0.22902	0.001771218	90	0.55411	0.004270559	135	0.94	0.006568378
1	0.0149	0	46	0.23686	0.001771218	91	0.56274	0.004319237	136	0.95098	0.006630266
2	0.01568	0	47	0.24313	0.001885608	92	0.56862	0.004362066	137	0.96313	0.006697759
3	0.0196	0	48	0.24705	0.001993401	93	0.5749	0.004419771	138	0.97137	0.006736664
4	0.02352	0	49	0.25568	0.002040033	94	0.58627	0.004471516	139	0.97607	0.006767731
5	0.02745	0	50	0.26235	0.002095567	95	0.59098	0.004513423	140	0.99019	0.006842966
6	0.03137	0	51	0.26666	0.002140145	96	0.60117	0.004582377	141	0.99647	0.006876656
7	0.03568	0	52	0.27451	0.002193231	97	0.60862	0.004632307	142	1	0.006937142
8	0.0396	0	53	0.27843	0.002277453	98	0.61568	0.004668182			
9	0.04705	0	54	0.28627	0.002286567	99	0.62588	0.004730622			
10	0.04705	0	55	0.29529	0.002376315	100	0.63372	0.004779003			
11	0.05294	0	56	0.30196	0.0024457	101	0.64235	0.004818236			
12	0.05529	0	57	0.30862	0.002466978	102	0.64784	0.00489258			
13	0.06274	0	58	0.31372	0.002546269	103	0.65568	0.004940402			
14	0.06392	0	59	0.32039	0.002607122	104	0.66666	0.004988851			
15	0.07058	0	60	0.32941	0.002631082	105	0.6749	0.005034751			
16	0.07843	0	61	0.33333	0.002705569	106	0.68588	0.005113059			
17	0.07843	0	62	0.34117	0.002781768	107	0.69137	0.005149748			
18	0.08274	0	63	0.34902	0.002793056	108	0.70352	0.005208308			
19	0.08901	0	64	0.35764	0.002881466	109	0.7098	0.005248295			
20	0.09058	0	65	0.36392	0.002928283	110	0.71803	0.005299868			
21	0.09803	0	66	0.37098	0.002995388	111	0.72705	0.00535991			
22	0.10196	0	67	0.38039	0.003064389	112	0.73725	0.005411137			
23	0.1098	0	68	0.38431	0.003088198	113	0.74352	0.005463092			
24	0.11372	0.000457313	69	0.39215	0.003135274	114	0.75451	0.005514526			
25	0.11803	0.000457313	70	0.3996	0.003194701	115	0.76196	0.005559859			
26	0.12549	0.000457313	71	0.40392	0.003265969	116	0.76902	0.005602954			
27	0.12941	0.000457313	72	0.41529	0.003329367	117	0.77843	0.005656825			
28	0.13333	0.000792113	73	0.42352	0.003385441	118	0.78823	0.005715664			
29	0.14117	0.000792113	74	0.42745	0.003431444	119	0.79607	0.005763025			
30	0.14509	0.000879672	75	0.43529	0.003452723	120	0.80823	0.005811835			
31	0.14902	0.000926005	76	0.44313	0.003524675	121	0.81568	0.005872694			
32	0.15411	0.001022601	77	0.45215	0.003571822	122	0.82392	0.005909961			
33	0.16078	0.001022601	78	0.45882	0.003629882	123	0.83411	0.005973243			
34	0.1647	0.001174887	79	0.46666	0.003687075	124	0.84352	0.00602215			
35	0.17254	0.001209967	80	0.47647	0.003757284	125	0.84862	0.006053067			
36	0.17647	0.001227124	81	0.48509	0.003798834	126	0.86156	0.006111524			
37	0.18352	0.001371979	82	0.49058	0.003853475	127	0.86313	0.006152691			
38	0.18823	0.001371979	83	0.4996	0.003907397	128	0.87607	0.00622365			
39	0.19215	0.001402137	84	0.50588	0.00396054	129	0.88627	0.006277122			
40	0.20392	0.001516772	85	0.5149	0.004013024	130	0.8949	0.006320305			
41	0.2047	0.001584225	86	0.52156	0.004064787	131	0.90509	0.006366546			
42	0.21176	0.00164891	87	0.52941	0.004090423	132	0.91451	0.006418828			
43	0.21803	0.001655238	88	0.53725	0.004166426	133	0.92235	0.006472727			
44	0.22352	0.001771218	89	0.54235	0.00421137	134	0.93137	0.0065185861			

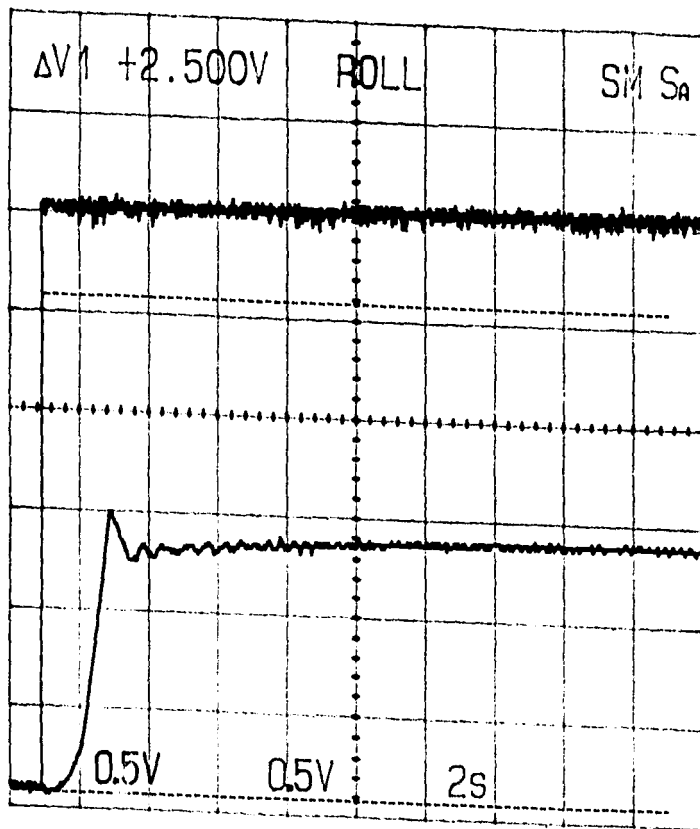
Rebozo, rayón 100%



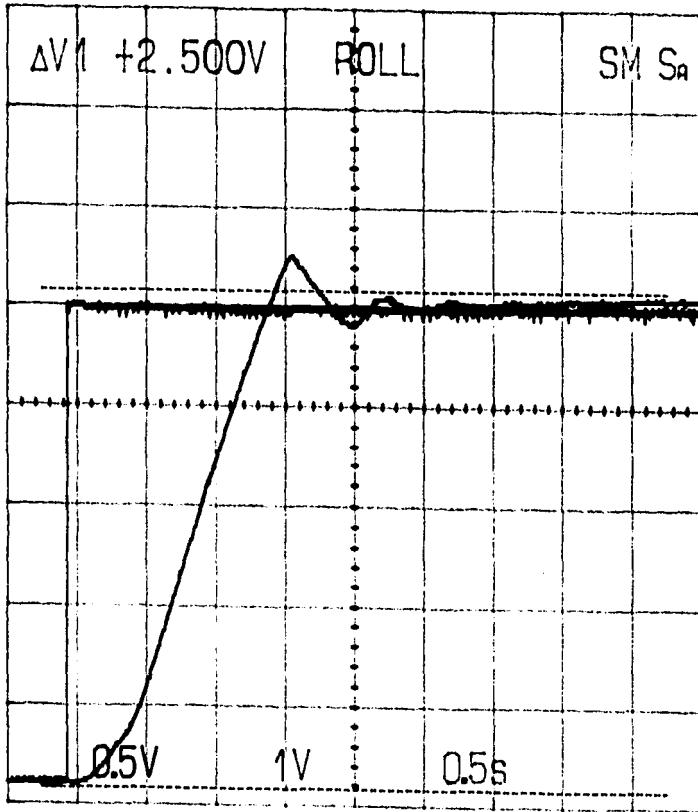
Rebozo, rayón 100%



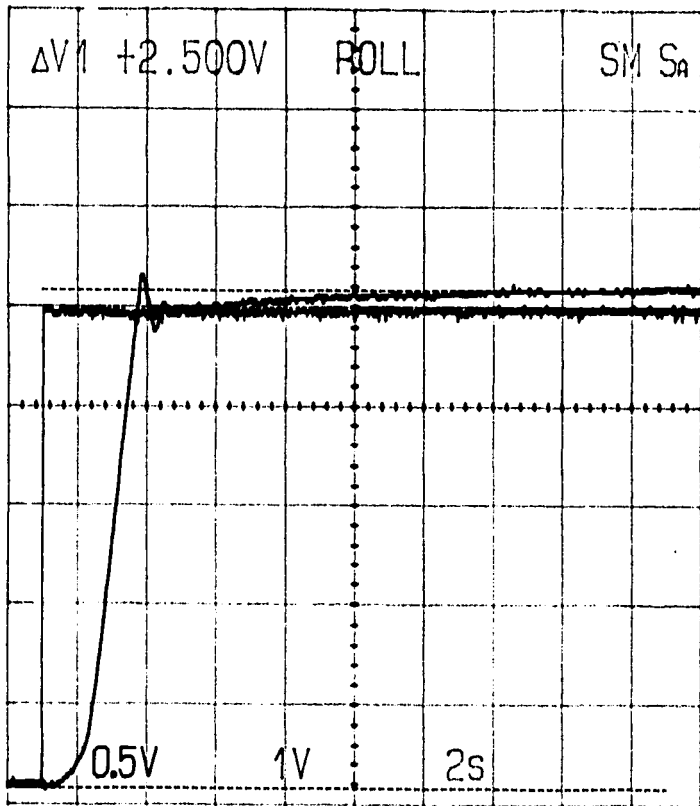
Rebozo, rayón 100%.



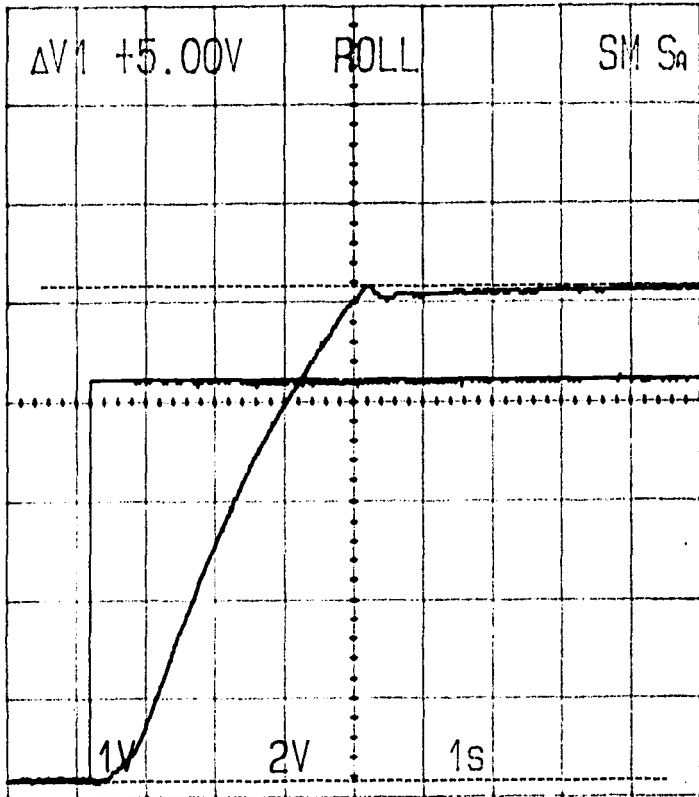
Rebozo, rayón 100%.



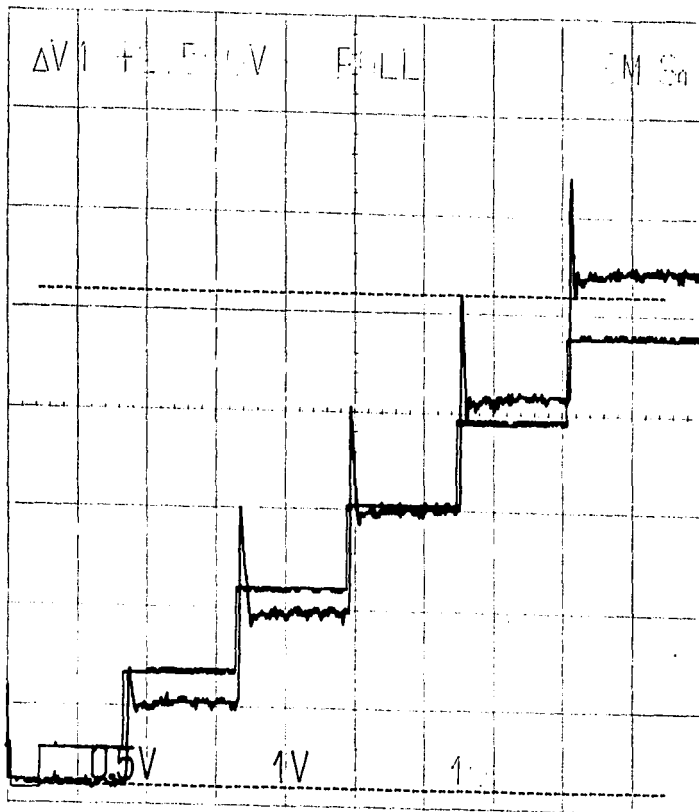
Rcbozo, rayón 100%.



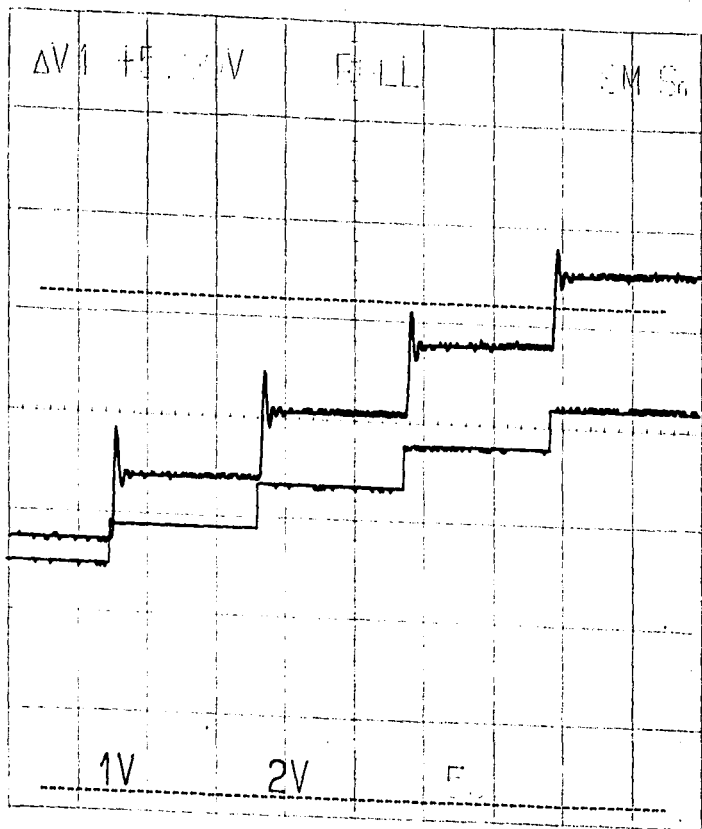
Rcbozo, rayón 100%.



Rebozo, rayón 100%.



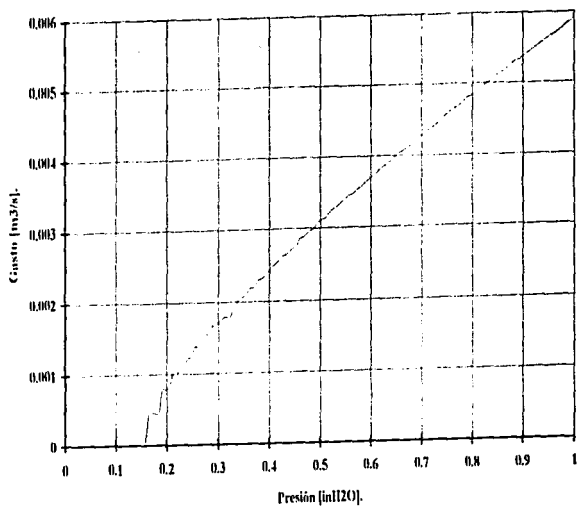
Rebozo, rayón 100%.



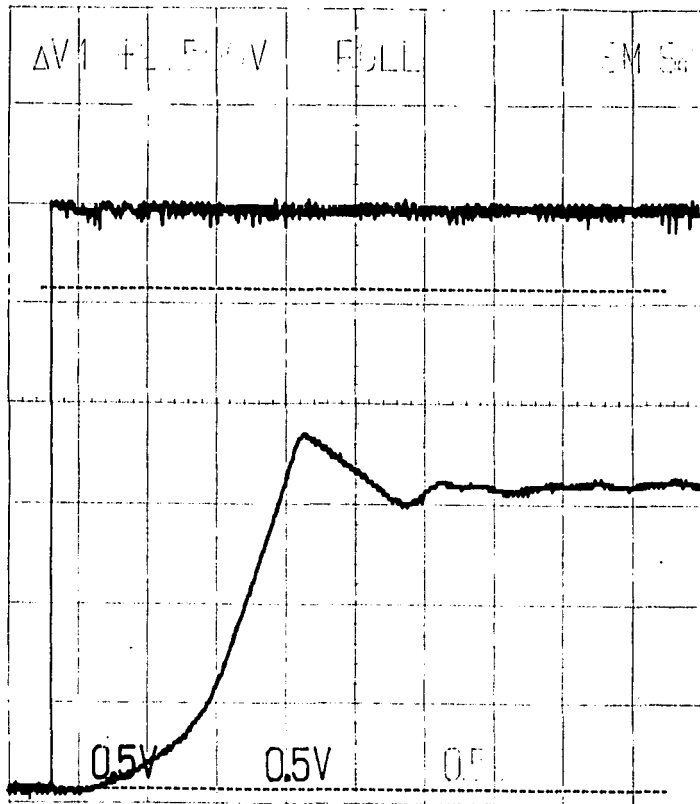
Tejido de punto alfelpado, 50 %acrílico y 50% poliéster.

	Presion [mil20]	Gasto [m3/s]		Presion [mil20]	Gasto [m3/s]		Presion [mil20]	Gasto [m3/s]
0	0.00078	0	45	0.30902	0.001771218	90	0.69019	0.004216348
1	0.01568	0	46	0.31529	0.001771218	91	0.69686	0.004265644
2	0.02235	0	47	0.32156	0.001771218	92	0.70588	0.004314377
3	0.02745	0	48	0.33019	0.001918508	93	0.71607	0.004376968
4	0.03215	0	49	0.33725	0.001993401	94	0.72549	0.004431581
5	0.03921	0	50	0.34549	0.002060496	95	0.73451	0.00448685
6	0.04705	0	51	0.35294	0.002095657	96	0.74509	0.004550336
7	0.0498	0	52	0.36078	0.002188443	97	0.75294	0.004596052
8	0.0549	0	53	0.36862	0.002193231	98	0.76156	0.004641355
9	0.06274	0	54	0.37647	0.002286567	99	0.77254	0.004686182
10	0.06784	0	55	0.38549	0.00231385	100	0.78078	0.004741658
11	0.07764	0	56	0.39372	0.002376315	101	0.79215	0.004803011
12	0.07843	0	57	0.40196	0.002462723	102	0.8	0.004822551
13	0.08431	0	58	0.4098	0.002508993	103	0.80823	0.004904285
14	0.09019	0	59	0.41607	0.002546269	104	0.81843	0.004946731
15	0.09803	0	60	0.42745	0.002627093	105	0.82705	0.004993018
16	0.1047	0	61	0.43529	0.002650873	106	0.83843	0.005049242
17	0.11058	0	62	0.44352	0.002713245	107	0.84705	0.005102827
18	0.11803	0	63	0.4549	0.002781768	108	0.85882	0.005159887
19	0.12509	0	64	0.45882	0.002855996	109	0.86627	0.005196245
20	0.12823	0	65	0.46784	0.002885104	110	0.87451	0.005258244
21	0.13647	0	66	0.4749	0.002928283	111	0.88274	0.005301813
22	0.14117	0	67	0.48627	0.002998888	112	0.89529	0.005356754
23	0.14902	0	68	0.4949	0.00306781	113	0.90509	0.005411337
24	0.15686	0	69	0.50235	0.003108453	114	0.91411	0.005461172
25	0.1647	0.000457313	70	0.5098	0.003135274	115	0.92588	0.005516427
26	0.17254	0.000457313	71	0.51764	0.003201261	116	0.93764	0.005576748
27	0.17647	0.000457313	72	0.52823	0.003265969	117	0.94352	0.005612275
28	0.18392	0.000457313	73	0.53725	0.003329367	118	0.95411	0.005662353
29	0.19058	0.000778781	74	0.54627	0.003385441	119	0.96313	0.005706512
30	0.19647	0.000792113	75	0.55411	0.003437552	120	0.97607	0.005773904
31	0.20392	0.00081808	76	0.56392	0.003467828	121	0.98431	0.005820821
32	0.21176	0.001022601	77	0.57254	0.003524675	122	0.99529	0.005874449
33	0.21843	0.001022601	78	0.58039	0.003571822	123	1	0.005911735
34	0.2247	0.001022601	79	0.58823	0.003629882			
35	0.23529	0.00118726	80	0.59843	0.003687075			
36	0.23921	0.001209967	81	0.60823	0.003743348			
37	0.24705	0.001235627	82	0.61568	0.003798834			
38	0.25411	0.001371979	83	0.62549	0.003853475			
39	0.26039	0.001371979	84	0.6349	0.003907397			
40	0.2698	0.001495954	85	0.64313	0.00396054			
41	0.2749	0.001516772	86	0.65098	0.004013024			
42	0.28902	0.001636182	87	0.66078	0.004064787			
43	0.28705	0.00164891	88	0.67294	0.004115942			
44	0.3	0.001692726	89	0.67764	0.004163908			

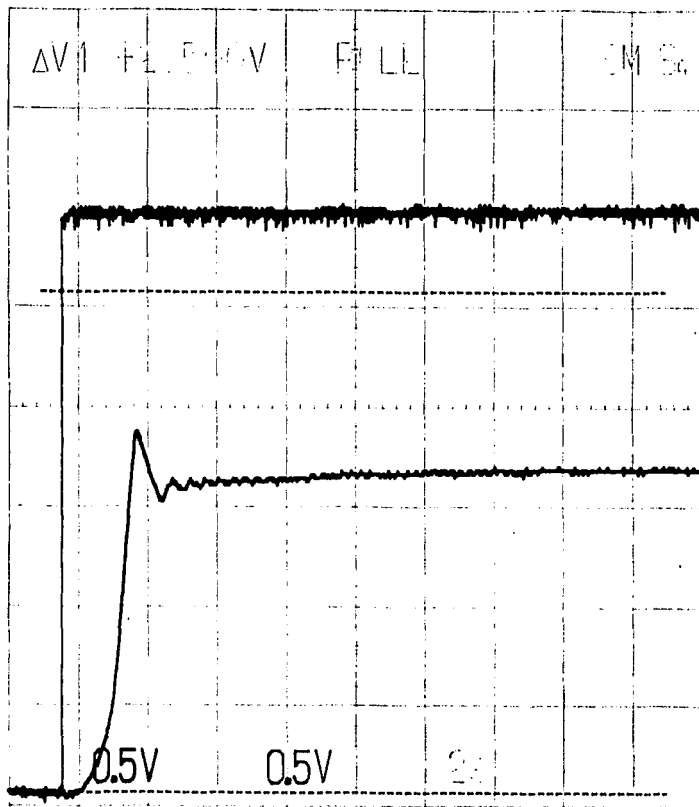
Tejido de punto afelpado, 50 %acrílico y 50% poliéster.



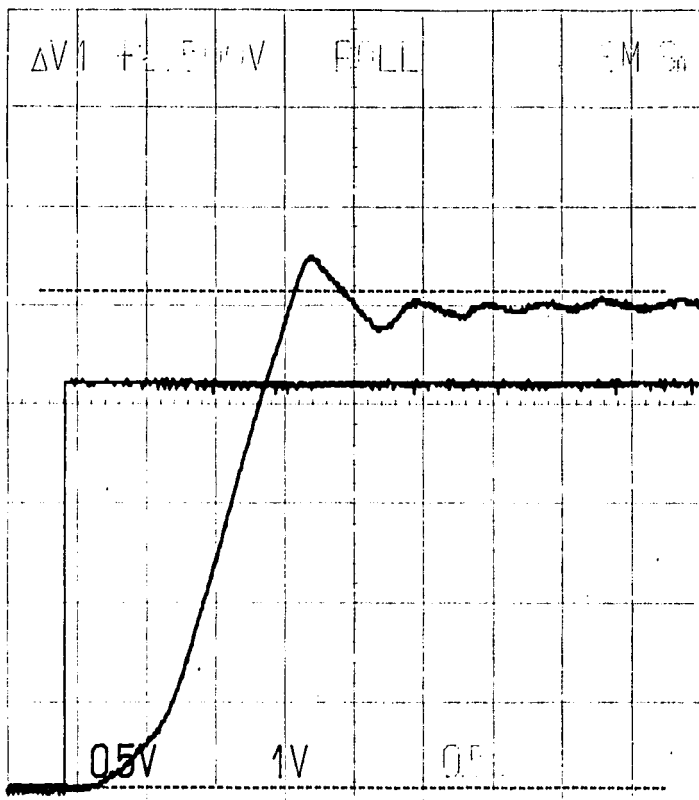
Tejido de punto afelpado, 50% acrílico y 50% poliéster.



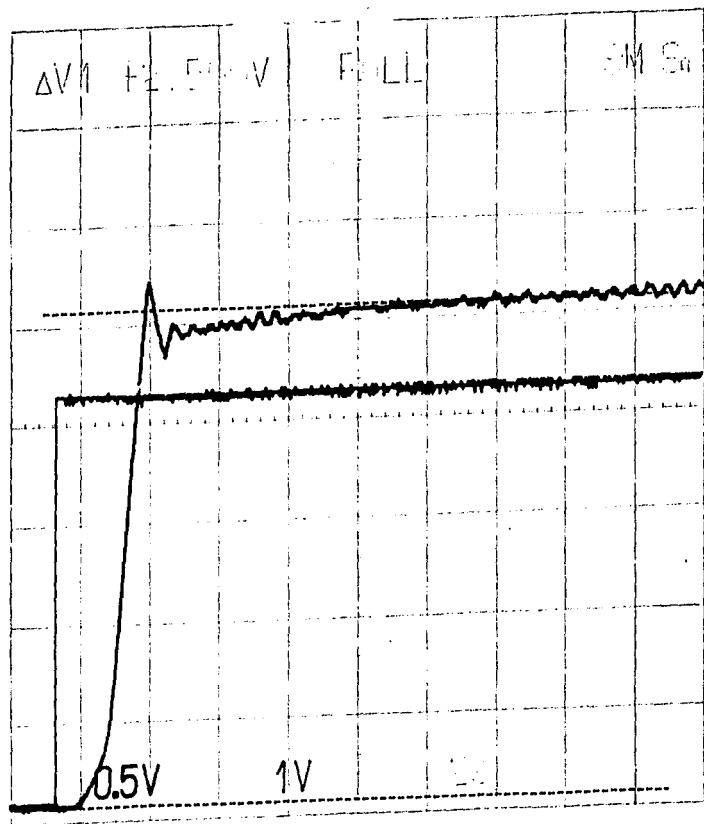
Tejido de punto afelpado, 50% acrílico y 50% poliéster.



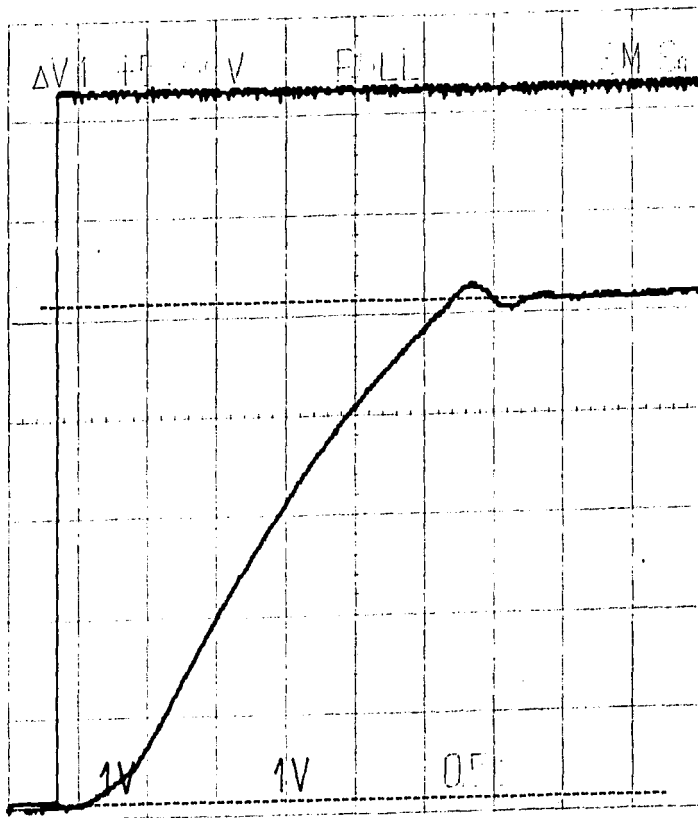
Tejido de punto alécpado, 50% acrílico y 50% poliéster.



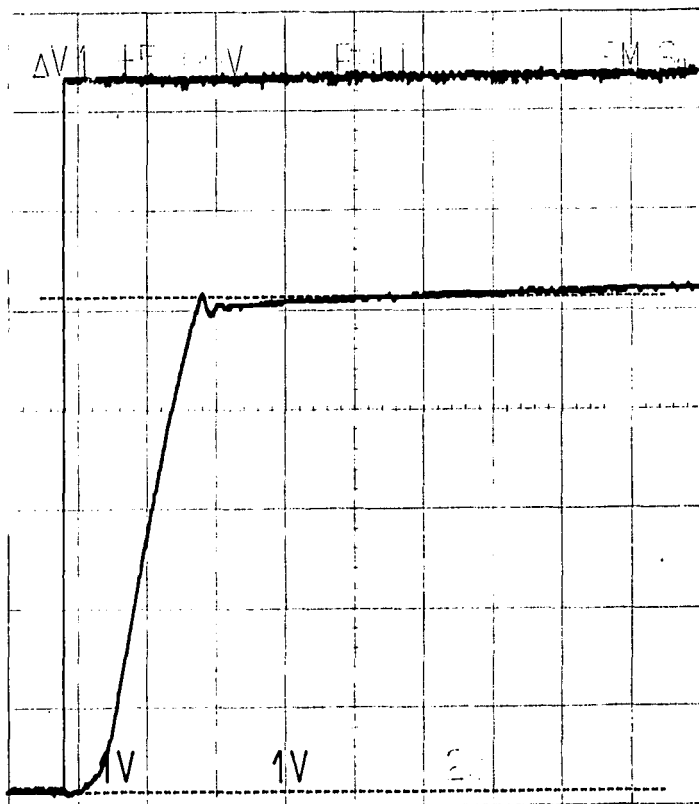
Tejido de punto afelpado, 50% acrílico y 50% poliéster.



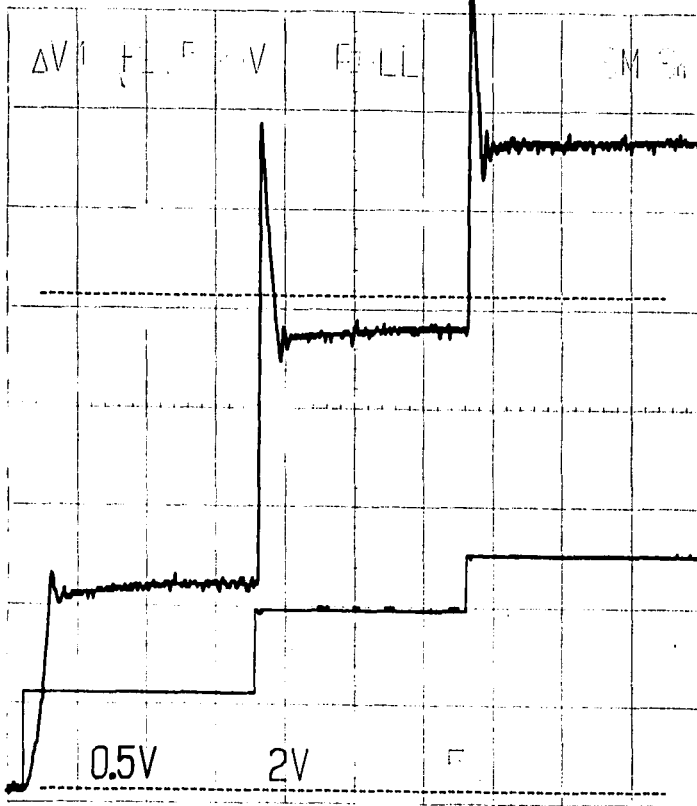
Tejido de punto afelpado, 50% acrílico y 50% poliéster.



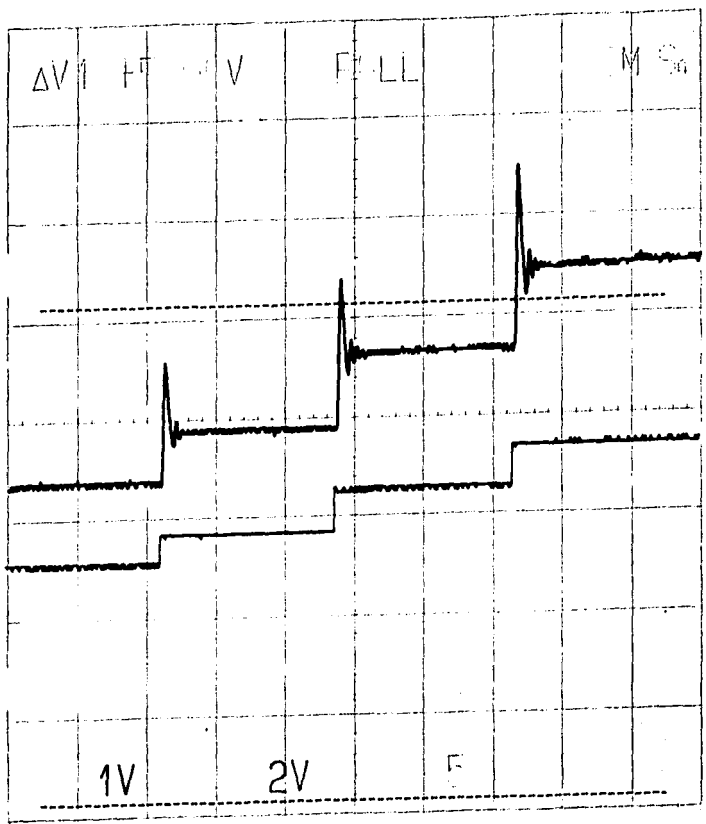
Tejido de punto afelpado, 50% acrílico y 50% poliéster.



Tejido de punto afelpado, 50% acrílico y 50% poliéster.

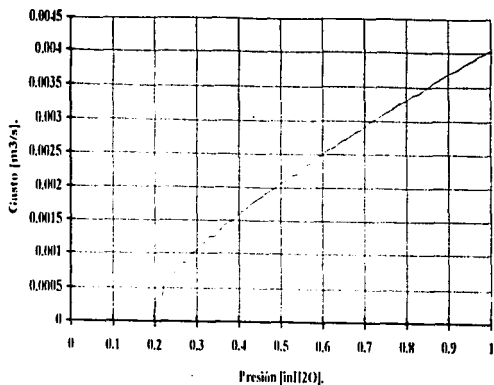


Tejido de punto afelpado, 50% acrílico y 50% poliéster.

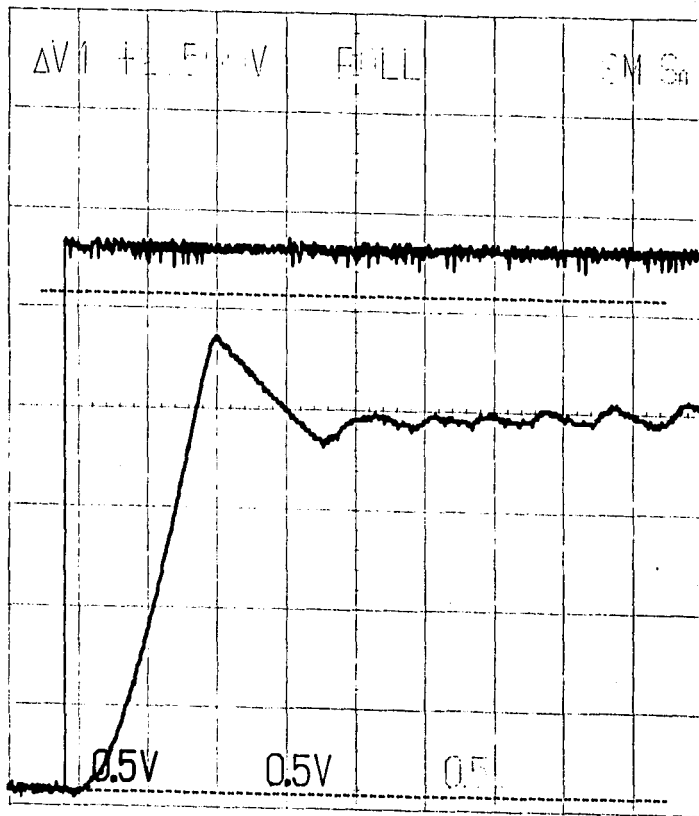


100% rayón estampado.

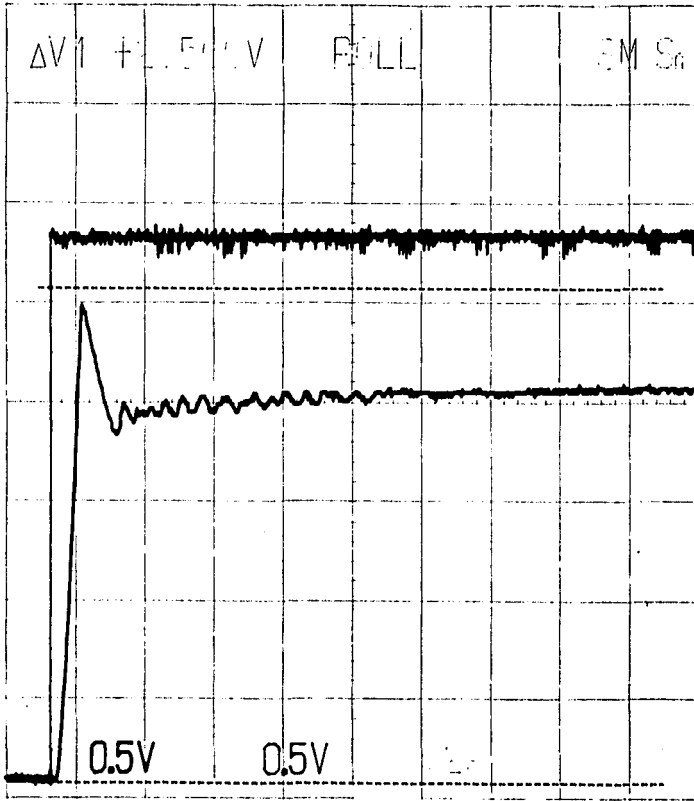
	Presión [mI20]	Gasto [m ³ /s]		Presión [mI20]	Gasto [m ³ /s]
0	0	0	45	0.44392	0.00181496
1	0.01568	0	46	0.45372	0.001857664
2	0.02117	0	47	0.46274	0.001910336
3	0.03137	0	48	0.47803	0.001967007
4	0.03921	0	49	0.4898	0.00201945
5	0.04705	0	50	0.50196	0.002070566
6	0.0545	0	51	0.51372	0.002120534
7	0.06117	0	52	0.52549	0.002178836
8	0.06901	0	53	0.54078	0.002216933
9	0.07843	0	54	0.55098	0.002307079
10	0.08627	0	55	0.5647	0.002354218
11	0.09411	0	56	0.57921	0.002402579
12	0.10235	0	57	0.59137	0.002464887
13	0.11019	0	58	0.60352	0.002523547
14	0.12156	0	59	0.61411	0.002566714
15	0.12941	0	60	0.62862	0.002611142
16	0.14117	0	61	0.64313	0.002682274
17	0.14784	0	62	0.6549	0.002724818
18	0.15529	0	63	0.66666	0.002768571
19	0.17058	0	64	0.68117	0.002837637
20	0.17921	0	65	0.69411	0.002874239
21	0.18627	0	66	0.70941	0.002962026
22	0.19411	0	67	0.72235	0.003012787
23	0.20392	0.000323369	68	0.73411	0.003049035
24	0.21372	0.000323369	69	0.74823	0.003110111
25	0.22352	0.000560992	70	0.76117	0.00316018
26	0.23529	0.000569348	71	0.77372	0.003217546
27	0.24509	0.000723076	72	0.78902	0.003279361
28	0.25451	0.000723076	73	0.80392	0.003313631
29	0.26666	0.000855576	74	0.81725	0.003376134
30	0.27451	0.000867708	75	0.83137	0.003407008
31	0.2847	0.000970127	76	0.84352	0.003469314
32	0.29803	0.001037839	77	0.85882	0.003530572
33	0.30627	0.001106115	78	0.87176	0.00358206
34	0.31921	0.001165956	79	0.88549	0.003629882
35	0.32941	0.001248253	80	0.90156	0.003691316
36	0.34039	0.001252433	81	0.91803	0.00374197
37	0.35137	0.001309582	82	0.93058	0.003789181
38	0.36235	0.001409573	83	0.9447	0.003841276
39	0.37294	0.001460586	84	0.96078	0.003893998
40	0.38588	0.001506399	85	0.96941	0.003939387
41	0.39882	0.00159081	86	0.98509	0.003984259
42	0.39764	0.001597367	87	1	0.004035114
43	0.41882	0.001683427			
44	0.42509	0.001729392			



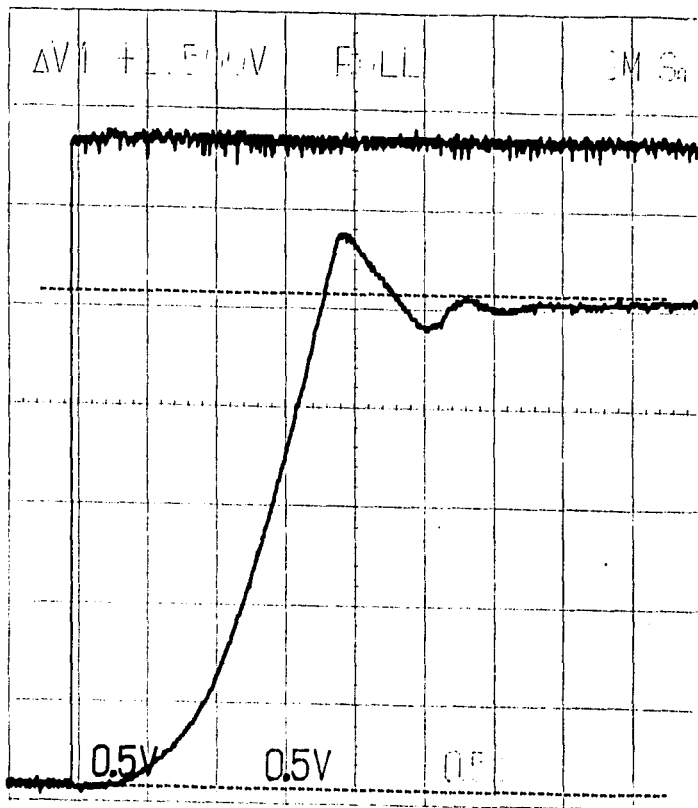
100% rayón estampado.



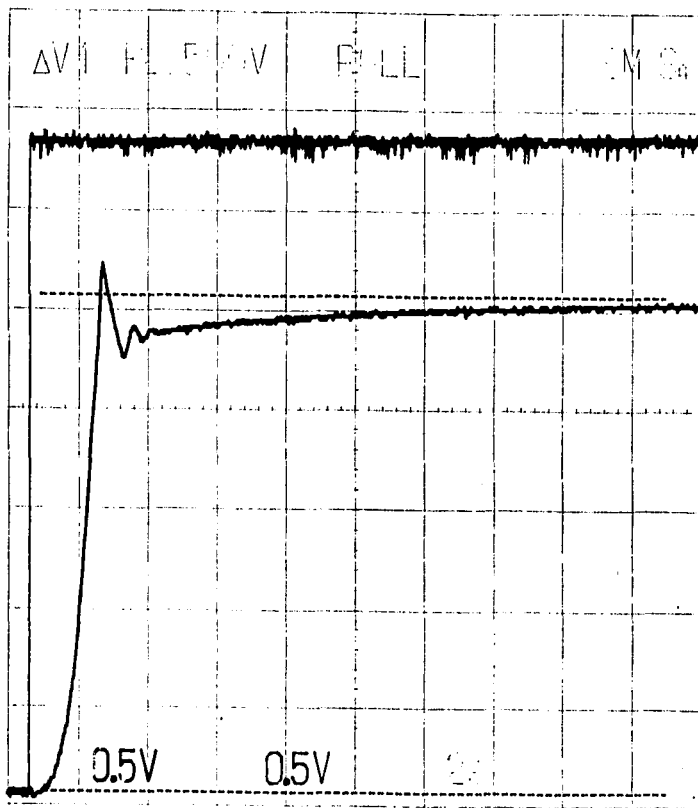
100% rayón estampado.



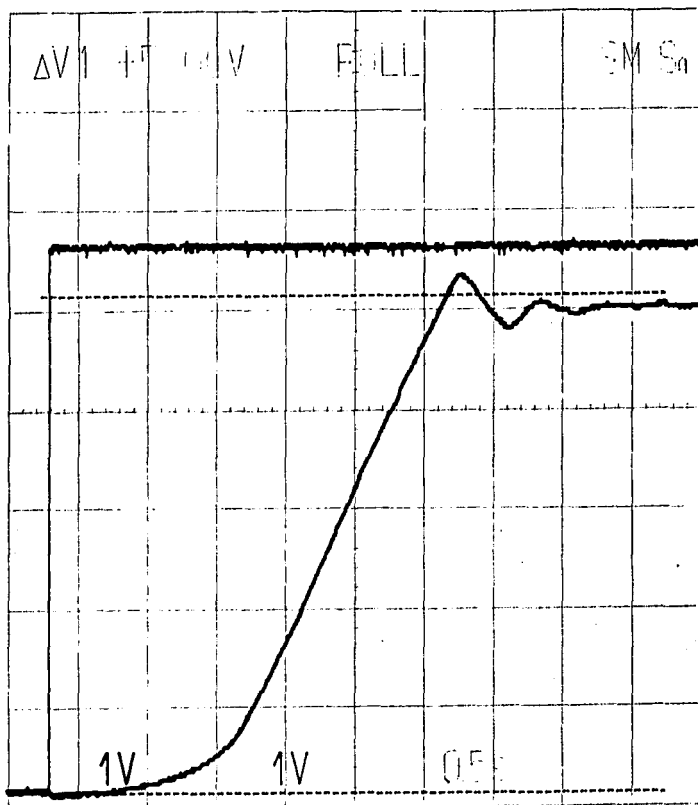
100% rayón estampado.



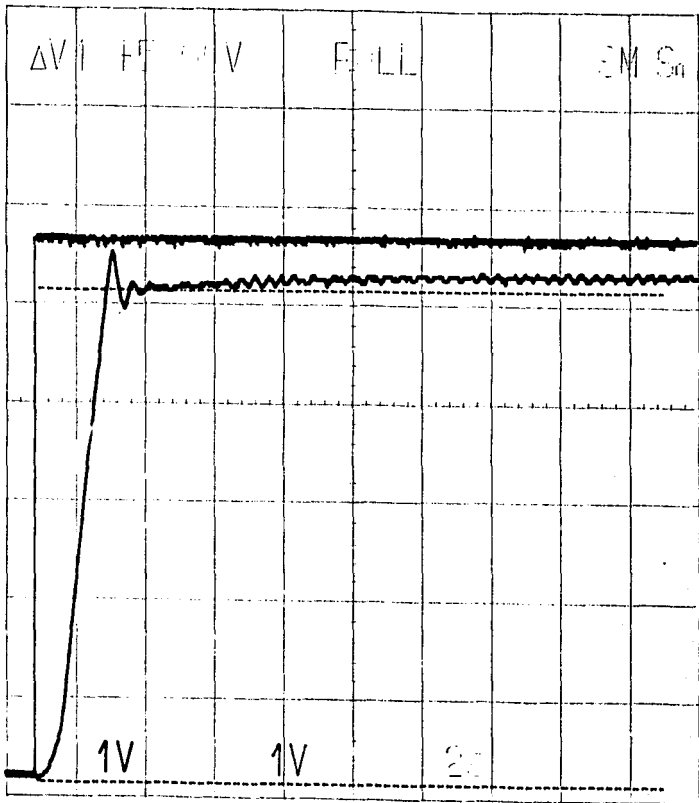
100% rayón estampado.



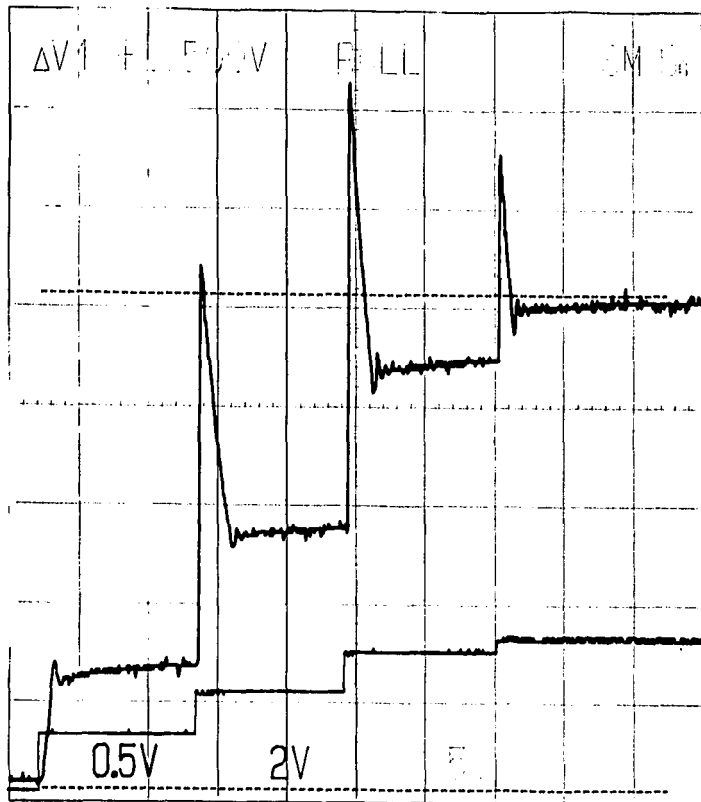
100% rayón estampado.



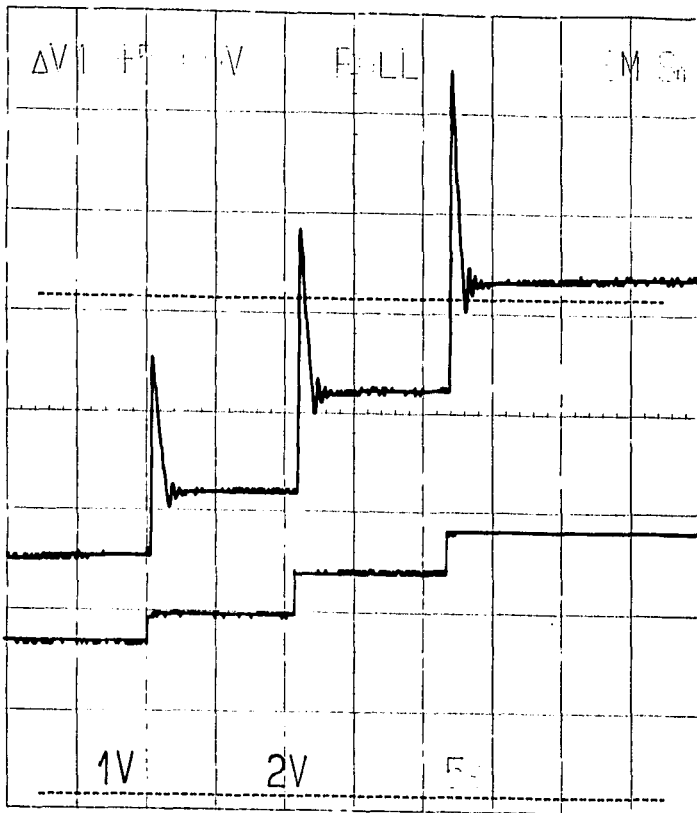
100% rayón estampado.



100% rayón estampado.

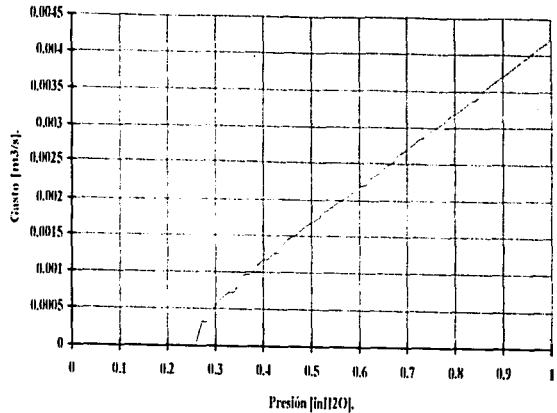


100% rayón estampado.

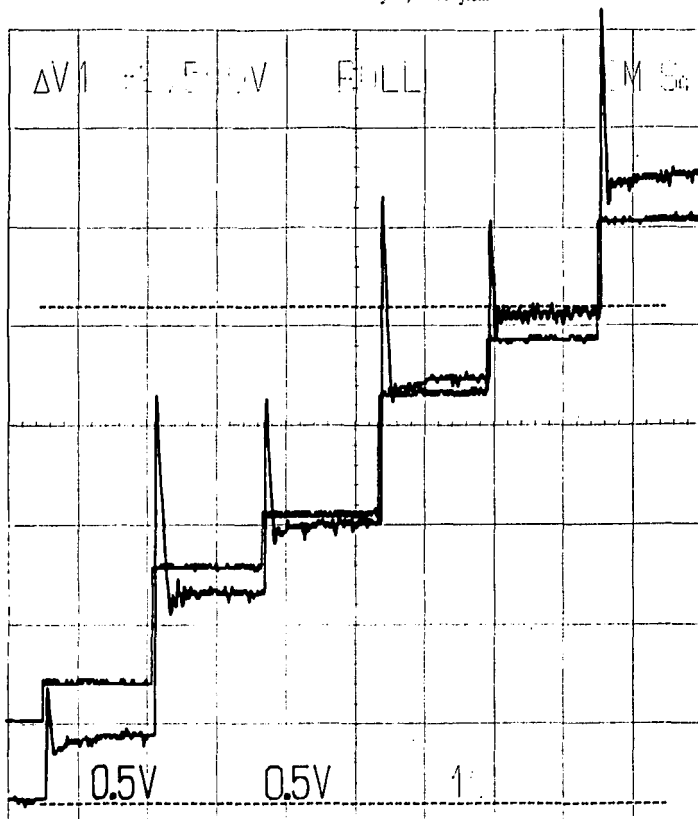


Lyera, 100 % nylon (tejido de punto).

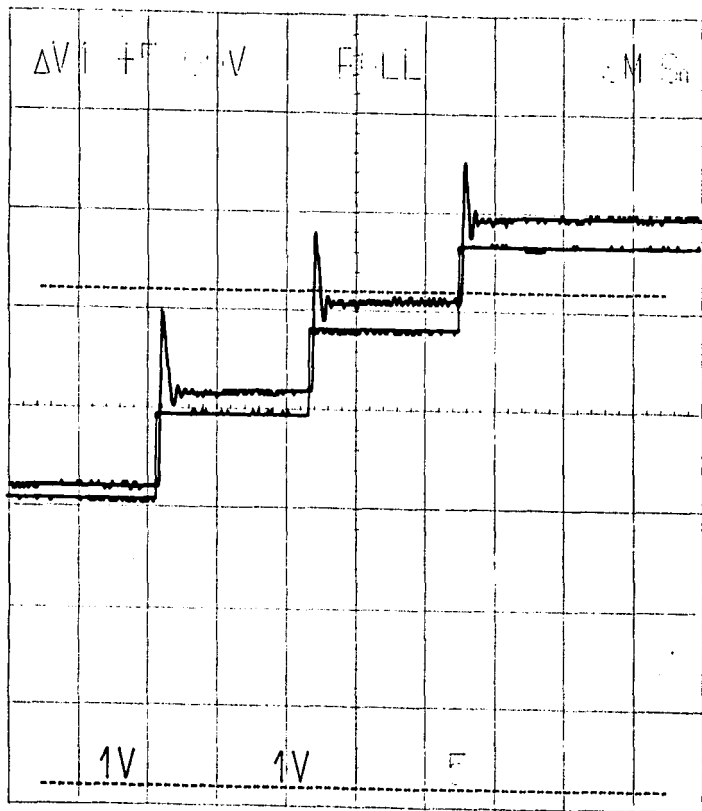
Presión [inH ₂ O]	Gasto [m ³ /s]	Presión [inH ₂ O]	Gasto [m ³ /s]	Presión [inH ₂ O]	Gasto [m ³ /s]
0	0.00039	0	45	0.51647	0.001741449
1	0.01568	0	46	0.52039	0.001800494
2	0.0247	0	47	0.53372	0.001857664
3	0.03529	0	48	0.54588	0.001915819
4	0.04705	0	49	0.55725	0.001967007
5	0.0549	0	50	0.56902	0.002024638
6	0.06666	0	51	0.58039	0.002070566
7	0.07843	0	52	0.59137	0.00214504
8	0.08784	0	53	0.6	0.002200352
9	0.09764	0	54	0.61372	0.002221659
10	0.1098	0	55	0.6247	0.002309389
11	0.11882	0	56	0.64156	0.002380725
12	0.13254	0	57	0.65908	0.002441408
13	0.14235	0	58	0.66196	0.002490217
14	0.15333	0	59	0.67451	0.002540118
15	0.16196	0	60	0.68235	0.002607122
16	0.17803	0	61	0.69294	0.002646913
17	0.18862	0	62	0.70588	0.002713245
18	0.20666	0	63	0.71529	0.002762915
19	0.21176	0	64	0.72784	0.002837637
20	0.21725	0	65	0.73921	0.002874239
21	0.23529	0	66	0.74941	0.002944328
22	0.24352	0	67	0.75607	0.002974364
23	0.25647	0	68	0.77019	0.003047285
24	0.27058	0.000323369	69	0.78039	0.003091593
25	0.28	0.000323369	70	0.7898	0.0031519
26	0.29451	0.000457313	71	0.80039	0.003194701
27	0.30196	0.000560092	72	0.81372	0.00325632
28	0.31451	0.000646739	73	0.82352	0.003313631
29	0.32784	0.000723076	74	0.83529	0.003369968
30	0.33725	0.000723076	75	0.84588	0.003407008
31	0.34941	0.000855576	76	0.85647	0.003481386
32	0.36078	0.000951827	77	0.86313	0.00352765
33	0.3749	0.000970127	78	0.8749	0.00358206
34	0.3847	0.001072512	79	0.88666	0.00363854
35	0.39921	0.001165956	80	0.89882	0.003701224
36	0.40784	0.001174887	81	0.90784	0.003743348
37	0.42117	0.001252433	82	0.91764	0.003794713
38	0.43137	0.001260752	83	0.92902	0.003852137
39	0.44274	0.001390897	84	0.93725	0.003895322
40	0.45843	0.00147817	85	0.94862	0.003955285
41	0.46392	0.001485424	86	0.95843	0.004001268
42	0.47882	0.001570962	87	0.96941	0.004054496
43	0.48627	0.001616891	88	0.98156	0.004118446
44	0.49843	0.001674087	89	0.99019	0.004162627



Lyra, 100% nylon.

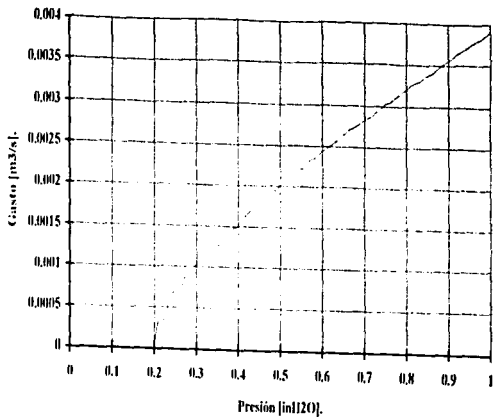


Lyera, 100% nylon.

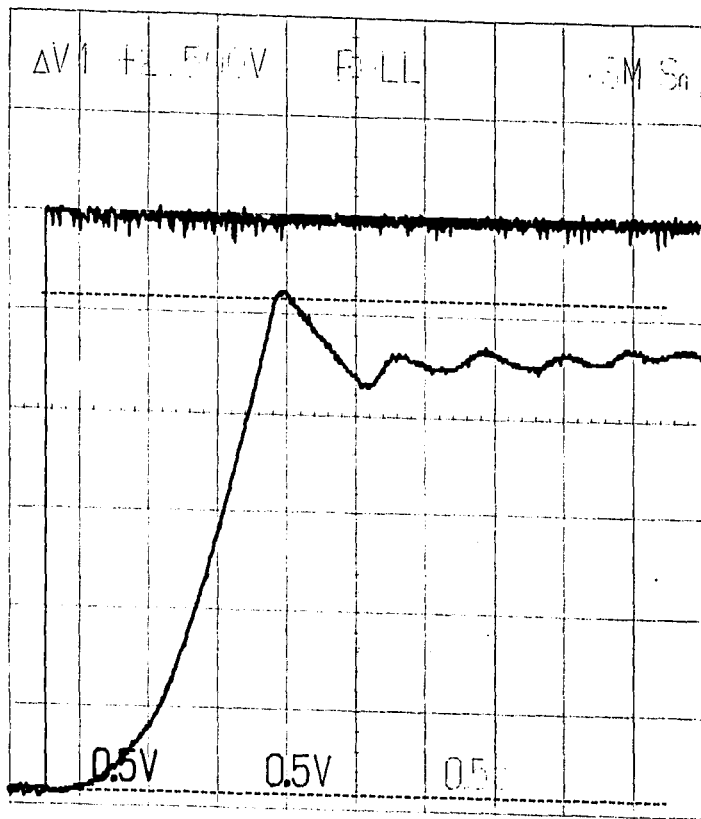


100% poliéster.

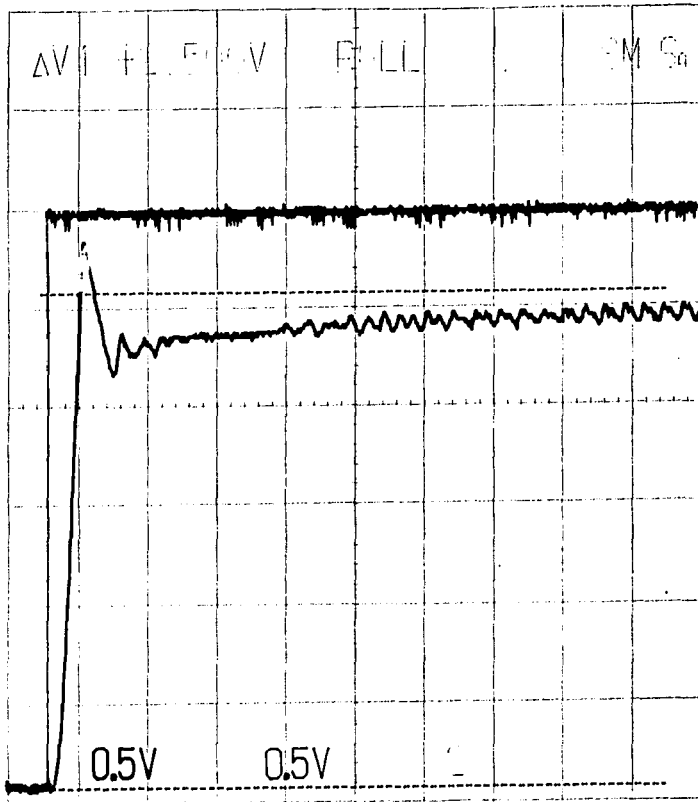
Presion [inI2O]	Gasto [m3/s]	Presion [inI2O]	Gasto [m3/s]
0	0	45	0.45176
1	0.01568	46	0.4647
2	0.02352	47	0.4796
3	0.03098	48	0.49411
4	0.03882	49	0.50588
5	0.04705	50	0.51764
6	0.0549	51	0.52941
7	0.06352	52	0.54311
8	0.07058	53	0.55686
9	0.08039	54	0.56862
10	0.08705	55	0.58431
11	0.09803	56	0.59803
12	0.1098	57	0.61176
13	0.11411	58	0.6247
14	0.12509	59	0.63529
15	0.13254	60	0.64862
16	0.14235	61	0.66039
17	0.15058	62	0.67451
18	0.15764	63	0.68666
19	0.17254	64	0.70509
20	0.18431	65	0.71725
21	0.18823	66	0.73176
22	0.20274	67	0.74823
23	0.20784	68	0.76078
24	0.2196	69	0.77058
25	0.23333	70	0.78549
26	0.24156	71	0.8
27	0.24705	72	0.81607
28	0.26156	73	0.83372
29	0.27058	74	0.84902
30	0.28235	75	0.86274
31	0.29294	76	0.87882
32	0.30549	77	0.89411
33	0.31568	78	0.90862
34	0.32941	79	0.92313
35	0.33725	80	0.94078
36	0.34902	81	0.95333
37	0.36078	82	0.96862
38	0.37176	83	0.98431
39	0.38196	84	1
40	0.39686		
41	0.4047		
42	0.42549		
43	0.43137		
44	0.44078		



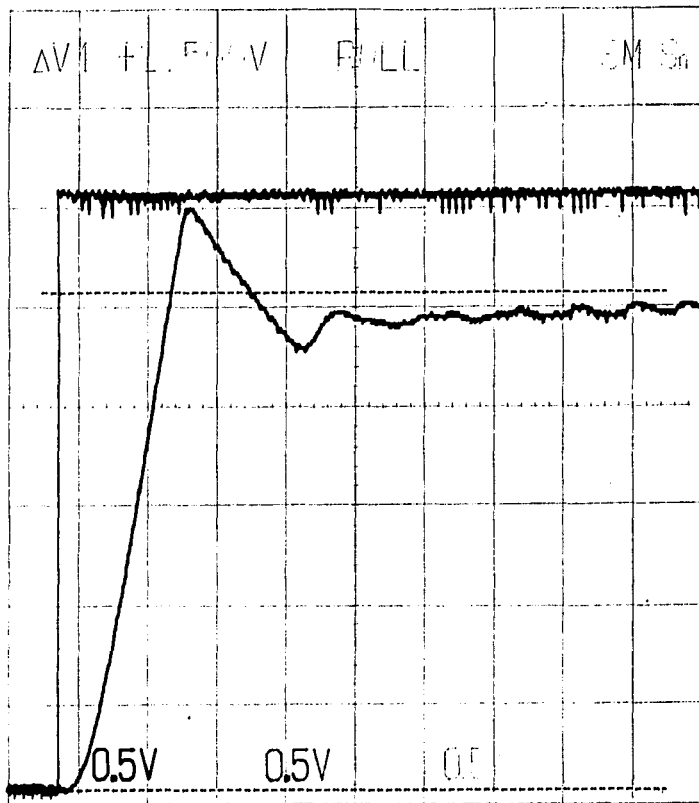
Poliéster 100%.



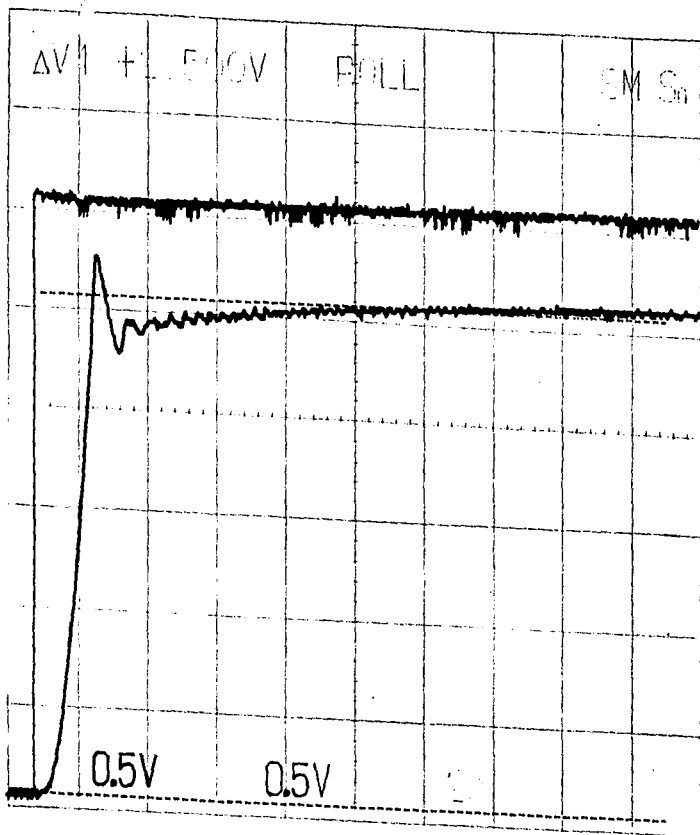
Poliéster 100%.



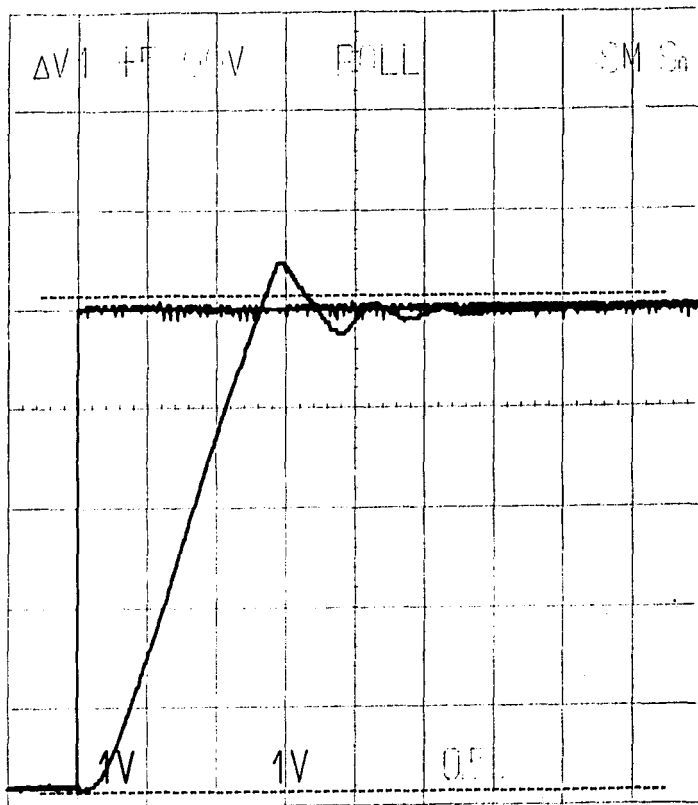
Poliéster 100%.



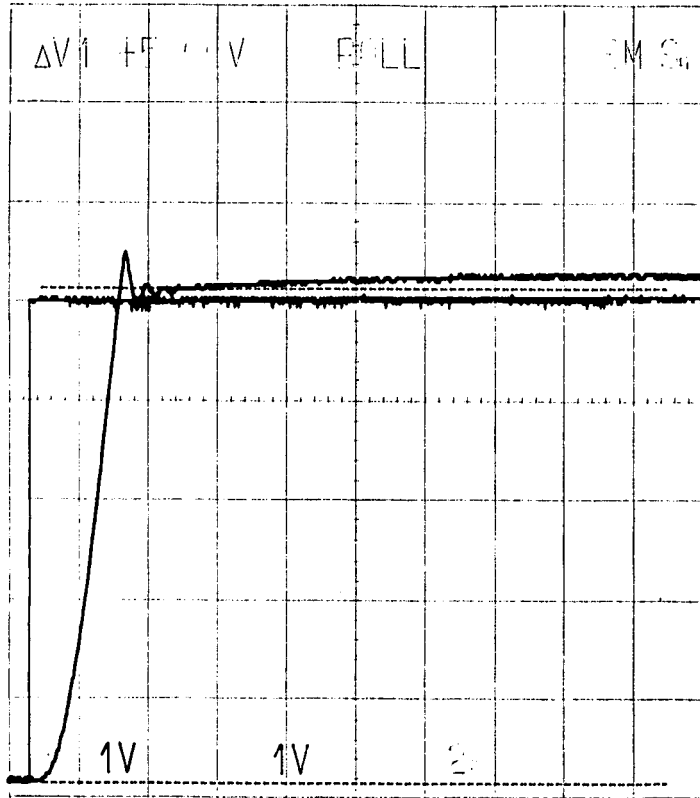
Poliéster 100%.



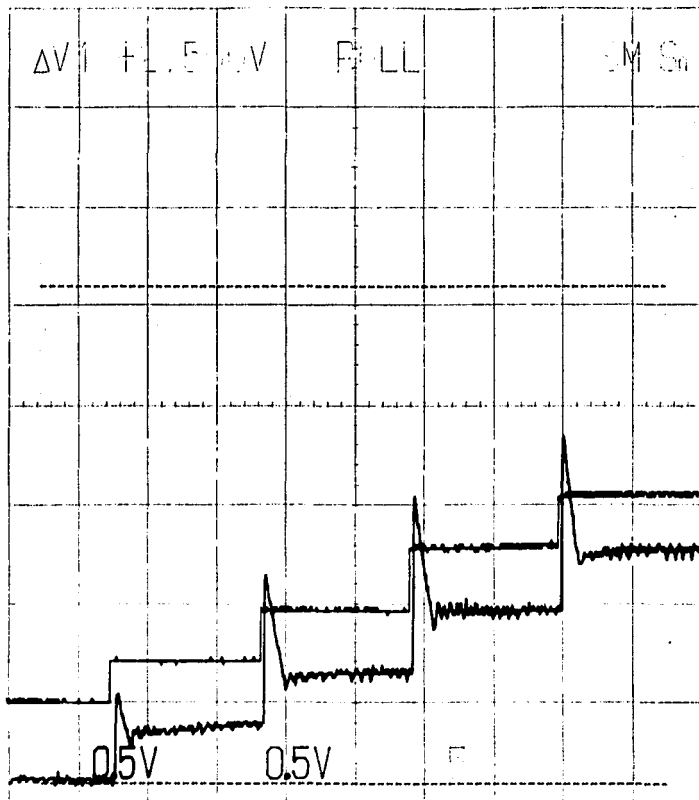
100% poliéster.



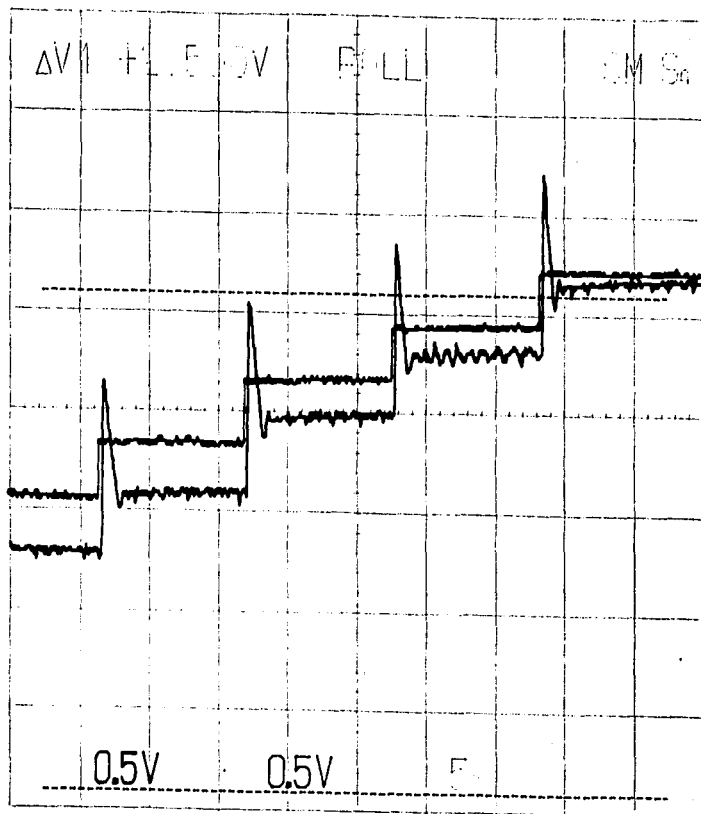
100% poliéster.



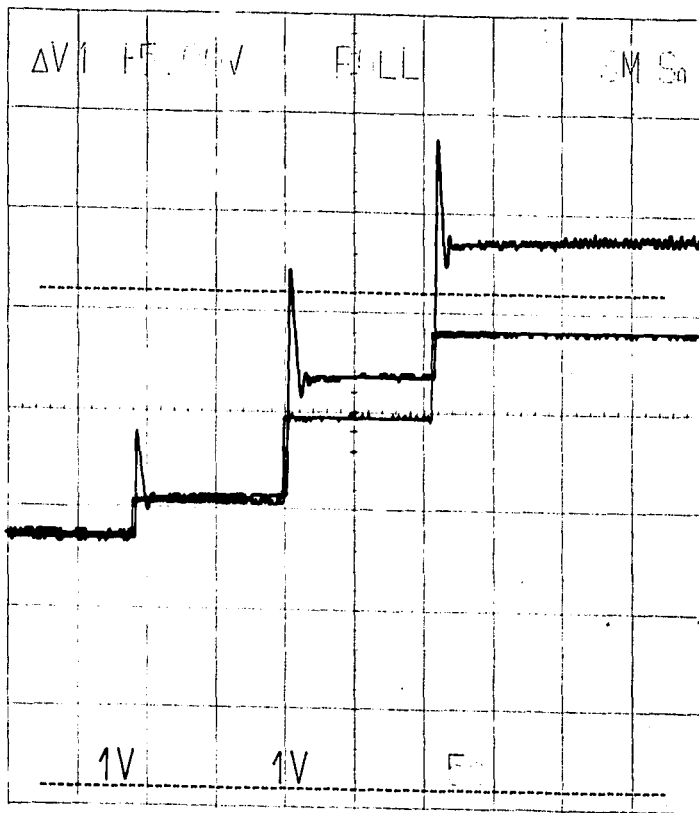
Poliéster 100%.



Políster 100%

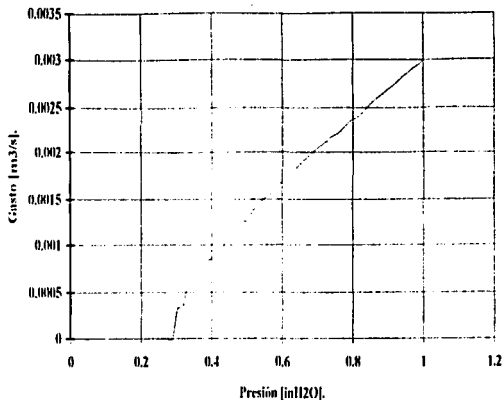


Poliéster 100%.

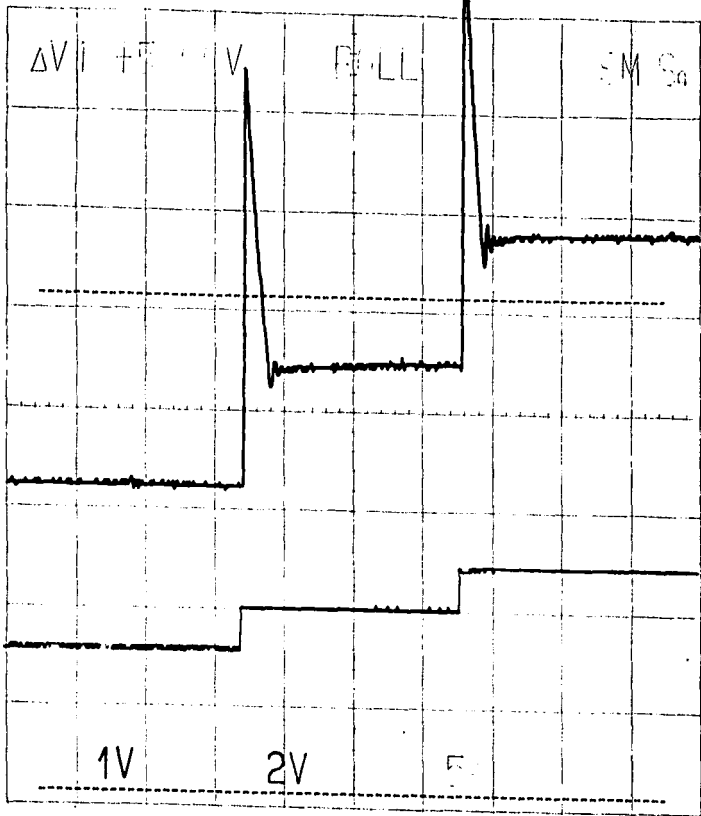


Paño.

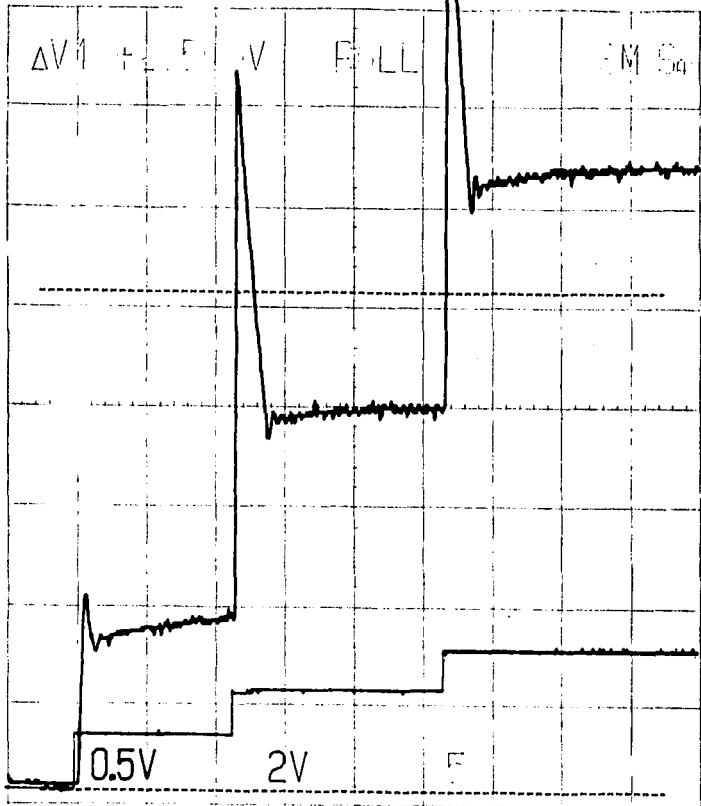
	Presion [in120]	Gasto [m ³ /s]		Presion [in120]	Gasto [m ³ /s]
0	0.00039	0	45	0.59529	0.001677206
1	0.01568	0	46	0.61411	0.001738445
2	0.02352	0	47	0.62902	0.001791764
3	0.03294	0	48	0.64588	0.001857664
4	0.04705	0	49	0.66235	0.001913126
5	0.0549	0	50	0.67725	0.001967007
6	0.06313	0	51	0.69215	0.00201945
7	0.0749	0	52	0.7098	0.002068075
8	0.08784	0	53	0.72549	0.002120534
9	0.09803	0	54	0.73764	0.002169269
10	0.1098	0	55	0.75882	0.002216933
11	0.12549	0	56	0.77647	0.00229115
12	0.13333	0	57	0.79215	0.002354218
13	0.14509	0	58	0.81176	0.002398209
14	0.15686	0	59	0.82588	0.002441408
15	0.17254	0	60	0.84039	0.00250488
16	0.18784	0	61	0.85882	0.002566714
17	0.19607	0	62	0.87451	0.002607122
18	0.20823	0	63	0.89451	0.002666653
19	0.22117	0	64	0.91333	0.002724818
20	0.2298	0	65	0.92745	0.002768571
21	0.24705	0	66	0.94274	0.002824702
22	0.2596	0	67	0.96078	0.002874239
23	0.27451	0	68	0.97372	0.00291575
24	0.28941	0	69	0.99803	0.002986651
25	0.30196	0.000323369	70	1	0.003045593
26	0.31568	0.000323369			
27	0.3298	0.000541107			
28	0.34509	0.000560092			
29	0.36039	0.000654771			
30	0.37137	0.000723076			
31	0.3898	0.000837027			
32	0.4	0.000855576			
33	0.41411	0.000964724			
34	0.43098	0.000970127			
35	0.44313	0.001072512			
36	0.45764	0.00112964			
37	0.47176	0.001164			
38	0.48627	0.00125433			
39	0.50196	0.001253433			
40	0.51882	0.001375784			
41	0.53294	0.001424327			
42	0.53882	0.001467725			
43	0.57098	0.001567631			
44	0.58039	0.001616891			



Paño. 95% lana y 5% nylon.

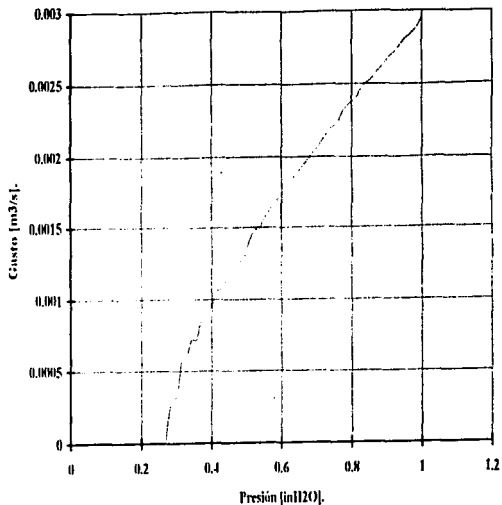


Paño, 95% lana y 5% nylon.

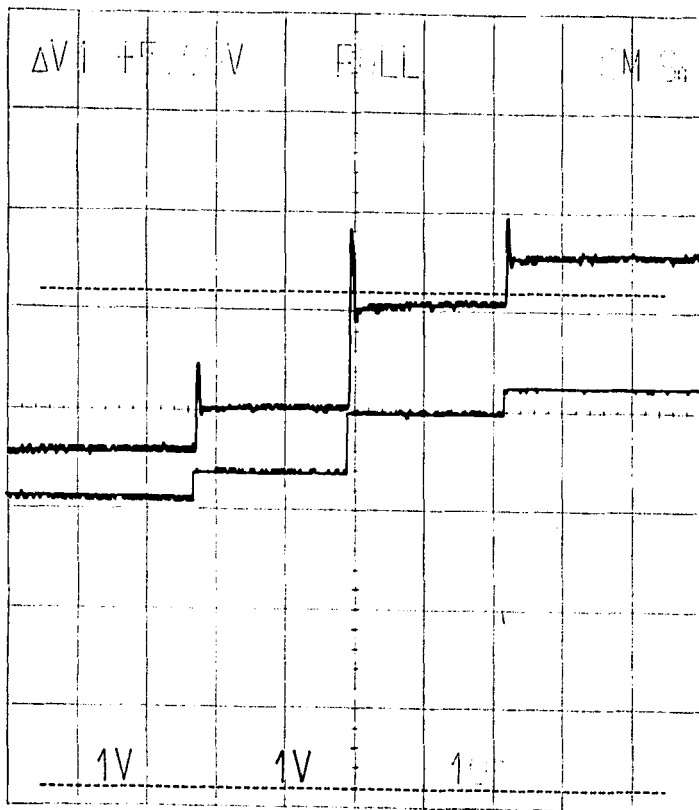


100% poliéster estampado.

	Presión [in12O]	Gasto [m3/s]		Presión [in12O]	Gasto [m3/s]
0	0.00078	0	45	0.5847	0.001689635
1	0.01686	0	46	0.60156	0.001741449
2	0.02705	0	47	0.61843	0.001800494
3	0.03921	0	48	0.63764	0.001857664
4	0.04705	0	49	0.65529	0.001913126
5	0.0549	0	50	0.67294	0.001967007
6	0.0647	0	51	0.68627	0.00201945
7	0.07843	0	52	0.70392	0.002070566
8	0.08509	0	53	0.72156	0.002135238
9	0.09764	0	54	0.74117	0.00219558
10	0.10902	0	55	0.75686	0.002216933
11	0.11882	0	56	0.77568	0.002309389
12	0.12941	0	57	0.79372	0.002354218
13	0.14117	0	58	0.81176	0.002400358
14	0.15294	0	59	0.83098	0.002481779
15	0.16352	0	60	0.85137	0.002525659
16	0.17725	0	61	0.86745	0.002566714
17	0.1898	0	62	0.88352	0.002609167
18	0.20627	0	63	0.90313	0.002668585
19	0.2149	0	64	0.92549	0.002724818
20	0.22313	0	65	0.94352	0.002787386
21	0.23882	0	66	0.95725	0.002815435
22	0.24705	0	67	0.98274	0.002874239
23	0.26666	0	68	0.9996	0.00295319
24	0.27843	0.000228618	69	1	0.002981408
25	0.29019	0.00032369			
26	0.30196	0.000339147			
27	0.31411	0.000560092			
28	0.32941	0.000622022			
29	0.34196	0.000723076			
30	0.35607	0.000723076			
31	0.36902	0.000849445			
32	0.38627	0.000891476			
33	0.4	0.000970127			
34	0.41568	0.001062721			
35	0.42784	0.001082231			
36	0.4447	0.001165956			
37	0.45882	0.001244059			
38	0.47411	0.001252433			
39	0.48902	0.001285399			
40	0.50627	0.001409573			
41	0.5247	0.001481901			
42	0.5349	0.001485424			
43	0.55686	0.001584225			
44	0.57725	0.001661552			

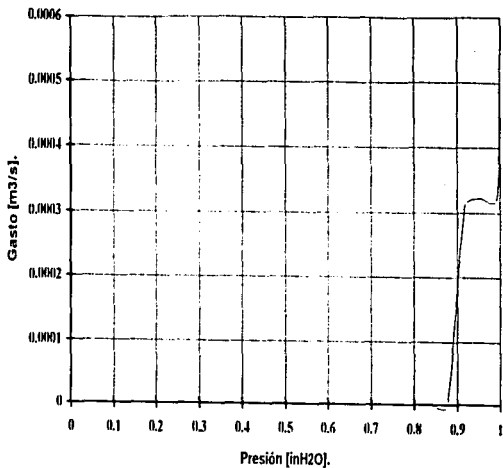


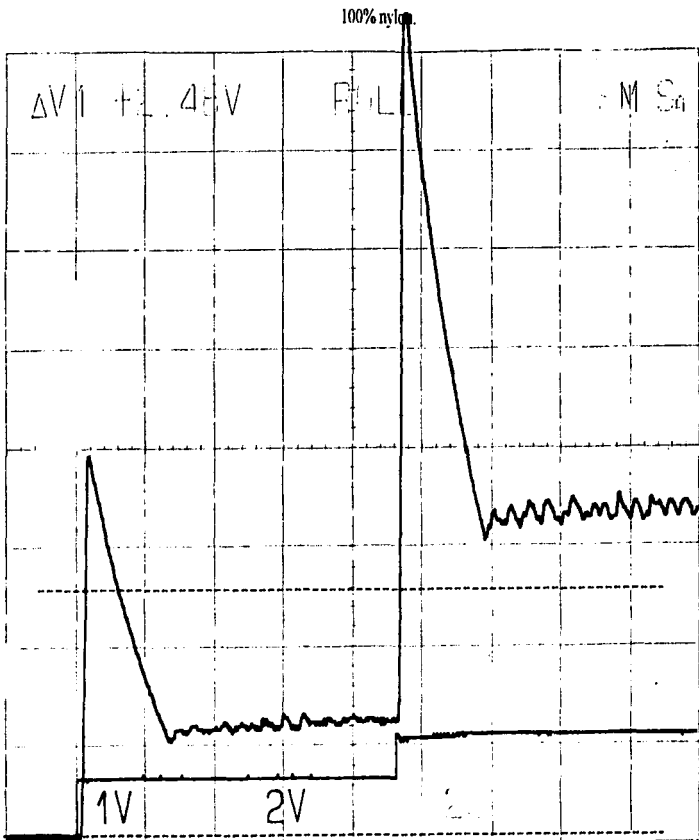
100% poliéster estampado.

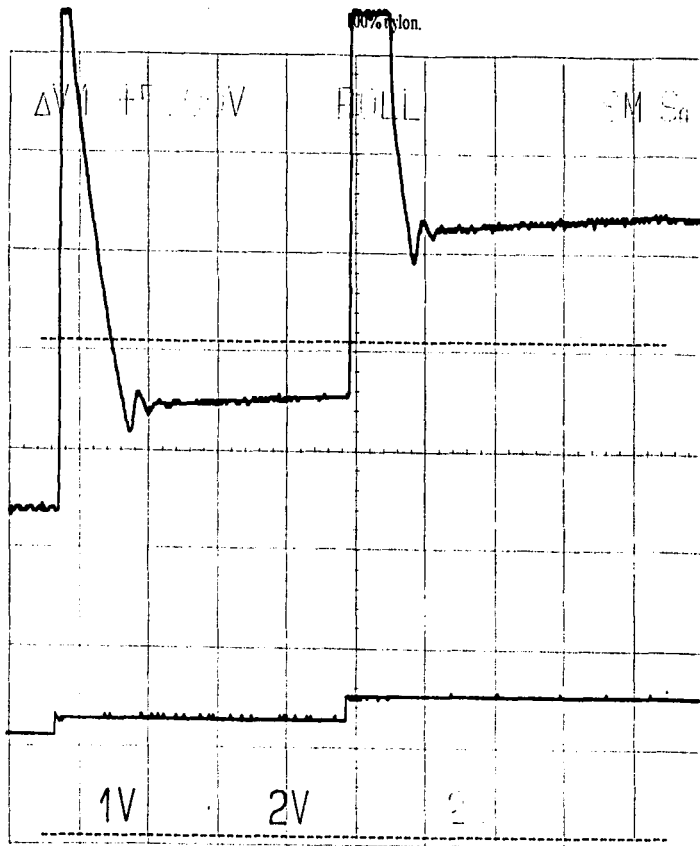


100% nylon.

	Presión [mH2O]	Gasto [m ³ /s]
0	0.00196	0
1	0.01882	0
2	0.02823	0
3	0.04078	0
4	0.0545	0
5	0.06862	0
6	0.08235	0
7	0.09686	0
8	0.11111	0
9	0.13019	0
10	0.14823	0
11	0.17058	0
12	0.18823	0
13	0.20862	0
14	0.22627	0
15	0.24705	0
16	0.27803	0
17	0.30235	0
18	0.34156	0
19	0.35215	0
20	0.39568	0
21	0.40823	0
22	0.43764	0
23	0.46392	0
24	0.49843	0
25	0.5298	0
26	0.56	0
27	0.59215	0
28	0.62666	0
29	0.65647	0
30	0.69137	0
31	0.72627	0
32	0.76549	0
33	0.80078	0
34	0.84039	0
35	0.87686	0
36	0.91725	0.000306781
37	0.95215	0.000323369
38	0.99294	0.000323369
39	1	0.000511275



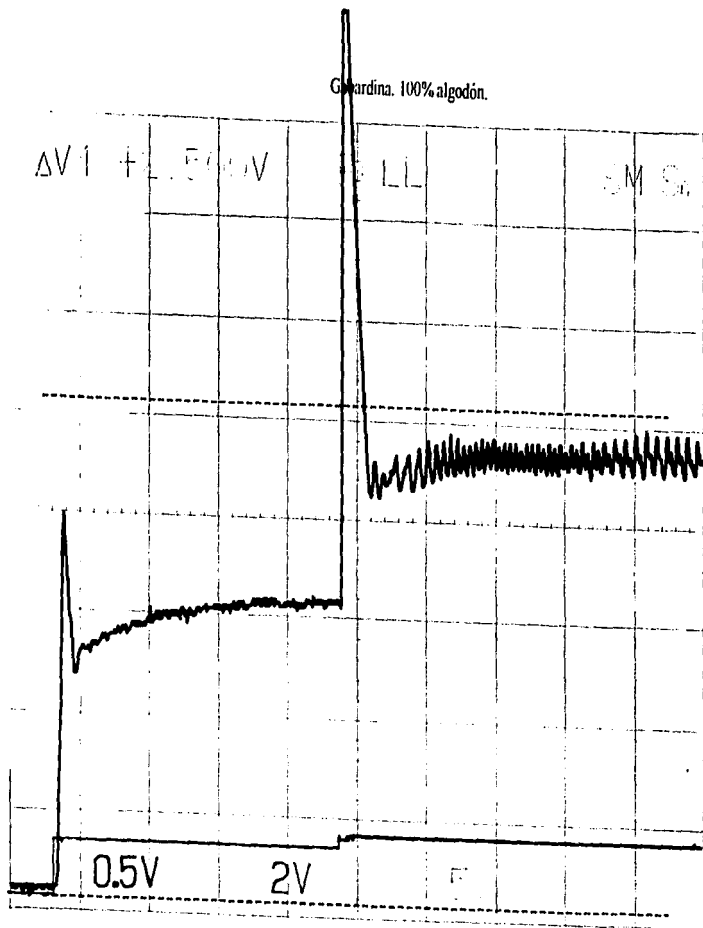




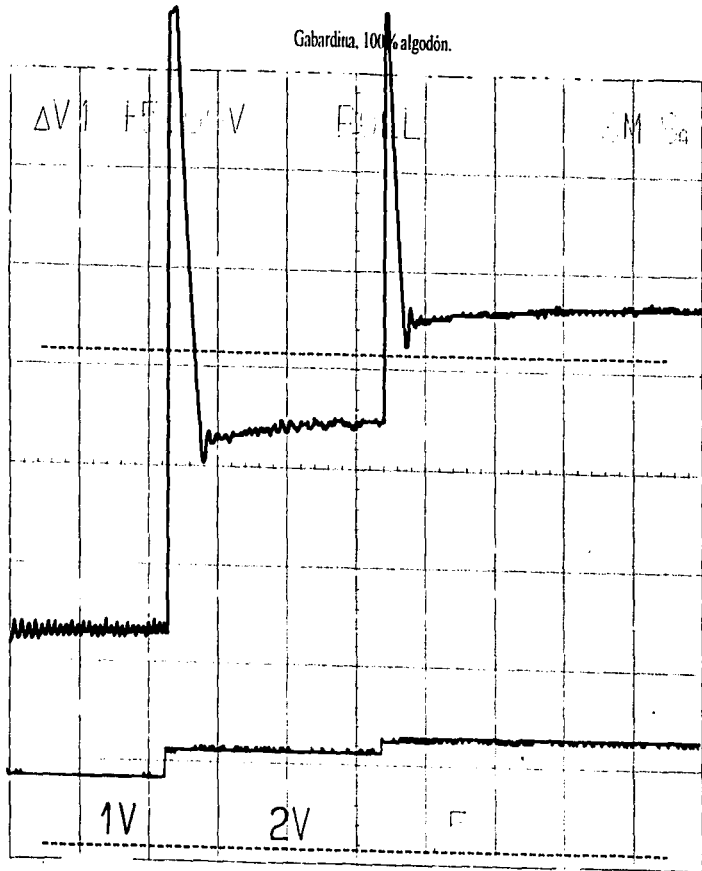
Gabardina, 100% algodón.

	Presión [mH ₂ O].	Gasto [m ³ /s].
0	0.00039	0
1	0.01043	0
2	0.03137	0
3	0.04549	0
4	0.05803	0
5	0.06666	0
6	0.08745	0
7	0.1098	0
8	0.12274	0
9	0.14117	0
10	0.16078	0
11	0.18823	0
12	0.20666	0
13	0.22784	0
14	0.25725	0
15	0.28039	0
16	0.31607	0
17	0.34549	0
18	0.38274	0
19	0.40156	0
20	0.43333	0
21	0.46666	0
22	0.50078	0
23	0.53372	0
24	0.57607	0
25	0.61294	0
26	0.65294	0
27	0.6898	0
28	0.72862	0
29	0.76862	0
30	0.80823	0
31	0.84902	0
32	0.89764	0
33	0.94117	0
34	0.98235	0
35	1	0

Guardina. 100% algodón.

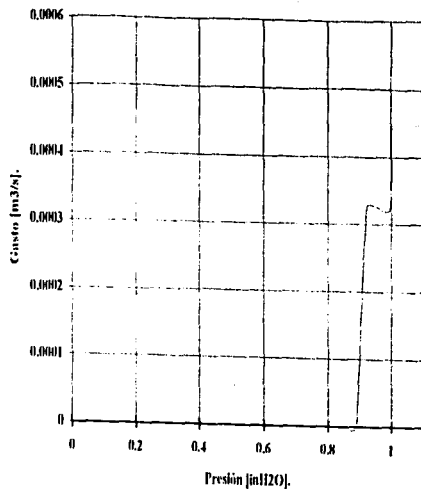


Gabardina, 100% algodón.

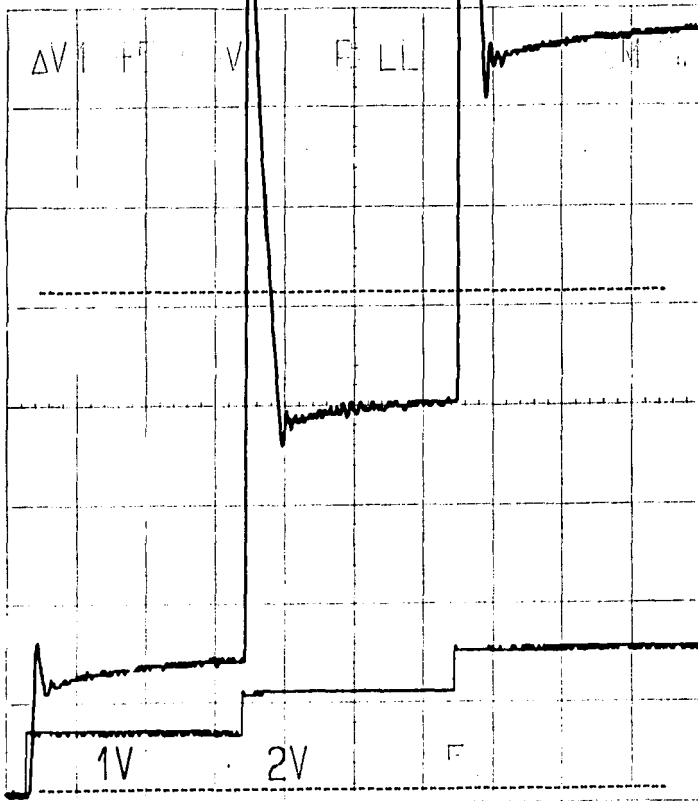


Ferro, 100% polister.

	Presion [inH2O]	Gasto [m ³ /s]
0	0.06039	0
1	0.01803	0
2	0.03137	0
3	0.04509	0
4	0.05529	0
5	0.06745	0
6	0.08392	0
7	0.09647	0
8	0.11725	0
9	0.14	0
10	0.15294	0
11	0.17254	0
12	0.19333	0
13	0.21294	0
14	0.23764	0
15	0.25843	0
16	0.28666	0
17	0.31176	0
18	0.33843	0
19	0.36078	0
20	0.39019	0
21	0.41882	0
22	0.45098	0
23	0.4749	0
24	0.51294	0
25	0.54352	0
26	0.57333	0
27	0.60549	0
28	0.64	0
29	0.67411	0
30	0.70588	0
31	0.73647	0
32	0.77686	0
33	0.81058	0
34	0.84784	0
35	0.88705	0
36	0.92215	0.000323369
37	0.95882	0.000323369
38	0.99803	0.000323369
39	1	0.000511275

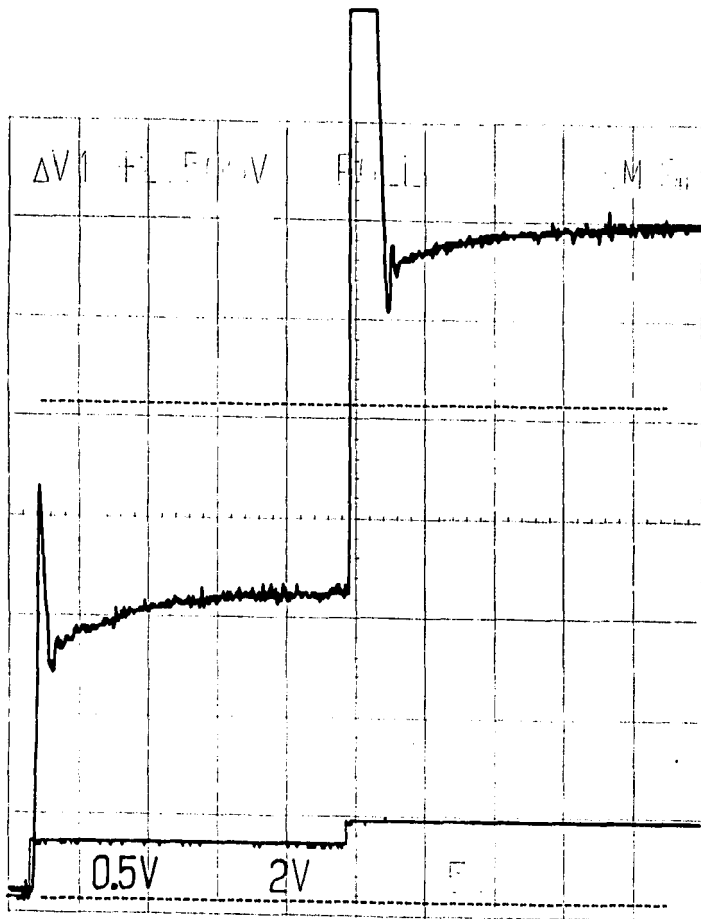


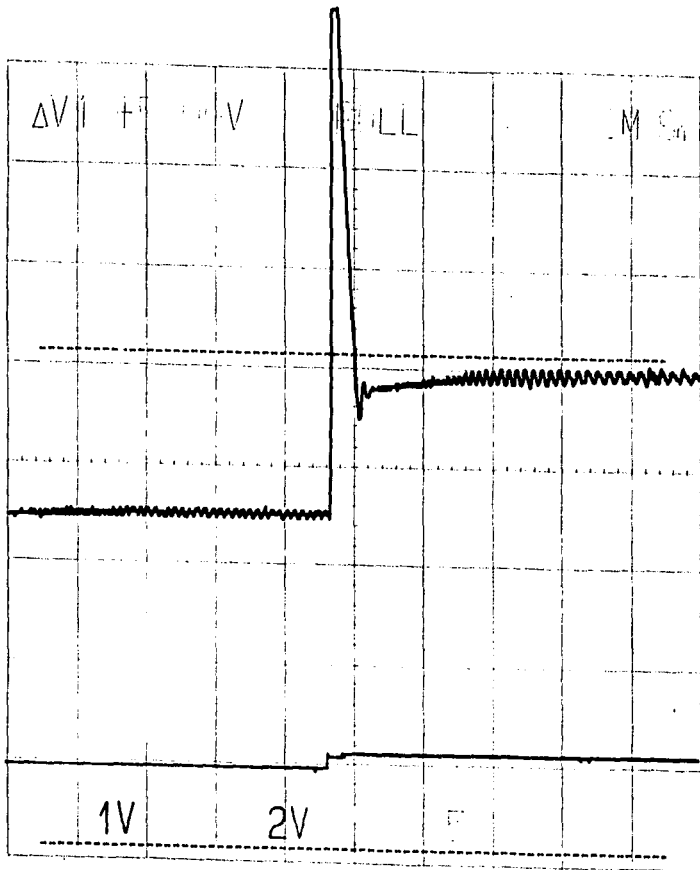
Ferro, 100% poliéster.



Seda 100%

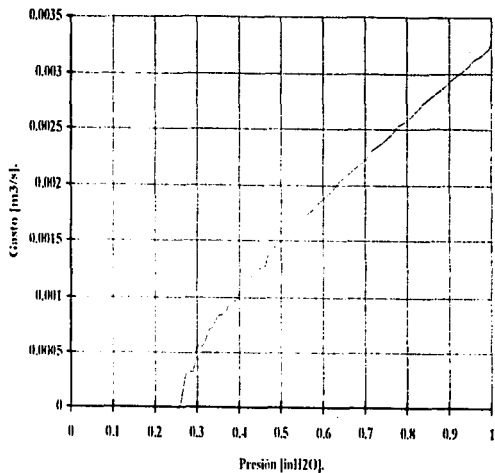
	Presión [m12O]	Gasto [m3/s]
0	0.00039	0
1	0.01607	0
2	0.0298	0
3	0.04705	0
4	0.06039	0
5	0.07058	0
6	0.08901	0
7	0.1098	0
8	0.13098	0
9	0.14902	0
10	0.17451	0
11	0.19647	0
12	0.22117	0
13	0.24705	0
14	0.27529	0
15	0.30705	0
16	0.34196	0
17	0.37529	0
18	0.40509	0
19	0.44	0
20	0.46549	0
21	0.51647	0
22	0.56078	0
23	0.60196	0
24	0.65137	0
25	0.69803	0
26	0.74313	0
27	0.78823	0
28	0.83725	0
29	0.88627	0
30	0.94156	0
31	0.98902	0
32	1	0

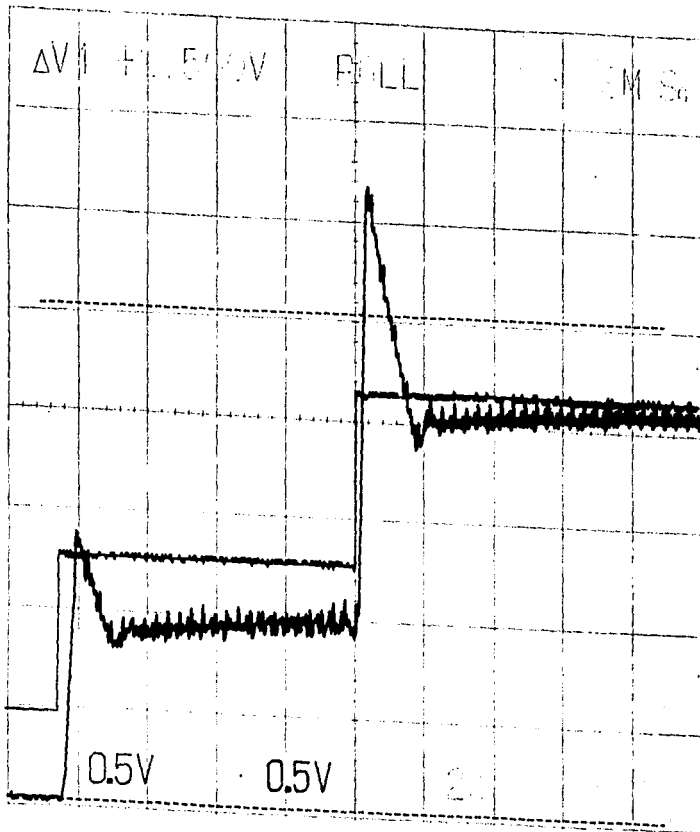


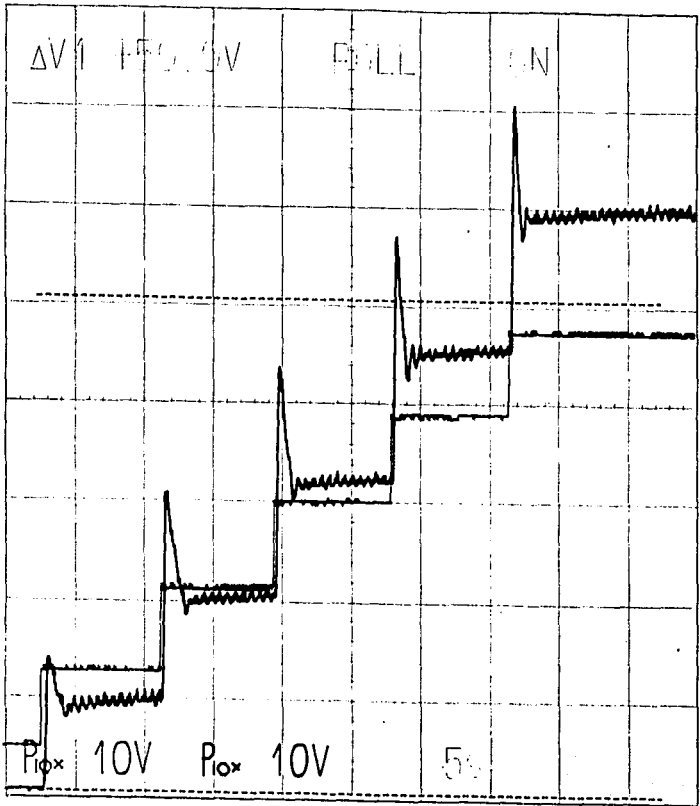


Franela, 100° algodón.

	Presión [inH2O]	Gasto [m ³ /s]		Presión [inH2O]	Gasto [m ³ /s]
0	0.00078	0	45	0.55607	0.001723337
1	0.01607	0	46	0.57568	0.001783911
2	0.02666	0	47	0.59019	0.001854848
3	0.03568	0	48	0.60862	0.001913126
4	0.04705	0	49	0.62313	0.001967007
5	0.0549	0	50	0.63529	0.00201945
6	0.06627	0	51	0.65137	0.002070566
7	0.07843	0	52	0.66666	0.002120534
8	0.08627	0	53	0.68235	0.002171644
9	0.09764	0	54	0.69764	0.002216933
10	0.1098	0	55	0.71372	0.0022604843
11	0.11764	0	56	0.73176	0.002304218
12	0.13098	0	57	0.74549	0.0023398299
13	0.14156	0	58	0.76117	0.002452088
14	0.15294	0	59	0.77607	0.002525659
15	0.1647	0	60	0.79372	0.002566714
16	0.17803	0	61	0.80784	0.002609167
17	0.18862	0	62	0.82313	0.002672513
18	0.1996	0	63	0.83921	0.002724818
19	0.21176	0	64	0.86	0.002787386
20	0.22705	0	65	0.87686	0.002839453
21	0.23686	0	66	0.89372	0.002892366
22	0.24705	0	67	0.90588	0.002946078
23	0.25921	0	68	0.9247	0.003000607
24	0.27451	0.000323369	69	0.94352	0.003055849
25	0.28941	0.000323369	70	0.95725	0.003115194
26	0.30196	0.000560092	71	0.97294	0.003155226
27	0.31372	0.000560092	72	0.99215	0.003217546
28	0.32941	0.000708472	73	1	0.003270756
29	0.33764	0.000723076			
30	0.3498	0.00083076			
31	0.3647	0.000855576			
32	0.38	0.000920127			
33	0.39215	0.000975499			
34	0.40588	0.001072512			
35	0.42313	0.001165956			
36	0.43137	0.001196938			
37	0.44352	0.001252433			
38	0.45882	0.001252433			
39	0.47098	0.001383361			
40	0.48823	0.001453399			
41	0.49607	0.001481901			
42	0.52235	0.001584225			
43	0.52941	0.001616891			
44	0.53882	0.00162977			







Información experimental.

Tipo de textil.	n/dif	Excusión [V].	Salida [m/H2O]	Kg.	C [V].	D [V].	E [V].	F [V].	Periodo T [s].	F [Ha].	Retardo.	d [s].	Tan [s].
Rayón est. 1 (turbulento)	0.50	2.757575758	0.393939394	0.142857143	0.484848485	0.121212121	0.090909091	0.060606061	1.151515152	0.868421053	0.121212121	0.96969697	
Rayón est. 2 (turbulento)	2.00	2.772727273	0.409090909	0.147540984	0.545454545	0.121212121	0.045454545	0.03030303	1.212121212	0.825	0.121212121	0.606060606	
Poliéster 1 (turbulento)	0.50	2.924242424	0.460606061	0.157512953	0.424242424	0.090909091	0.060606061	0.015151515	0.818181818	1.222222222	0.242424242	1.606060606	
Poliéster 2 (turbulento)	2.00	2.909090909	0.484848485	0.166666667	0.33030303	0.151515152	0.075757576	0.060606061	1.090909091	0.916666667	0.363636364	0.848484848	
Rayón 1 (turbulento)	0.50	2.787878788	0.263636364	0.094565217	0.454545455	0.060606061	0.03030303	0.003030303	1.348484848	0.741573034	0.121212121	0.590909091	
Rayón 2 (turbulento)	2.00	2.96969697	0.266666667	0.089795918	0.212121212	0.060606061	0.03030303	0.015151515	0.909090909		1.1	0.545454545	1.939393939
Tej. afelpado 1 (turbulento)	0.50	3.818181818	0.5	0.130952381	0.333333333	0.060606061	0.045454545	0.03030303	0.787878788	1.269230769	0.272727273	1.666666667	
Tej. afelpado 2 (turbulento)	2.00	2.954545455	0.327272727	0.110769231	0.303030303	0.090909091	0.045454545	0.03030303	0.96969697	1.03125	0.606060606	2.060606061	
Tej. afelpado 3 (turbulento)	0.50	2.96969697	0.312121212	0.105102041	0.303030303	0.090909091	0.03030303	0.015151515	1.03030303	0.970588235	0.303030303	1.696969697	
Acrílico 1 (turbulento)	0.50	2.954545455	0.224242424	0.075897436	0.303030303	0.03030303	0.257575758	0.021212121	1.090909091	0.916666667	0.227272727	1.242424242	
Acrílico 2 (turbulento)	2.00	2.954545455	0.23030303	0.077948718	0.181818182	0.060606061	0.024242424	0.003030303	0.96969697	1.03125	0.363636364	2.060606061	
Poliéster 1 (máximo P)	0.50	4.848484848	0.987878788	0.20375	0.515151515	0.242424242	0.121212121	0.060606061	0.696969697	1.434782609	0.121212121	0.515151515	
Poliéster 2 (máximo P)	2.00	4.909090909	1.036363636	0.211111111	0.363636364	0.242424242	0.090909091	0.03030303	0.606060606	1.65	0.363636364	2.787878788	
Rayón est. 1 (máximo P)	2.00	5.515151515	1.042424242	0.189010989	0.363636364	0.181818182	0.090909091	0.060606061	0.606060606	1.65	0.303030303	2.181818182	
Rayón est. 2 (máximo P)	0.50	5.454545455	0.981818182	0.18	0.363636364	0.181818182	0.060606061	0.042424242	0.575757576	1.736842105	0.46969697	2.848484848	
Tej. afelpado 1 (máximo P)	2.00	7.151515152	1.024242424	0.143220339	0.181818182	0.060606061	0.03030303	0.006060606	0.484848485	2.0625	0.545454545	3.878787879	
Tej. afelpado 2 (máximo P)	0.50	7.212121212	1.012121212	0.140336134	0.181818182	0.090909091	0.03030303	0.006060606	0.545454545	1.833333333	0.181818182	2.848484848	
Rayón 1 (máximo P)	1.00	8.181818182	1.006060606	0.122962963						0	0.363636364		1.545454545
Acrílico 1 (máximo P)	1.00	9.393939394	0.987878788	0.10516129						0	0.181818182		7.03030303
Acrílico 2 (máximo P)	2.00	5.515151515	0.515151515	0.093406593	0.181818182	0.060606061	0.045454545	0.03030303	0.606060606	1.65	0.424242424	2.424242424	
Rayón 1 (1/2 m/H2O)	0.50	4.848484848	0.496969697	0.1025	0.272727273	0.090909091	0.045454545	0.015151515	0.696969697	1.434782609	0.151515152	1.484848485	
Rayón 2 (1/2 m/H2O)	2.00	4.787878788	0.506060606	0.105696203	0.212121212	0.060606061	0.03030303	0.003030303	0.606060606	1.65	0.484848485	2.666666667	
Rayón est. 1 (1/2 m/H2O)	0.50	3.287878788	0.5	0.152073733	0.348484848	0.121212121	0.03030303	0.021212121	0.909090909	1.1	0.363636364	1.818181818	
Rayón e A. 2 (1/2 m/H2O)	2.00	3.272727273	0.493939394	0.150925926	0.393939394	0.121212121	0.060606061	0.03030303	0.848484848	1.178571429	0.484848485	2.060606061	
Poliéster 1 (1/2 m/H2O)	2.00	3.03030303	0.515151515	0.17	0.393939394	0.090909091	0.03030303	0.027272727	0.96969697	1.03125	0.448484848	1.696969697	
Poliéster 2 (1/2 m/H2O)	0.50	3	0.484848485	0.161616162	0.424242424	0.154545455	0.03030303	0.024242424	1.090909091	0.916666667	0.090909091	0.787878788	
Tej. afelpado 1 (1/2 m/H2O)	2.00	4.151515152	0.509090909	0.122627737	0.242424242	0.121212121	0.045454545	0.03030303	0.606060606	1.65	0.727272727	2.181818182	
Acrílico 1 (1/2 m/H2O)	0.50	5.454545455	0.490909091	0.09	0.151515152	0.090909091	0.03030303	0.024242424	0.606060606	1.65	0.484848485	2.878787879	
Acrílico 2 (1/2 m/H2O)	2.00	2.878787879	0.218181818	0.075789474	0.484848485	0.090909091	0.03030303	0.003030303	0.606060606	1.65	0.121212121	0.606060606	

Apéndice E. Sistema operativo para el sistema HC11.

A	equ	\$41	
B	equ	\$42	
C	equ	\$43	
D	equ	\$44	
E	equ	\$45	
F	equ	\$46	
G	equ	\$47	
H	equ	\$48	
I	equ	\$49	
J	equ	\$4a	
K	equ	\$4b	
L	equ	\$4c	
M	equ	\$4d	
N	equ	\$4e	
O	equ	\$4f	
P	equ	\$50	
Q	equ	\$51	
R	equ	\$52	
S	equ	\$53	
T	equ	\$54	
U	equ	\$55	
V	equ	\$56	
W	equ	\$57	
X	equ	\$58	
Y	equ	\$59	
Z	equ	\$5a	
a	equ	\$61	
b	equ	\$62	
c	equ	\$63	
d	equ	\$64	
e	equ	\$65	
f	equ	\$66	
g	equ	\$67	
h	equ	\$68	
i	equ	\$69	
j	equ	\$6a	
k	equ	\$6b	
l	equ	\$6c	
m	equ	\$6d	
n	equ	\$6e	
o	equ	\$6f	
p	equ	\$70	
q	equ	\$71	
r	equ	\$72	
s	equ	\$73	
t	equ	\$74	
u	equ	\$75	
v	equ	\$76	
w	equ	\$77	
x	equ	\$78	
y	equ	\$79	
z	equ	\$7a	
ESC	equ	\$1b	
EXPSIGN	EQU	0	EXPONENT SIGN (0=+, FF=-).
PWR10EXP	EQU	1	POWER 10 EXPONENT.

* REGISTROS ESPECIALES DEL HC11 E1
 DDRAM equ \$6001
 IRRAM equ \$6000

HPRIO	equ	\$103c
TMSK2	equ	\$1024
TFLG2	equ	\$1025
PACTL	equ	\$1026
ADCTL	equ	\$1030
ADR1	equ	\$1031
ADR2	equ	\$1032
ADR3	equ	\$1033
ADR4	equ	\$1034
OPTION	equ	\$1039
INIT	equ	\$103d
CONFIG	equ	\$103f
PORTD	equ	\$1008
DDR0	equ	\$1009
SPCR	equ	\$1028
BAUD	equ	\$102B
SCCR1	equ	\$102C
SCCR2	equ	\$102D
SCSR2	equ	\$102E
SCDR	equ	\$102F
TDR	equ	\$102F
RDR	equ	\$1035
BPROT	equ	\$1000
SYSREG	equ	\$1040
SYSREGEN	equ	\$1040

* ZONA DE VECTORES, MODO SINGLE(PSEUDO VECTORES) Y EXTENDIDO

*CMFVECTOR	equ	\$fc	;\$fMc	\$fb
*COPVECTOR	equ	\$fb	;\$fM8	\$f8
*ILLVECTOR	equ	\$f8	;\$fM6	\$f5
*SWIVECTOR	equ	\$f5	;\$fM4	\$f2
*XIRQVECTOR	equ	\$f2	;\$fM2	\$ej
*IRQVECTOR	equ	\$ec	;\$fM0	\$ec
*RTIVECTOR	equ	\$e9	;\$fFe	\$e9
*TIC1VECTOR	equ	\$e6	;\$fFec	\$e6
*TIC2VECTOR	equ	\$e3	;\$fFca	\$e3
*TIC3VECTOR	equ	\$e0	;\$fFcb	\$e0
*TOC1VECTOR	equ	\$dd	;\$fF6	\$dd
*TOC2VECTOR	equ	\$da	;\$fF4	\$da
*TOC3VECTOR	equ	\$d7	;\$fF2	\$d7
*TOC4VECTOR	equ	\$d4	;\$fF0	\$d4
*TOIVECTOR	equ	\$d1	;\$fFd	\$D1
*PAOVECTOR	equ	\$ce	;\$fFd	\$ce
*PAIVECTOR	equ	\$cb	;\$fFd8	\$cb
*SPIVECTOR	equ	\$c8	;\$fFd8	\$c8
*SCIVECTOR	equ	\$c5	;\$fFd6	\$c5

*CMFVECTOR	equ	\$fMc
*COPVECTOR	equ	\$fM8
*ILLVECTOR	equ	\$fM6
*SWIVECTOR	equ	\$fM4
*XIRQVECTOR	equ	\$fM2
*IRQVECTOR	equ	\$fM0
*RTIVECTOR	equ	\$fFec
*TIC1VECTOR	equ	\$fFec
*TIC2VECTOR	equ	\$fFca
*TIC3VECTOR	equ	\$fFcb
*TOC1VECTOR	equ	\$fF6
*TOC2VECTOR	equ	\$fF4
*TOC3VECTOR	equ	\$fF2
*TOC4VECTOR	equ	\$fF0
*TOIVECTOR	equ	\$fFd
*PAOVECTOR	equ	\$fFd
*PAIVECTOR	equ	\$fFd8
*SPIVECTOR	equ	\$fFd8
*SCIVECTOR	equ	\$fFd6

EOB	equ	\$00AA
-----	-----	--------

* MACROS

ERRORES PARA LAS FUNCIONES EN PUNTO FLOTANTE

```

FLTMTERR      EQU    1      /* floating point format error in ASCFLT */
OVFFERR      EQU    2      /* floating point overflow error */
UNFERR       EQU    3      /* floating point underflow error */
DIV0ERR      EQU    4      /* division by 0 error */
TOLGSMERR    EQU    5      /* number too large or small to convert to int. */
NSQRTERR     EQU    6      /* tried to take the square root of negative # */
TAN90ERR     EQU    7      /* TANGent of 90 degrees attempted */

```

* MASCARAS PARA EVENTOS BASE

```

XIRQFLAG     equ     $01      ;Mascaras para eventos base del sistema
NOTXIRQFLAG  equ     $f      ;
IROFLAG      equ     $02      ;parte baja del registro de 16 bits,
NOTIROFLAG   equ     $fd     ;
RTIFLAG      equ     $04      ;se hara AND con el registro B;
NOTRTIFLAG   equ     $fb     ;
TIC1FLAG     equ     $08
TIC2FLAG     equ     $10
TIC3FLAG     equ     $20
TOC1FLAG     equ     $40
TOC2FLAG     equ     $80
TOC3FLAG     equ     $01      ;parte alta del registro de 16 bits.se
TOC4FLAG     equ     $02      ;hara AND con el registro A.
TOC5FLAG     equ     $04
TOIFLAG      equ     $08
PAOFLAG      equ     $10
PAIFLAG      equ     $20
SPIFLAG      equ     $40
SCIFLAG      equ     $80
NOTSCIFLAG   equ     $7f

XIRQFLAG16   equ     $0001    ;Mascaras para eventos base del sistema
IROFLAG16    equ     $0002    ;parte baja del registro de 16 bits,
RTIFLAG16    equ     $0004    ;se hara AND con el registro B;
TIC1FLAG16   equ     $0008
TIC2FLAG16   equ     $0010
TIC3FLAG16   equ     $0020
TOC1FLAG16   equ     $0040
TOC2FLAG16   equ     $0080
TOC3FLAG16   equ     $0100    ;parte alta del registro de 16 bits.se
TOC4FLAG16   equ     $0200    ;hara AND con el registro A.
TOC5FLAG16   equ     $0400
TOIFLAG16    equ     $0800
PAOFLAG16    equ     $1000
PAIFLAG16    equ     $2000
SPIFLAG16    equ     $4000
SCIFLAG16    equ     $8000

```

* MENSAJES O EVENTOS BASE (INHERENTES AL SISTEMA)

```

EV_XIRQ      equ     $01      ;interrupcion pin XIRQ
EV_IRQ       equ     $02      ;interrupcion pin IRQ
EV_RTI       equ     $03      ;interrupcion RTI
EV_TIC1      equ     $04      ;interrupcion comparador 1
EV_TIC2      equ     $05      ;interrupcion comparador 2
EV_TIC3      equ     $06      ;interrupcion comparador 3
EV_TOC1      equ     $07      ;interrupcion comparador 1
EV_TOC2      equ     $08      ;interrupcion comparador 2
EV_TOC3      equ     $09      ;interrupcion comparador 3
EV_TOC4      equ     $0a      ;***** 4
EV_TOC5      equ     $0b      ;***** 5

```

EV_TOI	equ	\$0c	;"*****" sobreflujo
EV_PAO	equ	\$0d	;"* sobreflujo acumulador pulsos
EV_PAI	equ	\$0e	;"* acumulador de pulsos
EV_SPI	equ	\$0f	;"* comunicaciones SPI
EV_SCI	equ	\$10	;"* comunicaciones SCI

* SERVICIOS UNIVERSALES DEFINIDOS PARA ESTE SISTEMA

MSG	equ	\$00	;"mensaje
SWI	equ	\$01	;"servicios de SWI
SYS	equ	\$02	;"servicios de systema
STODEV	equ	\$03	;"servicios de disco
SCI	equ	\$04	;"servicios de comunicacion
RTI	equ	\$05	;"servicios de RTI
ADC	equ	\$06	;"servicios de conversion A/D1
LCD	equ	\$07	;"servicios de LCD
RTC	equ	\$08	;"servicios de reloj

* MENSAJES SECUNDARIOS (LOGICOS)

WM_INIT	equ	\$11	
WM_CREATE	equ	\$12	;"mensaje para crear ventana
WM_QUIT	equ	\$13	;"mensaje para cerrar ventana
WM_UP	equ	\$14	;"mensaje para subir en la ventana
WM_DOWN	equ	\$15	;"mensaje para continuar en la ventana
WM_PAINT	equ	\$16	;"mensaje para volver a pintar la ventana
WM_MAXIMIZE	equ	\$17	;"mensaje para volver a foreground
WM_MINIMIZE	equ	\$18	;"mensaje para enviar a background
SYS_INSTALL	equ	\$19	;"mensaje para instalar un driver
SYS_EV_NOMEM	equ	\$1a	;"mensaje: no hay memoria
SYS_EV_ILDESCR	equ	\$1b	;"mensaje: descriptor ilegal
CM_CREATE	equ	\$1c	;"mensaje para crear
CM_KILL	equ	\$1d	;"mensaje para destruir un shell
CM_NOTASK	equ	\$1e	;"mensaje para cambiar tarea
CM_RUN	equ	\$1f	;"mensaje para ejecutar algo
CM_STARTUP	equ	\$20	;"ejecutar rutinas de inicio
CM_EXIT	equ	\$21	
EV_RXCHAR	equ	\$22	;"mensaje: se recibio caracter
EV_TXCHAR	equ	\$23	;"mensaje: se termino de enviar caracter
EV_IDLELINE	equ	\$25	;"evento: Idle line detectado
EV_OVERRUN	equ	\$25	;"evento: sobreflujo de datos en SCI
EV_BTC	equ	\$26	;"evento: transmision de bloque completa
EV_BRC	equ	\$27	;"evento: recepcion de bloque completa
EV_RTCALARM	equ	\$28	;"mensaje: alarma del timer
EV_RTCPI	equ	\$29	;"mensaje: interrupcion periodica
EV_RTCUC	equ	\$2a	;"mensaje: termino ciclo de actualizacion
CM_INIT	equ	\$2b	;"mensaje: inicializar
CM_RESUME	equ	\$2c	;"mensaje: resumir
EV_ADCONV	equ	\$2d	;"conversion A/D terminada
EV_NOTFOUND	equ	\$2e	;"mensaje: algo no se encontro
EV_UNABLE	equ	\$2f	;"mensaje: incapaz de algo
EV_NOMEM	equ	\$30	;"mensaje: no hay memoria
EV_CMFAIL	equ	\$31	;"mensaje: no se ejecuto bien el comando

* FUNCIONES DE LCD

LCD_outtext	equ	\$01	
LCD_clear	equ	\$02	
LCD_esperar	equ	\$03	
LCD_wm_create	equ	\$04	
LCD_wm_quit	equ	\$05	
LCD_wm_up	equ	\$06	
LCD_wm_down	equ	\$07	
LCD_wm_paint	equ	\$08	
LCD_wm_minim	equ	\$09	
LCD_wm_maxim	equ	\$0a	
LCD_wm_dispatch	equ	\$0b	

```

LCD_display      equ      $0c

* FUNCIONES DE SYSTEMA
SYS_cm_run       equ      $01
SYS_cm_kill     equ      $02
SYS_cm_install  equ      $03
SYS_cm_xgtaak   equ      $04
SYS_cm_sleep    equ      $05
SYS_setsvect    equ      $06      ;int setvect(charNumServ,lpAddr)
SYS_getsvvect   equ      $07      ;int* getvect(ucharNumServ)
SYS_getpc       equ      $08
SYS_malloc      equ      $09
SYS_mfree       equ      $0a
SYS_getURT      equ      $0b
SYS_setURT      equ      $0c
SYS_sysmem      equ      $0d      ;int sysmem(lpStructMem)
SYS_load        equ      $0e      ;int load(lpstrName)
SYS_transfer    equ      $0f      ;void transfer(lpAddr)
SYS_enablenotif equ      $10      ;int enablenotif(intMascara,intHandler)
SYS_getsvvector equ      $11      ;int getsvvector(intMascara)
SYS_setsvvector equ      $12      ;int setsvvector(intMascara,int URT)
SYS_dispatch    equ      $13
SYS_IsForegnd   equ      $14      ;BOOL IsForegnd(intHandler)
SYS_terminate   equ      $15      ;BOOL terminate(void)
SYS_unload      equ      $16      ;BOOL Unload(lp Entry)

* Estructura MSG
.msg            equ      $0
.wparam        equ      $2
.lparam        equ      $4
.iparam        equ      $6

* FUNCIONES DE RTI
RTI_setRTIrate  equ      $01      ;setRTIrate(charRate)
RTI_enablenotif equ      $02      ;enablenotif(intURT)
RTI_ev_RTI      equ      $03

* FUNCIONES DE ADC
ADC_setmode     equ      $01      ;int BOOL setmode(char Mode)
ADC_enablenotif equ      $02      ;int BOOL enablenotif(int URT)
ADC_IsReady     equ      $03      ;int BOOL IsReady()

* FUNCIONES DE RELOJ
RTC_settime     equ      $01      ;settime(lpStructTime*)
RTC_gettime     equ      $02      ;gettime(lpStructTime*)
RTC_setalarm    equ      $03
RTC_getalarm    equ      $04
RTC_clearalarm  equ      $05
RTC_setmode     equ      $06
RTC_getRTCaddr  equ      $07
RTC_getPIR      equ      $08
RTC_setPIR      equ      $09
RTC_enablenotif equ      $0a
RTC_setvect     equ      $0b
RTC_getvect     equ      $0c

RTC.UCIRT       equ      $0e      ;desplazamientos sobre la direccion del reloj
RTC.AIURT       equ      $10      ;donde se encuentran estas variables.
RTC.PIURT       equ      $12

RTC.segundo     equ      $0
RTC.minuto      equ      $2      ;desplazamientos para leer la fecha
RTC.hora        equ      $4
RTC.dia         equ      $6
RTC.fecha       equ      $7
RTC.mes         equ      $8
RTC.ano         equ      $9

```

```

UCFLAG      equ      $01      ;mascara para ciclo de refresco
AIFLAG      equ      $02      ;mascara para alarma
PIFLAG      equ      $04      ;mascara para interrupcion periodica
NOTUCFLAG   equ      $fc      ;mascara para desactivar refresco
NOTAIFLAG   equ      $fd      ;mascara para desactivar alarma
NOTPIFLAG   equ      $fb      ;mascara para desactivar PI

* FUNCIONES DE SWI
SWI_setvect equ      $01
SWI_getvect equ      $02

* MACROS
TRUE        equ      $ff
FALSE       equ      $00
BACKGROUNDINACTIVO equ  $35
BACKGROUNDACTIVO   equ  $3f
FOREGNDINACTIVO    equ  $f5
FOREGNDACTIVO      equ  $ff
ACTIVO              equ  $0f
INACTIVO            equ  $05
SUSPENDIDO         equ  $0a
FOREGND            equ  $f0
BACKGROUND         equ  $50

* FUNCIONES DE DISPOSITIVO DE ALAMCENAMIENTO (ROM)
STODEV_getPDT    equ      $01
STODEV_search    equ      $02
STODEV_load      equ      $03

* FUNCIONES DE SCI
SCI_CloseComm    equ      $01
SCI_EnableNotif  equ      $02
SCI_GetError     equ      $03
SCI_GetEvMask    equ      $04
SCI_GetState     equ      $05
SCI_OpenComm     equ      $06
SCI_ReadComm     equ      $07
SCI_SetEvMask    equ      $08
SCI_SetState     equ      $09
SCI_XmitChar     equ      $0a
SCI_WriteComm    equ      $0b

BTC_FLAG        equ      $01      ;bandera de transmision de bloque completa
NOTBTC_FLAG     equ      $fc
BRC_FLAG        equ      $02      ;bandera de recepcion de bloque completa
NOTBRC_FLAG     equ      $fd
ERR_FLAG        equ      $04
NOTERR_FLAG     equ      $fb
OR_FLAG         equ      $08
NOTOR_FLAG      equ      $f7
IDLE_FLAG       equ      $10
RDRE_FLAG       equ      $20
TC_FLAG         equ      $40
TDRE_FLAG       equ      $80

IE_BADID        equ      $1
IE_BAUDRATE     equ      $2
IE_BYTESIZE     equ      $3
IE_DEFAULT      equ      $4
IE_HARDWARE     equ      $5
IE_MEMORY       equ      $6
IE_OPEN         equ      $7
IE_NOOPEN       equ      $8

CE_FRAME        equ      $1
CE_NOISE        equ      $2
CE_OVERRUN      equ      $4
CE_MODE         equ      $8

```

```
EVT_CHAR    equ    $3
EOT_CHAR    equ    $4
```

* Estructura DCB(desplazamientos)

```
.Id          equ    $0
.Baud       equ    $1
.Size      equ    $2
.EvtChar   equ    $3
.EofChar   equ    $4
.Verify    equ    $5
.Timeout   equ    $6
```

* FUNCIONES DE SISTEMA OPERATIVO *

```
SO_showtime equ    $01
```

* ESTRUCTURA TASK (descriptor de tarea)

```
TASK.Nombre equ    $0
TASK.Entry  equ    $10
TASK.Stack  equ    $12
TASK.URT    equ    $14
TASK.Ruptura equ    $16
TASK.Modos  equ    $18
```

* ESTRUCTURA CREATE (descriptor de proceso)

```
CREATE.Entry equ    $0
CREATE.Stack equ    $2
CREATE.URT   equ    $4
CREATE.Param equ    $6
CREATE.Nombre equ    $8
```

* ESTRUCTURA FDT (File Descriptor Table)

```
Descriptor  equ    $0      :fcc    "Reloc"
CompAddr    equ    $5      :fdb    #.msg
Entrypoint  equ    $7      :fdb    #.handler
Stacksize   equ    $9      :fdb    #S0000
ProgSize    equ    $b      :fdb    #.fin-#.DRV
RelocAddr   equ    $d      :fdb    #.
```

* ESTRUCTURA LCD DATA (Datos para desplegar en LCD)

```
*init.hwnd      fdb    0
*init.padre     fdb    0
*init.handler   fdb    0
*init.posic     fdb    0
*init.puntero   fdb    0
*init.buffer    fcc    "1234567890123456"
*               fcc    "1234567890123456"
```

```
.IRRAM        equ    $0
.DRRAM        equ    $1
NUMCARACT     equ    $2
CARACTP.RENG  equ    $6
```

```
.hwnd         equ    $0
.handler      equ    $2
.padre        equ    $4
.posic        equ    $6
.puntero      equ    $7
.buffer       equ    $9
```

* ESTRUCTURA TIME (para leer la hora)

```
.segundos     equ    $0
.minutos      equ    $1
.hora         equ    $2
.dia          equ    $3
.fecha        equ    $4
.mes          equ    $5
```

```

.ano          equ      $6

CM_INICIAR   equ      $65
CM_PAUSA     equ      $66

ORG $e000

BIOS
    sci
    ldaa      #$01
    staa     INIT
    tps      #57f          ;activa la instruccion stop
    anda
    tap
    lds      #$f00
    ldd      #SWI.handler
    std      SWIVECTOR
    ldd      #SYS.NIRQhandler ;todos los vectores apuntan a
    std      NIRQVECTOR      ;rti, supone que despues de re
    ldd      #SYS.IRQhandler
    std      IRQVECTOR       ;set todo está inhabilitado.
    ldd      #SYS.COPhandler
    std      COPVECTOR
    ldd      #SYS.CMFhandler
    std      CMFVECTOR
    ldd      #SYS.RTIhandler
    std      RTIVECTOR
    ldd      #SYS.SCIhandler
    std      SCIVECTOR
    ldd      #SYS.TOIhandler
    std      TOIVECTOR
    ldd      #SYS.ILLhandler
    std      ILLVECTOR       ;deshabilita comunicaciones seriales
    ldaa     #$0
    staa     SCCR2
    staa     SCCR1

    ldaa     #Se4          ;toi tiene la m s alta prioridad
    staa     HPRIO
    ldaa     #0
    staa     BPROT

clrmem
clrmem.c1 clr    ldx      #$100
0,x
inx
cmpx           #$200
beq           clrmem.cont1
bra           clrmem.c1

clrmem.cont1    ldaa     #5ff
staa         $110      ;protege registros especiales del HC11
staa         $1ff

    ldd      #SWI.handler ;desplazamiento sobre el objeto SWI
    std      SWIVECTOR   ;donde comienza SWI.handler
    ldx     #f000        ;esta zona transfiere el objeto SWI
    ldy     #$200       ;a la parte baja de memoria.
    *init.installSWI
    cmpx    #
    beq    init.vectsys
    ldaa   0,x
    staa  0,y
    inx
    iny
    bra   init.installSWI ;termina movimiento de memoria.

inicializacion  ldaa     #SYS          ;se coloca el vector de SYS
                psha
                ldx     #SYS.handler

```

```

pshx
tax
stx      SYS.wparam
ldaa    #SWI
ldab    #SWI_setvect      ;setvect(uchar Num,lp Handler)
swi
puls
pula
ldaa    #STODEV          ;se coloca el vector de STODEV
paha
ldx     #STODEV.handler
pshx
tax
stx      SYS.wparam
ldaa    #SWI
ldab    #SWI_setvect
swi
puls
pula
ldd     #CM_INIT        ;inicializa el sistema
std     SYS.msg
ldaa    #SYS
ldab    #MSG
swi
ldaa    #RTI            ;coloca el vector de RTI
paha
ldx     #RTI.handler
pshx
tax
stx      SYS.wparam
ldaa    #SWI
ldab    #SWI_setvect
swi
puls
pula
ldaa    #ADC            ;coloca el vector de ADC
paha
ldx     #ADC.handler
pshx
tax
stx      SYS.wparam
ldaa    #SWI
ldab    #SWI_setvect
swi
puls
pula
ldaa    #SCI            ;coloca el vector de SCI
paha
ldx     #SCI.handler
pshx
tax
stx      SYS.wparam
ldaa    #SWI
ldab    #SWI_setvect
swi
puls
pula

```

***** Verifica la existencia de pantalla LCD *****

```

ldaa    #S06
staa    IRRAM
jsr     init.sleep
ldaa    #S0c
staa    IRRAM      ;configura pantalla LCD
jsr     init.sleep
ldaa    #S3E
staa    IRRAM
jsr     init.sleep
ldaa    #S40

```



```

      staa      IRRAM
      jsr      init.sleep
      ldaa     #1
      staa     IRRAM
      jsr      init.sleep
      ldaa     #S31
              DDRAM
init.ciclo1
      staa     IRRAM
      clrb
      incb
      ldaa     IRRAM
      anda     #$80
      beq     init.cont1
      cmpb    #SIT
      blo     init.ciclo1
      ldx     #init.error1
      jsr     display
;no encontro pantalla LCD
      stop
      ldd     #SYS_INSTALL      ;mensaje para instalar rutinas de lcd
      std
      ldd     SYS.msg
      ldaa    #rutinaslcd      ;puntero a la cadena con nombre del
      ldaa    SYS.wparam      ;servicio a instalar
      staa    #LCD            ;numero de servicio que ocupara
      ldaa    SYS.lparam1
      ldaa    #SYS
      ldab    #MSG
      swi
      ldd     #WM_CREATE      ;se manda en tipo de mensaje
      std     SYS.msg
      ldx     #init.hwnd      ;direccion de los datos publicos
      stx     SYS.wparam      ;parametro tipo word
      ldx     #0              ;se envia el handler de esta ventana
      stx     SYS.lparam1     ;que es el BIOS(inicializacion).
      ldaa    #LCD            ;servicios de LCD
      ldab    #MSG           ;interprete de mensajes
      swi
      ldx     #init.hwnd
      pahx
      ldx     #lpstrTitulo    ;direccion del buffer
      pshx
      tax
      stx     SYS.wparam      ;servicios de LCD
      ldaa    #LCD
      ldab    #LCD_outtext   ;funcion LCD.outtext
      swi
      pulx
      pulx
      ldaa    #SYS
      ldab    #SYS_sleep
      swi
      jmp     init.RTC

```

***** Programa para revisar RTI *****

```

initprucbarti
      jmp     init.RTC
      ldaa    #LCD
      ldab    #LCD_clear
      swi
      ldaa    #LCD
      ldab    #LCD_outtext
      ldx     #init.mensajerti
      stx     init.buffer
      swi
      ldaa    PACTL
      anda    #S0
      oraa    #S03
      staa    PACTL
      ldaa    #S0            ;se crea una variable temporal
      psha    ;variable de cuenta

```

```

psha                                ;variable bandera
tax
ldaa                                ;activa RTI
staa                                #S40
ldd                                TMSK2
std                                #RTITOI,prueba
ldy                                RTIVECTOR
rti.prueba1                        #S21
jsr                                rti.prueba
ldaa                                PACTL
anda                                #S0
oraa                                #S02
staa                                PACTL
ldy                                #S11
rti.prueba2                        rti.prueba
jsr                                PACTL
ldaa                                #S0
anda                                #S01
oraa                                PACTL
staa                                PACTL
ldy                                #S9
rti.prueba3                        rti.prueba
jsr                                PACTL
ldaa                                #S0
anda                                PACTL
staa                                #S4
ldy                                #S4
rti.prueba4                        rti.prueba
jsr                                #RTI.handler
ldd                                RTIVECTOR
std                                TMSK2
ldaa                                #Sf
anda                                TMSK2
staa                                #initpruebatoi
jmp                                initpruebatoi

rti.prueba                          inc        numprueba
ldaa                                #S40
staa                                init.posic
ldd                                #init.mensaje3
std                                init.buffer
ldaa                                #LCD
ldab                                #LCD_outtext
swi
clr                                0,x
clr                                1,x
ldaa                                TFLG2
oraa                                #S40
staa                                TFLG2
ldaa                                #Sf
;las instrucciones decb, beq
;toaman ff veces 5 ciclos
;ciclos bra ff veces 3 ciclos
;que son 2037 ciclos, para
;16.38 ms se necesitan hacer
;16.08 veces (aprox 11h). Se
;genera por lo menos una int.
pruebarti.c1                        decb
beq                                pruebarti.cont1
pruebarti.cont1                    bra
dey                                pruebarti.c1
beq                                pruebarti.cont2
bra                                pruebarti.c1
pruebarti.cont2                    sei
ldaa                                0,x
crpsa                                #S1
beq                                pruebarti.fin
pruebarti.error                    ldd                                #init.rtierror
std                                init.buffer
ldaa                                #LCD
ldab                                #LCD_clear
swi
ldaa                                #LCD
ldab                                #LCD_outtext
swi
stop
pruebarti.fin                      ldd                                #init.ok
std                                init.buffer
ldaa                                #S4
staa                                init.posic

```

```

ldaa #LCD
ldab #LCD_outext
swi
ldaa #SYS
ldab #SYS_sleep
swi
rts

```

***** Se carga el driver del reloj*****

```

init.RTC      ldd #lpstrRTCname      ;instala el driver para el reloj
              std SYS.wparam ;en wparam se pasa la direccion
              ldd #SYS_INSTALL ;de la cadena que contiene el
              std SYS.msg    ;nombre del driver
              ldaa #RTC
              staa SYS.lparaml
              ldaa #SYS
              ldab #MSG
              swi
              ldaa #SYS
              ldab #SYS_sleep
              swi
              jmp      init.Systemem

```

***** Verifica la memoria del sistema*****

```

lpstrBloques fcc "Bloques de mem:"
fdb #EOB
totalmem     fcc "000"
fcc "tot "
freemem      fcc "000"
fcc " free"
fdb #EOB

numero      fcb #500
hexascii    tsy
            ldy 2,y      ;limpia area de resultados
            clr 2,y      ;unidades
            clr 1,y      ;decenas
            clr 0,y      ;centenas
            cmpa #0      ;verifica si el dato fuente es cero
            beq hexascii.salida ;si es cero no hace nada
            staa numero
            clra
            clrb

hexascii.cl  addb #50a
            cmpb numero
            bhi hexascii.fin1
            cmpb #54    ;esta convirtiendo un 0xFF
            beq hexascii.fin1
            inc 1,y
            ldaa 1,y    ;verifica se se ha generado acarreo
            cmpa #50a  ;"n centenas.
            bnc hexascii.cl
            inc 0,y    ;incrementa centenas.
            clr 1,y
            bra hexascii.cl

hexascii.fin1 subb #5a
            ldaa numero
            aba

hexascii.salida staa 2,y
            ldaa 2,y
            adda #530
            staa 2,y
            ldaa 1,y
            adda #530
            staa 1,y

```

```

        ldaa    0,y
        adda    #30
        staa   0,y
        rts

init.System    ldx    #init.hwnd
               pshx
               tax
               stx    SYS.wparam
               ldaa   #LCD
               ldab   #LCD_clear
               swi
               ldx    #lpstrBloques
               pshx
               tax
               stx    SYS.wparam
               ldaa   #LCD
               ldab   #LCD_outext
               swi
               pulx
               ldaa   #$0                ;crea estructura mem:char memfree, char totalmem
               psha   ;memfree
               psha   ;totalmem
               tax
               stx    SYS.wparam        ;pasa puntero a estructura mem
               ldaa   #SYS
               ldab   #SYS_systemem
               swi
               pula
               ldx    #totalmem
               pshx
               jar    hexascii
               pulx
               pula
               ldx    #freemem
               pshx
               jar    hexascii
               pulx
               ldx    #init.hwnd
               pshx
               ldx    #totalmem
               pshx
               tax
               stx    SYS.wparam
               ldaa   #16
               staa   init.posic
               ldaa   #LCD
               ldab   #LCD_outext
               swi
               pulx
               pulx
               ldaa   #SYS
               ldab   #SYS_sleep
               swi
               bra    init.so

***** Se carga el interprete del so *****

struct_create  fdb    $0000    ;Entry point
               fdb    $0000    ;Stack
               fdb    $0000    ;URT
               fdb    $0000    ;Puntero a parametros
               fdb    $0000    ;Puntero a nombre

init.so        ldx    #struct_create
               ldd   #SO_handler
               std   CREATE.Entry,X    ;manda el entry point
               ldd   #lpstrSOName

```

```

std      CREATE.Nombre,x ;manda puntero a cadena nombre
pahx
ldx      #128
pahx
tsx
stx      SYS.wparam
ldaa    #SYS
ldab    #SYS_malloc
swi
pulsx
pulsx
ldd      intRegreso
addd    #511
std      CREATE.Stack,x ;manda puntero al stack
ldd      #struct_create
std      SYS.wparam
ldd      #CN_RUN
std      SYS.msg
ldaa    #SYS
ldab    #MSG
swi
ldx      intRegreso
cmpd    #$0
beq     initso.cont1
ldx     #so.errormsg
jsr     display
stop
initso.cont1  ldx     #initso.e2
jsr     display
stop

so.errormsg  fcc     "No se corrio"
fcb     #EOB
initso.e2    fcc     "no hizo cambio"
fcb     #EOB

```

***** Termina inicializacion del HC11 !!!! *****

***** Funciones de init *****

```

init.sleep  ldaa    #SYS
           ldab    #SYS_sleep
           swi
           rts

LCD.display1  ldaa    #LCD
           ldab    #LCD.clear
           swi

display1.ciclo1  ldaa    0,x
           cmpa    #$0
           bnc    display1.cont
           ldab    1,x
           cmpb    #5AA
           bnc    display1.cont
           rts

display1.cont  staa    DDRAM
display1.ciclo2  ldaa    #LCD
           ldab    #LCD.espera
           swi
           inx
           bra    rts
           display1.ciclo1

```

***** Arca de datos de init *****
 * datos publicos de LCD

```

init.hwnd      fdb      0
init.padre     fdb      0
init.handler   fdb      0
init.posic     fcb      0
init.puntero   fdb      0
init.buffer    fcc      "1234567890123456"
               fcc      "1234567890123456"

```

```

lpstrTitulo fcc      "BIOS TMC v. 2.0"
               fcc      "Agosto de 1995"
               fdb      $00aa

mensaje      fcc      "hasta aqui"
               fcb      $0
               fcb      $aa

init.mensaje1 fcc      "Hasta el final"
               fcb      $0
               fcb      $aa

init.error1  fcc      "No hay LCD"
               fcb      $0
               fcb      $aa

init.rtierror fcc      "Error en RTI, system halted"
               fcb      $0
               fcb      $aa

init.toierror fcc      "Error en TOI, system halted"
               fcb      $0
               fcb      $aa

init.mensajerti fcc      "Verificando RTI"
               fcb      $0
               fcb      $aa

init.mensajetoi fcc      "Verificando TOI"
               fcb      $0
               fcb      $aa

init.mensaje3 fcc      "Prueba "
numprueba    fcb      $30
               fcc      ": .... "
               fcb      $0
               fcb      $aa

init.falla   fcc      "Falla"
               fcb      $0
               fcb      $aa

init.ok      fcc      "Ok. "
               fdb      #EOB

rutinaslcd   fcc      "LCD.DRV"
               fdb      #EOB

lpstrSOName  fcc      "SO.EXE"
               fdb      #EOB

lpstrSCIName fcc      "SCI.DRV"
               fdb      #EOB

```

```

***** DISCO IMAGINARIO *****
ORG      $e600

```

```

ROOTDIR     fcc      "RTC.DRV"
               fdb      #EOB
               fdb      #RTC.DRV
               fcc      "STODEV.DRV"
               fdb      #EOB
               fdb      #STODEV.DRV"
               fcc      "LCD.DRV"
               fdb      #EOB
               fdb      #LCD.DRV
               fcc      "SO.EXE"
               fdb      #EOB

```

```

fdb      #SO.EXE
fcc      "TESIS.ENE"
fdb      #EOB
fdb      #TESIS.ENE
fdb      $0000

.....   RTC .....

RTC.DRV      fcc      "RTC.DRV"
fdb      $00aa
           fcc      "reloc"
RTC.Descriptor      fdb      #RTC.DRV
RTC.CompAddr      fdb      #RTC.handler-#RTC.Descriptor
RTC.EntryPoint      fdb      $0000
RTC.Stacksize      fdb      #RTC.fin-#RTC.Descriptor
RTC.ProgramSize      fdb      #RTC.default+1
RTC.RelocAddr      fdb      #RTCa1+1
           fdb      #RTCa2+1
           fdb      #RTCa3+1
           fdb      #RTCa4+1
           fdb      #RTCa5+1
           fdb      #RTCa6+1
           fdb      #RTCa7+1
           fdb      #RTCa8+1
           fdb      #RTCa9+1
           fdb      #RTCa+1
           fdb      #RTCab+1
           fdb      #RTCa+1
           fdb      #RTCad+1
           fdb      #RTCa+1
           fdb      #RTCa+1
           fdb      #RTCa10+1
           fdb      #RTCa11+2
           fdb      #RTCa12+1
           fdb      #RTCa13+1
           fdb      #RTCa15+1
           fdb      #RTCa16+1
           fdb      #RTC.readtime+2
           fdb      #pruebartc+1
           fdb      #formatex+2
           fdb      #RTCinit.cont2+1
           fdb      #RTCinit.cont3+1
           fdb      #pruebartc.cont1+1
           fdb      #pruebartc.fin+1
           fdb      #RTC.getPIR+1
           fdb      #RTC.ev_IRQ+1
           fdb      #RTC.defaultmsg+1
           fdb      #RTC.default+1
           fdb      $0
           fdb      $55aa

RTC.FDEM      fdb

RTC.tabla      fdb      #RTC.settime      :funcion RTC_settime(lpStructime)
RTCi2          fdb      #RTC.gettime      :funcion  RTC_gettime(lpStructime)
RTCi3          fdb      #RTC.setalarm     :funcion  RTC_setalarm
RTCi4          fdb      #RTC.getalarm     :funcion  RTC_getalarm
RTCi5          fdb      #RTC.clearalarm   :funcion  RTC_clearalarm
RTCi6          fdb      #RTC.setmode      :funcion  RTC_setmode
RTCi7          fdb      #RTC.getRTCAddr   :funcion  RTC_getRTCAddr
RTCi8          fdb      #RTC.getPIR      :funcion  RTC_getPIR
RTCi9          fdb      #RTC.setPIR      :funcion  RTC_setPIR
RTCiA          fdb      #RTC.enablenotif  :funcion  RTC_enablenotif
RTCiB          fdb      #RTC.setvect     :funcion  RTC_setvect
RTCiC          fdb      #RTC.getvect     :funcion  RTC_getvect

RTC.intAddress      fdb      $A000
RTC.intRegreso      fdb      $0

```

```

*StructTime{
*   ucharSegundos
*   ucharMinutos
*   ucharHora
*   ucharDia
*   ucharFecha
*   ucharMes
*   ucharAco}

RTC.Structime      fcb      $0      ;Segundos
                  fcb      $0      ;Minutos
                  fcb      $0      ;Hora
                  fcb      $0      ;Dia
                  fcb      $0      ;Fecha
                  fcb      $0      ;Mes
                  fcb      $0      ;Aco

RTC.msgdefault     fec      "RTC msg default"
                  fdb      #EOB
RTCdef.mensaje     fec      "RTC.default"
                  fdb      #EOB
msgenablePI       fec      "EnablePI"
                  fdb      #EOB
msgenableAI       fec      "EnableAI"
                  fdb      #EOB
RTC.error         fec      "Fallo de RTC"
                  fdb      #EOB

RTC.handler        psha
                  tay
RTCcal            idx      #RTC.tabla
                  clr      0,y
                  cmpb    0,y
                  beq     RTC.casemsg
RTC.case          inc      0,y
                  cmpb    0,y
                  bne     RTC.Isdefault
                  pula    0,x
                  idx     0,x
                  jar
                  rts
RTC.Isdefault     cmpb    #RTC_getvect
                  bhi     RTC.default
                  inx
                  inx
RTC.default       bra     RTC.case
                  idx     #RTCdef.mensaje
                  jar
                  stop
                  rts

RTC.casemsg       pula    SYS.msg
                  ldd     #CM_INIT
                  cpd     RTC.msg1
                  bnc    RTC.init
                  bsr
                  rts
RTC.msg1          cpd     #EV_IRQ
                  bnc    RTC.defaultmsg
                  bsr
                  rts
RTC.defaultmsg    idx     #RTC.msgdefault
                  jsr
                  stop
                  rts

RTC.ev_IRQ        idx     RTC.intAddress
                  ldaa   $c,x
                  psha
                  anda   #$80

```



```

RTCev_IRQ.ev1      beq          RTCev_IRQ.fin
                   pula
                   psha
                   anda          #$10          ;verifica si es ciclo de refresco
                   beq          RTCev_IRQ.ev2
RTCa2              jar          RTC.readtime
                   ldd          #EV_RTCUC
                   std          SYS.msg
                   ldy          RTC.UCURT,x
                   pshy
                   bar          RTCdispatch
RTCev_IRQ.ev2     pula
                   psha
                   anda          #$20          ;verifica si es alarma.
                   beq          RTCev_IRQ.ev3
                   ldd          #EV_RTCALARM
                   std          SYS.msg
                   ldy          RTC.AJURT,x
                   pshy
                   bar          RTCdispatch
RTCev_IRQ.ev3     pula
                   psha
                   anda          #$40          ;verifica si es interrupcion periodica
                   beq          RTCev_IRQ.fin
                   ldd          #EV_RTCPI
                   std          SYS.msg
                   ldy          RTC.PIURT,x
                   pshy
                   bar          RTCdispatch
RTCev_IRQ.fin     pula
                   rts
RTCdispatch       tsy
                   iny
                   sty
                   ldaa
                   ldab
                   swi
                   rts
RTC.init          sci
                   ldx          #RTC.tabla-#RTC.Descriptor ;desplazamiento para tabla
                   pshx          ;crea variable temporal.
                   tsy
                   ldd          SYS.wparam          ;se lee la direccion de instalacion
                   addd         0,y              ;se obtiene la direccion real
                   xgdx          ;se pasa a IX la direccion de tabla
                   ldy          #RTC_getvect      ;numero de funciones en tabla.
                   ldd          0,x              ;este ciclo corrige las direcciones
                   subd         #RTC.Descriptor   ;de las funciones de la tabla
                   addd         SYS.wparam
                   std          0,x
                   dey
                   cmpy         #0
                   beq          RTCinit.cont1
                   inx
                   inx
                   bra          RTCinit.c1
RTCinit.c1        RTCinit.c1
                   pula
                   ldx          #RTC.RelocAddr-#RTC.Descriptor ;destruye variable temporal
                   ;desplazamiento a la lista de
                   xgdx          ;direcciones a corregir
                   addd         SYS.wparam        ;obtiene direccion fisica real de
                   xgdx          ;la tabla de direcciones
                   ldy          0,x              ;este ciclo modifica las direc-
                   cmpy         #$0000          ;busca marca de fin.   ;ciones en la lista.
                   beq          RTCinit.cont2

```

```

xgdy
subd
addd
xgdy
pshy
ldy
xgdy
subd
addd
puly
std
inx
inx
bra
RTCinit.cont2
RTCa3
ldd
crmpd
beq
ldx
jcr
stop

#RTC.Descriptor
SYS.wparam
;suma direccion sobre la que fue instalado el prog

0,y

#RTC.Descriptor
SYS.wparam

0,y

RTCinit.c2
pruebartc ;termina el ajuste de direcciones
RTC.intRegreso
#0 ;verifica si no hubo error.
RTCinit.cont3
#RTC.error1
display

rts
ldx
clr
clr
ldaa
staa
ldaa
anda
staa
ldx
pshx
ldx
pshx
tax
stx
ldaa
ldab
swi
pulx
pulx
rts

RTCinit.cont3
RTC.intAddress
RTC.UCURT,x ;limpia las URT.
RTC.AIURT,x
RTC.PIURT,x
#$20 ;limpia el registro a
$a,x
$b,x ;obtiene estado de SET para hacerlo cero.
#$FF
$b,x ;es decir, no hay actualizacion
#IRQFLAG16 ;pide al sistema la habilitacion
;del evento IRQ.

RTCa4
#RTC.handler

SYS.wparam
#SYS
#SYS_enablenotif

RTCipstrRTCname
fcc "RTC.DRV"
fdb #EOB
RTCinit.hwnd
fdb 0
RTCinit.handler
fdb 0
RTCinit.padre
fdb 0
RTCinit.posic
fdb 0
RTCinit.puntero
fdb 0
RTCinit.buffer
fcc "1234567890123456"
fcc "1234567890123456"

IpstrNohayrtc
fcc "No se encontro RTC"
fdb #EOB
IpstrBuscando
fcc "Buscando RTC"
fdb #EOB
IpstrEncontro
fcc "Se encontro RTC en:"
IDireccionrtc
fcc "0000h"
fdb #EOB
IpstrRTCname
fcc "RTC.DRV"
fdb #EOB

pruebartc
ldd #RTCinit.hwnd ;busca el reloj
std
ldd SYS.wparam
#0

```

```

std      SYS.lparaml
ldd      #WM_CREATE
std      SYS.msg
ldaa     #LCD
ldab     #MSG
RTCa16   swi
         idx
         pshx
RTCa5    idx
         pshx
         tax
         stx
         ldaa
         ldab
         swi
         pulx
         pulx
         idx
         bra

         #A000
         RTCa8      ;OJO: temporal, evito muestreo de memoria.

pruebartc.c1  ldy
               inx
               iny
               cmpy
               bis
               ldy
pruebartc.cont3  #0ccc
                 pruebartc.cont3
                 #0
                 #$ff66      ;evita la zona de vectores
                 pruebartc.cont1 ;evita la zona de registros internos
                 #SYSREG-1    ;del MCU
                 pruebartc.cont2
                 #SYSREGEN-1
                 pruebartc.c1
RTCa6        #pruebartc.c1      ;evita este programa.
RTCa7        pruebartc.cont2
pruebartc.cont2  bis
                 ldaa
                 psha
                 ldaa
                 staa
                 ldaa
                 cmpa
                 beq
                 cmpa
                 bnc
                 ldaa
                 ldx
                 ldaa
                 cmpa
                 bnc
                 pula
                 pshx
                 pula
                 pulb
                 subd
                 xgdx
RTCa8        stx
                 ldaa
                 anda
                 staa
                 bar
                 ldx
                 pshx
                 ldaa
                 staa
                 ldx
                 pshx
                 tax
                 stx
                 ldaa
                 ldab
         SYS.wparam
         #LCD
         #LCD_outtext

```

```

swi
pulsx
pulsx
ldaa #SYS
ldab #SYS_sleep
swi pruebartc.fin
bra
pruebartc.cont4 pula
staa
pruebartc.cont1 bra
ldx #lpstrNohayrtc
jsr display
stop

ldd #SFFFF
RTCab RTC.intRegreso
pruebartc.fin std
ldx #RTCinit.hwnd
stx SYS.wparam
ldd #WM_QUIT
std SYS.msg
ldaa #LCD
ldab #MSG
RTCcac ldd #S0
std RTC.intRegreso
rts

ldy #lDireccionrtc
formathex pshx
ldab #2
format.c1 pula
psha
rora
rora
rora
rora
anda #S0f
bar conversion
pula
anda #S0f
bar conversion
iny
bar
iny
decb
format.fin bnc
rts

conversion cmpa #S9
bla conversion.col1
suba #S9
adda #S40
staa 0,y
conversion.col1 adda #S30
staa 0,y
rts

ldy RTC.Structtime
RTCad ldx RTC.intAddress ;en x origen en y destino
ldaa RTC.segundo,x ;segundos
staa .segundos,y
ldaa RTC.minuto,x ;minutos
staa .minutos,y
ldaa RTC.hora,x ;horas
staa .hora,y
ldaa RTC.dia,x ;dia de la semana
staa .dia,y
ldaa RTC.fecha,x ;dia del mes

```

	staa	.fecha,y	
	ldaa	RTC.mes,x;mes	
	staa	.mes,y	
	ldaa	RTC.ano,x;a00	
	staa	.ano,y	
	rts		
RTC.settime	ldy	SYS.wparam	;int settime(lpStructTime)
	ldx	\$0,y	;direccion de la estructura time
RTCa6	ldx	RTC.intAddress	;en x origen, en y destino
	ldaa	0,x	;formato: segundos/minutos/horas
	staa	0,y	;dia de la semana/fecha/mes/a00
	ldaa	2,x	;segundos
	staa	1,y	;minutos
	ldaa	4,x	;horas
	staa	2,y	
	ldaa	6,x	;dia de la semana
	staa	3,y	
	ldaa	7,x	;fecha
	staa	4,y	
	ldaa	8,x	;mes
	staa	5,y	
	ldaa	9,x	;a00
	staa	6,y	
	rts		
RTC.gettime	ldy	SYS.wparam	;int gettime(lpStructTime)
	ldx	0,y	;obtiene puntero a estructura Time
RTCa7	ldx	RTC.Structime	;en x origen en y destino
	ldaa	.segundos,x	;segundos
	staa	.segundos,y	
	ldaa	.minutos,x	;minutos
	staa	.minutos,y	
	ldaa	.hora,x	;horas
	staa	.hora,y	
	ldaa	.dia,x	;dia de la semana
	staa	.dia,y	
	ldaa	.fecha,x	;dia del mes
	staa	.fecha,y	
	ldaa	.mes,x	;mes
	staa	.mes,y	
	ldaa	.ano,x	;a00
	staa	.ano,y	
	rts		
RTC.setalarm	rts		
RTC.getalarm	rts		
RTC.clearalarm	rts		
RTC.setmode	rts		
RTC.getRTCaaddr	rts		
RTC.getPIR	ldx	RTC.intAddress	
	ldaa	\$a,x	
	staa	intRegreso	
	rts		
RTC.setPIR	ldx	SYS.wparam	
	ldaa	\$0,x	
	anda	#SOF	
RTCa10	ldx	RTC.intAddress	
	staa	\$a,x	
	rts		
RTC.enablenotif	ldx	SYS.wparam	;void enablenotif(charMask,intURT)
RTCa11	ldy	RTC.intAddress	;se obtiene la direccion del RTC
	ldaa	\$2,x	;se obtiene la mascara

```

                psha
                anda          #$f0
                cmpa          #$f0
                beq           RTC.disable
                pula
                anda          #$7          ;aseguro no alterar mas bits.

RTCenable.UC   asra
                psha
                bcc          RTCenable.AI ;pregunto por UC
                idaa        $b,y
                oraa        #$10
                staa        $b,y
                ldd         $0,x
                std         RTC.UCIRT,y
                pula
                asra
                psha
                bcc          RTCenable.PI ;pregunto por AI
                idx         #mgenableAI
                jsr         display
                stop

                idaa        $b,y
                oraa        #$20
                staa        $b,y
                ldd         $0,x
                std         RTC.AIIRT,y
                pula
                asra
                bcc          RTCenable.fin ;pregunto por PI
                idx         #mgenablePI
                jsr         display
                stop

                idaa        $b,y
                oraa        #$40
                staa        $b,y
                ldd         $0,x
                std         RTC.PIIRT

RTCenable.fin  rts
RTC.disable    pula
                asra
                psha
                bcs          RTCdisable.AI ;pregunto por UC
                idaa        $b,y          ;si genero acarreo, no se apaga
                anda        #0xf
                staa        $b,y
                clr         RTC.UCIRT,y
                pula
                asra
                psha
                bcs          RTCdisable.PI ;pregunto por AI
                idaa        $b,y
                anda        #0xf
                staa        $b,y
                clr         RTC.AIIRT,y
                pula
                asra
                bcs          RTCdisable.fin ;pregunto por PI
                idaa        $b,y
                anda        #0xf
                staa        $b,y
                clr         RTC.PIIRT,y
                rts

RTC.disable.fin rts

RTC.setvect   rts
RTC.getvect   rts

```

```

RTC.dispatch      rts
RTC.fin           fcb          $0

*****          LCD          *****

LCD.DRV          fcc          "LCD.DRV"
                 fdb          $00aa
                 fcc          "reloc"
LCD.Descriptor   fcc          #LCD.DRV
LCD.CompAdrr     fdb          #LCD.handler-#LCD.Descriptor
LCD.EntryPoint   fdb          $0000
LCD.StackSize    fdb          #LCD.fin-#LCD.DRV
LCD.ProgamSize   fdb          #LCDa1+1
LCD.RelocAddr    fdb          #LCDa2+2
                 fdb          #LCDa3+1
                 fdb          #LCDa4+1
                 fdb          #LCDa5+1
                 fdb          #LCDa6+1
                 fdb          #LCDa7+2
                 fdb          #LCDa8+2
                 fdb          #LCDa9+1
                 fdb          #outext.contl+1
                 fdb          #outext.fin1+1
                 fdb          #LCD.espera+1
                 fdb          #LCD.default+1
                 fdb          #LCD.down+1
                 fdb          #LCD.minimize+1
                 fdb          #LCD.maximize+1
                 fdb          #LCD.dispatch+1
                 fdb          #LCD.msgdefault+1
                 fdb          #0000

LCD.FDEM         fcb          $55      ;File Descriptor End Mark
                 fcb          $aa

LCD.address      fdb          #$G000

LCD.tabla        fdb          #LCD.outtext      ;funcion outtext(lpstrBuffer)    01
                 f2          #LCD.clear        ;funcion clear                    02
                 f3          #LCD.espera      ;funcion esperar                  03
                 f4          #LCD.create      ;funcion wm_create                04
                 f5          #LCD.quit       ;funcion wm_quit                  05
                 f6          #LCD.up         ;funcion wm_up                    06
                 f7          #LCD.down      ;funcion wm_down                  07
                 f8          #LCD.paint     ;funcion wm_paint                 08
                 f9          #LCD.minimize   ;funcion wm_minimize              09
                 fa          #LCD.maximize   ;funcion wm_maximize              0a
                 fb          #LCD.dispatch   ;funcion wm_dispatch               0b
                 fc          #LCD.display   ;funcion temporal                  0c

LCD.msgdef       fcc          "LCD msgDef"
                 fdb          #EOB

LCD.datos fdb    $0          ;direccion de los datos publicos

LCD.handler      psha
LCDa1            ty
                 ldx          #LCD.tabla
                 clr          0,y
                 cmpb        0,y
                 bne         LCD.case
                 bra         LCD.casemsg
LCD.case         inc          0,y
                 cmpb        0,y
                 bne         LCD.Isdefault
                 ldx          0,x
                 jr          pula
                 pula

```

```

LCD.ladefault    rts          #LCD_display
                 cmph          LCD.default
                 bhi
                 xgdx
                 addd          #2
                 xgdx
LCD.default      bra          LCD.case
                 ldz          #def.mensaje
                 jsr          display
                 swi
                 stop
def.mensaje      rts          "LCD.default"
                 fcc          $0
                 fcb
                 fcb

LCD.casemsg      pula
                 ldd          SYS_msg          ;se inhabilitan mensajes
                 cpd          #CM_INIT        ;carga el tipo de mensaje
                 bne          LCD.msg1
                 bsr          LCD.init
LCD.msg1         rts          #WM_CREATE      ;mensajes que se procesan
                 bne          LCD.msg2
                 ldz          6,x
                 jsr          0,x
                 rts
LCD.msg2         cpd          #WM_QUIT
                 bne          LCD.msg3
                 ldz          $,x
                 jsr          0,x
                 rts
LCD.msg3         cpd          #WM_UP
                 bne          LCD.msg4
                 ldz          $a,x
                 jsr          0,x
                 rts
LCD.msg4         cpd          #WM_DOWN
                 bne          LCD.msg5
                 ldz          $c,x
                 jsr          0,x
                 rts
LCD.msg5         cpd          #WM_PAINT
                 bne          LCD.msg6
                 ldz          $e,x
                 jsr          0,x
                 rts
LCD.msg6         cpd          #WM_MINIMIZE
                 bne          LCD.msg7
                 ldz          $10,x
                 jsr          0,x
                 rts
LCD.msg7         cpd          #WM_MAXIMIZE
                 bne          LCD.msgdefault
                 ldz          $12,x
                 jsr          0,x
                 rts
LCD.msgdefault   lcl          #LCD.msgdef     ;funcion que envia el mensaje a URT
                 jsr          display
                 stop
                 rts

LCD.init         ldz          #LCD.tabla-#LCD.Descriptor ;desplazamiento para tabla
                 psbx         ;crea variable temporal.
                 tay
                 ldd          SYS.wparam      ;se lee la direccion de instalacion
                 addd          0,y           ;se obtiene la direccion real
                 xgdx         ;se pasa a IX la direccion de tabla
                 ldz          #LCD_display    ;num funciones

```



```

LCDinit.c1      ldd      0,x
                subd   #LCD.Descriptor
                addd   SYS.wparam
                std    0,x
                dey    crmpy
                beq    LCDinit.cont1
                inx
                inx
                bra    LCDinit.c1
LCDinit.cont1   pulx
                ldx    #LCD.RelocAddr-#LCD.Descriptor ;destruye variable temporal
                addd   SYS.wparam
                xgdx
LCDinit.c2      ldy      0,x ;ciclo para modificar direcciones
                crmpy #S0000
                beq    LCDinit.fin
                xgdy
                subd   #LCD.Descriptor
                addd   SYS.wparam
                xgdy
                pahy
                ldy    0,y
                xgdy
                subd   #LCD.Descriptor
                addd   SYS.wparam
                puly
                std    0,y
                inx
                inx
                bra    LCDinit.c2
LCDinit.fin     rts

*init.hwnd      fdb      0
*init.handler   fdb      0
*init.padre     fdb      0
*init.posic     fcb      0
*init.puntero   fdb      0
*init.buffer    fcc      "1234567890123456"
*               fcc      "1234567890123456"

*hwnd          equ      $0
*handler       equ      $2
*padre         equ      $4
*posic         equ      $6
*puntero       equ      $7
*buffer        equ      $9

LCD.create      ldx      SYS.wparam ;en IX direccion de la nueva estructura de datos
LCDa2           ldy      LCD.datos ;se obtiene la direccion de la estructura del padre
                sty    padre,x ;guarda esta direccion en el nuevo bloque(en padre)
LCDa3           stx    LCD.datos ;La ventana hija es ahora la activa
                stx    .hwnd,x
                ldd    SYS.lparam1
                std    .handler,x
                clr    .posic,x ;limpia posicion
                clr    .puntero,x
                pahx
                tax
                stx    SYS.wparam
                ldaa  #LCD
                ldab  #LCD_clear ;limpia pantalla
                swi
                pulx
                rts

outtext.tmp     fdb      $0
LCD.outtext     ldy      SYS.wparam
                ldx    2,y ;obtiene puntero a datos del objeto ventana

```

```

LCDa9      #ix      LCD.datos
           0,y
LCDa4      clr      outtext.tmp      ;obtiene puntero a texto
           ldab     .posic,x          ;lee posicion
           cmpb    #NUMCARACT-1      ;verifica si esta en final de posicion
           bis     outtext.cont1
           rts
outtext.cont1 #tab     outtext.tmp+1      ;salva posicion
           xgdx    #gdx
           addd    #buffer
LCDa5      addd    outtext.tmp      ;IX apunta a buffer
           xgdx
           psbh
           ldd     0,y
           cpd     #EOB
           beq    outtext.fin2
           pulb
           cmpb    #NUMCARACT-1
           beq    outtext.fin1
           cmpb    #CARACTP.RENG-1 ;deja el caracter 16 libre
           beq    outtext.cont2
           staa   0,x
outtext.cont2 #incb
           incb
           inx
           bra     outtext.c1
outtext.fin1 #ldx     LCD.datos
           stab   .posic,x
           xgdx
           addd    #buffer
           addd    #NUMCARACT-1
           xgdx
           ldab   #62
           stab   0,x
           bra     outtext.fin
outtext.fin2 #tab
LCDa6      bra     LCD.datos
           ldx     SYS.wparam
           stx
           stab   .posic,x
           ldd     #WM_PAINT
           std     SYS.msg
           ldaa   #LCD
           ldab   #MSG
           swi
           rts
LCD.clear   #ldy     SYS.wparam      ;direccion de buffer
           ldx     0,y
           clr     .posic,x
           xgdx
           addd    #buffer
           xgdx
           psbh
           ldab   #NUMCARACT      ;numero de caracteres
           ldaa   #S20
           staa   0,x
           decb
           beq    clear.fin
           inx
           bra     clear.c1
clear.c1    ldaa   #S0
           pulx
           staa   0,x
           ldx     0,y
           stx     SYS.wparam
           ldd     #WM_PAINT
           std     SYS.msg
           ldaa   #LCD
           ldab   #MSG
           swi

```

```

rtz
LCD.espera      ldx      LCD.address
LCD.cl         ldaa     .IRRAM,x
              anda     #$80
              bne     LCD.cl
              rts

LCD.up         ldx      #up.mensaje
              jsr     display
              stop

up.mensaje     rts
              fcc     "LCD.up"
              $0
              Saa

LCD.down       ldx      #down.mensaje
              jsr     display
              stop

down.mensaje   rts
              fcc     "LCD.down"
              $0
              Saa

LCD.paint      ldx      SYS.wparam
              ldaa     #SYS
              ldab     #SYS_IsForegnd ;pregunta si el proceso actual esta
              swi     ;en foreground.
              ldd     intRegreso
              cmpd    #FALSE
              beq     paint.rts
LCDa7          ldy     LCD.address
              ldaa     #$01
              staa    .IRRAM,y
              #.buffer

              0,x ;verifica si hay que limpiar nada mas.
              #0
              beq     paint.fin
              ldab    #NUMCARACT
              cmpb   #0
              beq     paint.fin
              ldaa   0,x

              #LCD
              #LCD_esperar

              .DRRAM,y

              #CARACTP.RENG
              #paint.cl

              #LCD
              #LCD_esperar

              #Sc0
              .IRRAM,y
              bra     paint.cl
              ldaa   #LCD
              ldab   #LCD_esperar

paint.fin      swi

paint.rts     rts

```

```

LCD.minimize      ldx      #minim.mensaje
                  bsr      LCD.display
                  stop
minim.mensaje     rts
                  fcc      "LCD.Minimize"
                  $0
                  Saa
LCD.maximize      ldx      #maxim.mensaje
                  bsr      LCD.display
                  stop
maxim.mensaje     rts
                  fcc      "Maximize"
                  $0
                  Saa

*init.hwnd        fcc      "00"
*init.padre       fcc      "00"
*init.posic       fcb      0
*init.buffer      fcc      "00"
*init.puntero     fcc      "00"

LCD.quit          ldx      SYS.wparam
LCDa8             ldy      .padre,x           ;se obtiene el padre de esta ventana
                  LCD.datos           ;se hace la ventana activa
                  sty      SYS.wparam
                  ldd      #WMT_PAINT    ;vuelve a pintar la ventana
                  std      SYS.msg       ;padre
                  ldaa     #LCD
                  ldab     #MSG
                  swi
                  rts

LCD.dispatch     ldx      #dis.mensaje
                  bsr      LCD.display
                  stop
dis.mensaje       fcc      "LCD.dispatch"
                  $0
                  Saa

LCD.display       ldaa     #LCD
                  ldab     #LCD.clear

display.ciclo1   ldaa     0,x
                  cmpa     #50
                  bne     display.cont
                  ldab     1,x
                  cmpb     #5AA
                  bne     display.cont
                  rts
display.cont     staa     DDRAM
display.ciclo2   ldaa     #LCD
                  ldab     #LCD.espera
                  swi
                  inx
                  bra     display.ciclo1
                  rts

LCD.fin          fcb      0

*****          TESIS.EXE
TESIS.EXE       fcc      "TESIS.EXE"

```

	fdb	\$00aa	
TES.Descriptor	fcc	"reloc"	
TES.CompAdrr	fdb	#TESIS.EXE	
TES.Entrypoint	fdb	#TES.handler-#TES.Descriptor	
TES.Stacksize	fdb	\$0000	
TES.ProgramSize	fdb	#TES.fin-#TESIS.EXE	
TES.RelocAddr	fdb	#TES.handler+1	
	fdb	#TESa1+1	
	fdb	#TESa2+2	
	fdb	#TESa3+1	
	fdb	#TESa4+1	
	fdb	#TESa5+1	
	fdb	#TESa6+1	
	fdb	#TESa7+1	
	fdb	#TESa8+1	
	fdb	#TESa9+2	
	fdb	#TESaa+1	
	fdb	#TESab+1	
	fdb	#TESac+2	
	fdb	#TESad+1	
	fdb	#TESae+1	
	fdb	#TESaf+1	
	fdb	#TESa10+1	
	fdb	#TESa11+1	
	fdb	#TESa12+1	
	fdb	#TESa13+1	
	fdb	#TESa14+1	
	fdb	#TESa15+2	
	fdb	#TESa16+1	
	fdb	#TESa17+1	
	fdb	#TESa18+1	
	fdb	#TESa19+1	
	fdb	#TESa1a+1	
	fdb	#TESa1b+1	
	fdb	#TESa1c+1	
	fdb	#TESa1d+1	
	fdb	#TESa23+1	
	fdb	#TES.Pausa+1	
	fdb	#TES.Exit+1	
	fdb	#TES.trans+1	
	fdb	#ev_RTI.err2+1	
	fdb	#TES.Iniciar+1	
	fdb	\$0000	
TES.FDEM	fdb	\$55aa ;File Descriptor End Mark	
TES.tabla	fdb	#TES.ev_RTI ;funcion respuesta a evento RTI	
	fdb	#TES.Iniciar	
	fdb	#TES.Pausa	
TES.tabla.fin	fdb	#TES.Exit	
tesis.hwnd	fdb	\$00	
tesis.handler	fdb	0	
tesis.padre	fdb	\$00	
tesis.posic	fcc	\$0	
tesis.puntero	fcc	\$00	
tesis.buffer	fcc	"1234567890123456"	
	fcc	"1234567890123456"	
TES.Commid	fdb	\$0	
byteAccion	fdb	\$0	
TES.SCI	fcc	"SCI"	
dato	fcc	"P,"	
o		1234567890123456	
dato1	fcc	"000 [lnH]"	
	fcc	"K,"	
dato2	fcc	"000 [cfm]"	
	fdb	#EOB	
num_bloque	fcc	0	

```

bloque      fcb      #EVT_CHAR
            fcc      "000"
            fcb      $10
            fcc      "000"
bloque.0    fcb      $10
            fcb      "000"
bloque.1    fcb      $10
            fcb      #EOT_CHAR
            fcc      "Tesis default msg"
            fcb      #EOB
TES.defaultmsg fcb      #EOT_CHAR
            fcb      "No se abrio Comm"
            fcb      #EOB
TES.err1    fcb      "Error en la transmision"
            fcb      #EOB
TES.err2    fcb      #EOB

TES.handler ldx      #TES.tabia
            cmpb     #MSG
            beq      TES.msgs
TESa1       jsr      #TES.msgdefault
            display
            stop
            rts

TES.msgs    ldd      SYS.msg
            #CM_INIT
            cpd      TES.msg2
            bnc      TES.init
            cli
            bra      TES.Idle
TES.msg2    cmpd     #EV_RT1
            bnc      TES.msg3
            ldx      0,x
            jsr      TES.Idle
            bra      #CM_INICIAR
TES.msg3    cmpd     #TES.msg4
            bnc      TES.msg4
            ldx      2,x
            jsr      0,x
            bra      TES.Idle
TES.msg4    cmpd     #CM_PAUSA
            bnc      TES.msg5
            ldx      4,x
            jsr      0,x
            bra      TES.Idle
TES.msg5    cmpd     #CM_EXIT
            bnc      TES.msg6
            ldx      6,x
            jsr      0,x
            bra      TES.Idle
TES.msg6    cmpd     #CM_STARTUP
            bnc      TES.msg7
            ldx      2,x
            jsr      0,x
            bra      TES.Idle
TES.msg7    cmpd     #CM_RESUME
            bnc      TES.msgdefault
            ldx      SYS.wparam
            ldd     SYS.lparam1
            jmp     0,x

TES.msgdefault ldx      #TES.defaultmsg
            jsr      display
            ldx      SYS.msg
            jsr      showIX
            stop

TES.Idle    nop
TESa2      ldy      #tesis.hwnd
            clr
            ldd     #CM_XGTASK

```

```

        std      SYS.msg
        ldd      #S0      ;cambio no forzado
        std      SYS.wparam
        ldea    #SYS
        ldab    #MSG
        swi
        cli
        wai
        bra     TES.Idle

TES.inlt      sei
              idx
              pahx
              tay
              ldd
              SYS.lparaml      ;se lee una referencia la direccion de instalacion que es el entry point
              subd    #TES.handler-#TES.Descriptor ;se obtiene la direccion real de instalacion
              std      SYS.lparaml
              addd    0,y      ;se obtiene la direccion real
              xgdx    ;se pasa a IX la direccion de tabla
              ldy    #TES.tabla.fln-TES.tabla
              0,x
              subd    #TES.Descriptor
              addd    SYS.lparaml
              std      0,x
              dey
              cmpy    #S0
              beq    TESinit.cont1
              inx
              bra     TESinit.c1

TESinit.c1   ldd      TESinit.c1      ;destruye variable temporal
              pulx    #TES.RelocAddr-#TES.Descriptor
              ldx
              xgdx    SYS.lparaml
              addd
              xgdx
              ldy
              cmpy    0,x
              beq    #S0000
              xgdy    TESinit.cont2
              subd    #TES.Descriptor
              addd    SYS.lparaml
              xgdy
              pshy
              ldy
              xgdy
              subd    #TES.Descriptor
              addd    SYS.lparaml
              puly
              std      0,y
              inx
              inx
              bra     TESinit.c2

TESinit.cont2 ldd      TESinit.c2
              ldd      #WM_CREATE      ;crea una ventana para esta apli
              std      SYS.msg         ;cacion
              ldd      #Size.lword
              std      SYS.wparam
              ldd      #TES.handler
              std      SYS.lparaml
              ldea    #LCD
              ldab    #MSG

TESa3        swi
              ldea    #S3      ;se ajusta el intervalo de
              .paha    ;interrupcion del RTI
              tax
              slx
              ldea    SYS.wparam
              ldab    #RTI
              idab    #RTI_setRTIrate

```

```

swi
pula

TESa10      ldx      #TES.SCI
            pshx
            tax
            stx
            ldaa
            ldab
            swi
            pulx
            ldd
            cmpa
            bne
            cmpb
            bne
            stx
            ldx
            pshx
            tax
            stx
            ldaa
            ldab
            swi
            ldaa
            ldab
            swi
            ldaa
            ldab
            swi
            pulx
            ldaa
            pha
            tax
            stx
            ldaa
            ldab
            swi
            pula
            jdaa
            staa
            rts

TESa11
TESa5      ldx      #TES.handler ;se habilita la notificacion del
            pshx                    ;evento RTI.
            tax
            stx
            ldaa
            ldab
            swi
            ldaa
            ldab
            swi
            ldaa
            ldab
            swi
            pulx
            ldaa
            pha
            tax
            stx
            ldaa
            ldab
            swi
            pula
            jdaa
            staa
            rts

TESinit.err1 ldx      #TES.err1
            jsr
            stop

TES.Iniciar  clr      num_bloque
TESa1d      clr      byteAccion

TES.Kc      fcb      #131
TES.KcMN    fdb      #SC5F3
TES.KcMNSGN fdb      #B600
TES.MUESTREO fcb      #124
TES.MUESTREOMN fdb      #B312
TES.MUESTREOMNSGN fdb      #6E00
TES.TAU     fcb      #128
TES.TAUMN   fdb      #C31E
TES.TAUMNSGN fdb      #B800
TES.ERROR   fcb      #128
TES.ERRORMN fdb      #0
TES.ERRORMNSGN fdb      #0
TES.ERR     fcb      #128
TES.ERRMN   fdb      #0
TES.ERRMNSGN fdb      #0
TES.V       fcb      #128
TES.VMN     fdb      #0
TES.VMNSGN  fdb      #0
TES.AREA1   fcb      #121 ;exponente de PI

```



```

TES.AREA1MN          fdb      #S84B5
TES.AREA1MNSGN      fdb      #S5400
TES.AREA2           fcb      #116
TES.AREA2MN         fdb      #S7C5
TES.AREA2MNSGN      fdb      #SAC00
TES.DENS            fcb      #129
TES.DENSMN          fdb      #S51E
TES.DENSMNSGN       fdb      #S800
TES.FACTOR           fcb      #128
TES.FACTORMN        fdb      #S9A1C
TES.FACTORMNSGN     fdb      #SAC00
TES.PASCAL           fcb      #136
TES.PASCALMN        fdb      #SFD7
TES.PASCALMNSGN     fdb      #S0A00
TES.FLTASC          rmb      14
TES.FPACCEX         rmb      1
TES.FPACCMN         rmb      3
TES.FPACCSGN        rmb      1
TES.cuenta          fcb      50

TES.ajuste          pahx
                   pshy
                   ldab      #7
TESajuste.c1       ldaa      0,x
                   cmpa      #0
                   beq       TESajuste.fin
                   staa      0,y
                   inx
                   iny
                   decb
                   bne       TESajuste.c1
TESajuste.fin      pulx
                   rts

TES.ev_RTI          equ      *
TESa7              inc      TES.cuenta
TESa8              ldaa      TES.cuenta
                   cmpa      #S4
                   bhi      TESev_RTI.cont1

TESev_RTI.cont1    ldaa      #ADC
                   ldab      #ADC_IsReady
                   swi
                   ldd      intRegreso
                   cpd      #FALSE
                   beq      TESev_RTI.cont1
                   ldd      #255 ;se convierte 255 a float
                   std      FPACC1MN+1
                   jsr      UNT2FLT
                   ldx      #TES.FPACCEX
                   jsr      PUTFPAC1 ;se pasa el dato a PFACC2
                   jsr      GETFPAC2
                   ldx      #ADR1 ;se convierte el dato de la
                   pahx ;conversion A/D a float
                   clra
                   ldab      0,x
                   std      FPACC1MN+1
                   jsr      UNT2FLT
                   jsr      FLTDIV ;se obtiene el valor de presion:
                                ;Lectura/255;
TESa14             ldx      #TES.FLTASC ;convierte el resultado de la
                   jsr      FLTASC ;division a caracteres ASCII
TESa9              ldy      #dato1
TESa23             jsr      TES.ajuste
                   pulx
                   inx

```

```

      clra
      ldab      0,x
      std       FPACC1MN+1
      jsr      UINT2FLT
      jsr      FLTDIV
      jsr      TFR1TO2
      ldd      #15
      std       FPACC1MN+1      ;convierte la lectura del CAD a
      jsr      UINT2FLT      ;flotante
      jsr      FLTMUL

TESa17  ldx     #TES.PASCAL
      bar     TES.FP2ACC2
      jsr     FLTMUL      ;convierte InH2O a Pascales.
      jsr     TFR1TO2     ;realiza la operacion para
      ldd     #2          ;obtener gasto.
      std     FPACC1MN+1
      jsr     UINT2FLT
      jsr     FLTMUL

TESa18  ldx     #TES.DENS
      bar     TES.FP2ACC2
      jsr     FLTDIV
      jsr     FLTSQR
      ldx     #TES.AREA2
      bar     TES.FP2ACC2
      jsr     FLTMUL

TESa19  ldx     #TES.FACTOR
      bar     TES.FP2ACC2
      jsr     FLTMUL      ;gasto en m3/s

TESa1a  ldx     #TES.AREA1
      bar     TES.FP2ACC2
      jsr     FLTDIV      ;permeabilidad en m/s
      jsr     TFR1TO2
      ldd     #100
      std     FPACC1MN+1
      jsr     UINT2FLT
      jsr     FLTMUL      ;permeabilidad en cm/s

TESa16  ldx     #TES.FLTASC
      jsr     FLTASC

TESa15  ldy     #dato2
TESa1b  jsr     TES.ajuste

      bra     TESa1c

TES.FP2ACC1  pshx
      ldy     #FPACC1EX
      ldab    #5
TESFP2ACC1.c1  ldaa    0,x
      staa   0,y
      inx
      iny
      decb
      bne   TESFP2ACC1.c1
      pulx
      rts

TES.FP2ACC2  pshx
      ldy     #FPACC2EX
      ldab    #5
TESFP2ACC2.c1  ldaa    0,x
      staa   0,y
      inx
      iny
      decb
      bne   TESFP2ACC2.c1
      pulx
      rts

TESa1c  ldaa    byteAccion

```

TESaa	bnc	TESad	
TESab	inc	num_bloque	
TESac	ldx	#num_bloque	
	ldy	#bloque	
	pshy		;salva puntero al buffer de datos
	iny		
	bar	TESconversion	;la siguiente secuela de instrucciones
	ldx	#ADR1	;es para convertir los numeros de los
	xgdy		;registros del ADC en caracteres y
	addd	#bloque.1-#bloque.0	;enviarlos por el puerto de comunicaciones
	xgdy		
	bar	TESconversion	
	inx	;ADR2	
	xgdy		
	addd	#bloque.1-#bloque.0	
	xgdy		
	bar	TESconversion	
	puly	;recupera puntero al buffer de datos	
	bar	TES.trans	
TESad	clr	TES.cuenta	
TESae	ldx	#tesis.hwnd	
	clr	.posic,x	
TESa6	ldx	#tesis.hwnd	
TESaf	pshx		
	ldx	#dato	
	pshx		
	tsx		
	stx	SYS.wparam	
	ldaa	#LCD	
	ldab	#LCD_outtext	
	swi		
	pulx		
TESev_RTI.fin	rts		
TESconversion	pshx		
	ldaa	0,x	;carga el dato a convertir y lo guarda en pila
	psha		
	psha		;crea variable temporal con el numero a convertir
	tsx		
	clr	0,y	;unidades
	clr	1,y	;decenas
	clr	2,y	;centenas
	cmpa	#0	;verifica si el dato fuente es cero
	beq	TESconver.fin2	;si es cero no hace nada
TESconver.c1	clra		
	clrb		
	addb	#\$0a	
	cmpb	0,x	
	bhl	TESconver.fin1	
	cmpb	#\$4	;esta convirtiendo un 0xFF
	beq	TESconver.fin1	
	inc	1,y	;incrementa decenas
	ldaa	1,y	;verifica se se ha generado acarreo
	cmpa	#\$0a	;en centenas.
	blo	TESconver.c1	
	inc	0,y	;incrementa centenas.
	clr	1,y	
TESconver.fin1	bra	TESconver.c1	
	subb	#\$a	
	stab	0,x	
	ldaa	1,x	;obtiene el numero a convertir
	suba	0,x	
	staa	2,y	
TESconver.fin2	ldaa	2,y	
	adda	#\$30	
	staa	2,y	

```

        idaa      1.y
        adda     #$30
        staa    1.y
        ldaa    0.y
        adda     #$30
        staa    0.y
        pula
        pulx
        rts

TES.trans      idx      TES.CommId
               pshx
               pshy
               ldaa     #14
               psha
               tax
               stx
               ldaa     SYS.wparam
               ldab     #SCI
               swi
               pula
               puly
               pulx
               ldd
               cpd
               bnc
               rts
               idx      #TES.err2
               jsr
               rts

cv_RTI.err2    idx      #TES.err2
               jsr
               rts

TES.Pausa      idx      #byteAccion
               ldaa     0,x
               cmpa     #CM_PAUSA
               beq      TESPausa.1
               ldaa     #CM_PAUSA
               staa     0,x
               rts

TESPausa.1    idaa     #0
               staa     0,x
               rts

TES.Exit       idx      #TES.SCI
               pshx
               tax
               stx
               ldaa     SYS.wparam
               ldab     #SCI
               swi
               pulx
               idx      #tesis.hwnd
               stx      SYS.wparam
               ldd      #WM_QUIT
               std      SYS.msg
               ldaa     #LCD
               ldab     #MSG
               swi
               idx      #S0
               pshx
               tax
               stx
               ldaa     SYS.wparam
               ldab     #RTI
               swi
               ldab     #RTI_enablenotif
               swi
               ldab     #ADC
               ldab     #ADC_enablenotif
               swi
               pulx
               ldaa     #SYS

```

```

ldab      #SYS_terminate
swi

TES.fin    fdb      #EOB

*****
SERVICIOS DE SWI

*MSG      equ      $00      ;mensaje
*SWI      equ      $01      ;servicios de SWI
*SYS      equ      $02      ;servicios de systema
*STODEV   equ      $03      ;servicios de disco
*SCI      equ      $04      ;servicios de comunicacion
*ADC      equ      $05      ;servicios de ADC
*LCD      equ      $06
*RTC      equ      $07      ;servicios de reloj

ORG        $200

SWI.tabla  fdb      #SWI.iret ;servicio de SWI
           fdb      #SWI.iret ;servicio de SYS
           fdb      #SWI.iret ;servicio de STODEV
           fdb      #SWI.iret ;servicio de SCI
           fdb      #SWI.iret ;servicio de RTI
           fdb      #SWI.iret ;servicio de ADC
SWI.ultimo fdb      #SWI.iret ;servicio de LCD
           fdb      #SWI.iret ;servicio de RTC

SWI.caso   fcb      $0

SWI.handler bsr      SWI.aavemsg
           cmpa   #1      ;verifica si es funcion de SWI
           bcq   SWI.case1

SWI.case2  idx      #SWI.tabla;busca el vector del servicio
           clr   SWI.caso      ;pedido.

SWI.case.c1 inc      SWI.caso
           cmpx  #SWI.ultimo
           bhi  SWI.default
           cmpa  SWI.caso
           bne  SWI.case.cont1
           idx  0,x
           jsr  0,x
           bra  SWI.iret

SWI.case.cont1 inx
           inx
           bra  SWI.case.c1

SWI.default xgdx
           jsr  showIX
           tsy
           idx  $f,y
           jsr showIX
           idx  #swidefault
           jsr  display
           stop

swidefault fcc      "SWI default"
           fdb      #EOB

SWI.case1  cmpb   #SWI_getvect
           bne  SWI.fun2
           bar  SWI.setvect
           bra  SWI.iret

SWI.fun2   cmpb   #SWI_getvect      ;getVect(charNumSer,lpAdr)
           bne  SWI.default
           bar  SWI.getvect
           bra  SWI.iret

SWI.aavemsg puly
           idx  SYS.msg

```



```

TOC3URT      fdb      #SYS.handler
TOC4URT      fdb      #SYS.handler
TOC5URT      fdb      #SYS.handler
TOIURT       fdb      #SYS.handler
PAOURT       fdb      #SYS.handler
PAIURT       fdb      #SYS.handler
SPIURT       fdb      #SYS.handler
SCIURT       fdb      #SYS.handler

PDT.intDescr fcc      "Reloc"
PDT.CompAddr fdb      00
PDT.intEntry fdb      00
PDT.intStackSize fdb    00
PDT.intProgSize fdb    00

*TASK.Nombre equ      $0
*TASK.Entry  equ      $10
*TASK.Stack  equ      $12
*TASK.URT    equ      $14
*TASK.Ruptura equ     $16
*TASK.Modos  equ      $18

SYS.task1    fcc      "Nombre7890123456"
             fdb      $0      ;entrypoint
             fdb      $0      ;stack
             fdb      $0      ;URT
             fdb      $0      ;Ruptura
             fcb      $0      ;Modos

SYS.task2    fcc      "0123456789abcdef"
             fdb      $0      ;entrypoint
             fdb      $0      ;stack
             fdb      $0      ;URT
             fdb      $0      ;Ruptura
             fcb      $0      ;modos

SYS.task3    fcc      "1234567890123456"
             fdb      $0      ;entry
             fdb      $0      ;stack
             fdb      $0      ;URT
             fdb      $0      ;Ruptura
             fcb      $0      ;modos

SYS.task4    fcc      "0123456789abcdef"
             fdb      $0      ;entry
             fdb      $0      ;stack
             fdb      $0      ;URT
             fdb      $0      ;Ruptura
             fcb      $0      ;modos

lpstrProgName fdb      $00
lpDirecInstal fdb      $00

SYS.tabla    fdb      #SYS.run      ;funcion de respuesta
             fdb      #SYS.kill     ;funcion de respuesta
             fdb      #SYS.install  ;funcion de respuesta
             fdb      #SYS.xgtask   ;funcion de respuesta
             fdb      #SYS.sleep    ;funcion sys_sleep
             fdb      #SYS.setsysvect ;funcion sys_setsysvect(servicio,direcc)
             fdb      #SYS.getsysvect ;funcion sys_getsysvect
             fdb      #SYS.getpc     ;funcion sys_getpc
             fdb      #SYS.malloc    ;funcion sys_malloc(intBytes)
             fdb      #SYS.mfree     ;funcion sys_mfree
             fdb      #SYS.getURT    ;funcion int_sys_getURT(lpVector)
             fdb      #SYS.setURT    ;funcion int_sys_setURT(lpVector)
             fdb      #SYS.sysatmem  ;funcion int_sys_sysatmem(lpStructMem)
             fdb      #SYS.load      ;funcion int_sys_load(lpstrName)
             fdb      #SYS.transfer  ;funcion void_sys_transfer(lpEntryPoint)
             fdb      #SYS.enablenotif ;funcion void_sys_enablenotif(intMascara)
             fdb      #SYS.getevvector ;funcion int_getevvector(intMascara)
             fdb      #SYS.setevvector ;funcion int_setevvector(intMascara)
             fdb      #SYS.dispatch  ;funcion void_dispatch(intURT)
             fdb      #SYS.IsForegnd ;BOOL IsForegnd(int(handler)

```

```

sysfl5      fdb      #SYS.terminate      ;BOOL terminate(void)
sysfl6      fdb      #SYS.unload      ;BOOL unload(lp Entry)

SYS.EV_FLAG      fdb      #0000      ;SCI|SPI|PAI|PAO|ITO|OC5|OC4|OC3|OC2|OC1|IC3|IC2|IC1|RTI|IRQ|XIRQ
                : 15| 14| 13| 12| 11| 10| 9 | 8| 7| 6| 5| 4| 3| 2| 1| 0|

SYS.caso     fcb      $0

SYS.handler  cmpb     #MSG
            bne     SYS.case1
            bra     SYS.case0
SYS.case1    ldx     #SYS.tabla
            clr     SYS.caso
SYS.scase.c1 inc     SYS.caso
            cmpx   #sysfl6
            bhi   SYS.default
            cmpb   SYS.scase.cont1
            bnc   ldx     0,x
            jsr   jsr     0,x
            rts
SYS.scase.cont1 inx
            inx
            bra   SYS.scase.c1
SYS.default  ldx     SYS.scase.c1
            jsr   #sysdefault
            stop
            rts
sysdefault  fcc     "Sys default"
            fdb     #EOB

SYS.case0    ldx     #SYS.tabla
            ldd     SYS.msg
            cpd     #CM_INIT
            bne   SYS.msg1
            bar   SYS.init
            rts
SYS.msg1     cpd     #SYS_INSTALL
            bne   SYS.msg2
            ldx   4,x
            jsr   0,x
            rts
SYS.msg2     cpd     #CM_RUN
            bne   SYS.msg3
            ldx   0,x
            jsr   0,x
            rts
SYS.msg3     cpd     #CM_XGTASK
            bne   SYS.msgdefault
            ldx   6,x
            jsr   0,x
            rts
SYS.msgdefault ldx     #msgdefault
            jsr   display
            stop
msgdefault  fcc     "SYS Mensaje default"
            fdb     #EOB

SYS.msajereso ldx     #mensajereso
            jsr   display
            stop
            rts
mensajereso fcc     "SYS.msajereso"
            fdb     #EOB

SYS.sleep    ldy     #S3d1      ;1 segundo
sleep.ciclo  ldab    #SIT
            decb                    ;ciclo toma 2037 ciclos

```



```

sleep.cont1      beq      sleep.cont1
                 bra      sleep.ciclo
                 beq      sleep.fin
                 bra      sleep.ciclo
sleep.fin        rts

cuenta          fcb      $00
                 fcb      0
                 fcb      $AA

SYS.init        idx      #$100
                 ldaa     #$1F
                 staa     $1F,x
                 ldd      #BIOS.fin
                 ldx      #$100
                 idiv     #0
                 cmpd     sysinit.cont4
                 beq      inx
                 xgdx     ldx
                 ldaa     #$01
                 staa     0,x
                 decb     sysinit.cont1
                 beq      inx
                 inca     sysinit.c4
                 bra      inx
                 cmpx     #$1F
                 beq      sysinit.cont2
                 ldaa     0,x
                 cmpa     #$1F
                 beq      sysinit.cont1
                 psbx     ;salva IX para obetnerlo despues
                 psbx     ;transfiere IX a IY
                 puly     xgdy
                 ldx      #$8
                 lsl     ;se hacen corrimientos para obtener la direccion
                 dex     ;fisica del bloque
                 bnc     sysinit.c3
                 pulx     xgdy
                 lsl     #5
                 ldaa     0,y
                 psba     ;los 256 bytes de cada bloque
                 ldaa     #$55
                 staa     0,y
                 ldaa     0,y
                 cmpa     #$55
                 bnc     sysinit.cont3
                 ldaa     #$aa
                 staa     0,y
                 ldaa     0,y
                 cmpa     #$aa
                 bnc     sysinit.cont3
                 pula     sysinit.c2
                 staa     #00
                 iny     #5
                 decb     sysinit.c2
                 bnc     #00
                 ldaa     0,x
                 bra      sysinit.cont1
                 pula     ;en este bloque no hay RAM, indica en la
                 staa     ;tabla de RAM que no hay RAM ahi con 0xFF
                 ldaa     0,y
                 staa     #$fc
                 staa     0,x

```

```

sysinit.cont2      bra      sysinit.cont1      ;va al siguiente bloque.
ldy                #SYS.msagregao ;no hay URT, intURT apunta a rti
sty
ldx                #SYS.XIRQhandler
stx                XIRVECTOR
ldx                #SYS.IRQhandler
stx                IRQVECTOR
ldx                #SYS.RTIhandler
stx                RTIVECTOR
ldx                #SYS.TOIhandler
stx                TOIVECTOR
ldx                #SYS.ILLhandler
stx                ILLVECTOR
ldx                #SYS.COPhandler
stx                COPVECTOR
ldx                #SYS.SCIhandler
stx                SCIVECTOR
ldx                #SYS.SPIhandler
stx                SPIVECTOR
ldx                #SYS.TIC1handler
stx                TIC1VECTOR
ldx                #SYS.TIC2handler
stx                TIC2VECTOR
ldx                #SYS.TIC3handler
stx                TIC3VECTOR
ldx                #SYS.TOC1handler
stx                TOC1VECTOR
ldx                #SYS.TOC2handler
stx                TOC2VECTOR
ldx                #SYS.TOC3handler
stx                TOC3VECTOR
ldx                #SYS.TOC4handler
stx                TOC4VECTOR
ldx                #SYS.TOC5handler
stx                TOC5VECTOR
ldx                #SYS.PAOhandler
stx                PAOVECTOR
ldx                #SYS.PAIhandler
stx                PAIVECTOR
rts

INESPERADA        ldx                #inesp
jmr                display
inexp              stop
fcc                "interrupcion no iden"
fdb                #EOB

SYSrun.intReg      fdb                #0
SYSrun.intReg2     fdb                #0
SYSrunmsg          fcc                "No hay entry"
fcb                #EOB
*struct_create    fdb                $0000      ;Entry point
*                  fdb                $0000      ;Stack
*                  fdb                $0000      ;URT
*                  fdb                $0000      ;Puntero a parametros
*                  fdb                $0          ;puntero a nombre
SYS.run            ldy                SYS.wparam      ;lee puntero a estructura Create
ldx                #SYS.task1
SYSrun.c1          cpx                #SYS.task4      ;busca un Entry libre en la tabla
bhi                SYSrun.er1      ;no hay Entry libre
ldd                TASK.Entry.x      ;obtiene el entry point de la tabla, si esta
cmpd               #0                ;ocupado(distinto de cero) tal entry, busca otro task en tabla
beq                SYSrun.cont1
xgdx
addd               #SYS.task2-#SYS.task1 ;calcula la direccion del siguiente task
xgdx

SYSrun.er1        bra                SYSrun.c1
ldx                #SYSrunmsg
jmr                display
stop

```

```

SYSrun.cont1    pshx                ;guarda puntero al task libre de la tabla de tareas
                pshy                ;guarda puntero a struct_create
                ldv                ;obtiene el puntero al nombre del proceso
                ldaa                ;16 caracteres a transferir
                cmpa                #CREATE.Nombre.y
                beq                #$10
                ldab                #SY$0
                stab                SY$run.cont2
                iny                0,y
                inx                ;transfiere el nombre al entry
                deca                0,x

SYSrun.cont2    bra                SY$run.c2
                ldx                #SYS.task1
                ldaa                TASK.Mod0,x
                cmpa                ;busca proceso activo
                beq                #S01
                cmpa                #ACTIVO
                beq                ;pregunta por proceso activo
                cmpa                SY$run.cont3
                beq                ;desactiva el proceso activo
                cmpx                #SUSPENDIDO
                bha                SY$run.cont4
                xgdx                #SYS.task4
                addd                ;no hay proceso activo
                xgdx                #SYS.task2-#SYS.task1
                bra                SY$run.c3

SYSrun.cont3    tsy                ;decremento iy para evitar las variables temporales
                iny                ;dejandolo a nivel de entrada de esta funcion.
                iny
                iny
                iny
                sty                TASK.Stack,x
                ldd                intURT
                std                ;guarda la pila del proceso
                ldaa                TASK.URT,x
                staa                ;obtiene URT del proceso
                ldd                #BACKGNDINACTIVO
                staa                ;lo hace backg e inactivo
                ldd                TASK.Mod0,x
                std                $13,y
                ldaa                TASK.Ruptura,x
                staa                ;obtiene punto de ruptura del proceso que lo llamo
                staa                #BACKGNDINACTIVO
                staa                ;lo hace backg e inactivo
                staa                TASK.Mod0,x

SYSrun.cont4    puly                ;obtiene puntero a struct
                pulx                ;obtiene puntero al tasklibre
                clr                TASK.Ruptura,x
                ldd                CREATE.Entry,y
                std                ;punto de entrada
                std                TASK.Entry,x
                std                intRegreso
                ldd                CREATE.URT,y
                std                ;URT
                ldd                TASK.URT,x
                std                ;pila
                std                CREATE.Stack,y
                ldaa                TASK.Stack,x
                staa                #FOREGNDACTIVO ;lo hace activo en foreground
                staa                TASK.Mod0,x

SYSrun.cont5    pulx                ;obtiene regreso de esta subrutina
                clr                ;regreso de esta subrutina
                stx                SY$run.intReg
                stx                SY$run.intReg2
                pula                ;regreso del servicio
                ora                ;obtiene CCR
                ldx                #S10
                ldy                CREATE.Entry,y
                dey                ;desactiva interrupciones
                tye
                pshx                ;pasa a IY el contenido de IX(Entry)
                puly                ;empuja regreso para rti
                pshx                #S0
                pshx                ;empuja IY
                pshx                ;empuja IX
                pshx                ;empuja reg D
                psha                ;empuja CCR

```

```

Idx      #CM_INIT
pshx
Idx      #$0                ;empuja SYS.msg
pshx    SYS.wparam        ;empuja SYS.wparam
pshy    SYS.iparam1      ;empuja SYS.iparam1 ;que es el Entry.
pshx    SYS.iparam        ;empuja SYS.iparam
Idx      SYSrun.intReg2
pshx    SYSrun.intReg     ;empuja regreso de servicio
Idx      SYSrun.intReg
pshx    SYSrun.intReg     ;empuja regreso de subrutina
rts

SYS.kill      rts

nomemmensaje  fcc      "No hay memoria"
illmensaje    fdb      #EOB
              fcc      "Descriptor il"
              fdb      #EOB

SYS.install   Idx      SYS.wparam
              stx      lpstrProgName
              pshx    #PDT.intDescr
              Idx      #PDT.intDescr
              pshx    ;se envia direcc de una region de memoria
              tax
              stx      SYS.wparam
              ldaa    #STODEV
              ldab    ;para almacenaje del PDT
              swi     ;en wparam se conserva la direc
              pulx    ;cion de la cadena con el nombre
              pulx    ;del archivo.
              ldaa    0,x
              cmpa    #$72
              bnc     descrip.case1
              ldaa    1,x
              cmpa    #$65
              bnc     illdescrip
              ldaa    2,x
              cmpa    #$6c
              bnc     illdescrip
              ldaa    3,x
              cmpa    #$6f
              bnc     illdescrip
              ldaa    4,x
              cmpa    #$63
              bnc     illdescrip
              ldx     PDT.intProgSize
              pshx
              tax
              stx      SYS.wparam
              ldaa    #SYS
              ldab    #SYS_malloc
              swi
              pulx
              ldx     intRegreso
              stx     lpDirecInstal
              cmpx    #$$fff
              beq     install.error1
              ldx     lpDirecInstal
              pshx
              ldx     lpstrProgName
              pshx
              tax
              stx      SYS.wparam
              ldaa    #STODEV
              ldab    #STODEV_load
              swi
              pulx
              pulx

```



```

xgtask_err3      ldx      #xgtask.error3
                 jr        display
xgtask_noproc    ldx      #xgtask.error2
                 jsr      display
SYS_xgtask       pshx
                 tsy
xgtask_c1        ldx      #SYS.task1
                 ldaa     TASK.Modo,x
                 anda     #S0f
                 cmpa     #ACTIVO
                 beq      xgtask.cont1
                 cmpa     #SUSPENDIDO
                 beq      xgtask.cont2
                 cmpx     #SYS.task4
                 bhs     xgtask.noxx
                 xgdx     addd
                 xgdx     bra
                 xgdx     ldd
                 xgdx     cmpd
                 xgdx     bne
                 xgdx     bra
xgtask_cont1     stx
                 iny
                 iny
                 sty     TASK.Stack,x
                 ldd     $13,y
                 std     TASK.Ruptura,x
                 dey
                 ldd     intURT
                 std     TASK.URT,x
                 ldd     SYS.wparam
                 ldaa     TASK.Modo,x
                 anda     #Sf0
                 ora     #INACTIVO
                 staa     TASK.Modo,x
xgtask_c2        xgdx
                 addd     #SYS.task2-#SYS.task1
                 xgdx
xgtask_cont4     ldaa     TASK.Modo,x
                 anda     #S0f
                 cmpa     #INACTIVO
                 beq      xgtask.activa
                 cmpx     0,y
                 beq      xgtask.noproc
                 cmpx     #SYS.task4
                 blo      xgtask_c2
                 ldx     #SYS.task1
                 bra     xgtask.cont4
xgtask_activa    ldd     SYS.wparam
                 cpd     #S0
                 beq     xgtask.cont5
                 ldd     TASK.Entry,x
                 cmpd    SYS.wparam
                 bne     xgtask.c2
xgtask_cont5     puly
                 ldaa     TASK.Modo,x
                 psha
                 ldd     SYS.wparam
                 cpd     #S0
                 beq     xgtask.cont7
                 cpd     TASK.Entry,y
                 bne     xgtask.cont7
                 pula
                 anda     #Sf0

```

: ciclo: busca proceso activo
: pregunta por proceso activo
: busca en el siguiente proceso
: Un proceso es suspendido cuando se genera un mensaje para
: el estando activo, tiene que resumir en el punto donde se suspendio
;
: no encontro proceso activo.

: guarda el puntero al TASK activo en variable temporal
: dejo la pila a nivel de entrada
: de esta funcion. POR DEFINICION
: guarda la pila del proceso
: obtiene punto de ruptura
: guarda punto de ruptura
: justo otra vez el apuntador a
: la pila
: obtiene URT del proceso
: lo guarda

: obtiene estado
: lo hace inactivo

: ciclo: busca proceso por activar

: busca siguiente proceso para habilitarlo
: pregunta si esta inactivo
: verifica si llego al task que fue activo (ya revise todos los demas entrias
: de la tabla)

: ve si se trata de un cambio obli
: gado (por algun evento)
: si wparam es cero, es cambio conse
: cutivo
: verifica si es el entry que se

: destruyo variable temporal y paso a lY el entry de la tabla que pidio el
: el cambio.

: verifica cambio obligatorio

: verifica si el proceso pedido es el mismo que el que estaba activo

```

        oraa          #SUSPENDIDO
        bra          xgtask.cont9
xgtaak.cont7      pula
        anda        #S0
        oraa        #ACTIVO           ;lo hace activo, respeta plano de ejecucion
xgtaak.cont8      staa          TASK.Mod0,x
        ldy         TASK.Stack,x
        tys
xgtaak.cont11     ldd          TASK.Ruptura,x           ;se obtiene el pc antes de ruptura
        std         $8,y           ;se envia como parametro
        ldd         $4,y           ;se obtiene el registro D antes de la ruptura
        ldd         $4,y           ;se envia como parametro
        ldd         TASK.Entry,x   ;se hace el regreso a Entry al encontrar rti
        std         $13,y
        #0
        std         ;registro D
        ldd         $d,y           ;se deja en el registro D ceros(mensaje)
        ldx         #CM_RESUME     ;mensaje default: resume
        cpx         SYS.wparam     ;verifica cambio forzado
        beq         #S0
        ldatb       xgtaak.cont6
        orab       #$10
        stab       $c,y
xgtaak.cont6     ldd         $c,y           ;lee el mensaje a desplegar
        std         SYS.lparaml    ;deja mensaje en SYS.msg
xgtaak.cont9     rts
        staa        TASK.Mod0,x
        ldy         TASK.Stack,x
        tys
        xgdy
        addd        #S14           ;apunto al principio de esta interrupcion
        xgdy       ;para duplicar el stack y hacer el regreso
xgtaak.c3        dab           ;sin danar el antiguo regreso. Esto es solo
        ldaa       0,y           ;para programas suspendidos.
        psha
        dey
        decb
        beq        xgtaak.cont10
        bra        xgtaak.c3
xgtaak.cont10   tay
        bra        xgtaak.cont11
xgtaak.err4     ldx         #xgtaak.error4
        jar         display
xgtaak.error1   fcc         "No hay p. activo"
        fdb        #EOB
xgtaak.error2   fcc         "No hay procesos"
        fdb        #EOB
xgtaak.error3   fcc         "Mensaje anidado"
        fdb        #EOB
xgtaak.error4   fcc         "Destinatario incorrecto"
        fdb        #EOB

setsysvectcaso fcb         S0
SYS.setsysvect ldx         #SYS.tabla;setvect(ucharNumSer,lpVector)
        ldy         SYS.wparam
        clr        setsysvectcaso
        inc        setsysvectcaso
setsys.c1       ldaa       $2,y
        cmpa       setsysvectcaso
        bne       setsys.cont1
        ldd       S0,y
        std       0,x
setsys.fin     rts
setsys.cont1   cmpx       sysfl4
        beq       setsys.fin
        inc       setsysvectcaso
        inx
        inx
        bra       setsys.c1

```

```

SYS.getsysvect      ldx      #SYS.tabla;lp getsysvect(ucharNumSer);
                   ldy
                   clr
                   inc
                   ldaa
getsys.c1           cmpa      SYS.wparam
                   bne      setsysvectcaso
                   ldd      setsysvectcaso
                   ldd      2,y
                   std      setsysvectcaso
                   rts
getsys.fin          setsysvectcaso
getsys.cont1       ldd      getsys.cont1
                   ldd      0,x
                   std      intRegreso
                   rts
                   cmpx      sayfl4
                   beq      getsys.fin
                   inc
                   inx
                   inx
                   bra      getsys.c1

SYS.getpc           tax
                   ldd      7,x
                   ldx      9,x
                   std      0,x
                   rts

intNumbloq         fcb      $0
intNumbloq1        fcb      $0

SYS.malloc         ldy      SYS.wparam
                   ldd      $0,y           ;obtiene el num de bytes
                   ldx      #256
                   idiv      ;en IX esta el num de bloque que el prog
cmpd               beq      ;necesita. Si hay residuo se incrementa en
                   ldd      #0             ;un bloque m s
                   std      malloc.cont1

malloc.cont1       xgdx      ;en b estan ahora el numero de bloques
                   stab      intNumbloq1
                   pahb      ;salva b para usarlo posteriormente.
                   ldx      #$100         ;direccion de MAT
malloc.c1          ldab      intNumbloq1
malloc.c2          inx
                   cmpx      ;en este bloque se buscan el num
                   beq      # $1ff       ;de bloques de memoria CONTIGUOS
                   ldaa      malloc.error1 ;requeridos
                   cmpa      0,x
                   bne      # $00       ;verifica si esta libre
                   decb      malloc.c1

malloc.cont2       beq      malloc.cont2
                   bra      malloc.c2
                   dec      intNumbloq1
                   xgdx      intNumbloq
                   subd      intNumbloq
                   xgdx      intNumbloq
                   inc      intNumbloq1
                   ldab      intNumbloq1
malloc.c3          ldaa      # $01
                   staa      0,x
                   decb      malloc.cont3
                   beq      intNumbloq
                   inx
                   inca
malloc.cont3       bra      malloc.c3
                   ldaa      malloc.c3
                   staa      # $ff       ;marca el ultimo bloque con 0xff
                   dec      0,x
                   inx
                   ldaa      intNumbloq
                   xgdx      intNumbloq
                   subd      # $100
                   subd      # $8
malloc.c4          ldd      ;se obtiene el primer bloque de
                   ldx      ;la memoria reservada. Con los dos
                   ldd      ;corrimientos se obtiene la direc-

```



```

dex
bnc      malloc.c4
xgdx
pula
paix
ldab
;obtiene numero de bloques
;guarda direccion de inicio
malloc.c5      #SfT
malloc.c6      0,x
clr
decb
beq      malloc.cont4
inx
bra      malloc.c6
malloc.cont4   deca
beq      malloc.cont5
malloc.cont5   bra
pula
xgdx
std
rts
intRegreso ;cion fisica del primer bloque

malloc.error1  pulb
ldx      #SfT
stx
rts
intRegreso

SYS.mfree      rts

SYS.getURT     rts

SYS.setURT     rts

SYS.dispatch  ldx      SYS.wparam      ;void dispatch(intURT)
ldx      0,x                ;obtiene receptaculo del mensaje.
beq      SYSdispatch.fin
stx
ldx      SYS.wparam
SYS.dispatch  SYS.msg
stx      SYS.lparaml
ldx      #S0
stx      SYS.lparam
ldd      #CM_XGTASK
std      SYS.msg
ldaa     #SYS
ldab     #MSG
SYSdispatch.fin  swi
rts

SYS.systemem  ldx      SYS.wparam ;se obtiene punero a parametros.
systemem.c1   ldy      #SfT      ;parametros, estructura mem
iny
cmpy      #S200
beq      systemem.fin
ldaa     0,y
cmpa     #Sfe
beq      systemem.c1
inc      $0,x
cmpa     #S0
systemem.cont1  bnc     systemem.cont1
bra      systemem.c1
systemem.cont1  inc     $1,x
systemem.fin   bra      systemem.c1
rts

loadill       fcc      "Invalid descriptor"
load.ill      fdb     #EOB
ldx           #loadill
jsr           display
stop
rts

SYS.load      ldy      SYS.wparam

```

```

idx      $0,y
stx      lpstrProgName
pahx
ldx      #PDT.intDescr
pahx
tax
        SYS.wparam
        #STODEV
ldaa     #STODEV_getPDT
ldab
swi
pulx
puly
idd
cpd
bnc
idd
jmp

load.cont1
ldaa     0,x
cmpa    #$72
bnc
ldaa     1,x
cmpa    #$65
bnc
ldaa     2,x
cmpa    #$66
bnc
ldaa     3,x
cmpa    #$6f
bnc
ldaa     4,x
cmpa    #$63
bnc
idx      PDT.intProgSize
pahx
tax
stx      SYS.wparam
ldaa     #SYS
ldab     #SYS_malloc
swi
pulx
ldx      intRegreso
stx      lpDirecInstal
cmpx    #FFFF
bnc
ldx      lpDirecInstal
pahx
ldx      lpstrProgName
pahx
tax
stx      SYS.wparam
ldaa     #STODEV
ldab     #STODEV_load
swi
pulx
pulx
xgdx
addd
rts

load.fin
        PDT.intEntry
        intRegreso
        ;en IX: direccion de instalacion
        ;para encontrar el punto de entrada se suma el desplazamiento
        ;PDT.intEntryPoint

load.error1
ldx      #loaderror1
jnr
stop
idd
std
ldd
std
ldaa     #SYS
ldab     #SYS_dispatch

```

;se envia direc de una region de memoria

;para almacenaje del PDT
;en wparam se conserva la direccion de la cadena con el nombre

;del archivo.

intRegreso
#FFFF
load.cont1
#FFFF
load.fin

0,x
#\$72
load.ill
1,x
#\$65
load.ill
2,x
#\$66
load.ill
3,x
#\$6f
load.ill
4,x
#\$63
load.ill
PDT.intProgSize

SYS.wparam
#SYS
#SYS_malloc

intRegreso
lpDirecInstal
#FFFF
load.error1
lpDirecInstal
lpstrProgName

SYS.wparam
#STODEV
#STODEV_load

PDT.intEntry
intRegreso
;en IX: direccion de instalacion
;para encontrar el punto de entrada se suma el desplazamiento
;PDT.intEntryPoint

#loaderror1
display
#EV_NOMEM
SYS.msg
#lpstrProgName
SYS.wparam
#SYS
#SYS_dispatch

```

        rts

loadererrorl    fcc      "Error, no mem!"
                fdb      #EOB

SYS.transfer    ldy      SYS.wparam ;transfiere el control a la direccion
                ldx      $0,y      ;indicada en la pila. Lo hace hasta en
                stx      $1,y      ;contrar el rti.
                rts

SYS.enablenotif  ldy      SYS.wparam      ;enablenotif(intMask,intURT)
                ldd      SYS.EV_FLAG
                orab     $3,y      ;hace OR con la mascara
                andb     #S03      ;asegura no alterar otras banderas
                std      SYS.EV_FLAG
                ldd      $2,y      ;obtiene otra vez la mascara de los parametros
                pshb
                andb     #S1
                beq      SYSnotif2 ;XIRQ
                bty
                ldaa     2,y      ;obtiene el CCR anterior de la interr
                anda     #Sbf
                staa     2,y
                ldy      SYS.wparam
                ldc      $2,y
                pshx
                idx      $0,y
                pshx
                bsr      SetVector
                pulx
                pulx
                andb     #S2      ;IRQ
                andb     SYSnotif2
                beq      SYSnotif2
                idx      $2,y
                pshx
                ldc      $0,y      ;lee URT
                pshx
                bsr      SetVector
                bsr
                pulx
                pulx
                rts

SYSnotif2
                rts

SYSnotif2fin
                rts

SetVector
                tsx
                inx
                stx
                SYS.wparam
                stx
                idaa     #SYS
                ldab     #SYS_setevector
                awi
                rts

getevector.tmp  fcb      $0      ;getevector(intMask)
SYS.getevector  ldy      SYS.wparam
                ldx      #XIRQURT
                ldab     #S01
                stab     getevector.tmp
                ldd      0,y      ;ciclo para el byte menos signifi
                andb     getevector.tmp ;catico de Mask.
                bnc     SYSgetev.cont1 ;dio con la mascara.
                inx
                inx
                ldab     getevector.tmp
                lsib
                bcs     SYSgetev.cont2 ;brinca al ciclo para el byte mas significativo
                stab
                bra     SYSgetev.c1

```

```

SYSgetev.cont2      ldaa      #$1
                    staa      getevvector.tmp
SYSgetev.c2         ldd      0,y
                    andb     getevvector.tmp
                    bne      SYSgetev.cont1
                    inx
                    inx
                    ldaa     getevvector.tmp
                    lsl     ;ciclo para el byte mas significativo
                    bcs     ;dio con la mascara
                    staa     getevvector.tmp
                    bra     SYSgetev.c2
                    ;la mascara es cero.
SYSgetev.cont1     ldd      0,x
                    std      intRegreso
                    rts
                    ;obtiene el vector.
SYSgetev.error     ldd      #NIT
                    std      intRegreso
                    rts
                    ;NIT
                    ;intRegreso
setevvector.tmp    fcb      $0
SYS.setevvector    ldy      SYS.wparam
                    ldx      #XIRQURT
                    ldab     ;setevvector(intMask,lpURT)
                    stab     ;direccion de la tabla con las URT
                    ldd      #$01
                    andb     setevvector.tmp
                    bne     SYSsetev.c1
                    inx
                    inx
                    ldab     setevvector.tmp
                    lab     ;obtiene la prueba
                    bcs     ;recorre bit
                    stab     SYSsetev.cont2
                    ldd      setevvector.tmp
                    ldd      SYSsetev.c1
                    ldd      #$01
                    andb     ;si se genero acarreo, el evento pedido no pertenece a la parte baja del
                    bne     ;de banderas
                    inx
                    inx
                    ldab     setevvector.tmp
                    lab     ;guarda la mascara de prueba
                    bcs     ;cierra ciclo
                    stab     SYSsetev.c2
                    ldd      #$2,y
                    andb     setevvector.tmp
                    bne     SYSsetev.cont1
                    inx
                    inx
                    ldab     setevvector.tmp
                    lab     ;ahora prueba con la parte alta del registro de banderas
                    bcs     ;prueba la mascara prueba
                    stab     setevvector.tmp
                    ldd      SYSsetev.c1
                    ldd      #$01
                    andb     ;guarda la mascara
                    bne     ;siguiente vector
                    inx
                    inx
                    ldab     setevvector.tmp
                    lab     ;hace corrimiento
                    bcs     ;error no encontro ninguna
                    staa     SYSsetev.error
                    staa     setevvector.tmp
                    bra     SYSsetev.c2
                    ldd      $0,y
                    std      0,x
                    rts
                    ;obtiene direccion del handler
                    ;lo guarda en el vector corres.
SYSsetev.error     ldx      #SYSseteverror
                    jar      display
                    stop
                    ldd      #NIT
                    std      intRegreso
                    rts
                    ;NIT
                    ;intRegreso
SYSseteverror     fcc      "SYSseteverror"
                    fdb      #EOB
                    ;EOB
subintRegreso     fcb      $0
                    ;$0
* XIRQ **
SYS.XIRQhandler   ldx      #XIRQmensaje
                    jar      display
                    stop
                    bra     SYS.XIRQhandler
                    tpa     ;temporal, se habilita stop
                    anda    #$7f
                    tap
                    ldx      #XIRQmensaje

```

```

        jsr      display
        stop
        ldd     SYS.EV_FLAG
        anda   #$0
        andb   #XIRQFLAG
        beq    XIRQ.fin
        ldd     #EV_XIRQ
        std     SYS.msg
        ldab   #MSG
        ldx    XIRQURT
        jmp    0,x
XIRQ.fin
XIRQmensaje        rti      "XIRQ"
                   fcc     #EOB

* IRQ **
SYS.IRQhandler     ldx     intRegreso
                   jsr     subintRegreso
                   ldd     SWI.savemsg
                   ldd     SYS.EV_FLAG
                   anda   #$0
                   andb   #IRQFLAG
                   beq    SYS.IRQfin
                   ldd     #EV_IRQ
                   std     SYS.msg
                   ldx    IRQURT
                   ldab   #MSG
                   jsr    0,x
                   ldx    subintRegreso
                   stx    intRegreso
                   jmp    SWI.iret

SYS.IRQfin

* RTI **
SYS.RTIhandler     ldx     intRegreso
                   stx    subintRegreso
                   jsr     SWI.savemsg
                   ldaa   TFLG2
                   oraa   #$40
                   staa   TFLG2
                   ldd     SYS.EV_FLAG
                   anda   #$0
                   andb   #RTIFLAG
                   beq    SYS.RTIfin
                   ldd     #EV_RTI
                   std     SYS.msg
                   ldx    RTIURT
                   ldab   #MSG
                   jsr    0,x
                   ldx    subintRegreso
                   stx    intRegreso
                   jmp    SWI.iret

SYS.RTIfin

* TIC1 **
SYS.TIC1handler    ldx     #mensajetic1
                   jsr     display
                   stop
                   ldd     SYS.EV_FLAG
                   anda   #$0
                   andb   #TIC1FLAG
                   beq    SYS.TIC1fin
                   ldx    TIC1URT
                   stx    SYS.wparam
                   ldd     #EV_TIC1
                   std     SYS.msg
                   ldaa   #SYS
                   ldab   #MSG
                   swi
                   rti
                   fcc     "TIC1"
                   fcb     #EOB

SYS.TIC1fin
mensajetic1

```

```

* TIC2 **
SYS.TIC2handler  ldx      #mensajetic2
                  jsr      display
                  stop
                  ldd      SYS.EV_FLAG
                  anda     #$0
                  andb     #TIC2FLAG
                  beq      SYS.TIC2fin
                  ldx      TIC2URT
                  stx      SYS.wparam
                  ldd      #EV_TIC2
                  std      SYS.msg
                  ldaa     #SYS
                  ldab     #MSG
SYS.TIC2fin      rti
mensajetic2      fcc      "TIC2"
                  fdb      #EOB

* TIC3 **
SYS.TIC3handler  ldx      #mensajetic3
                  jsr      display
                  stop
                  ldd      SYS.EV_FLAG
                  anda     #$0
                  andb     #TIC3FLAG
                  beq      SYS.TIC3fin
                  ldx      TIC3URT
                  stx      SYS.wparam
                  ldd      #EV_TIC3
                  std      SYS.msg
                  ldaa     #SYS
                  ldab     #MSG
SYS.TIC3fin      rti
mensajetic3      fcc      "TIC3"
                  fdb      #EOB

* TOC1 **
SYS.TOC1handler  ldx      #mensajetic1
                  jsr      display
                  ldd      SYS.EV_FLAG
                  anda     #$0
                  andb     #TOC1FLAG
                  beq      SYS.TOC1fin
                  ldx      TOC1URT
                  stx      SYS.wparam
                  ldd      #EV_TOC1
                  std      SYS.msg
                  ldaa     #SYS
                  ldab     #MSG
SYS.TOC1fin      rti
mensajetic1      fcc      "TOC1"
                  fdb      #EOB

* TOC2 **
SYS.TOC2handler  ldx      #mensajetic2
                  jsr      display
                  stop
                  ldd      SYS.EV_FLAG
                  anda     #$0

```

```

                                andb    #TOC2FLAG
                                beq     SYS.TOC2fin
                                ldx     TOC2URT
                                stx     SYS.wparam
                                ldd     #EV_TOC2
                                std     SYS.msg
                                ldaa   #SYS
                                ldab   #MSG
                                swi
                                rti
                                fcc    "TOC2"
                                fdb    #EOB

SYS.TOC2fin
mensajetoc2

* TOC3 **
SYS.TOC3handler    ldx     #mensajetoc3
                   jsr     display
                   stop
                   ldd     SYS.EV_FLAG
                   andb   #S0
                   anda   #TOC3FLAG
                   beq   SYS.TOC3fin
                   ldx   TOC3URT
                   stx   SYS.wparam
                   ldd   #EV_TOC3
                   std   SYS.msg
                   ldaa  #SYS
                   ldab  #MSG
                   swi
                   rti
                   fcc   "TOC3"
                   fdb   #EOB

SYS.TOC3fin
mensajetoc3

* TOC4 **
SYS.TOC4handler    ldx     #mensajetoc4
                   jsr     display
                   stop
                   ldd     SYS.EV_FLAG
                   anda   #S0
                   anda   #TOC4FLAG
                   beq   SYS.TOC4fin
                   ldx   TOC4URT
                   stx   SYS.wparam
                   ldd   #EV_TOC4
                   std   SYS.msg
                   ldaa  #SYS
                   ldab  #MSG
                   swi
                   rti
                   fcc   "TOC4"
                   fdb   #EOB

SYS.TOC4fin
mensajetoc4

* TOC5 **
SYS.TOC5handler    ldx     #mensajetoc5
                   jsr     display
                   stop
                   ldd     SYS.EV_FLAG
                   anda   #S0
                   anda   #TOC5FLAG
                   beq   SYS.TOC5fin
                   ldx   TOC5URT
                   stx   SYS.wparam
                   ldd   #EV_TOCS
                   std   SYS.msg
                   ldaa  #SYS
                   ldab  #MSG

```

```

SYS.TOC5fin      swi
mensajetoc5     rti          "TOC5"
                fcc          #EOB
                fdb

* TOI **
SYS.setPR       idx          SYS.wparam
                ldaa         $0,x
                anda         #50f
                oraa         TMSK2
                staa         TMSK2
                rts

SYS.enableTOI   idx          SYS.wparam
                ldab         $0,x
                cmpb         #50
                beq          SYS.disableTOI
                ldaa         TMSK2
                oraa         #580
                staa         TMSK2
                rts

SYS.disableTOI  idx          TMSK2
                anda         #57f
                staa         TMSK2
                rts

SYS.TOIhandler  idx          #mensajetoi
                jsr          display
                stop
                ldaa         TFLG2
                oraa         #580
                staa         TFLG2
                ldd          SYS.EV_FLAG
                andb         #50
                anda         #TOIFLAG
                beq          SYS.TOIfin
                ldd          #EV_TOI
                std          SYS.msg
                ldaa         #0 ;notifico servicio 0 funcion 0 (mensaje)
                ldab         #0
                ldx          TOIURT
                jmp          0,x

SYS.TOIfin      mensajetoi  rti          "TOI ev"
                fcc          #EOB
                fdb

* PAO **
SYS.PAOhandler  idx          #mensajepao
                jsr          display
                stop
                ldd          SYS.EV_FLAG
                andb         #50
                anda         #PAOFLAG
                beq          SYS.PAOfin
                ldx          PAOURT
                stx          SYS.wparam
                ldd          #EV_PAO
                std          SYS.msg
                ldaa         #SYS
                ldab         #MSG

SYS.PAOfin      mensajepao  rti          "PAO"
                fcc          #EOB
                fdb

* PAI **
SYS.PAIhandler  idx          #mensajepai
                jsr          display

```



```

stop
ldd
andb
anda
beq
ldx
stx
ldd
std
ldaa
ldab
swi
rli
fcc
fdb

SYS.PAIfin
mensajepai

"PAI"
#EOB

* SPI **
SYS.SPHandler

ldx
jsr
stop
ldd
andb
anda
beq
ldx
stx
ldd
std
ldaa
ldab
swi
rli
fcc
fdb

#mensajespi
display
SYS.EV_FLAG
#$0
#SPIFLAG
SYS.SPifin
SPIURT
SYS.wparam
#EV_SPI
SYS.msg
#SYS
#MSG

"SPI"
#EOB

* SCI **
SYS.SCIhandler

ldx
stx
jsr
ldd
andb
anda
beq
ldd
std
ldx
cpv
beq
ldab
jsr
ldx
stx
jmp

intRegreso
subintRegreso
SWI.aavemag
SYS.EV_FLAG
#$0
#SCIFLAG
SYS.SCIfin
#EV_SCI
SYS.msg
SCIURT
#$0
SYS.SCIfin
#MSG
0,x
subintRegreso
intRegreso
SWI.iret

;salva estructura de mensajes
;verifica que este acitvado el evento
;obtiene el URT

SYS.SCIfin

* COP **
SYS.COPhandler

ldx
jsr
stop
rli
fcc
fcb
fcb

#COP.mensaje1
display
"COP FAIL"
$0
$aa

* ILL **
SYS.ILLhandler

ldx
jsr

#ILL.mensaje1
display

```

```

tax
ldx
jsr
ldx
jsr
nop
stop
ILL.c1      ILL.c1
ILL.mensaje1 fcc      "Illegal OpCode"
             fcb      $0
             fcb      Saa

* CMF **
SYS.CMFHandler ldx      #CMF.mensaje1
              jsr      display

CMF.mensaje1  stop
             fcc      "CMF fail"
             fcb      $0
             fcb      Saa

SYS.IsForegnd ldy      #SYS.task1      ;puntero a lista de tareas
IsFor.c1      ldaa     TASK.Modoy    ;ciclo para buscar proceso activo
             anda    #SOF          ;pregunta por activo
             cmpa    #ACTIVO
             beq     IsFor.cont1; se encontro el proceso activo
             cmpa    #SUSPENDIDO
             beq     IsFor.cont1
             xgdy    #SYS.task2-#SYS.task1 ;calcula la direcc. del sig. entry de
             addd    ;la tabla de tareas
             xgdy    cmpy
             bhi    IsFor.cont2  ;verifica si llego al fin de la tabla
             bra    IsFor.c1     ;no hay procesos. Autoriza el foregnd.
IsFor.cont1   ldaa     TASK.Modoy    ;verifica que este en Foreground
             anda    #SOF
             cmpa    #FOREGND
             bne    IsFor.cont3  ;no esta en foregnd.
IsFor.cont2   ldd     #TRUE
             std     intRegreso
IsFor.cont3   ldd     #FALSE
             std     intRegreso
             rts

SYSter.msg1   fcc      "Exit fail"
SYSter.msg2   fcb      #EOB
             fcc      "No proc"
             fcb      #EOB

SYS.terminate ldx      #SYS.task1
SYSter.c1     ldaa     TASK.Modoy    ;ciclo: busca proceso activo
             anda    #SOF
             cmpa    #ACTIVO      ;pregunta por proceso activo
             beq     SYSter.cont1 ;busca en el siguiente proceso
             cmpa    #SUSPENDIDO ;Un proceso es suspendido cuando se genera un mensaje para
             beq     SYSter.cont1 ;ese estando activo, tiene que resumir en el punto donde se suspendio
             cmpx    #SYS.task4
             bhs    SYSter.err1  ;no encontro proceso activo.
             xgdx    #SYS.task2-#SYS.task1
             addd
             xgdx
             bra    SYSter.c1
SYSter.err1   ldx      #SYSter.msg1
             jsr      display

SYSter.cont1  ldy      TASK.Entry,x
             pshy
             tsy

```

```

sty      SYS.wparam
ldaa    #SYS
ldab    #SYS_unload
swi
puly
pahx
puly
ldaa    #S10           ;16 caracteres a limpiar
ldab    #S20
cmpa    #S0
SYSter.c2  beq      SYSter.cont2
          stab
          iny
          deca
          bra      SYSter.c2
SYSter.cont2  ldd      #S0
          std      TASK.Ruptura,x
          std      TASK.Entry,x
          std      TASK.URT,x
          std      TASK.Stack,x
          clr      TASK.Modo,x
SYSter.c3  xgdx     ;ciclo: busca proceso por activar
          addd    #SYS.task2-#SYS.task1
          xgdx
SYSter.cont3  ldaa    TASK.Modo,x      ;busca siguiente proceso para habilitarlo
          anda    #S0f
          cmpa    #INACTIVO          ;pregunta si esta inactivo
          beq     SYSter.activa
          cmpx    0,y
          beq     SYSter.noproc      ;verifica si llego al task que fue activo(ya reviso todos los demas entrias
          cmpx    #SYS.task4        ;de la tabla)
          blo     SYSter.c3
          ldx     #SYS.task1
          bra     SYSter.cont3
SYSter.activa  anda    #S0
          oraa    #FOREGNDACTIVO ;lo hace activo
          staa   TASK.Modo,x
          idy    TASK.Stack,x
          tys
          ldd    TASK.Ruptura,x      ;se obtiene el pc antes de ruptura
          std    $8,y                ;se envia como parametro
          ldd    $4,y                ;se obtiene el registro D antes de la ruptura
          std    $4,y                ;se envia como parametro
          ldd    TASK.Entry,x        ;se hace el regreso a Entry al encontrar rti
          std    $13,y
          ldd    #0                  ;registro D
          std    $d,y                ;se deja en el registro D ceros(mensaje)
          ldd    #CM_RESUME          ;mensaje default: resume
          std    $a,y                ;deja mensaje en SYS.msg
SYSter.noproc  ldx     #SYSter.msg2
          jsr    display
          stop
SYS.unload  ldx     SYS.wparam
          ldd    0,x
          tab
          clra
          inca
          xgdx
          ldaa   0,x
          cmpa   #Sff
          bnc   SYSunload.c1
          bra   SYSunload.fin
SYSter.unload.c1  ldx     0,x
          ldaa   #Sff
          cmpa   #Sff
          bnc   SYSunload.c1
          ;en este ciclo se busca el fin de la cadena
          ;D contiene 1XX donde XX es la direccion base del bloque
          ;en IX esta la direccion de la MAT que le corresponde el bloque que se
          ;quiere liberar
          ;verifica si es bloque unico, si no lo es, busca inicio ;de la cadena de
          ;bloques

```

```

SYSunload.c2      clr      0,x
                  dex
                  ldaa     0,x
                  cmps    #S1
                  beq     SYSunload.fin
                  bra     SYSunload.c2
SYSunload.fin     clr      0,x
                  rts

catchexcep       ldx     #SYS.NIRQ1
                  stx     NIRQVECTOR
                  ldx     #SYS.IRQ1
                  stx     IRQVECTOR
                  ldx     #SYS.RTI1
                  stx     RTIVECTOR
                  ldx     #SYS.TOI1
                  stx     TOIVECTOR
                  ldx     #SYS.ILL1
                  stx     ILLVECTOR
                  ldx     #SYS.COP1
                  stx     COPVECTOR
                  ldx     #SYS.SCI1
                  stx     SCIVECTOR
                  ldx     #SYS.SPI1
                  stx     SPIVECTOR
                  ldx     #SYS.TIC11
                  stx     TIC1VECTOR
                  ldx     #SYS.TIC21
                  stx     TIC2VECTOR
                  ldx     #SYS.TIC31
                  stx     TIC3VECTOR
                  ldx     #SYS.TOC11
                  stx     TOC1VECTOR
                  ldx     #SYS.TOC21
                  stx     TOC2VECTOR
                  ldx     #SYS.TOC31
                  stx     TOC3VECTOR
                  ldx     #SYS.TOC41
                  stx     TOC4VECTOR
                  ldx     #SYS.TOC51
                  stx     TOC5VECTOR
                  ldx     #SYS.PAI1
                  stx     PAIVECTOR
                  ldx     #SYS.PAIJ
                  stx     PAJVECTOR
                  rts

ORG              $1040

NIRQ1            fcc     "NIRQ"
                  fdb     #EOB
IRQ1             fcc     "IRQ"
                  fdb     #EOB
RTI1            fcc     "RTI"
                  fdb     #EOB
TOI1            fcc     "TOI"
                  fdb     #EOB
ILL1            fcc     "ILL"
                  fdb     #EOB
COP1            fcc     "COP"
                  fdb     #EOB
SCI1            fcc     "SCI"
                  fdb     #EOB
SPI1            fcc     "SPI"
                  fdb     #EOB
TIC11           fcc     "TIC1"
                  fdb     #EOB
TIC21           fcc     "TIC2"
                  fdb     #EOB
TIC31           fcc     "TIC3"

```

```

TOC11      fdb      #EOB
           fcc      "TOC1"
           fdb      #EOB
TOC21      fcc      "TOC2"
           fdb      #EOB
TOC31      fcc      "TOC3"
           fdb      #EOB
TOC41      fcc      "TOC4"
           fdb      #EOB
TOC51      fcc      "TOC5"
           fdb      #EOB
PAO1       fcc      "PAO1"
           fdb      #EOB
PAI1       fcc      "PAI1"
           fdb      #EOB

SYS.XIRQ1  idx      #XIRQ1
           bra      ddisplay
SYS.IRQ1   idx      #IRQ1
           bra      ddisplay
SYS.RTI1   idx      #RTI1
           bra      ddisplay
SYS.TO11   idx      #TO11
           bra      ddisplay
SYS.ILL1   idx      #ILL1
           bra      ddisplay
SYS.COP1   idx      #COP1
           bra      ddisplay
SYS.SCI1   idx      #SCI1
           bra      ddisplay
SYS.SPI1   idx      #SPI1
           bra      ddisplay
SYS.TIC11  idx      #TIC11
           bra      ddisplay
SYS.TIC21  idx      #TIC21
           bra      ddisplay
SYS.TIC31  idx      #TIC31
           bra      ddisplay
SYS.TOC11  idx      #TOC11
           bra      ddisplay
SYS.TOC21  idx      #TOC21
           bra      ddisplay
SYS.TOC31  idx      #TOC31
           bra      ddisplay
SYS.TOC41  idx      #TOC41
           bra      ddisplay
SYS.TOC51  idx      #TOC51
           bra      ddisplay
SYS.PAO1   idx      #PAO1
           bra      ddisplay
SYS.PAI1   idx      #PAI1
           bra      ddisplay
ddisplay  jsr      display
           stop

debug.temp fcb      $0
debug      paha
           ldaa
           staa      debug.temp
           pula
           rts

stop       paha
           ldaa      debug.temp
           cmpa      #SO
           beq      stop.fin
           ldx
           jsr      #hastaaqui
           stop
           pula

stop.fin

```

```

rtx
display      ldaa    #S01
             staa   IRRAM
             bsr    despera
ddisplay.cic1 ldaa    0,x
             cmpa   #S0
             bne   ddisplay.cont
*
*           OJO:
rtx
rtx          ;rtx agregado para terminar
             ;cadenas en NULL
             ltab   1,x
             cmpb   #SAA
             bne   ddisplay.cont
ddisplay.cont rtx
             staa   DDRAM
             bsr    despera
             inx
             bra   ddisplay.cic1
despera      ldaa    IRRAM
             anda   #S80
             bne   despera
             rtx
showIX       pahx
             pahy
             paha
             pahb
             pahx
             jar    sleep
             pulx
             jar    SYSformathex
             idx
             jar    #SYSCadena
             jar    display
             jar    sleep
             pulb
             pula
             puly
             pulx
             rtx
showPC       pahx
             tax
             idx   4,x
             bar   showIX
             pulx
             rtx
sleep        ldaa    #SYS
             ltab   #SYS_sleep
             swi
             rtx
SYSCadena   fcc    "0000"
             fdb
             ldy
             pahx
             ltab   #2
             pula
             paha
             rora
             rora
             rora
             rora
             anda   #S0f
             bsr    SYSconversion
             pula
             anda   #S0f

```

```

                iny
                br  SYSconversion
                iny
                decb
                bne  SYSformat.c1
SYSformat.fin   rts

SYSconversion   cmpa   #S9
                bit   SYSconver.col
                suba  #S9
                adda  #S40
                staa  0,y
                rts
SYSconver.col   adda  #S30
                staa  0,y
                rts

SYS.fin         fcb    $0

..... Interprete del sistema .....

SO.EXE         fcc    "SOTMC.EXE"
                fdb    $00aa

SO.tabla       fdh    #SO.showtime
                fdb    #SO.startup

tesisfile      fcc    "TESIS.EXE"
                fdb    #EOB

SO.defaultmsg  fcc    "SO default msg"
                fdb    #EOB

Idleerror      fcc    "no se cargo Tesis"
                fdb    #EOB

SO.handler     ldx    #SO.tabla
                cmpb  #MSG
                beq   SO.msgs
                rts

SO.msgs        idd    SYS.msg
                cpd   #CM_INIT
                bne  SO.msg2
                jsr  SO.init
                cli
                jmp  SO.Idle

SO.msg2        cmpd  #EV_RTCUC
                bne  SO.msg3
                ldx  0,x
                jsr  0,x
                bra  SO.Idle

SO.msg3        cmpd  #CM_STARTUP
                bne  SO.msg4
                ldx  2,x
                jsr  0,x
                bra  SO.Idle

SO.msg4        cmpd  #CM_RESUME
                bne  SO.msgdefault
                ldx  SYS.wparam
                idd  SYS.lparam1
                jmp  0,x

SO.msgdefault  ldx  #SO.defaultmsg
                jsr  display
                ldx  SYS.msg
                jsr  showIX
                stop

SO.temp        fcb    $0

```

```

SO.Idle      ldaa    #SYS
             ldab    #SYS_IsForegnd
             swi
             ldd
             cmpd   #FALSE
             beq    SOIdle.cont1
SOa13       ldaa    SOtemp
             cmpa   #$F
             beq    SOIdle.cont1
             ldx    so.hwmd
             pshx
             tax
             stx    SYS.wparam
             ldaa   #LCD
             ldab   #LCD_clear
             swi
             pulx
             ldx    #so.SCI
             pshx
             tax
             stx    SYS.wparam
             ldaa   #SCI
             ldab   #SCI_OpenComm
             swi
             pulx
             ldx    #SO.handler
             pshx
             tax
             stx    SYS.wparam
             ldaa   #SCI
             ldab   #SCI_EnableNotif
             swi
             pulx
             ldaa   #$F
             staa   SO.temp
             ldd
             std   #SO
             ldd   SYS.wparam
             std   #CM_XGTASK
             ldd   SYS.msg
             std   #SYS
             ldaa   #MSG
             ldab
             swi
             cli
             wai
             bra    SO.Idle
Idle.fin
SO.init
SOinit.cont2 sei
             ldd   #WM_CREATE
             std   SYS.msg
             ldd   #so.hwmd
             std   SYS.wparam
             ldd   #SO.handler
             std   SYS.lparam1
             ldd   #LCD
             ldaa   #MSG
             ldab
             swi
             ldx    #so.SCI
             pshx
             tax
             stx    SYS.wparam
             ldaa   #SCI
             ldab   #SCI_OpenComm
             swi
             pulx
             ldx    #UCFLAG; mascara para habilitar ciclo de refresco
             pshx
             ldx    #SO.handler
             tax
             stx    SYS.wparam

```



```

ldaa #RTC
ldab #RTC_enablenotif ;se habilita el evento
swi
puls
pula
pshx
tax
sx
ldaa #SCI
ldab #SCI_EnableNotif
swi
puls
ldaa #SfF
staa SO.temp
rts

so.hwnd fdb $00
so.handler fdb $00
so.padre fdb $00
so.posic fcb $0
so.puntero fdb $00
so.buffer fcc "1234567890123456"
fcc "1234567890123456"

so.dia fcc "Lu "
so.fecha fcc "00"
fcc " "
so.hora fcc "00"
fcc ":"
so.minutos fcc "00"
fcc ":"
so.segundos fcc "00"
so.cursor fcc "C:"
so.SCI fcc #EOB
fcc #SCI"

*.hwnd fdb 0
*.handler fdb 0
*.padre fdb 0
*.posic fcb 0
*.buffer fdb 0
*.puntero fdb 0

*struct_create fdb $0000 ;Entry point
* fdb $0000 ;Stack
* fdb $0000 ;URT
* fdb $0000 ;Puntero a parametros
* fdb $0000 ;Puntero a nombre

*StructTime{
* ucharAoo
* ucharMes
* ucharFecha
* ucharDia
* ucharHora
* ucharMinutos
* ucharSegundos
*

SOformathex pshy
paha
rora
rora
rora
rora
anda #Sof
bar SOconversion
pula
anda #Sof
iny
bar SOconversion

```

```

SOformat.fin      puly
                  rts

SOconversion      cmpa      #$9
                  bna      $Oconver.col
                  suba     #$9
                  adda     #$40
                  staa     0,y
                  rts

SOconver.col      adda      #$30
                  staa     0,y
                  rts

SO.showtime       paha      ;crea estructura Time
                  pahx
                  pahx
                  pahx
                  tax
                  pahx
                  tax
                  stx      SYS.wparam
                  ldaa     #RTC
                  ldab     #RTC_gettime
                  swi
                  pulx
                  ldaa     .segundos,x
                  ldy     #so.segundos
                  bsr      SOformathex
                  ldaa     .minutos,x
                  dey     ;IY apunta a minutos
                  dey
                  bsr      SOformathex
                  ldaa     .hora,x
                  dey     ;IY apunta a horas
                  dey
                  bsr      SOformathex
                  ldaa     .fecha,x
                  dey     ;IY apunta a fecha
                  dey
                  bsr      SOformathex
                  ldaa     .dia,x
                  ldx     #so.dia
                  pahx     ;guardo IX para usuario mas adelante.
                  cmpa     #$1
                  bnc     showtime.lun
                  ldd     #$446f
                  std     0,x
                  bra     showtime.pint
showtime.lun      cmpa     #$2
                  bnc     showtime.mar
                  ldd     #$4c75
                  std     0,x
                  bra     showtime.pint
showtime.mar      cmpa     #$3
                  bnc     showtime.mic
                  ldd     #$4d61
                  std     0,x
                  bra     showtime.pint
showtime.mic      cmpa     #$4
                  bnc     showtime.jue
                  ldd     #$4d59
                  std     0,x
                  bra     showtime.pint
showtime.jue      cmpa     #$5
                  bnc     showtime.vie
                  ldd     #$4e75
                  ;Ju en hex

```

```

std      0,x
bra      showtime.pint
showtime.vic  cmpa      #56
          bnc      showtime.sab
          ldd      #5669          :Vi en hex
          std      0,x
          bra      showtime.pint
showtime.sab  cmpa      #57
          bnc      showtime.pint
          ldd      #5361          :Sa en hex
showtime.pint std      0,x
          ldx      #so.hwnd
          clr      .posic,x
          pulv
          pshx
          pshy
          tsx
          stx      SYS.wparam
          ldaa     #LCD
          ldab     #LCD_outtext
          swi
          pulx
          pulx
          pulx          :destruye estructura time
          pulx
          pulx
          pula
          rts
SO.tesis.exe  fcc      "Tesis.exe"
          fdb      #EOB
SO.startup    ldx      #so.hwnd
          pshx
          ldx      #SO.tesis.exe
          pshx
          tsx
          stx      SYS.wparam
          ldaa     #LCD
          ldab     #LCD_outtext
          swi
          pulx
          pulx
          ldaa     #SYS
          ldab     #SYS_sleep
          swi
          pshx
          pshx
          pshx
          pshx
          ldx      #tesisfile
          pshx
          tsx
          stx      SYS.wparam
          ldaa     #SYS
          ldab     #SYS_load
          swi
          pulx
          ldd      intRegreso
          cpd      #SMT
          bnc      startup.cont1
          ldx      #ldleerror
          jsr      display
          stop
startup.cont1 tsx
          std      CREATE.Entry,x
          ldd      #tesisfile
          std      CREATE.Nombre,x
          :obten puntero a estructura create
          :manda el entry point
          :manda puntero a cadena nombre

```

```

        Idy          #128
        pshy
        tsy
        sty          SYS.wparam
        ldaa        #SYS          ;malloc regresa puntero al principio
        ldab        #SYS_malloc   ;del bloque, hay que ajustarlo al
        swi                ;final para poder usarlo de pila.
        puly
        ldd          intRegreso
        cpd          #FFFF
        tne          startup.cont2
        ldx          #Idleerror
        jr          display
        stop

startup.cont2      clr          SO.temp
                  psha
                  pshb
                  pshx
                  idx          #so.SCI
                  pshx
                  tx
                  stx          SYS.wparam
                  ldaa        #SCI
                  ldab        #SCI_CloseComm ;deshabilita la notificacion de SCI
                  swi
                  pulx
                  pulb
                  pula
                  addd
                  std          CREATE.Stack,x ;manda puntero al stack
                  stx          SYS.wparam     ;manda puntero a struct_create.
                  ldd          #CM_RUN
                  std          SYS.msg
                  ldaa        #SYS
                  ldab        #MSG
                  swi                ;destruye estructura
                  pulx
                  pulx
                  pulx
                  pulx
                  rts

*struct_create
*
*
*
*
SO.fin            nop

.....           RTI.DRV .....

RTI.DRV          fcc          "RTI.DRV"
RTI.URT          fdb          $0
RTI.tabla        fdb          #RTI.setRTIrate ;setRTIrate(char Rate)
                  fdb          #RTI.enablenotif ;enablenotif(intURT)
                  fdb          #RTI.ev_rti   ;funcion de respuesta

RTI.msgdefault   fcc          "RTI msgdef"
                  fdb          #EOB
RTI.fundefault   fcc          "RTI fundef"
                  fcc          #EOB
                  fdb          #EOB

RTI.handler      psha

```

```

                tsy
                ldx      #RTI.tabla
                clr      0,y
                cmpb    0,y
RTI.case       beq      RTI.casemsg
                inc      0,y
                cmpb    RTI.lndefault
                bnc     pula
                ldx      0,x
                jsr      0,x
                rts
RTI.lndefault  cmpb    #RTI_enablenotif
                bhi     RTI.default
                inx
                inx
                bra     RTI.case
RTI.default    ldx      #RTI.fundefault
                jsr      display
RTI.stop1     stop
                bra     RTI.stop1

RTI.casemsg    pula
                ldd     SYS_msg
                cpd     #CM_INIT
                bnc     RTI.msg1
                bar     RTI.init
                rts
RTI.msg1       cpd     #EV_RTI
                bnc     RTI.defaultmsg
                bar     RTI.ev_rti
                rts
RTI.defaultmsg ldx      #RTI.msgdefault
                jsr      display
RTI.stop2     stop
                bra     RTI.stop2

RTI.init       rts

RTI.ev_rti     ldx      RTI.URT
                cmpx    #S0
                beq     ev_rti.fin
                psbx
                tsx
                msl
                ldd     SYS.wparam
                ldd     #EV_RTI
                std     SYS_msg
                ldaa   #SYS
                ldab   #SYS_dispatch
                swi
                pulx
                rts

ev_rti.fin     ldx      #ev_rti.mens
                jsr      display
                stop
                bra     RTI.stop1

ev_rti.mens    fcc     "URT cero"
                fdb     #EOB

RTIactRTIrate ldx      SYS.wparam
                ldx     0,x
                anda   #S03
                orna   PACTL
                staa   PACTL
                rts

RTIerror1     fcc     "RTI en uso"
                fdb     #EOB

```

```

RTI.enablenotif    ldx      SYS.wparam      ;enable(intURT)
                  ldx      0,x          ;obtiene URT
                  cmpx     #$0
                  beq      RTIenable.dis ;si URT es cero, deshabilita.
                  ldd      RTI.URT
                  cpd      #$0
                  bne      RTIenable.err1
                  stx      RTI.URT
                  ldd      SYS.EV_FLAG  #RTIFLAG
                  orab     SYS.EV_FLAG
                  anda     #$0
                  andb     #RTIFLAG
                  xgdx
                  pshx
                  ldx      #RTI.handler
                  pshx
                  tax
                  stx      SYS.wparam
                  ldaa     #SYS
                  idab     #SYS_setevvector
                  swi
                  pulx
                  pulx
                  ldaa     TMSK2
                  oraa     #$40
                  staa     TMSK2
RTIenable.dis     rts
                  ldaa     TMSK2
                  anda     #$bf
                  staa     TMSK2
                  ldaa     TFLG2
                  oraa     #$40
                  staa     TFLG2
                  ldd      #$0
                  std      RTI.URT
                  rts

RTImensaj1        fcc      "RTI disable"
                  fdb      #EOB

                  clr      RTI.URT
                  ldaa     TMSK2
                  anda     #$bf
                  staa     TMSK2
                  ldd      SYS.EV_FLAG
                  andb     #NOTRTIFLAG
                  std      SYS.EV_FLAG
RTIenable.fin     ldx      #RTIerror1
RTIenable.err1    jar
                  stop

*****          ADC *****
ADC.DRV           fcc      "ADC.DRV"
ADC.tabla         fdb      #ADC.setmode      ;int BOOL setmode(uchar Mode)
                  fdb      #ADC.enablenotif   ;int BOOL enablenotif(uint URT)
                  fdb      #ADC.IsReady      ;int BOOL IsReady()

ADC.msgdefault    fcc      "ADC msgdef"
                  fdb      #EOB

ADC.URT           fdb      $0

ADC.error1        fcc      "ADC already in use"
                  fdb      #EOB

```

```

ADC.handler      pula
                 tay
                 ldx      #ADC.tabla
                 clr      0,y
                 cmph    ADC.casemsg
ADC.case         beq      ADC.casemsg
                 inc     0,y
                 cmph    ADC.ldefault
                 bne     ADC.ldefault
                 pula
                 ldx     0,x
                 jsr     0,x
                 rts

ADC.ldefault     cmph    #ADC_IsReady
                 bhi     ADC.default
                 inx
                 inx
ADC.default      bra     ADC.case
                 ldx     #ADC.msgdefault
                 jsr     display
ADC.stop1        stop
                 bra     ADC.stop1

ADC.casemsg      pula
                 ldd     SYS.msg
                 cpd     #CM_INIT
                 bne     ADC.defaultmsg
                 bsr     ADC.init
ADC.defaultmsg   ldx     #ADC.msgdefault
                 jsr     display
ADC.stop2        stop
                 bra     ADC.stop2

ADC.init         rts

ADC.setmode      ldx     SYS.wparam
                 ldaa    0,x
                 anda    #$3f
                 staa   ADCTL      ;aseguro no alterar mas bits que los posibles
                 rts

ADC.enablenotif  ldx     SYS.wparam      ;int BOOL enablenotif(uint URT)
                 ldx     0,x
                 cmpx   #$0
                 beq    ADC.disable
                 ldd    ADC.URT
                 cpd    #$0
                 bne   ADCenable.err1
                 stx   ADC.URT
                 clr   ADCTL      ;limpia modo del ADC
                 oraa  OPTION      ;prende ADC
                 staa  OPTION
ADC.disable      rts
                 ldd   #$0          ;limpia URT y apaga ADC
                 std   ADC.URT
                 ldaa  OPTION
                 anda  #$7f
                 staa  OPTION
ADCenable.err1   rts
                 ldd   #FALSE
                 std   intRegreso
                 ldx   #ADC.error1
                 jsr   display
                 stop
                 rts

ADC_IsReady      ldx     #FALSE
                 ldaa   ADCTL

```

```

anda      #SBO
beg       ADCIsReady.fin
ldx       #TRUE
stx       iniRegreso
rst

```

..... SCI

*Como opera SCI: SCI es dependiente del sistema en cuanto a la habilitacion
 *de los eventos. SCI llama al sistema para habilitar su interrupcion. SCI
 *identifica el tipo de evento que se genera. Por eventos SCI se entiende:
 *se envia caracter, se recibio caracter, y los que se puedan derivar de estas
 *acciones como son sobreflujo, marco, y ruido en la linea, que son de recepcion.
 *Sin embargo se puede generar un nuevo evento que es sobreflujo en transmision.
 *Este error se genera por software, es decir, solo se genera cuando se intenta
 *enviar un nuevo dato cuando no se ha desocupado el registro de transmision.
 *Puede verse que se definen dos eventos nuevos: transmision de bloque completada
 *y recepcion de bloque completada, estos eventos se generan por software y
 *no por hardware. La habilitacion de BTC se hace cuando se ha transmitido el
 *ultimo dato de una secuencia y BRC se genera cuando se ha recibido el ultimo
 *dato de una secuencia.

```

SCI.DRV      fec      "SCI.DRV"
             fdb      $00aa

SCI.tabla    fdb      #SCI.CloseComm
SCI12        fdb      #SCI.EnableNotif
SCI13        fdb      #SCI.GetError      ;int GetError();
SCI14        fdb      #SCI.GetEvMask     ;byte GetEvMask(byte Mask);
SCI15        fdb      #SCI.GetState
SCI16        fdb      #SCI.OpenComm      ;OpenComm(lpstr Dev)
SCI17        fdb      #SCI.ReadComm
SCI18        fdb      #SCI.SetEvMask
SCI19        fdb      #SCI.SetState     ;SetState(lpstructDCB DCB)
SCI1a        fdb      #SCI.XmitChar
SCI1b        fdb      #SCI.WriteComm    ;int WriteComm(char ComId,lp Buffer,char NumBytes)

```

```

*Struct DCB
{
*   uchar   Id      ;identificador
*   uchar   Baud    ;velocidad
*   uchar   Size    ;tamaño de dato
*   uchar   EvtChar ;caracter para reconocer evento
*   uchar   EofChar ;caracter para terminar evento
*   uchar   Verify  ;envia caracter recibido
*   uint    Timeout ;tiempo de espera para la respuesta
}

```

```

SCI.Id       fcb      0
SCI.Baud     fcb      0
SCI.Size     fcb      0
SCI.EvtChar  fcb      0
SCI.EofChar  fcb      0
SCI.Verify   fcb      0
SCI.Timeout  fdb      0

SCI.EV_MASK  fcb      $0      ;TDRE|TC| RDRF|IDLE|OR|ERR|BRC|BTC|
             ; 7| 6| 5| 4| 3| 2| 1| 0|

SCI.FLAGS    fcb      $0

SCI.STATUS   fcb      $0

SCI.RDR      fcb      $0

SCI.URT      fdb      $0

TX.handler   fdb      $0

```



```

RX.handler      fdb      #Rx.Handler
SCI.lasterror   fdb      $0
SCI.intRegreso  fdb      $0
SCI.handler     psha
                tay
                ldx      #SCI.tabla
                clr      0,y
                cmpb    0,y
                beq     SCI.casemsg      ;se trata de un mensaje.
                inc     0,y
                cmpb    SCI.lisdefault
                bne     pula
                ldx     0,x
                jsr     0,x
                rts
SCI.lisdefault  cmpb    #SCI_WriteComm
                bhi     SCI.default
                inx
                bra     SCI.case
SCI.default     ldx     #SCI.msgdefault
                jsr     display
SCI.stop1      stop
                bra     SCI.stop1
SCI.msgdefault  fdb      "SCI default"
                fdb      #EOB
SCI.casemsg    pula
                ldaa    SCSR2           ;se lee el status de SCI
                staa    SCI.STATUS      ;se guarda el estado en la variable
                anda    #SCI.STATUS
                psha
                anda    #0f
                beq     SCIhandler.c1   ;en el bloque siguiente se activan
                ldaa    SCI.FLAGS       ;banderas de error si hubo NF, FE u
                oraa    #ERR_FLAG       ;OR. Si es OR ademas se activa
                staa    SCI.FLAGS      ;ERR.
SCIhandler.c1  pula
                anda    #0f             ;se eliminan las banderas de NF y FE
                oraa    SCI.FLAGS       ;se aplican las banderas de BRC y BTC
                staa    SCI.FLAGS
                ldd     SYS.msg
                cmpd    #CM_INIT
                bne     SCImsg2
                bar     SCI.init
                rts
SCImsg2        cpd     #EV_SCI
                bne     SCICasedefault   ;Se trata de un evento de SCI??
                bar     SCIEv           ;si es evento SCI ve a SCIEv
SCICasedefault ldx     #SCIDefault
                jsr     display
SCIDefault     fcc     "SCI msgdefault"
                fdb     #EOB
SCIEv          ldaa    SCI.FLAGS        ;Que evento es???
                psha
                anda    #TDRE_FLAG      ;verifico evento TDRE
                beq     SCIEv_RX
                ldx     TX.handler
                cmpx    #0
                beq     SCIEv_RX        ;verifico si el receptor es SCI
                jsr     0,x
SCIev_RX       pula
                anda    #RDRF_FLAG      ;verifico evento RDRF

```

```

    beq      SCIlev.cont1
    ldab    RDR
    stab    SCI.RDR           ;se salva el caracter recibido.
    ldx     RX.handler       ;Rx.handler indica si es para SCI.
    cmpx   #$0               ;verifico si el receptor es SCI.
    beq     SCIlev.cont1    ;si no lo es envio mensaje.
SCIlev.cont1
    jsr     0,x
    ldaa   SCI.EV_MASK
    anda   SCI.FLAGS
    beq     SCIlev.fin
    ldx     SCI.URT
    cp     #$0
    beq     SCIlev.fin
    ldd    #EV_SCI
    std    SYS.msg
    pshx
    tax
    stx     SYS.wparam
    ldaa   #SYS
    ldab   #SYS_dispatch
    swi
    pulx
SCIlev.fin
    rts
SCI.init
    rts
SCI.EnableNotif
    ldy     SYS.wparam       ;int EnableNotif(int URT)
    ldx     0,y              ;obtiene URT
    cmpx   #$0              ;verifica si es deshabilitacion
    beq     SCI.disable
    stx     SCI.URT
    ldd    SYS.EV_FLAG       ;activa la bandera de SYS
    oraa   #SCIFLAG
    std    SYS.EV_FLAG
    andb   #SCIFLAG         ;pone la mascara de SCI para
    anda   #SCIFLAG        ;poner el vector
    xgdx
    pshx
    ldx     #SCI.handler
    pshx
    tax
    stx     SYS.wparam
    ldaa   #SYS
    ldab   #SYS_setevvector
    swi
    pulx
    pulx
    ldaa   SCCR2
    oraa   #S24             ;habilita Recepcion. por si llegan mensajes de tipo ESC
    staa   SCCR2
    ldd    #$0
    std    intRegreso
    rts
SCI.disable
    clr
    ldaa   SCCR2
    ldaa   SCSR2
    ldd    RDR
    ldd    SYS.EV_FLAG       ;apaga la banderas de SYS
    anda   #NOTSCIFLAG
    ldd    SYS_TV_FLAG
    ldd    #$0
    std    SCI.URT
    ldd    #$0
    std    intRegreso
    rts
SCI.GetEvMask
    ldx     SYS.wparam
    ldab   0,x              ;obtiene la mascara
    andb   SCI.FLAGS      ;verifica banderas activas con la mascara
    clra

```



```

GetError.cont2      ldd      intRegreso
                   orab    #CE_NOISE
                   std     intRegreso

GetError.fin        fcc      GetError.fin
                   ldd      intRegreso
                   orab    #CE_OVERRUN
                   std     intRegreso
                   ldaa    SCI_FLAGS
                   anda    #NOTERR_FLAG&NOTOR_FLAG
                   staa    SCI_FLAGS
                   rts

Open.err1           fcc      "No existe COM"
                   fdb     #EOB

Open.err2           fcc      "COM en uso"
                   fdb     #EOB

SCI.OpenComm        ldy      SYS.wparam      ;OpenComm(lpstr Dev)
                   ldx      0,y              ;se obtiene el puntero a Dev
                   ldaa    0,x
                   cmpa    #'S'             ;verifica de que dispositivo se trata
                   bnc     OpenComm.err1
                   ldaa    1,x
                   cmpa    #'C'
                   bnc     OpenComm.err1
                   ldaa    2,x
                   cmpa    #'I'
                   bnc     OpenComm.err1
                   ldaa    SCCR2             ;verifica si esta prendido ya.
                   psha
                   anda    #S0c
                   bnc     OpenComm.err2
                   pula
                   oraa    #S0c
                   staa    SCCR2            ;prende el SCI
                   ldaa    #S30
                   staa    BAUD             ;ajusta maxima velocidad a 9600, actual 4800
                   ldaa    SCI_FLAGS       ;por DEFINICION.
                   oraa    #BTC_FLAG
                   staa    SCI_FLAGS
                   clr     SCI_STATUS
                   cir     SCI_EV_MASK
                   ldaa    SCSR2
                   ldaa    RDR
                   ldaa    #SCI
                   ldaa    #S0
                   std     intRegreso

OpenComm.err1      ldd      #IE_HARDWARE     ;error: no se dispone.
                   std     intRegreso

                   ldx     #Open.err1
                   jsr     display

OpenComm.err2      rts
                   pula
                   ldd      #IE_OPEN
                   std     intRegreso       ;error: ya se abrio

                   ldx     #Open.err2
                   jsr     display

                   rts

Close.err1         fcc      "Error en closecomm"
                   fdb     #EOB

```



```

        idaa      2,y      ;obtiene num de bytes
        dcca
        cmpa     #S0      ;verifica si hay mas bytes por transmitir
        beq     WriteComm.fin
        staa    2,y      ;actualiza el num de bytes restantes
        inx
        stx     0,y      ;incrementa el puntero del buffer
        rts     ;salva el puntero del buffer

WriteComm.fin
        clr     0,y      ;limpia puntero y bytes
        clr     1,y
        clr     2,y
        ldd     #S0
        std     TX.handler      ;limpia vector de servicio
        idaa   SCCR2
        anda   #S3f           ;limpia TIE y TCIE
        staa   SCCR2
        idaa   SCI.FLAGS      ;activa la bandera de terminacion de trans
        oraa   #BTC_FLAG     ;mision de bloque
        staa   SCI.FLAGS
        rts

Rx.error1
        fcc     "Error 4"

Rx.Mensaje
        fdb     #EOB
        fdb     $0           ;UINT Mensaje
        fdb     $0           ;WORD wparam
        fdb     $0           ;DWORD lparam

Rx.Msg
        fcc     "!!!"
        fcc     "!!!"
        fcc     "!!!!!"

Rx.MsgFlag
        fcb     $0
Rx.Bytes
        fcb     $0
Rx.Puntero
        fdb     $0
Rx.edoprevio
        fcb     $0
Rx.Handler
        idaa   #RDRF_FLAG|ERR_FLAG|OR_FLAG|IDLE_FLAG ;limpia eventos de recepcio
        paha
        tax
        idaa   SYS.wparam      ;errores y estado Idle.
        idaa   #SCI
        idab   #SCI_GetEvMask  ;se llama la funcion para limpiar
        swi     ;las banderas del evento

        pula
        ldd     intRegreso
        andb   #ERR_FLAG|OR_FLAG ;verifico si hubo error, si asi fue
        #0    ;no se toma ninguna accion, y se
        cmpd   #S0           ;limpia el error.
        beq
        idaa   Rx.cont1
        idab   #SCI
        swi     #SCI_GetError

        idx
        jsr     intRegreso
        showTX

        idx
        stx     #Rx.Msg
        clr     Rx.Puntero
        clr     Rx.Bytes      ;limpia bytes porque se perdio el mensaje completo
        clr     Rx.MsgFlag
        idaa   Rx.edoprevio
        staa   SCCR2

Rx.cont1
        idaa   SCI.RDR      ;Verifica si es secuencia ESC
        cmpa   #ESC        ;se es ESC, se prepara para recibir
        bne
        clr     Rx.cont2    ;el mensaje, que consiste de 14 bytes.
        stx     #Rx.Bytes
        idx     #Rx.Msg
        stx     Rx.Puntero
        idaa   #S3f
        staa   Rx.MsgFlag
        idaa   SCCR2      ;se salva el estado de SCCR2 en

```

	staa	Rx.edoprevio	;Rx.edoprevio; con la finalidad de
	anda	#\$3f	;deactivar la transmision, si
	staa	SCCR2	;la hay, evitando sobreflujo.
Rx.cont2	Rta		
	ldab	Rx.MsgFlag	
	crmpb	#\$0	
Rx.cont3	beq	Rx.fin	
	ldx	Rx.Puntero	
	staa	0,x	
	inx		
	stx	Rx.Puntero	
	inc	Rx.Bytes	
	ldab	Rx.Bytes	
	crmpb	#16	
	blo	Rx.fin	
	idd	#Rx.Msg	
	std	Rx.Puntero	
	clr	Rx.Bytes	
	ldaa	Rx.MsgFlag	
	staa	Rx.edoprevio	
	ldx	SCCR2	
	ldy	#Rx.Msg	
	pbay	#Rx.Mensaje	
	bar		
	puly	Rx.ConvMsg	
	ldx		
	pahx	SVS.msg	
	ldx	SVS.wparam	
	pahx		
	ldx	SVS.lparam	
	pahx		
	ldx	SVS.lparaml	
	stx	.msg,y	
	ldx	SVS.msg	
	stx	.wparam,y	
	ldx	SVS.wparam	
	ldx	.lparam,y	
	stx	SVS.lparam	
	ldx	.lparaml,y	
	stx	SVS.lparaml	
	ldx	SCI.URT	
	pahx		
	stx		
	stx	SVS.wparam	
	ldaa	#SVS	
	ldab	#SVS_dispatch	
	swi		
	pulk		
	pulk		
	stx	SVS.lparaml	
	pulk		
	stx	SVS.lparam	
	pulk		
	stx	SVS.wparam	
	pulk		
	stx	SVS.msg	
	rts		
Rx.fin			
Rx.ConvMsg	ldab	#\$2	
RxConvMsg.c1	clrb		
	bar	RxASCToByte.c1	
	asia		
	asia		
	asia		
	paha		
	pulb		

```

inx
bar
aba
staa
iny
inx
pulb
decfb
bnc
rts
RxASCtoByte.c1
RxConvMsg.c1

RxASCtoByte.c1      ldaa    0,x
                    cmpa    #'0'
                    blo     ASCtoByte.fin
                    cmpa    #'9'
                    bhi     ASCtoByte.cont1
                    suba    #'0'
                    bra     ASCtoByte.fin
ASCtoByte.cont1     cmpa    #'F'
                    bhi     ASCtoByte.fin
                    cmpa    #'A'
                    blo     ASCtoByte.fin
                    cmpa    #'F'
                    bhi     ASCtoByte.cont2
                    suba    #$37
                    bra     ASCtoByte.fin
ASCtoByte.cont2     cmpa    #'a'
                    blo     ASCtoByte.fin
                    suba    #$57
ASCtoByte.fin       rts

SCI.fin             nop

*****            STODEV *****
STODEV.DRV         fcc     "STODEV.DRV"

STODEV.tabla       fdb     #STODEV.getPDT      :function STODEV_getPDT(lpstrName,lpStructPDT)
                    fdb     #STODEV.search      :function lp STODEV_search(lpstrName)
                    fdb     #STODEV.load        :function STODEV_load(lpstrName,lpAddr)

STODEV.handler     ldx     #STODEV.tabla
                    cmpb    #MSG
                    bnc     stodev.case1
                    bra     stodev.case0
stodev.case1       cmpb    #STODEV.getPDT
                    bnc     stodev.case2
                    ldx     0,x
                    jsr     0,x
                    rts
stodev.case2       cmpb    #STODEV.search
                    bnc     stodev.case3
                    ldx     2,x
                    jsr     0,x
                    rts
stodev.case3       cmpb    #STODEV.load
                    bnc     stodev.default
                    ldx     4,x
                    jsr     0,x
                    rts
stodev.default     rts
stodev.case0       ldd     SYS.msg
                    cmpd    #CM_INIT
                    bnc     stodev.msg2
                    bar     stodev.init
stodev.msg2        rts
                    rts

```



```

stodev.init      ldd      #STODEV.getPDT
                 std      0,x
                 ldd      #STODEV.search
                 std      2,x
                 ldd      #STODEV.load
                 std      4,x
                 rts

STODEV.getPDT   ldy      SYS.wparam      ;getPDT(lpstrName,lpStrcutPDT)
                 ldx      $2,y
                 pshx
                 tax
                 stx      SYS.wparam
                 ldaa     #STODEV
                 ldab     #STODEV_search
                 swi
                 pulx
                 ldx      intRegreso
                 cpx      #$FFF
                 bne     getPDT.cont1
                 ldd      #FFF
                 std
                 bra     getPDT.fin

getPDT.cont1    dex
getPDT.c2 inx   ;busca el final del nombre en el
                 ;archivo.
                 ldaa     0,x
                 cmpa     #00
                 bne     getPDT.c2
                 ldaa     1,x
                 cmpa     #$aa
                 bne     getPDT.c2
                 inx      ;se coloca en la PDT propiamente
                 inx
                 ldy      $0,y      ;obtiene direccion destino
                 ldab     #14      ;num de datos de PDT a transferir
                 decb
                 beq     getPDT.fin
                 ldaa     0,x
                 staa     0,y

getPDT.c3       bra     getPDT.c3
getPDT.fin      rts
getPDT.error    rts

STODEV.load     pshx      ;crea dos variables temporales
                 pshx
                 ldy      SYS.wparam      ;STODEV_load(lpMem,lpstrName)
                 ldx      $0,y      ;obtiene puntero al nombre del archivo
                 pshx
                 tax
                 stx      SYS.wparam
                 ldaa     #STODEV      ;se busca el archivo
                 ldab     #STODEV_search
                 swi
                 pulx
                 ldx      intRegreso      ;regresa puntero al archivo
                 cpx      #FFF
                 bne     load.cont5
                 ldd      #FFF
                 std
                 bra     STOload.fin

load.cont5      dex
load.cont6      sty      SYS.wparam      ;vuelve a colocar el puntero a parametros.
                 inx

```

```

        ldaa    0,x                ;busca fin de nombre
        cmpa   #$00
        bnc    load.cont6
        ldaa   load.cont6
        cmpa   1,x
        bnc    load.cont6
        inx
        inx
        tay
        stx    ProgSize,x        ;guarda puntero origen
        ldx    ProgSize,x        ;obtiene tamaño de programa
        pshx
        ldy    SYS.wparam
        ldx    $2,y              ;obtiene direccion de memoria
        tay
        iny
        iny
        stx    2,y               ;guarda en variable temporal
load.c2 pulx    ;obtiene tamaño de programa
        dex
        beq    STOload.fin       ;verifica si ha transferido todos los bytes
        pshx
        ldx    0,y               ;obtiene puntero al archivo(origen)
        ldaa   0,x               ;lee byte origen
        inx
        stx    0,y               ;salva puntero
        ldx    2,y               ;obtiene puntero destino
        staa   0,x               ;guarda en destino
        inx
        stx    2,y               ;guarda puntero destino
STOload.fin bra    load.c2
        pulx
        puls
        rts
STODEV.search ldy    SYS.wparam    ;lp STODEV_search(lpstrName)
        ldy    $0,y             ;puntero al nombre del prog.
        psxy
        ldx    #ROOTDIR        ;direccion de ROOT
search.c1 ldaa   0,x
        cmpa   #$00             ;comienza a buscar en ROM, busca
        bnc    search.cont1     ;el fin de ROOT.
        ldaa   1,x
        cmpa   #$00
        bnc    search.cont1
        ldd   #SINT             ;llego al final de ROOT sin en
        std   intRegreso        ;encontrar nada
search.cont1 rts
        ldaa   0,x
        cmpa   0,y             ;busca coincidencia con el nombre
        bnc    search.cont3
        iny
search.cont2 ldaa   0,y
        cmpb   #$0             ;busca el fin de cadena en el
        bnc    search.cont4     ;nombre.
        ldaa   1,y
        cmpb   #Saa
        bnc    search.cont4
        bra   search.cont5
search.cont3 puly
        psxy
search.cont4 inx
        bra   search.c1
search.cont5 inx
        ldaa   0,x
        cmpa   #00
        bnc    search.cont3
        ldaa   1,x
        ;verifica que efectivamente
        ;se llego al fin del nombre en
        ;ROOT, evitando coincidencias par-
        ;ciales. Si no se alcanzo el
        ;nombre exactamente sigue la bus-
        ;queda.

```

```

        cmpa      #Saa
        bne      search.con13
        inx
        inx      ;apunta a la direccion donde esta
                ;el archivo.
        ldx      0,x
        stx      intRegreso
        puly
        rts

hastaaqui      fcc      "hasta aqui"
                fdb      #EOB

```

***** Rutinas para Punto Flotante¹ *****

```

FPACC1EX      RMB      1      FLOATING POINT ACCUMULATOR #1..
FPACC1MN      RMB      3
MANTSGN1      RMB      1      MANTISSA SIGN FOR FPACC1 (0+., FF-.).
FPACC2EX      RMB      1      FLOATING POINT ACCUMULATOR #2.
FPACC2MN      RMB      3
MANTSGN2      RMB      1      MANTISSA SIGN FOR FPACC2 (0+., FF-.).
FLTSTRING     RMB      14     Cadena para conversion ASCII

```

***** Conversion ASCII a FLOAT *****

```

ASCFLT EQU *
PSHX JSR PSHPAC2      SAVE POINTER TO ASCII STRING.
LDX   LDX #0          SAVE FPACC2.
PSHX JSR #0           PUSH ZEROS ON STACK TO INITIALIZE LOCALS.
STX   STX FPACC1EX    ALLOCATE 2 BYTES FOR LOCALS.
CLR   CLR MANTSGN1    CLEAR FPACC1.
TSY   TSY             MAKE THE MANTISSA SIGN POSITIVE INITIALLY.
LDX   LDX 6,Y        POINT TO LOCALS.
LDAA  LDAA 0,X       GET POINTER TO ASCII STRING.
ASCFLT1 JSR NUMERIC   GET 1ST CHARACTER IN STRING.
BCS   BCS ASCFLT4    IS IT A NUMBER.
*     *               YES. GO PROCESS IT.
*     *
*     * LEADING MINUS SIGN ENCOUNTERED?
*     *
ASCFLT2 CMPA #'-'     NO. IS IT A MINUS SIGN?
BNE   BNE ASCFLT3    NO. GO CHECK FOR DECIMAL POINT.
COM   COM MANTSGN1   YES. SET MANTISSA SIGN. LEADING MINUS BEFORE?
INX   INX            POINT TO NEXT CHARACTER.
LDAA  LDAA 0,X       GET IT.
JSR   JSR NUMERIC    IS IT A NUMBER?
BCS   BCS ASCFLT4    YES. GO PROCESS IT.
*     *
*     * LEADING DECIMAL POINT?
*     *
ASCFLT3 CMPA #'.'     IS IT A DECIMAL POINT?
BNE   BNE ASCFLT5    NO. FORMAT ERROR.
INX   INX            YES. POINT TO NEXT CHARACTER.
LDAA  LDAA 0,X       GET IT.
JSR   JSR NUMERIC    MUST HAVE AT LEAST ONE DIGIT AFTER D.P.
BCC   BCC ASCFLT5    GO REPORT ERROR.
JMP   JMP ASCFLT11   GO BUILD FRACTION.
*     *
*     * FLOATING POINT FORMAT ERROR
*     *
ASCFLT3 INS          DE-ALLOCATE LOCALS.
JSR   JSR PULFPAC2   RESTORE FPACC2.
PULX  PULX           GET POINTER TO TERMINATING CHARACTER IN STRING.
LDAA  LDAA #FLTFMTER FORMAT ERROR.
SEC   SEC            SET ERROR FLAG.

```

¹Transcripción directa de: Motorola Inc., AN974 "MC68HC11" Floating Point Package".


```

ASCFLT12      STAA      ADDA      FPACC1EX      FINAL TOTAL PWR 10 EXPONENT.
              BRA       FPACC1EX  SAVE RESULT.
              .         FINISH    GO FINISH UP CONVERSION.
.
.
.
PRE-DECIMAL POINT NON-DIGIT FOUND, IS IT A DECIMAL POINT?
ASCFLT10      CMPA      #.        IS IT A DECIMAL POINT?
              BNE      ASCFLT7    NO. GO CHECK FOR THE EXPONENT.
              INX      .           YES. POINT TO NEXT CHARACTER.
.
.
.
POST DECIMAL POINT PROCESSING
ASCFLT11      LDAA      0,X       GET NEXT CHARACTER.
              JSR      NUMERIC?   IS IT NUMERIC?
              BCC      ASCFLT7    NO. GO CHECK FOR EXPONENT.
              BSR      ADDNXTD     YES. ADD IN THE DIGIT.
              INX      .           POINT TO THE NEXT CHARACTER.
              BCS      ASCFLT8     IF OVER FLOW, FLUSH REMAINING DIGITS.
              DEC      FPACC1EX    ADJUST THE 10 POWER EXPONENT.
              BRA      ASCFLT11    PROCESS ALL FRACTIONAL DIGITS.
.
.
.
ADDNXTD       LDAA      FPACC1MN   GET UPPER 8 BITS.
              STAA     FPACC2MN    COPY INTO FPACC2.
              LDD      FPACC1MN+1  GET LOWER 16 BITS OF MANTISSA.
              STD      FPACC2MN+1  COPY INTO FPACC2.
              LSLD     .           MULT. BY 2.
              ROL      FPACC1MN    OVERFLOW?
              BCS      ADDNXTD1    YES. DON'T ADD THE DIGIT IN.
              LSLD     .           MULT BY 4.
              ROL      FPACC1MN    OVERFLOW?
              BCS      ADDNXTD1    YES. DON'T ADD THE DIGIT IN.
              ADDD     FPACC2MN+1  BY 5.
              PSHA     .           SAVE A.
              LDAA     FPACC1MN    GET UPPER 8 BITS.
              ADCA     #0          ADDIN POSSABLE CARRY FROM LOWER 16 BITS.
              ADDA     FPACC2MN    ADD IN UPPER 8 BITS.
              STAA     FPACC1MN    SAVE IT.
              PULA     .           RESTORE A.
              BCS      ADDNXTD1    OVERFLOW? IF SO DON'T ADD IT IN.
              LSLD     .           BY 10.
              ROL      FPACC1MN
              STD      FPACC1MN+1  SAVE THE LOWER 16 BITS.
              BCS      ADDNXTD1    OVERFLOW? IF SO DON'T ADD IT IN.
              LDAB     0,X         GET CURRENT DIGIT.
              SUBB     #530        MAKE IT BINARY.
              CLRA     .           16-BIT
              ADDD     FPACC1MN+1  ADD IT IN TO TOTAL.
              STD      FPACC1MN+1  SAVE THE RESULT.
              LDAA     FPACC1MN    GET UPPER 8 BITS.
              ADCA     #0          ADD IN POSSIBLE CARRY. OVERFLOW?
              BCS      ADDNXTD1    YES. COPY OLD MANTISSA FROM FPACC2.
              STAA     FPACC1MN   NO. EVERYTHING OK.
              RTS      .           RETURN.
ADDNXTD1      LDD      FPACC2MN+1  RESTORE THE ORIGINAL MANTISSA BECAUSE
              STD      FPACC1MN+1  OF OVERFLOW.
              LDAA     FPACC2MN
              STAA     FPACC1MN
              RTS      .           RETURN.
.
.
.
.
.
.
NOW FINISH UP CONVERSION BY MULTIPLYING THE RESULTANT MANTISSA
BY 10 FOR EACH POSITIVE POWER OF 10 EXPONENT RECIEVED OR BY .1
(DIVIDE BY 10) FOR EACH NEGATIVE POWER OF 10 EXPONENT RECIEVED.
FINISH      EQU      *
              STX     6,Y         SAVE POINTER TO TERMINATING CHARACTER IN STRING.

```

	LDX	#FPACC1EX	POINT TO FPACC1.
	JSR	CHCK0	SEE IF THE NUMBER IS ZERO.
	BEQ	FINISH3	QUIT IF IT IS.
	LDAA	FPACC1EX	GET THE POWER 10 EXPONENT.
	STAA	PWR10EXP.Y	SAVE IT.
	LDAA	#\$0+24	SET UP INITIAL EXPONENT (# OF BITS + BIAS).
	STAA	FPACC1EX	
	JSR	FPNORM	GO NORMALIZE THE MANTISSA.
	TST	PWR10EXP.Y	IS THE POWER 10 EXPONENT POSITIVE OR ZERO?
	BEQ	FINISH3	IT'S ZERO, WE'RE DONE.
	BPL	FINISH1	IT'S POSITIVE MULTIPLY BY 10.
	LDX	#CONSTP1	NO. GET CONSTANT '1' (DIVIDE BY 10).
	JSR	GETFPAC2	GET CONSTANT INTO FPACC2.
	NEG	PWR10EXP.Y	MAKE THE POWER 10 EXPONENT POSITIVE.
	BRA	FINISH2	GO DO THE MULTIPLIES.
FINISH1	LDX	#CONST10	GET CONSTANT '10' TO MULTIPLY BY.
	JSR	GETFPAC2	GET CONSTANT INTO FPACC2.
FINISH2	JSR	FLTMUL	GO MULTIPLY FPACC1 BY FPACC2, RESULT IN FPACC1.
	DEC	PWR10EXP.Y	DECREMENT THE POWER 10 EXPONENT.
	BNE	FINISH2	GO CHECK TO SEE IF WE'RE DONE.
FINISH3	INS		DE-ALLOCATE LOCALS.
	INS		
	JSR	PULFPAC2	RESTORE FPACC2.
	PULX		GET POINTER TO TERMINATING CHARACTER IN STRING.
	RTS		RETURN WITH NUMBER IN FPACC1.
	.		
	NUMERIC	EQU	
	CMPA	#0	IS IT LESS THAN AN ASCII 0?
	BLO	NUMERIC1	YES. NOT NUMERIC.
	CMPA	#9	IS IT GREATER THAN AN ASCII 9?
	BHI	NUMERIC1	YES. NOT NUMERIC.
	SEC		IT WAS NUMERIC. SET THE CARRY.
	RTS		RETURN.
NUMERIC1	CLC		NON-NUMERIC CHARACTER. CLEAR THE CARRY.
	RTS		RETURN.
	.		
	FPNORM	EQU	
	LDX	#FPACC1EX	POINT TO FPACC1.
	BSR	CHCK0	CHECK TO SEE IF IT'S 0.
	BEQ	FPNORM3	YES. JUST RETURN.
	TST	FPACC1MN	IS THE NUMBER ALREADY NORMALIZED?
	BMI	FPNORM3	YES. JUST RETURN..
FPNORM1	LDD	FPACC1MN+1	GET THE LOWER 16 BITS OF THE MANTISSA.
FPNORM2	DEC	FPACC1EX	DECREMENT THE EXPONENT FOR EACH SHIFT.
	BEQ	FPNORM4	EXPONENT WENT TO 0. UNDERFLOW.
	LSLD		SHIFT THE LOWER 16 BITS.
	ROL	FPACC1MN	ROTATE THE UPPER 8 BITS. NUMBER NORMALIZED?
	BPL	FPNORM2	NO. KEEP SHIFTING TO THE LEFT.
	RTS	FPACC1MN+1	PUT THE LOWER 16 BITS BACK INTO FPACC1.
FPNORM3	CLC		SHOW NO ERRORS.
			YES. RETURN.
FPNORM4	SEC		FLAG ERROR.
	RTS		RETURN.
	.		
	CHCK0	EQU	
	PSHB		CHCK0 CHECKS FOR ZERO IN FPACC POINTED TO BY X.
	PSHA		SAVE D.
	LDD	0,X	... FPACC EXPONENT & HIGH 8 BITS.
	BNE	CHCK01	NOT ZERO. RETURN.
	LDD	2,X	CHECK LOWER 16 BITS.
CHCK01	PULA		
	PULB		RESTORE D.
	RTS		RETURN WITH CC SET.
	.		
	CONSTP1	FCB	\$7D,\$4C,\$CC,\$CD 0.1 DECIMAL
	CONST10	FCB	\$84,\$20,\$00,\$00 10.0 DECIMAL
	*****	Conversion FLOAT to ASCII	*****
	FLTASC	EQU	

	PSHX		SAVE THE POINTER TO THE STRING BUFFER.
	LDX	#FPACC1EX	POINT TO FPACC1.
	JSR	CHCK0	IS FPACC1 0?
	BNE	FLTASC1	NO. GO CONVERT THE NUMBER.
	PULX		RESTORE POINTER.
	LDD	#\$3000	GET ASCII CHARACTER + TERMINATING BYTE.
	STD	0,X	PUT IT IN THE BUFFER.
	RTS		RETURN.
FLTASC1	LDX	FPACC1EX	SAVE FPACC1.
	PSHX		
	LDX	FPACC1MN+1	
	PSHX		
	LDA	MANTSGN1	
	PSHA		
	JSR	PSHFPAC2	SAVE FPACC2.
	LDX	#0	
	PSHX		ALLOCATE LOCALS.
	PSHX		
	PSHX		SAVE SPACE FOR STRING BUFFER POINTER.
	TSY		POINT TO LOCALS.
	LDX	15,Y	GET POINTER FROM STACK.
	LDA	#\$20	PUT A SPACE IN THE BUFFER IF NUMBER NOT NEGATIVE.
	TST	MANTSGN1	IS IT NEGATIVE?
	BEQ	FLTASC2	NO. GO PUT SPACE.
	CLR	MANTSGN1	MAKE NUMBER POSITIVE FOR REST OF CONVERSION.
	LDA	#-	YES. PUT MINUS SIGN IN BUFFER.
FLTASC2	STAA	0,X	
	INX		POINT TO NEXT LOCATION.
	STX	0,Y	SAVE POINTER.
FLTASC5	LDX	#N9999999	POINT TO CONSTANT 9999999.
	JSR	GETFPAC2	GET INTO FPACC2.
	JSR	FLTCMP	COMPARE THE NUMBERS. IS FPACC1 > 9999999?
	BHI	FLTASC3	YES. GO DIVIDE FPACC1 BY 10.
	LDX	#P9999999	POINT TO CONSTANT 9999999.
	JSR	GETFPAC2	MOVE IT INTO FPACC2.
	JSR	FLTCMP	COMPARE NUMBERS. IS FPACC1 > 9999999.?
	BHI	FLTASC4	YES. GO CONTINUE THE CONVERSION.
	DEC	2,Y	DECREMENT THE MULT./DIV. COUNT.
	LDX	#CONST10	NO. MULTIPLY BY 10. POINT TO CONSTANT.
FLTASC6	JSR	GETFPAC2	MOVE IT INTO FPACC2.
	JSR	FLTMUL	
	BRA	FLTASC5	GO DO COMPARE AGAIN.
FLTASC3	INC	2,Y	INCREMENT THE MULT./DIV. COUNT.
	LDX	#CONSTP1	POINT TO CONSTANT ".1".
	BRA	FLTASC6	GO DIVIDE FPACC1 BY 10.
FLTASC4	LDX	#CONSTP5	POINT TO CONSTANT OF ".5".
	JSR	GETFPAC2	MOVE IT INTO FPACC2.
	JSR	FLTADD	ADD .5 TO NUMBER IN FPACC1 TO ROUND IT.
	LDAB	FPACC1EX	GET FPACC1 EXPONENT.
	SUBB	#\$81	TAKE OUT BIAS +1.
	NEGB		MAKE IT NEGATIVE.
	ADDB	#23	ADD IN THE NUMBER OF MANTISSA BITS -1.
	BRA	FLTASC17	GO CHECK TO SEE IF WE NEED TO SHIFT AT ALL.
FLTASC7	LSR	FPACC1MN	SHIFT MANTISSA TO THE RIGHT BY THE RESULT (MAKE
	ROR	FPACC1MN+1	THE NUMBER AN INTEGER).
	ROR	FPACC1MN+2	
	DECB		DONE SHIFTING?
FLTASC17	BNE		FLTASC7 NO. KEEP GOING.
	LDA	#1	GET INITIAL VALUE OF "DIGITS AFTER D.P." COUNT.
	STAA	3,Y	INITIALIZE IT.
	LDA	2,Y	GET DECIMAL EXPONENT.
	ADDA	#8	ADD THE NUMBER OF DECIMAL +1 TO THE EXPONENT.
			WAS THE ORIGINAL NUMBER > 9999999?
	BMI	FLTASC8	YES. MUST BE REPRESENTED IN SCIENTIFIC NOTATION.
	CMPLA	#8	WAS THE ORIGINAL NUMBER < 1?
	BHS	FLTASC8	YES. MUST BE REPRESENTED IN SCIENTIFIC NOTATION.
	DECA		NO. NUMBER CAN BE REPRESENTED IN 7 DIGITS.
	STAA	3,Y	MAKE THE DECIMAL EXPONENT THE DIGIT COUNT BEFORE
			THE DECIMAL POINT.
	LDA	#2	SETUP TO ZERO THE DECIMAL EXPONENT.

FLTASC8	SUBA	#2		SUBTRACT 2 FROM THE DECIMAL EXPONENT.
	STAA	2,Y		SAVE THE DECIMAL EXPONENT.
	TST	3,Y		DOES THE NUMBER HAVE AN INTEGER PART? (EXP. >0)
	BGT	FLTASC9		YES. GO PUT IT OUT.
	LDA	#'		NO. GET DECIMAL POINT.
	LDX	0,Y		GET POINTER TO BUFFER.
	STAA	0,X		PUT THE DECIMAL POINT IN THE BUFFER.
	INX			POINT TO NEXT BUFFER LOCATION.
	TST	3,Y		IS THE DIGIT COUNT TILL EXPONENT =0?
	BEQ	FLTASC18		NO. NUMBER IS <1
	LDA	#0'		YES. FORMAT NUMBER AS .0XXXXXXXX
	STAA	0,X		PUT THE 0 IN THE BUFFER.
	INX			POINT TO THE NEXT LOCATION.
FLTASC18	STX	0,Y		SAVE NEW POINTER VALUE.
FLTASC9	LDX	#DEC DIG		POINT TO THE TABLE OF DECIMAL DIGITS.
	LDA	#7		INITIALIZE THE THE NUMBER OF DIGITS COUNT.
	STAA	5,Y		
FLTASC10	CLR	4,Y		CLEAR THE DECIMAL DIGIT ACCUMULATOR.
FLTASC11	LDD	FPACCMN+1		GET LOWER 16 BITS OF MANTISSA.
	SUBD			SUBTRACT LOWER 16 BITS OF CONSTANT.
	STD	FPACCMN+1		SAVE RESULT.
	LDA	FPACCMN		GET UPPER 8 BITS.
	SBCA	0,X		SUBTRACT UPPER 8 BITS.
	STAA	FPACCMN		SAVE RESULT. UNDERFLOW?
	BCS	FLTASC12		YES. GO ADD DECIMAL NUMBER BACK IN.
	INC	4,Y		ADD 1 TO DECIMAL NUMBER.
	BRA	FLTASC11		TRY ANOTHER SUBTRACTION.
FLTASC12	LDD	FPACCMN+1		GET FPACCI MANTISSA LOW 16 BITS.
	ADD	1,X		ADD LOW 16 BITS BACK IN.
	STD	FPACCMN+1		SAVE THE RESULT.
	LDA	FPACCMN		GET HIGH 8 BITS.
	ADCA	0,X		ADD IN HIGH 8 BITS OF CONSTANT.
	STAA	FPACCMN		SAVE RESULT.
	LDA	4,Y		GET DIGIT.
	ADDA	#30		MAKE IT ASCII.
	PSHX			SAVE POINTER TO CONSTANTS.
	LDX	0,Y		GET POINTER TO BUFFER.
	STAA	0,X		PUT DIGIT IN BUFFER.
	INX			POINT TO NEXT BUFFER LOCATION.
	DEC	3,Y		SHOULD WE PUT A DECIMAL POINT IN THE BUFFER YET?
	BNE	FLTASC16		NO. CONTINUE THE CONVERSION.
	LDA	#'		YES. GET DECIMAL POINT.
	STAA	0,X		PUT IT IN THE BUFFER.
	INX			POINT TO THE NEXT BUFFER LOCATION.
FLTASC11	STX	0,Y		SAVE UPDATED POINTER.
	PULX			RESTORE POINTER TO CONSTANTS.
	INX			POINT TO NEXT CONSTANT.
	INX			
	INX			
	DEC	5,Y		DONE YET?
	BNE	FLTASC10		NO. CONTINUE CONVERSION OF "MANTISSA".
	LDX	0,Y		YES. POINT TO BUFFER STRING BUFFER.
FLTASC13	DEN			POINT TO LAST CHARACTER PUT IN THE BUFFER.
	LDA	0,X		GET IT.
	CMPA	#30		WAS IT AN ASCII 0?
	BEQ	FLTASC13		YES. REMOVE TRAILING ZEROS.
	INX			POINT TO NEXT AVAILABLE LOCATION IN BUFFER.
	LDAB	2,Y		DO WE NEED TO PUT OUT AN EXPONENT?
	BEQ	FLTASC15		NO. WE'RE DONE.
	LDA	#E'		YES. PUT AN 'E' IN THE BUFFER.
	STAA	0,X		
	INX			POINT TO NEXT BUFFER LOCATION.
	LDA	#+'		ASSUME EXPONENT IS POSITIVE.
	STAA	0,X		PUT PLUS SIGN IN THE BUFFER.
	TSTB			IS IT REALLY MINUS?
	BPL	FLTASC14		NO. IS'S OK AS IS.
	NEGB			YES. MAKE IT POSITIVE.
	LDA	#-'		PUT THE MINUS SIGN IN THE BUFFER.
	STAA	0,X		
FLTASC14	INX			POINT TO NEXT BUFFER LOCATION.

	STX	0,Y		SAVE POINTER TO STRING BUFFER.
	CLRA			SET UP FOR DIVIDE.
	LDX	#10		DIVIDE DECIMAL EXPONENT BY 10.
	IDIV			SAVE REMAINDER.
	PSHB			PUT QUOTIENT IN D.
	XGDX			MAKE IT ASCII.
	ADDB	#\$30		GET POINTER.
	LDX	0,Y		PUT NUMBER IN BUFFER.
	STAB	0,X		POINT TO NEXT LOCATION.
	INX			GET SECOND DIGIT.
	PULB			MAKE IT ASCII.
	ADDB	#\$30		PUT IT IN THE BUFFER.
	STAB	0,X		POINT TO NEXT LOCATION.
FLTASC15	INX	CLR	0,X	TERMINATE STRING WITH A ZERO BYTE.
	PULX			CLEAR LOCALS FROM STACK.
	PULX			
	PULX			
	JSR	PULFPAC2		RESTORE FPACC2.
	PULA			
	STAA	MANTSGN1		RESTORE FPACC1.
	PULX			
	STX	FPACC1MN+1		
	PULX			
	STX	FPACC1EX		POINT TO THE START OF THE ASCII STRING.
	PULX			RETURN.
	RTS			
.				
.				
DECDIG	EQU	.		
	FCB	\$0F,\$42,\$40		DECIMAL 1,000,000
	FCB	\$01,\$86,\$A0		DECIMAL 100,000
	FCB	\$00,\$27,\$10		DECIMAL 10,000
	FCB	\$00,\$03,\$E8		DECIMAL 1,000
	FCB	\$00,\$00,\$64		DECIMAL 100
	FCB	\$00,\$00,\$0A		DECIMAL 10
	FCB	\$00,\$00,\$01		DECIMAL 1
.				
.				
P9999999	EQU	.		CONSTANT 9999999.9
	FCB	\$94,\$74,\$23,\$FE		
.				
N9999999	EQU	.		CONSTANT 99999999.
	FCB	\$98,\$18,\$96,\$7F		
.				
CONSTP5	EQU	.		CONSTANT .5
	FCB	\$80,\$00,\$00,\$00		
.				
.				
FLTCMP	EQU	.		
	TST	MANTSGN1		IS FPACC1 NEGATIVE?
	BPL	FLTCMP2		NO. CONTINUE WITH COMPARE.
	TST	MANTSGN2		IS FPACC2 NEGATIVE?
	BPL	FLTCMP2		NO. CONTINUE WITH COMPARE.
	LDD	FPACC2EX		YES. BOTH ARE NEGATIVE SO COMPARE MUST BE DONE
	CPD	FPACC1EX		BACKWARDS. ARE THEY EQUAL SO FAR?
	BNE	FLTCMP1		NO. RETURN WITH CONDITION CODES SET.
	LDD	FPACC2MN+1		YES. COMPARE LOWER 16 BITS OF MANTISSAS.
	CPD	FPACC1MN+1		
FLTCMP1	RTS			RETURN WITH CONDITION CODES SET.
FLTCMP2	LDAA	MANTSGN1		GET FPACC1 MANTISSA SIGN.
	CMPA	MANTSGN2		BOTH POSITIVE?
	BNE	FLTCMP1		NO. RETURN WITH CONDITION CODES SET.
	LDD	FPACC1EX		GET FPACC1 EXPONENT & UPPER 8 BITS OF MANTISSA.
	CPD	FPACC2EX		SAME AS FPACC2?
	BNE	FLTCMP1		NO. RETURN WITH CONDITION CODES SET.
	LDD	FPACC1MN+1		GET FPACC1 LOWER 16 BITS OF MANTISSA.
	CPD	FPACC2MN+1		COMPARE WITH FPACC2 LOWER 16 BITS OF MANTISSA.
	RTS			RETURN WITH CONDITION CODES SET.

***** Conversion UINT & FLOAT *****

UINT2FLT	EQU	*	POINT TO FPACCI.
LDX	#FPACCIEX		IS IT 0?
CLR	1,X		clear the upper 8-bits of the mantissa. Changed 05/26/93 G.S.D.
clr	4,x		limpia el signo.
JSR	CHCK0		IS IT ALREADY 0?
BNE	UINTFLT1		NO. GO CONVERT.
RTS			YES. JUST RETURN.
UINTFLT1	LDA	#\$98	GET BIAS + NUMBER OF BITS IN MANTISSA.
STAA	FPACCIEX		INITIALIZE THE EXPONENT
JSR	FPNORM		GO MAKE IT A NORMALIZED FLOATING POINT VALUE.
CLC			NO ERRORS.
RTS			RETURN.

***** Conversion SINT & FLOAT *****

SINT2FLTEQU	CLR	MANTSGN1	initialize the FPACCI mantissa sign to zero (positive) Added 12/17/91 G.S.D.
CLR	FPACCI1MN		Clear the upper 8-bits of the FPACCI mantissa. Added 12/17/91 G.S.D.
LDD	FPACCI1MN+1		GET THE LOWER 16 BITS OF FPACCI MANTISSA.
PSHA			SAVE SIGN OF NUMBER.
BPL	SINTFLT1 IF		POSITIVE JUST GO CONVERT.
COMA			MAKE POSITIVE.
COMB			
ADDD	#1		TWO'S COMPLEMENT.
STD	FPACCI1MN+1		PUT IT BACK IN FPACCI MANTISSA.
SINTFLT1 BSR	UINT2FLT		GO CONVERT.
PULA			GET SIGN OF ORIGINAL INTEGER.
LDAB	#SFF		GET "MINUS SIGN"
TSTA			WAS THE NUMBER NEGATIVE?
BPL	SINTFLT2		NO. RETURN.
STAB	MANTSGN1		YES. SET FPACCI SIGN BYTE.
SINTFLT2 CLC			NO ERRORS.
RTS			RETURN.

***** Conversion FLOAT & UINT *****

FLT2INT	EQU	*	POINT TO FPACCI.
LDX	#FPACCIEX		IS IT 0?
JSR	CHCK0		YES. JUST RETURN.
BEQ	FLT2INT3		GET FPACCI EXPONENT.
LDAB	FPACCIEX		IS THERE AN INTEGER PART?
CMPEB	#\$81		NO. GO PUT A 0 IN FPACCI.
BLO	FLT2INT2		IS THE NUMBER NEGATIVE?
TST	MANTSGN1		YES. GO CONVERT NEGATIVE NUMBER.
BMI	FLT2INT1		IS THE NUMBER TOO LARGE TO BE MADE AN INTEGER?
CMPEB	#\$90		YES. RETURN WITH AN ERROR.
BHI	FLT2INT4		SUBTRACT THE BIAS PLUS THE NUMBER OF BITS.
SUBB	#\$98		MAKE THE NUMBER AN INTEGER.
FLT2INT5 LSR	FPACCI1MN		
ROR	FPACCI1MN+1		
ROR	FPACCI1MN+2		
INCB			DONE SHIFTING?
BNE	FLT2INT5		NO. KEEP GOING.
CLR	FPACCIEX		ZERO THE EXPONENT (ALSO CLEARS THE CARRY).
RTS			
FLT2INT1 CMPEB	#\$8F		IS THE NUMBER TOO SMALL TO BE MADE AN INTEGER?
BHI	FLT2INT4		YES. RETURN ERROR.
SUBB	#\$98		SUBTRACT BIAS PLUS NUMBER OF BITS.
BSR	FLT2INT5		GO DO SHIFT.
LDD	FPACCI1MN+1		GET RESULTING INTEGER.
COMA			MAKE IT NEGATIVE.
COMB			
ADDD	#1		TWO'S COMPLEMENT.
STD	FPACCI1MN+1		SAVE RESULT.
CLR	MANTSGN1		CLEAR MANTISSA SIGN. (ALSO CLEARS THE CARRY)
RTS			RETURN.
FLT2INT4 LDAA	#TOLGSMER		NUMBER TOO LARGE OR TOO SMALL TO CONVERT TO INT.
SEC			FLAG ERROR.
RTS			RETURN.
FLT2INT2 LDD	#0		

STD FPACC1EX ZERO FPACC1.
 STD FPACC1MN+1 (ALSO CLEARS THE CARRY)
 FLT2INT3 RTS RETURN.

***** Multiplicacion en punto flotante *****

FLTMUL EQU *
 JSR PSHFPAC2 SAVE FPACC2.
 LDX #FPACC1EX POINT TO FPACC1
 JSR CHCK0 CHECK TO SEE IF FPACC1 IS ZERO.
 BEQ FPMULT3 IT IS. ANSWER IS 0.
 LDX #FPACC2EX POINT TO FPACC2.
 JSR CHCK0 IS IT 0?
 BNE FPMULT4 NO. CONTINUE.
 CLRA CLEAR D.
 CLRBB
 STD FPACC1EX MAKE FPACC1 0.
 STD FPACC1MN+1
 BRA FPMULT3 RETURN.
 FPMULT4 LDAA MANTSGN1 GET FPACC1 EXPONENT.
 EORA MANTSGN2 SET THE SIGN OF THE RESULT.
 STAA MANTSGN1 SAVE THE SIGN OF THE RESULT.
 LDAA FPACC1EX GET FPACC1 EXPONENT.
 ADDA FPACC2EX ADD IT TO FPACC2 EXPONENT.
 BPL FPMULT1 IF RESULT IS MINUS AND
 BCC FPMULT2 THE CARRY IS SET THEN:
 FPMULT5 LDAA #OVFERR OVERFLOW ERROR.
 SEC SET ERROR FLAG.
 BRA FPMULT6 RETURN.
 FPMULT1 BCS FPMULT2 IF RESULT IS PLUS & THE CARRY IS SET THEN ALL OK.
 LDAA #UNFERR ELSE UNDERFLOW ERROR OCCURED.
 SEC FLAG ERROR.
 BRA FPMULT6 RETURN.
 FPMULT2 LDAA #S80 ADD 128 BIAS BACK IN THAT WE LOST.
 STAA FPACC1EX SAVE THE NEW EXPONENT.
 JSR UMULT GO MULTIPLY THE "INTEGER" MANTISSAS.
 TST FPACC1EX WAS THERE AN OVERFLOW ERROR FROM ROUNDING?
 BEQ FPMULT5 YES. RETURN ERROR.
 FPMULT3 CLC SHOW NO ERRORS. Moved the FPMULT3 label to this instruction from the TST
 instruction above. G.S.D. 12/20/91
 FPMULT6 JSR PULFPAC2 RESTORE FPACC2.
 RTS
 *
 *
 UMULT EQU *
 LDX #0
 PSHX CREATE PARTIAL PRODUCT REGISTER AND COUNTER.
 PSHX POINT TO THE VARIABLES.
 TSY SET COUNT TO THE NUMBER OF BITS.
 LDAA #24
 STAA 0,X
 UMULT1 LDAA FPACC2MN+2 GET THE L.S. BYTE OF THE MULTIPLIER.
 LSRA PUT L.S. BIT IN CARRY.
 BCC UMULT2 IF CARRY CLEAR, DON'T ADD MULTIPLICAND TO P.P.
 LD DD FPACC1MN+1 GET MULTIPLICAND L.S. 16 BITS.
 ADDD ADD TO PARTIAL PRODUCT.
 STD 2,X SAVE IN P.P.
 LDAA FPACC1MN GET UPPER 8 BITS OF MULTIPLICAND.
 ADCA 1,X ADD IT W/ CARRY TO P.P.
 STAA 1,X SAVE TO PARTIAL PRODUCT.
 UMULT2 ROR 1,X ROTATE PARTIAL PRODUCT TO THE RIGHT.
 ROR 2,X
 ROR 3,X
 ROR FPACC2MN SHIFT THE MULTIPLIER TO THE RIGHT 1 BIT.
 ROR FPACC2MN+1
 ROR FPACC2MN+2
 DEC 0,X
 BNE UMULT1 DONE YET?
 TST 1,X NO. KEEP GOING.
 BMI UMULT3 DOES PARTIAL PRODUCT NEED TO BE NORMALIZED?
 NO. GET ANSWER & RETURN.

	LSL	FPACC2MN	GET BIT THAT WAS SHIFTED OUT OF P.P REGISTER.
	ROL	3,X	PUT IT BACK INTO THE PARTIAL PRODUCT.
	ROL	2,X	
	ROL	1,X	
	DEC	FPACC1EX	FIX EXPONENT.
UMULT3	TST	FPACC2MN	DO WE NEED TO ROUND THE PARTIAL PRODUCT?
	BPL	UMULT4	NO. JUST RETURN.
	LDD	2,X	YES. GET THE LEAST SIGNIFIGANT 16 BITS.
	ADDD	#1	ADD 1.
	STD	2,X	SAVE RESULT.
	LDAA	1,X	PROPAGATE THROUGH.
	ADCA	#0	
	STAA	1,X	
	BCC	UMULT4	IF CARRY CLEAR ALL IS OK.
	ROR	1,X	IF NOT OVERFLOW. ROTATE CARRY INTO P.P.
	ROR	2,X	
	ROR	3,X	
UMULT4	INC	FPACC1EX	UP THE EXPONENT.
	INS		TAKE COUNTER OFF STACK.
	PULX		GET M.S. 16 BITS OF PARTIAL PRODUCT.
	STX	FPACC1MN	PUT IT IN FPACC1.
	PULA		GET L.S. 8 BITS OF PARTIAL PRODUCT.
	STAA	FPACC1MN+2	PUT IT IN FPACC1.
	RTS		RETURN.
***** Suma en punto flotante *****			
FLTADD	EQU	*	
	JSR	PSHFPAC2	SAVE FPACC2.
	LDX	#FPACC2EX	POINT TO FPACC2
	JSR	CHCK0	IS IT ZERO?
	BNE	FLTADD1	NO. GO CHECK FOR 0 IN FPACC1.
FLTADD6	CLC		NO ERROR.
FLTADD10	JSR		PULFPAC2 RESTORE FPACC2.
	RTS		ANSWER IN FPACC1. RETURN.
FLTADD1	LDX	#FPACC1EX	POINT TO FPACC1.
	JSR	CHCK0	IS IT ZERO?
	BNE	FLTADD2	NO. GO ADD THE NUMBER.
FLTADD4	LDX	FPACC2EX	ANSWER IS IN FPACC2. MOVE IT INTO FPACC1.
	STD	FPACC1EX	
	LDD	FPACC2MN+1	MOVE LOWER 16 BITS OF MANTISSA.
	STD	FPACC1MN+1	
	LDAA	MANTSGN2	MOVE FPACC2 MANTISSA SIGN INTO FPACC1.
	STAA	MANTSGN1	
	BRA	FLTADD6	RETURN.
FLTADD2	LDAA	FPACC1EX	GET FPACC1 EXPONENT.
	CMPA	FPACC2EX	ARE THE EXPONENTS THE SAME?
	BEQ	FLTADD7	YES. GO ADD THE MANTISSA'S.
	SUBA	FPACC2EX	NO. FPACC1EX-FPACC2EX. IS FPACC1 > FPACC2?
	BPL	FLTADD3	YES. GO CHECK RANGE.
	NEGA		NO. FPACC1 < FPACC2. MAKE DIFFERENCE POSITIVE.
	CMFA	#23	ARE THE NUMBERS WITHIN RANGE?
	BHI	FLTADD4	NO. FPACC2 IS LARGER. GO MOVE IT INTO FPACC1.
	TAB		PUT DIFFERENCE IN B.
	ADDB	FPACC1EX	CORRECT FPACC1 EXPONENT.
	STAB	FPACC1EX	SAVE THE RESULT.
	LDX	#FPACC1MN	POINT TO FPACC1 MANTISSA.
	BRA	FLTADD5	DENORMALIZE FPACC1 FOR THE ADD.
FLTADD3	CMFA	#23	ARE THE NUMBERS WITHIN RANGE?
	BHI	FLTADD6	NO. ANSWER ALREADY IN FPACC1. JUST RETURN.
	LDX	#FPACC2MN	POINT TO THE MANTISSA TO DENORMALIZE.
FLTADD5	LSR	0,X	SHIFT THE FIRST BYTE OF THE MANTISSA.
	ROR	1,X	THE SECOND.
	ROR	2,X	AND THE THIRD.
	DECA		DONE YET?
	BNE	FLTADD5	NO. KEEP SHIFTING.
FLTADD7	LDAA	MANTSGN1	GET FPACC1 MANTISSA SIGN.
	CMFA	MANTSGN2	ARE THE SIGNS THE SAME?
	BEQ	FLTADD11	YES. JUST GO ADD THE TWO MANTISSAS.
	TST	MANTSGN1	NO. IS FPACC1 THE NEGATIVE NUMBER?
	BPL	FLTADD8	NO. GO DO FPACC1-FPACC2.

	BSR	FLTSUB1	INVERT SIGN.
	JSR	FLTADD	GO DO FLOATING POINT ADD.
FLTSUB1	LDAA	MANTSGN2	GET FPACC2 MANTISSA SIGN.
	EORA	#SF	INVERT THE SIGN.
	STAA	MANTSGN2	PUT BACK.
	RTS		RETURN.
***** Division en punto flotante *****			
FLTDIV	EQU	*	
	LDX	#FPACC2EX	POINT TO FPACC2.
	JSR	CHKDIV0	IS THE DIVISOR 0?
	BNE	FLTDIV1	NO. GO SEE IF THE DIVIDEND IS ZERO.
	LDAA	#DIVOERR	YES. RETURN A DIVIDE BY ZERO ERROR.
	SEC		FLAG ERROR.
	RTS		RETURN.
FLTDIV1	LDX	#FPACC1EX	POINT TO FPACC1.
	JSR	CHK0	IS THE DIVIDEND 0?
	BNE	FLTDIV2	NO. GO PERFORM THE DIVIDE.
	CLC		YES. ANSWER IS ZERO. NO ERRORS.
	RTS		RETURN.
FLTDIV2	JSR	PSHFPAC2	SAVE FPACC2.
	LDAA	MANTSGN2	GET FPACC2 MANTISSA SIGN.
	EORA	MANTSGN1	SET THE SIGN OF THE RESULT.
	STAA	MANTSGN1	SAVE THE RESULT.
	LDX	#0	SET UP WORK SPACE ON THE STACK.
	PSHX		
	PSHX		
	PSHX		
	LDAA	#24	PUT LOOP COUNT ON STACK.
	PSHA		
	TSX		SET UP POINTER TO WORK SPACE.
	LDD	FPACC1MN	COMPARE FPACC1 & FPACC2 MANTISSAS.
	CPD	FPACC2MN	ARE THE UPPER 16 BITS THE SAME?
	BNE	FLTDIV3 NO.	YES. COMPARE THE LOWER 8 BITS.
	LDAA	FPACC1MN+2	
	CMPA	FPACC2MN+2	
FLTDIV3	BIHS	FLTDIV4	IS FPACC2 MANTISSA > FPACC1 MANTISSA? NO.
	INC	FPACC2EX	ADD 1 TO THE EXPONENT TO KEEP NUMBER THE SAME.
	*		; DID OVERFLOW OCCUR?
	BNE	FLTDIV14	NO. GO SHIFT THE MANTISSA RIGHT 1 BIT.
FLTDIV8	LDAA	#OVFERR	YES. GET ERROR CODE.
	SEC		FLAG ERROR.
FLTDIV6	PULX		REMOVE WORKSPACE FROM STACK.
	PULX		
	INS		
	JSR	PULFPAC2	RESTORE FPACC2.
	RTS		RETURN.
FLTDIV4	LDD	FPACC1MN+1	DO AN INITIAL SUBTRACT IF DIVIDEND MANTISSA IS
	SUBD	FPACC2MN+1	GREATER THAN DIVISOR MANTISSA.
	STD	FPACC1MN+1	
	LDAA	FPACC1MN	
	SBCA	FPACC2MN	
	STAA	FPACC1MN	
	DEC	0,X	SUBTRACT 1 FROM THE LOOP COUNT.
FLTDIV14	LSR	FPACC2MN	SHIFT THE DIVISOR TO THE RIGHT 1 BIT.
	ROR	FPACC2MN+1	
	ROR	FPACC2MN+2	
	LDAA	FPACC1EX	GET FPACC1 EXPONENT.
	LDAB	FPACC2EX	GET FPACC2 EXPONENT.
	NEGB		ADD THE TWO'S COMPLEMENT TO SET FLAGS PROPERLY.
	ABA		
	BMI	FLTDIV5	IF RESULT MINUS CHECK CARRY FOR POSS. OVERFLOW.
	BCS	FLTDIV7	IF PLUS & CARRY SET ALL IS OK.
	LDAA	#UNFERR	IF NOT, UNDERFLOW ERROR.
	BRA	FLTDIV6	RETURN WITH ERROR.
FLTDIV5	BCS	FLTDIV8	IF MINUS & CARRY SET OVERFLOW ERROR.
FLTDIV7	ADDA	#S1	ADD BACK BIAS+1 (THE '1' COMPENSATES FOR ALGOR.)
	STAA	FPACC1EX	SAVE RESULT.
FLTDIV9	LDD	FPACC1MN	SAVE DIVIDEND IN CASE SUBTRACTION DOESNT GO.

STD	4,X	
LDAA	FPACC1MN+2	
STAA	6,X	
LDD	FPACC1MN+1	GET LOWER 16 BITS FOR SUBTRACTION.
SUBD	FPACC2MN-1	SAVE RESULT.
STD	FPACC1MN-1	GET HIGH 8 BITS.
LDAA	FPACC1MN	
SBCA	FPACC2MN	
STAA	FPACC1MN	
BPL	FLTDIV10	SUBTRACTION WENT OK. GO DO SHIFTS.
LDD	4,X	RESTORE OLD DIVIDEND.
STD	FPACC1MN	
LDAA	6,X	
STAA	FPACC1MN+2	
FLTDIV10ROL	3,X	ROTATE CARRY INTO QUOTIENT.
ROL	2,X	
ROL	1,X	
LRL	FPACC1MN+2	SHIFT DIVIDEND TO LEFT FOR NEXT SUBTRACT.
ROL	FPACC1MN-1	
ROL	FPACC1MN	
DEC	0,X	DONE YET?
BNE	FLTDIV9	NO. KEEP GOING.
COM	1,X	RESULT MUST BE COMPLEMENTED.
COM	2,X	
COM	3,X	
LDD	FPACC1MN+1	DO 1 MORE SUBTRACT FOR ROUNDING.
SUBD	FPACC2MN+1	(DON'T NEED TO SAVE THE RESULT.)
LDAA	FPACC1MN	(NO NEED TO SAVE THE RESULT.)
SBCA	FPACC2MN	GET LOW 16 BITS.
LDD	2,X	IF IT DIDN'T GO RESULT OK AS IS.
BCC	FLTDIV11	CLEAR THE CARRY.
CLC		GO SAVE THE NUMBER.
BRA	FLTDIV13	ROUND UP BY 1.
FLTDIV11ADDD	#1	PUT IT IN FPACCI.
FLTDIV13STD	FPACC1MN+1	GET HIGH 8 BITS.
LDAA	1,X	
ADCA	#0	SAVE RESULT.
STAA	FPACC1MN	CARRY CLEAR ANSWER OK.
BCC	FLTDIV12IF	IF NOT OVERFLOW. ROTATE CARRY IN.
ROR	FPACC1MN	
ROR	FPACC1MN+1	
ROR	FPACC1MN+2	
INC	FPACCIEX	Compensate the exponent for rotate right. Added 12/17/91 G.S.D.
BNE	FLTDIV12	if the exponent didn't go to zero, the answer's OK.
JMP	FLTDIV8	if not an overflow occurred.
FLTDIV12CLC		NO ERRORS.
JMP	FLTDIV6	RETURN.
***** Raiz cuadrada en punto flotante *****		
FLTSQR	EQU	
LDX	#FPACCIEX	POINT TO FPACCI.
JSR	CHK0	IS IT ZERO?
BNE	FLTSQR1	NO. CHECK FOR NEGATIVE.
RTS		YES. RETURN.
FLTSQR1	TST	IS THE NUMBER NEGATIVE?
BPL	FLTSQR2	NO. GO TAKE ITS SQUARE ROOT.
LDAA	#NSQRTERR	YES. ERROR.
SEC		FLAG ERROR.
RTS		RETURN.
FLTSQR2	JSR	SAVE FPACC2.
LDAA	#4	GET ITERATION LOOP COUNT.
PSHA		SAVE IT ON THE STACK.
LDX	FPACC1MN+1	SAVE INITIAL NUMBER.
PSHX		
LDX	FPACCIEX	
PSHX		
TSY		POINT TO IT.
BSR	TFR1TO2	TRANSFER FPACC1 TO FPACC2.
LDAA	FPACC2EX	GET FPACCI EXPONENT.
SUBA	#580	REMOVE BIAS FROM EXPONENT.

COM	MANTSGN1	YES. SET SIGN TO NEGATIVE.
GETFP11	#S80	RESTORE MOST SIGNIFICANT BIT IN MANTISSA.
ORAB	FPACC1EX	PUT IN FPACC1.
GETFP12	2,X	GET LOW 16-BITS OF THE MANTISSA.
STD	FPACC1MN+1	PUT IN FPACC1.
STD		RETURN.
RTS		
.		
.		
GETFPAC2	EQU	
LDD	0,X	GET THE EXPONENT & HIGH BYTE OF THE MANTISSA.
BEQ	GETFP22	IF NUMBER IS 0, SKIP SETTING THE MS BIT.
CLR	MANTSGN2	SET UP FOR POSITIVE NUMBER.
TSTB		IS NUMBER NEGATIVE?
BPL	GETFP21	NO. LEAVE SIGN ALONE.
COM	MANTSGN2	YES. SET SIGN TO NEGATIVE.
GETFP21	#S80	RESTORE MOST SIGNIFICANT BIT IN MANTISSA.
ORAB	FPACC2EX	PUT IN FPACC1.
GETFP22	2,X	GET LOW 16-BITS OF THE MANTISSA.
LDD	FPACC2MN+1	PUT IN FPACC1.
STD		RETURN.
STD		
RTS		
PUTFPAC1	EQU	
LDD	FPACC1EX	GET FPACC1 EXPONENT & UPPER 8 BITS OF MANT.
TST	MANTSGN1	IS THE NUMBER NEGATIVE?
BMI	PUTFP11	YES. LEAVE THE M.S. BIT SET.
ANDB	#57F	NO. CLEAR THE M.S. BIT.
PUTFP11	0,X	SAVE IT IN MEMORY.
STD	FPACC1MN+1	GET L.S. 16 BITS OF THE MANTISSA.
LDD		
STD	2,X	RETURN.
RTS		
.		
.		
PUTFPAC2	EQU	
LDD	FPACC2EX	GET FPACC1 EXPONENT & UPPER 8 BITS OF MANT.
TST	MANTSGN2	IS THE NUMBER NEGATIVE?
BMI	PUTFP21	YES. LEAVE THE M.S. BIT SET.
ANDB	#57F	NO. CLEAR THE M.S. BIT.
PUTFP21	0,X	SAVE IT IN MEMORY.
STD	FPACC2MN+1	GET L.S. 16 BITS OF THE MANTISSA.
LDD		
STD	2,X	RETURN.
RTS		
FP2ACC1	pshx	
	ldy	#FPACC1EX
	ldab	#5
FP2ACC1.c1	ldaa	0,x
	staa	0,y
	inx	
	iny	
	decb	
	bnc	FP2ACC1.c1
	pulx	
	rts	
FP2ACC2	pshx	
	ldy	#FPACC2EX
	ldab	#5
FP2ACC2.c1	ldaa	0,x
	staa	0,y
	inx	
	iny	
	decb	
	bne	FP2ACC2.c1
	pulx	
	rts	
ACC2MEM	pshx	
	ldy	#FPACC2EX
	ldab	#5
ACC2MEM.c1	ldaa	0,y
	staa	0,x

```
inx
iny
decb
bnc
puls
rts
ACC2MEM.c1
```

ACC1MEM

```
pahx
idy
ldab
ldaa
staa
inx
iny
decb
bnc
puls
rts
#FPACC1EX
#5
0,y
0,x
```

ACC1MEM.c1

ACC1MEM.c1

BIOS.fin

nop

Apéndice F . Programa del sistema HC11 para la obtención de las curvas de permeabilidad.

* REGISTROS ESPECIALES DEL HC11 E1

DDRAM	equ	\$6001
IRRAM	equ	\$6000
HPRIO	equ	\$103c
TMSK2	equ	\$1024
TFLG2	equ	\$1025
PACTL	equ	\$1026
ADCTL	equ	\$1030
ADR1	equ	\$1031
ADR2	equ	\$1032
ADR3	equ	\$1033
ADR4	equ	\$1034
OPTION	equ	\$1039
INIT	equ	\$103d
CONFIG	equ	\$103f
PORTD	equ	\$1008
DDR0	equ	\$1009
SPCR	equ	\$1028
BAUD	equ	\$102B
SCCR1	equ	\$102C
SCCR2	equ	\$102D
SCSR2	equ	\$102E
SCDR	equ	\$102F
TDR	equ	\$102F
RDR	equ	\$102F
BPROT	equ	\$1035
SYSREG	equ	\$1000
SYSREGEN0	equ	\$1040

* ZONA DE VECTORES, MODO SINGLE(PSEUDOVECTORES) Y EXTENDIDO

CMFVECTOR	equ	\$fe	;\$ffe	\$fc
COPVECTOR	equ	\$fb	;\$ffa	\$fb
ILLVECTOR	equ	\$f8	;\$ff8	\$F8
SWIVECTOR	equ	\$f5	;\$ff6	\$F5
XIRQVECTOR	equ	\$f2	;\$ff4	\$f2
IRQVECTOR	equ	\$ef	;\$ff2	\$ej
RTIVECTOR	equ	\$ec	;\$ff0	\$ec
TIC1VECTOR	equ	\$e9	;\$ffe	\$e9
TIC2VECTOR	equ	\$e6	;\$ffc	\$e6
TIC3VECTOR	equ	\$e3	;\$ffa	\$e3
TOC1VECTOR	equ	\$e0	;\$ff8	\$e0
TOC2VECTOR	equ	\$dd	;\$ff6	\$dd
TOC3VECTOR	equ	\$da	;\$ff4	\$da
TOC4VECTOR	equ	\$d7	;\$ff2	\$d7
TOC5VECTOR	equ	\$d4	;\$ff0	\$d4
TOIVECTOR	equ	\$d1	;\$ffde	\$D1
PAOVECTOR	equ	\$cc	;\$ffdc	\$cc
PAIVECTOR	equ	\$cb	;\$ffdc	\$cb

```

SPIVECTOR equ $c8 ;$ffd8 $c8
SCIVECTOR equ $c5 ;$ffd6 $C5

*CMFVECTOR equ $ffc ;Vectores para modo extendido.
*COPVECTOR equ $ffa
*ILLVECTOR equ $ff8
*SWIVECTOR equ $ff6
*XIRQVECTOR equ $ff4
*IRQVECTOR equ $ff2
*RTIVECTOR equ $ff0
*TIC1VECTOR equ $ffe
*TIC2VECTOR equ $ffc
*TIC3VECTOR equ $ffa
*TOC1VECTOR equ $ffc8
*TOC2VECTOR equ $ffc6
*TOC3VECTOR equ $ffc4
*TOC4VECTOR equ $ffc2
*TOC5VECTOR equ $ffc0
*TOIVECTOR equ $ffd8
*PAOVECTOR equ $ffd8
*PAIVECTOR equ $ffda
*SPIVECTOR equ $ffd8
*SCIVECTOR equ $ffd6

ESC equ $1b
EOT equ $4
EOB equ $00AA
MOTOR equ $8000

ORG $2000

lds #$f000
ldaa #$06
staa IRRAM
jsr sleep
ldaa #$0c
staa IRRAM ;Configura pantalla LCD.
jsr sleep
ldaa #$38
staa IRRAM
jsr sleep
ldaa #$40
staa IRRAM
jsr sleep
ldaa #1
staa IRRAM
jsr sleep
ldaa #$31
staa DDRAM
jsr sleep
ldx #Mensajc1
jsr display

ldaa OPTION
oraa #$80
staa OPTION

```

	ldaa	#\$30	
	staa	BAUD	
	ldaa	#\$0	
	oraa	#\$0C	
	staa	SCCR2	
ciclo0	clra		
	staa	MOTOR	
	ldx	#Esperando	
ciclo6	jsr	display	
	ldab	SCSR2	;Espera recibir el primer 1b de la PC. Una vez recibido
	andb	#\$20	;empieza a enviar.
	beq	ciclo6	
	ldab	RDR	
	cmpb	#\$1b	
	beq	cont1	
	bra	ciclo6	
ciclo1	ldab	SCSR2	;Verifica si llego un 1b para
	andb	#\$20	;poner la cuenta a ceros y empezar de nuevo.
	beq	cont2	
	ldab	RDR	
	cmpb	#\$1b	
	beq	cont1	
	cmpb	#\$4	
	beq	ciclo0	
cont1	bra	cont2	
cont2	clra		
	staa	MOTOR	
	psha		
	jsr	SYsformathex	
	ldy	0,y	
	sty	Voltaje	
	jsr	sleep	
	jsr	sleep	
	jsr	sleep	
ciclo5	ldab	#10	
	pshb		
	ldaa	#\$10	
ciclo2	staa	ADCTL	
	ldaa	ADCTL	
	anda	#\$80	
	beq	ciclo2	
	ldy	#Dato1	
	ldx	#ADR1	
	pshx		
	jsr	TESconversion	
	pulx		
	ldy	#Gasto	
	jsr	TESconversion	
	ldx	#ADR2	
	ldy	#Dato2	
	pshx		
	jsr	TESconversion	
	pulx		

```

ldy      #Presion
jsr      TESconversion
tsx
inx
ldy      #Numero
jsr      TESconversion
ldx      #Gasto
ldab     #13
ciclo4   ldaa    0,x
ciclo3   ldaa    TDR
         andaa   SCSR2
         beq     #$40
         inx
         decb
         bnc    ciclo4
         pulb
         decb
         bnc    ciclo5
         ldaa   #EOT
ciclo7   staa   TDR
         ldaa   SCSR2
         andaa  #$40
         beq   ciclo7
         ldx   #Mensaje1
         jsr   display
         pula
         inca
         cmpa  #$00
         beq  cont3
ciclo3   jmp   ciclo1
         ldx  #FindePrueba
         jsr  display
         jsr  sleep
         jmp  ciclo0

SYSCadena   fcc    "00"
SYSSformatex  fdb    #EOB
SYSSformat.c1 ldy    #SYSCadena ;Prepara un hexadccimal para ASCII.
             psha
             pula
             rora
             rora
             rora
             rora
             anda   #$0f
             bsr   SYSSconversion
             pula
             anda   #$0f
             iny
             bsr   SYSSconversion
SYSSformat.fin  dcy
               rts

```

```

SYSconversion  cmpa  #$9          ;Convierte el valor hexadecimal a decimal ASCII.
                bls   SYSconver.co1
                suba  #$9
                adda  #$40
                staa  0,y
SYSconver.co1  rts
                adda  #$30
                staa  0,y
                rts

TESconversion  pshx
                ldaa  0,x          ;Carga el dato a convertir y lo guarda en pila.
                psha
                psha          ;Crea variable temporal con el numero a convertir.
                tsx
                clr   0,y          ;unidades
                clr   1,y          ;decenas
                clr   2,y          ;centenas
                cmpa  #0          ;Verifica si el dato fuente es cero,
                beq   TESconver.fin2 ;si es cero no hace nada
                clra
                clrb
TESconver.c1   addb  #$0a
                cmpb  0,x
                bhi   TESconver.fin1
                cmpb  #$4          ;Está convirtiendo un 0xFF.
                beq   TESconver.fin1
                inc   1,y          ;Incrementa decenas.
                ldaa  1,y          ;Verifica se se ha generado acarreo
                cmpa  #$0a          ;en centenas.
                blo   TESconver.c1
                inc   0,y          ;Incrementa centenas.
                clr   1,y
                bra   TESconver.c1
TESconver.fin1 subb  #$a
                stab  0,x
                ldaa  1,x          ;Obtiene el numero a convertir.
                suba  0,x
                staa  2,y
TESconver.fin2 ldaa  2,y
                adda  #$30
                staa  2,y
                ldaa  1,y
                adda  #$30
                staa  1,y
                ldaa  0,y
                adda  #$30
                staa  0,y
                pula
                pula
                pulx
                rts

sleep  ldy  #$3d1          ;1 segundo
        ldab #$ff

```

sleep.ciclo	dec b bcq bra	sleep.cont1 sleep.ciclo	;Ciclo toma 2037 ciclos
sleep.cont1	dey beq bra	sleep.fin sleep.ciclo	;Cont1 toma RegY veces 2037 mas RegY ;veces 7 ciclos más (RegY - 1) ;tres ciclos. Ecuación: $2037y+7y+3(y-1)=$;2 M. $y=977.04$ (3D1hex).
sleep.fin	rts		
display	ldaa staa	#S01 IRRAM	;Desplegado de datos en la pantalla.
ddisplay.ciclo1	bsr ldaa cmpa bnc ldab cmpb bnc rts	despera 0,x #\$0 ddisplay.cont 1,x #\$AA ddisplay.cont	
ddisplay.cont	staa bsr inx bra	DDRAM despera ddisplay.ciclo1	
despera	ldaa anda bnc rts	IRRAM #\$80 despera	
Mensaje1	fcc	"D1:"	
Dato1	fcc	"000 "	
Dato2	fcc	"D2:"	
Voltaje	fcc	"000 "	
Gasto	fcc	"fT"	
Presion	fcc	#EOB	
Numero	fcc	"000"	
FindcPrueba	fcc	#S20	
Esperando	fcc	"000"	
	fcc	#S0d	
	fcc	#S0a	
	fcc	"Fin de Prueba"	
	fdb	#EOB	
	fcc	"Esperando..."	
	fdb	#EOB	

Apéndice G . Programa para Windows.

1. Cuerpo principal, prueba2.cpp.

```

#include <owl\applicat.h>
#include <owl\framewin.h>
#include <owl\dialog.h>
#include <owl\button.h>
#include <owl\radiobut.h>
#include <owl\combobox.h>
#include <owl\groupbox.h>
#include <owl\menu.h>
#include <owl\static.h>
#include <owl\diobjec.h>
#include <owl>window.h>
#include <owl\control.h>
#include <string.h>
#include <c:\bc4\windows\prueba2.h>

#define NUMERO_DE_DATOS 14

#include <c:\bc4\windows\comm.cpp>
#include <c:\bc4\windows\config.cpp>
#include <c:\bc4\windows\iniciar.cpp>
#include <c:\bc4\windows\sistema.cpp>

class TMenuItemEnabler:public TCommandEnabler
{
public:
TMenuItemEnabler(UINT Id,HWND hwndReceiver=0);
TCommandEnabler(Id,hwndReceiver){}

void SetCheck(int){};
void SetText(LPCSTR){};
};

class TVentanaPrincipal:public TWindow
{
TComm *PuertoSerial;
TMenuItemEnabler *ItemEnabler;
public:
TVentanaPrincipal(HINSTANCE hInstance,TWindow* parent=0)
{
Init(parent,0,0);
PuertoSerial=new TComm(140,140,EOT_CHAR,0);
ItemEnabler=new TMenuItemEnabler(CM_INICIAR_MEDICION,HWindow);
}
~TVentanaPrincipal()
{
delete PuertoSerial;
delete ItemEnabler;
}

void SetupWindow();
void CmArchivoDestino();
void CmVerArchivo();
void CmIniciarMediccion();
void CmIniciarMediccion(TCommandEnabler&);
void CmSistemaHC1();
void CmConfiguracionPC();
void CmHelp();
void CmVersion();
void CmSalir();
BOOL CanClose();

DECLARE_RESPONSE_TABLE(TVentanaPrincipal);

```

```

};
DEFINE_RESPONSE_TABLE1(TVentanaPrincipal, TWindow)
    EV_COMMAND(CM_ARCHIVO_DESTINO, CmArchivoDestino),
    EV_COMMAND(CM_VER_ARCHIVO, CmVerArchivo),
    EV_COMMAND(CM_INICIAR_MEDICION, CmIniciarMedicion),
    EV_COMMAND(CM_SISTEMA_HC11, CmSistemaHC11),
    EV_COMMAND(CM_CONFIGURACION_PC, CmConfiguracionPC),
    EV_COMMAND(CM_VERSION, CmVersion),
    EV_COMMAND(CM_AYUDA, CmHelp),
    EV_COMMAND(CM_SALIR, CmSalir),
    EV_COMMAND_ENABLE(CM_INICIAR_MEDICION, CmElIniciarMedicion),
END_RESPONSE_TABLE;

void TVentanaPrincipal::CmSalir()
{
    PuertoSerial->Close();
    Destroy();
}

BOOL TVentanaPrincipal::CanClose()
{
    PuertoSerial->Close();
    return TRUE;
}

void TVentanaPrincipal::CmElIniciarMedicion(TCommandEnabler&)
{
    ItemEnabler->Enable(FALSE);
}

void TVentanaPrincipal::SetupWindow()
{
    TWindow::SetupWindow();
    PuertoSerial->Inicializar();
}

void TVentanaPrincipal::CmHelp()
{
    MessageBox("Opcion de Ayuda");
}

void TVentanaPrincipal::CmArchivoDestino()
{
}

void TVentanaPrincipal::CmVerArchivo()
{
}

void TVentanaPrincipal::CmIniciarMedicion()
{
    if(PuertoSerial->Id()<0)
        switch(PuertoSerial->Id())
        {
            case IE_HARDWARE:
                MessageBox(strcat("No existe puerto:", PuertoSerial->Port), "ERROR", MB_ICONSTOP);
                break;
            case IE_OPEN:
                MessageBox(strcat("Puerto en uso:", PuertoSerial->Port), "ERROR!", MB_ICONSTOP);
                break;
            case IE_NOOPEN:
                MessageBox(strcat("No se pudo abrir:", PuertoSerial->Port), "ERROR!", MB_ICONSTOP);
                break;
            case IE_MEMORY:
                MessageBox("No hay memoria", "ERROR!", MB_ICONSTOP);
                break;
            default:
                MessageBox("Parametros Invalidos", "CUIDADO!", MB_ICONSTOP);
        }
}
else

```

```

        {
            TDialogoIniciarMedicion IniciarMedicion(this,TResId(DIALOGO_MEDICION),PuertoSerial);
            IniciarMedicion.Execute();
        }
    }

void TVentanaPrincipal::CmConfiguracionPC()
{
    TDialogoConfigurarPC ConfigurarPC(this,TResId(DIALOGO_PUERTO),PuertoSerial);
    ConfigurarPC.Execute();
}

void TVentanaPrincipal::CmSistemaHC11()
{
    if(PuertoSerial->Id()<0)
        switch(PuertoSerial->Id())
        {
            case IE_HARDWARE:
                MessageBox(strcat("No existe puerto:",PuertoSerial->Port),"ERROR",MB_ICONSTOP);
                break;
            case IE_OPEN:
                MessageBox(strcat("Puerto en uso:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
                break;
            case IE_NOOPEN:
                MessageBox(strcat("No se pudo abrir:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
                break;
            case IE_MEMORY:
                MessageBox("No hay memoria","ERROR!",MB_ICONSTOP);
                break;
            default:
                MessageBox("Parametros Invalidos","CUIDADO!",MB_ICONSTOP);
        }
    else
        {
            TDialogoSistema Sistema_HC11(this,TResId(DIALOGO_SISTEMA_HC11),PuertoSerial);
            Sistema_HC11.Execute();
        }
}

void TVentanaPrincipal::CmVersion()
{
}

class TMiAplicacion:public TApplication
{
    TFrameWindow *Frame;
    TVentanaPrincipal *VentanaPrincipal;
public:
    TMiAplicacion(const char far * nombre):TApplication(nombre)
    {
        VentanaPrincipal= new TVentanaPrincipal(HInstance);
        Frame=new TFrameWindow(0,nombre,VentanaPrincipal);
    }
    void InitMain Window()
    {
        SetMain Window(Frame);
        Frame->AssignMenu(TResId(MENU_PRINCIPAL));
        Frame->SetIcon(Module,TResId(ICONO_PRINCIPAL));
    }
    ~TMiAplicacion()
    {
        delete Frame;
    }
};

int OwlMain(int,char*[])
{
    TMiAplicacion TAplicacion1("Prueba para windows");
    TAplicacion1.Run();
    return 0;
}

```

}

2. Tcomm.cpp

```

#define EVT_CHAR 0x3
#define EOT_CHAR 0x4
#define ACK_CHAR 0x6
#define ESC_CHAR 0x1b

class TComm
{
int ComId;
tagDCB dcb;
COMSTAT ComStat;
UINT cbInQueue,cbOutQueue;
public:
char Port[5];
TComm(UINT InBytes=1,UINT OutBytes=1,char EvChar=0x0,char EofChar=0x0)
{
for(int i=0;i<5;i++)
Port[i]=0;
cbInQueue=InBytes;
cbOutQueue=OutBytes;
ObtenParametros();
if(EvtChar==0)
{
dcb.EvtChar=0;
dcb.fChEvt=0;
}
else
{
dcb.fChEvt=1;
dcb.EvtChar=EvtChar;
}
if(EofChar==0)
{
dcb.fBinary=1;
dcb.EofChar=0;
}
else
{
dcb.fBinary=0;
dcb.EofChar=EofChar;
}
}

int Close();
int EnableNotification(HWND,int,int);
int Flush(int);
int GetError();
int GetEventMask(int);
int GetState();
int Read(void far*,int);
UINT far* SetEventMask(unsigned int);
int TransmitChar(unsigned char);
int UngetChar(char);
int Write(const void FAR *,int cbWrite);

int Id();
void ObtenParametros();
void GuardaParametros();
void ObtenDCB(tagDCB*);
void ObtenCOMSTAT(tagCOMSTAT*);
void TransfiereDCB(tagDCB*);

int Inicializar();
int ReadChar(void far*);
};

```

```

int TComm::Close()
{
    return CloseComm(ComId);
}

int TComm::Flush(int intQueue)
{
    return FlushComm(dcb.Id,intQueue);
}

int TComm::Id()
{
    return ComId;
};

int TComm::GetState()
{
    return GetCommState(ComId,&dcb);
}

UINT far* TComm::SetEventMask(unsigned int intEvento)
{
    UINT far*regreso;
    regreso=SetCommEventMask(dcb.Id,intEvento);
    return regreso;
};

int TComm::EnableNotification(HWND hwndVentana,int intInBytes=-1,int intOutBytes=-1)
{
    return EnableCommNotification(dcb.Id,hwndVentana,intInBytes,intOutBytes);
};

int TComm::TransmitChar(unsigned char chCaracter)
{
    return TransmitCommChar(dcb.Id,chCaracter);
};

int TComm::Read(void far *lpBuffer,int Bytes)
{
    return ReadComm(dcb.Id,lpBuffer,Bytes);
};

int TComm::ReadChar(void far *lpCaracter)
{
    return ReadComm(dcb.Id,lpCaracter,1);
}

int TComm::GetEventMask(int intEvento)
{
    return GetCommEventMask(dcb.Id,intEvento);
};

int TComm::GetError()
{
    return GetCommError(dcb.Id,&ComStat);
}

void TComm::ObtenerParametros()
{
    UINT BaudRate;
    GetProfileString("HC11","Port","COM1",Port,5);
    BaudRate=GetProfileInt("HC11","BaudRate",9600);
    switch(BaudRate)
    {
        case 9600:
            dcb.BaudRate=CBR_9600;
            break;
        case 4800:
            dcb.BaudRate=CBR_4800;
    }
}

```

```

        break;
    case 2400:
        dcb.BaudRate=CBR_2400;
        break;
    case 1200:
        dcb.BaudRate=CBR_1200;
        break;
    case 600:
        dcb.BaudRate=CBR_600;
        break;
    case 300:
        dcb.BaudRate=CBR_300;
        break;
    case 110:
        dcb.BaudRate=CBR_110;
        break;
    default:
        dcb.BaudRate=CBR_9600;
    }
    dcb.ByteSize=GetProfileInt("HC11","ByteSize",8);
    dcb.StopBits=GetProfileInt("HC11","StopBits",ONESTOPBIT);
    dcb.Parity=GetProfileInt("HC11","Parity",NOPARITY);
}

void TComm::GuardaParametros()
{
    char byte[2];
    WriteProfileString("HC11","Port",Port);
    switch(dcb.BaudRate)
    {
        case CBR_9600:
            WriteProfileString("HC11","BaudRate","9600");
            break;
        case CBR_4800:
            WriteProfileString("HC11","BaudRate","4800");
            break;
        case CBR_2400:
            WriteProfileString("HC11","BaudRate","2400");
            break;
        case CBR_1200:
            WriteProfileString("HC11","BaudRate","1200");
            break;
        case CBR_600:
            WriteProfileString("HC11","BaudRate","600");
            break;
        case CBR_300:
            WriteProfileString("HC11","BaudRate","300");
            break;
        case CBR_110:
            WriteProfileString("HC11","BaudRate","110");
            break;
        default:
            WriteProfileString("HC11","BaudRate","9600");
    }
    wprintf(byte,"%d",(int)dcb.ByteSize);
    WriteProfileString("HC11","ByteSize",byte);
    wprintf(byte,"%d",(int)dcb.StopBits);
    WriteProfileString("HC11","StopBits",byte);
    wprintf(byte,"%d",(int)dcb.Parity);
    WriteProfileString("HC11","Parity",byte);
}

int TComm::Inicializar()
{
    GuardaParametros();
    ComId=OpenComm(Port,cbInQueue,cbOutQueue);
    dcb.Id=ComId;
    SetCommState(&dcb);
    return ComId;
}

```

```
int TComm::ObtenDCB(tagDCB *destino)
{
    *destino=dcb;
    return 0;
};

int TComm::ObtenCOMSTAT(tagCOMSTAT *destino)
{
    *destino=ComStat;
    return 0;
}

void TComm::TransfiereDCB(tagDCB *origen)
{
    dcb=*origen;
}

int TComm::UngetChar(char chUnget)
{
    return UngetCommChar(ComId,chUnget);
}

int TComm::Write(const void FAR *lpvBuf,int cbWrite)
{
    return WriteComm(ComId,lpvBuf,cbWrite);
}
```

3. Configurar.cpp

```

class TDialogoConfigurarPC:public TDialog
{
    tagDCB deb;
    char Puerto[5];
    TComm *PuertoSerial;
public:
    void SetupWindow();
    void EvHandlerParidad(WPARAM);
    void EvHandlerBitsdeStop(WPARAM);
    void EvHandlerPuerto(WPARAM);
    void CmOk();
    void CmCancel();
    void EvCBNSelChangeVelocidad();
    void EvCBNSelChangeBitsdeDatos();
    TDialogoConfigurarPC(TWindow *parent, TResId resId, TComm *Puerto)
        :TDialog(parent, resId), TWindow(parent)
    {
        Velocidad=new TComboBox(this, IDC_COMBOBOXVELOCIDAD);
        BitsdeDatos=new TComboBox(this, IDC_COMBOBOXBITSDEDATOS);
        ParidadGroup=new TGroupBox(this, PARIDADGROUP);
        BitsdestopGroup=new TGroupBox(this, BITSDESTOPGROUP);
        PuertoGroup=new TGroupBox(this, PUERTOGROUP);
        Impar=new TRadioButton(this, IDC_RADIOBUTTONIMP, ParidadGroup);
        Par=new TRadioButton(this, IDC_RADIOBUTTONPAR, ParidadGroup);
        Ninguna=new TRadioButton(this, IDC_RADIOBUTTONNINGUNA, ParidadGroup);
        Uno=new TRadioButton(this, IDC_RADIOBUTTONUNO, BitsdestopGroup);
        UnoyMedio=new TRadioButton(this, IDC_RADIOBUTTONUNOYMEDIO, BitsdestopGroup);
        Dos=new TRadioButton(this, IDC_RADIOBUTTONDOS, BitsdestopGroup);
        Com1=new TRadioButton(this, IDC_COM1, PuertoGroup);
        Com2=new TRadioButton(this, IDC_COM2, PuertoGroup);
        Com3=new TRadioButton(this, IDC_COM3, PuertoGroup);
        Com4=new TRadioButton(this, IDC_COM4, PuertoGroup);
        PuertoSerial=Puerto;
    }
    ~TDialogoConfigurarPC()
    {
        delete Velocidad;
        delete PuertoGroup;
        delete BitsdeDatos;
        delete ParidadGroup;
        delete Com1;
        delete Com2;
        delete Com3;
        delete Com4;
        delete BitsdestopGroup;
        delete Impar;
        delete Par;
        delete Ninguna;
        delete Uno;
        delete UnoyMedio;
        delete Dos;
    }
    TComboBox *Velocidad,*BitsdeDatos;
    TGroupBox *PuertoGroup,*ParidadGroup,*BitsdestopGroup;
    TRadioButton
    *Com1,*Com2,*Com3,*Com4,*Impar,*Par,*Ninguna,*Uno,*UnoyMedio,*Dos;DECLARE_RESPONSE_TABLE(TDialogoConfigurarPC);
};

DEFINE_RESPONSE_TABLE1(TDialogoConfigurarPC, TDialog)
    EV_COMMAND(BOTON_OK, CmOk),
    EV_COMMAND(BOTON_CANCEL, CmCancel),
    EV_CHILD_NOTIFY_ALL_CODES(PARIDADGROUP, EvHandlerParidad),
    EV_CHILD_NOTIFY_ALL_CODES(BITSDESTOPGROUP, EvHandlerBitsdeStop),
    EV_CHILD_NOTIFY_ALL_CODES(PUERTOGROUP, EvHandlerPuerto),
    EV_CBN_SELCHANGE(IDC_COMBOBOXVELOCIDAD, EvCBNSelChangeVelocidad),
    EV_CBN_SELCHANGE(IDC_COMBOBOXBITSDEDATOS, EvCBNSelChangeBitsdeDatos),
END_RESPONSE_TABLE;

```



```

void TDialogoConfigurarPC::SetupWindow()
{
    TDialog::SetupWindow();
    PuertoSerial->Close();
    PuertoSerial->ObtenDCB(&dcb);
    Velocidad->AddString("9600");
    Velocidad->AddString("4800");
    Velocidad->AddString("2400");
    Velocidad->AddString("1200");
    Velocidad->AddString("600");
    Velocidad->AddString("300");
    BitsdeDatos->AddString("8");
    BitsdeDatos->AddString("7");
    BitsdeDatos->AddString("6");
    BitsdeDatos->AddString("5");
    BitsdeDatos->AddString("4");
    switch(dcb.BaudRate)
    {
        case(CBR_9600):
            Velocidad->SetText("9600");
            break;
        case(CBR_4800):
            Velocidad->SetText("4800");
            break;
        case(CBR_2400):
            Velocidad->SetText("2400");
            break;
        case(CBR_1200):
            Velocidad->SetText("1200");
            break;
        case(CBR_600):
            Velocidad->SetText("600");
            break;
        case(CBR_300):
            Velocidad->SetText("300");
            break;
    }
    switch(dcb.Parity)
    {
        case(EVENPARITY):
            Impar->Check();
            break;
        case(ODDPARITY):
            Par->Check();
            break;
        case(NOPARITY):
            Ninguna->Check();
            break;
    }
    switch(dcb.ByteSize)
    {
        case(4):
            BitsdeDatos->SetText("4");
            break;
        case(5):
            BitsdeDatos->SetText("5");
            break;
        case(6):
            BitsdeDatos->SetText("6");
            break;
        case(7):
            BitsdeDatos->SetText("7");
            break;
        case(8):
            BitsdeDatos->SetText("8");
            break;
    }
    switch(dcb.StopBits)
    {

```

```

        case(ONESTOPBIT):
            Uno->Check();
            break;
        case(ONESSTOPBITS):
            UnoyMedio->Check();
            break;
        case(TWOSTOPBITS):
            Dos->Check();
            break;
    }
    strcpy(Puerto,PuertoSerial->Port);
    if(strcmp(Puerto,"COM1")==0)
        Com1->Check();
    else
        if(strcmp(PuertoSerial->Port,"COM2")==0)
            Com2->Check();
        else
            if(strcmp(PuertoSerial->Port,"COM3")==0)
                Com3->Check();
            else
                Com4->Check();
    }

void TDialogoConfigurarPC::EvHandlerPuerto(WPARAM)
{
    if(Com1->GetCheck()==BF_CHECKED)
        strcpy(Puerto,"COM1");
    if(Com2->GetCheck()==BF_CHECKED)
        strcpy(Puerto,"COM2");
    if(Com3->GetCheck()==BF_CHECKED)
        strcpy(Puerto,"COM3");
    if(Com4->GetCheck()==BF_CHECKED)
        strcpy(Puerto,"COM4");
}

void TDialogoConfigurarPC::EvHandlerParidad(WPARAM)
{
    if(Par->GetCheck()==BF_CHECKED)
        dcb.Parity=ODDPARITY;
    if(Impar->GetCheck()==BF_CHECKED)
        dcb.Parity=EVENPARITY;
    if(Ninguna->GetCheck()==BF_CHECKED)
        dcb.Parity=NOPARITY;
}

void TDialogoConfigurarPC::EvHandlerBitsdeStop(WPARAM)
{
    if(Uno->GetCheck()==BF_CHECKED)
        dcb.StopBits=ONESTOPBIT;
    if(UnoyMedio->GetCheck()==BF_CHECKED)
        dcb.StopBits=ONESSTOPBITS;
    if(Dos->GetCheck()==BF_CHECKED)
        dcb.StopBits=TWOSTOPBITS;
}

void TDialogoConfigurarPC::EvCBNSelChangeVelocidad()
{
    int indice;
    indice=Velocidad->GetSelIndex();
    switch(indice)
    {
        case(0):
            dcb.BaudRate=CBR_9600;
            break;
        case(1):
            dcb.BaudRate=CBR_4800;
            break;
        case(2):
            dcb.BaudRate=CBR_2400;
            break;
    }
}

```

```

    case(3):
        dcb.BaudRate=CBR_1200;
        break;
    case(4):
        dcb.BaudRate=CBR_600;
        break;
    case(5):
        dcb.BaudRate=CBR_300;
        break;
    }
}

void TDialogoConfigurarPC::EvCBNSelChangeBitsdeDatos()
{
    int indice;
    indice=BitsdeDatos->GetSelIndex();
    switch(indice)
    {
        case(0):
            dcb.ByteSize=8;
            break;
        case(1):
            dcb.ByteSize=7;
            break;
        case(2):
            dcb.ByteSize=6;
            break;
        case(3):
            dcb.ByteSize=5;
            break;
    }
}

void TDialogoConfigurarPC::CmOk()
{
    int intError;
    PuertoSerial->TransfiereDCB(&dcb);
    strcpy(PuertoSerial->Port,Puerto);
    intError=PuertoSerial->Inicializar();
    if(intError<0)
        switch(intError)
        {
            case IE_HARDWARE:
                MessageBox(strcat("No existe puerto:",PuertoSerial->Port),"ERROR",MB_ICONSTOP);
                break;
            case IE_OPEN:
                MessageBox(strcat("Puerto en uso:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
                break;
            case IE_NOPEN:
                MessageBox(strcat("No se pudo abrir:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
                break;
            case IE_MEMORY:
                MessageBox("No hay memoria","ERROR!",MB_ICONSTOP);
                break;
            default:
                MessageBox("Parametros Invalidos","CUIDADO!",MB_ICONSTOP);
        }
}

TDialog::CmOk();
}

void TDialogoConfigurarPC::CmCancel()
{
    int intError;
    intError=PuertoSerial->Inicializar();
    if(intError<0)
        switch(intError)
        {
            case IE_HARDWARE:
                MessageBox(strcat("No existe puerto:",PuertoSerial->Port),"ERROR",MB_ICONSTOP);
                break;

```

```
case IE_OPEN:
    MessageBox(stcat("Puerto en uso:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
    break;
case IE_NOPEN:
    MessageBox(stcat("No se pudo abrir:",PuertoSerial->Port),"ERROR!",MB_ICONSTOP);
    break;
case IE_MEMORY:
    MessageBox("No hay memoria","ERROR!",MB_ICONSTOP);
    break;
default:
    MessageBox("Parametros Invalidos","CUIDADO!",MB_ICONSTOP);
}
TDialog::CmCancel();
}
```

4. Sistema.cpp

```

char far* strformat(char far* ,int,int,int);
char far* strformat(char far* Cadena,int Tamano,int PuntoDecimal,int Signo)
{
char CadenaTemporal[33];
char far* Puntero=Cadena;
char far* InicioCadena=Cadena;
int i=1;
if(PuntoDecimal<0)
{
CadenaTemporal[i]='.';
for(i++;PuntoDecimal<0;PuntoDecimal++,i++)
CadenaTemporal[i]='0';
for(,*Puntero!=0;i++,Puntero++)
CadenaTemporal[i]=*Puntero;
}
else
{
for(;PuntoDecimal>0;i++,Puntero++,PuntoDecimal--)
{
if(*Puntero==0)
return NULL;
CadenaTemporal[i]=*Puntero;
}
for(CadenaTemporal[i]='.',i++;*Puntero!=0;i++,Puntero++)
CadenaTemporal[i]=*Puntero;
}
if(Signo==0)
CadenaTemporal[0]='.';
else
CadenaTemporal[0]='-';
for(i=0;i<Tamano;i++,Cadena++)
*Cadena=CadenaTemporal[i];
*Cadena=0;
return InicioCadena;
}

class TDialogoSistema:public TDialog
{
HFILE HArchivo;
tagOFSTRUCT tagOFSTRUCTArchivo;
struct structPorCampo
{
char Presion[4];
char Gasto[4];
char Numero[3];
char RetornodeCarro;
char RegresodeCarro;
};
TComm *PuertoSerial;
TStatic *Gasto,*Presion,*Numero;
TButton *BotonIniciarCaptura,*BotonOk;
public:
union unionDates
{
char PorBloque[130];
struct PorCampo PorCampo[10];
} Buffer;
char Registro[130];
void CminiciarCaptura();
void CmOk();
LRESULT EvWMCommNotify(WPARAM,LPARAM);
TDialogoSistema(TWindow* Padre,TResId Resid,TComm* Puerto):
TDialog(Padre,Resid)
{
PuertoSerial=Puerto;
Gasto=new TStatic(this,ID_STATIC_GASTO,7);
Presion=new TStatic(this,ID_STATIC_PRESION,7);
}
}

```

```

Numero=new TStatic(this, ID_STATIC_NUMERO, 3);
BotonIniciarCaptura=new TButton(this, ID_BUTTON_INICIAR_CAPTURA);
BotonOk=new TButton(this, ID_BUTTON_OK);
HArchivo=OpenFile("results.txt", &tagOFSTRUCT Archivo, OF_CREATE|OF_WRITE);
if(HArchivo==HFILE_ERROR)
    MessageBox("Error en archivo temporal");
}

void SetupWindow()
{
    PuertoSerial->EnableNotification(HWindow, 130, 130);
    PuertoSerial->SetEventMask(EV_RXFLAG|EV_ERR|EV_TXEMPTY);
    TDialog::SetupWindow();
};

~TDialogoSistema()
{
    delete Gasto;
    delete Presion;
    delete Numero;
    delete BotonIniciarCaptura;
    delete BotonOk;
}

DECLARE_RESPONSE_TABLE(TDialogoSistema);
};
DEFINE_RESPONSE_TABLE1(TDialogoSistema, TDialog)
    EV_COMMAND(ID_BUTTON_INICIAR_CAPTURA, CmlniciarCaptura),
    EV_COMMAND(ID_BUTTON_OK, CmOk),
    EV_MESSAGE(WM_COMMNOTIFY, EvWmCommNotify),
END_RESPONSE_TABLE;

void TDialogoSistema::CmlniciarCaptura()
{
    int intError;
    PuertoSerial->GetEventMask(EV_RXCHAR|EV_TXEMPTY|EV_RXFLAG|EV_ERR);
    PuertoSerial->GetError();
    PuertoSerial->Flush(1);
    PuertoSerial->Flush(0);
    if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
    {
        MessageBox("Error en transmision");
        intError=PuertoSerial->GetError();
    }
}

void TDialogoSistema::CmOk()
{
    int intRespuesta;
    PuertoSerial->TransmitChar(EOT_CHAR);
    intRespuesta=MessageBox("Guardar los datos?", "Sistema HC11", MB_YESNOCANCEL|MB_ICONQUESTION);
    switch(intRespuesta)
    {
        case IDNO:
            OpenFile("results.txt", &tagOFSTRUCT Archivo, OF_DELETE);
            PuertoSerial->SetEventMask(0);
            PuertoSerial->EnableNotification(0, 0, 0);
            CmCancel();
        case IDYES:
            _lclose(HArchivo);
            PuertoSerial->SetEventMask(0);
            PuertoSerial->EnableNotification(0, 0, 0);
            CmCancel();
            break;
        case IDCANCEL:
            break;
        default:
            break;
    }
}
}

```

```

LRESULT TDialogoSistema::EvWMCommNotify(WPARAM,LPARAM)
{
    tagCOMSTAT ComStat;
    UINT CommEvent,intError;
    char *Caracter=NULL;
    int intSigno,intDec;
    double dNumero=0,dPresion=0,dGasto=0;
    struct Tipo2
    {
        char strNumero[8],strPresion[8],strGasto[9];
    };
    union Buffer2
    {
        char Bloque[25];
        struct Tipo2 PorCampos;
    }Resultados;
    CommEvent=PuertoSerial->GetEventMask(EV_RXCHAR|EV_TXEMPTY|EV_RXFLAG|EV_ERR);
    intError=PuertoSerial->GetError();
    PuertoSerial->ObtenCOMSTAT(&ComStat);
    if(intError>0)
        CommEvent=CommEvent|EV_ERR;
    if((CommEvent&EV_RXFLAG)&&ComStat.cbInQue>=130)
    {
        PuertoSerial->Read(Buffer.PorBloque,ComStat.cbInQue);
        Numero->SetText(Buffer.PorCampo[0].Numero);
        for(int i=0;i<=10;i++)
        {
            dNumero=dNumero+atof(Buffer.PorCampo[i].Numero)/*13.5/255;*/
            dPresion=dPresion+atof(Buffer.PorCampo[i].Presion)/255;
            dGasto=dGasto+atof(Buffer.PorCampo[i].Gasto)*7.5/255;
        }
        dNumero=dNumero/10;
        dPresion=dPresion/10;
        dGasto=dGasto/10;
        strcpy(Resultados.PorCampos.strNumero,ecvt(dNumero,6,&intDec,&intSigno),7);
        sprintf(Resultados.PorCampos.strNumero,7,intDec,intSigno);
        strcpy(Resultados.PorCampos.strPresion,ecvt(dPresion,6,&intDec,&intSigno),7);
        sprintf(Resultados.PorCampos.strPresion,7,intDec,intSigno);
        strcpy(Resultados.PorCampos.strGasto,ecvt(dGasto,6,&intDec,&intSigno),7);
        sprintf(Resultados.PorCampos.strGasto,7,intDec,intSigno);
        Gasto->SetText(Resultados.PorCampos.strGasto);
        Presion->SetText(Resultados.PorCampos.strPresion);
        Caracter=NULL=strchr(Resultados.PorCampos.strNumero,0x0);
        *Caracter=NULL=0x20;
        Caracter=NULL=strchr(Resultados.PorCampos.strPresion,0x0);
        *Caracter=NULL=0x20;
        Caracter=NULL=strchr(Resultados.PorCampos.strGasto,0x0);
        *Caracter=NULL=0x0;
        *(Caracter=NULL+1)=0xa;
        _write(HArchivo,Resultados.Bloque,25);
    }
    else
        if(CommEvent&EV_ERR)
        {
            PuertoSerial->Flush(1);
            PuertoSerial->Flush(0);
        }
        else
        {
            PuertoSerial->Flush(1);
            PuertoSerial->Flush(0);
        }
}
return 0;
}

```

S.- Iniciar.cpp

```

#define ALMACENAR 01
#define PAUSA 02
#include <string.h>
#include <math.h>

extern char far* strFormat(char far*,int,int,int);

class TDialogoIniciarMediccion:public TDialog
{
    struct charMSG
    {
        char intMsg[4];
        char wParam[4];
        char lParam[8];
    }Mensaje;
    int Comando;
    UINT intIndice;
    struct tipo1
    {
        char EvtChar;
        char NumdeMuestra[4];
        char Dato1[4];
        char Dato2[4];
        char EvtChar;
    };
    union tipo2
    {
        BYTE Bloque[14];
        struct tipo1 PorDato;
    }Buffer;
    BYTE Buffer2[88];
    BYTE byteAccionFlag;
    char strCadena [33];
    float fPresion_Textil,fPresion_Placa,fGasto,fPermeabilidad,
        fUnidades_de_Presion,fUnidades_de_Gasto,fUnidades_de_Permabilidad;
    int indec,instain;
    TStatic *Presion_Textil,*Presion_Placa,*Gasto,*Permeabilidad,*Bloque,
        *Status,*Error,*Unidades_Presion_Textil,*Unidades_Presion_Placa,
        *Unidades_Gasto,*Unidades_Permabilidad;
    TButton *BotonConectar,*BotonIniciar,*BotonPausa,*BotonCerrar,*BotonUnidades;
    TComm *PuertoSerial;
    HFILE HArchivo;
    tagOFSTRUCT tagOFSTRUCT Archivo;
public:
    BOOL unidades;
    BOOL boolPausaFlag;
    BOOL boolConectarFlag;
    BOOL boolIniciarFlag;
    void CmConectar();
    void CmPausa();
    void CmCerrar();
    void CmIniciar();
    void CmUnidades();
    void SetupWindow();
    LRESULT EvWMCommNotify(WPARAM,LPARAM);
    TDialogoIniciarMediccion(TWindow* padre,TResId ResId,TComm* Puer:~)
        TDialog(padre,ResId)
    {
        byteAccionFlag=NULL;
        boolIniciarFlag=FALSE;
        boolConectarFlag=TRUE;
        boolPausaFlag=FALSE;
        boolIniciarFlag=FALSE;
        Unidades_Presion_Textil=new TStatic(this,UNIDADES_PRESION_TEXTIL,7);
        Unidades_Presion_Placa=new TStatic(this,UNIDADES_PRESION_PLACA,7);
        Unidades_Gasto=new TStatic(this,UNIDADES_GASTO,7);
        Unidades_Permabilidad=new TStatic(this,UNIDADES_PERMEABILIDAD,11);
        Presion_Textil=new TStatic(this,PRESION_TEXTIL,7);
        Presion_Placa=new TStatic(this,PRESION_PLACA,7);
    }
}

```



```

Gasto=new TStatic(this,GASTO,7);
Permeabilidad=new TStatic(this,PERMEABILIDAD,7);
Bloque=new TStatic(this,BLOQUE,3);
Status=new TStatic(this,STATUS,20);
Error=new TStatic(this,ID_ERROR,20);
BotonCerrar=new TButton(this,BOTON_CERRAR);
BotonConectar=new TButton(this,BOTON_CONECTAR);
BotonPausa=new TButton(this,BOTON_PAUSA);
BotonIniciar=new TButton(this,BOTON_INICIAR);
BotonUnidades=new TButton(this,BOTON_UNIDADES);
PuertoSerial=Puerto;
unidades=1;
fUnidades_de_Presion=248.84;
fUnidades_de_Gasto=0.00211888000329;
fUnidades_de_Permecabilidad=1.968503937;
}
~TDialogoIniciarMedicion()
{
delete Presion_Textil;
delete Presion_Placa;
delete Gasto;
delete Permeabilidad;
delete Unidades_Presion_Textil;
delete Unidades_Presion_Placa;
delete Unidades_Gasto;
delete Unidades_Permecabilidad;
delete Bloque;
delete Status;
delete Error;
delete BotonConectar;
delete BotonPausa;
delete BotonIniciar;
delete BotonCerrar;
}
DECLARE_RESPONSE_TABLE(TDialogoIniciarMedicion);
};

DEFINE_RESPONSE_TABLE1(TDialogoIniciarMedicion,TDialog)
EV_COMMAND(BOTON_CERRAR,CmCerrar),
EV_COMMAND(BOTON_CONECTAR,CmConectar),
EV_COMMAND(BOTON_PAUSA,CmPausa),
EV_COMMAND(BOTON_INICIAR,CmIniciar),
EV_COMMAND(BOTON_UNIDADES,CmUnidades),
EV_MESSAGE(WM_COMMNOTIFY,EvWMCCommNotify),
END_RESPONSE_TABLE;

void TDialogoIniciarMedicion::SetupWindow()
{
TDialog::SetupWindow();
}

void TDialogoIniciarMedicion::CmUnidades()
{
if(unidades==1)
{
Unidades_Presion_Textil->SetText("[Pa]");
Unidades_Presion_Placa->SetText("[Pa]");
Unidades_Gasto->SetText("[cm3/s]");
Unidades_Permecabilidad->SetText("[cm3/cm2f]");
fUnidades_de_Presion=1;
fUnidades_de_Gasto=1;
fUnidades_de_Permecabilidad=1;
unidades=0;
}
else
{
Unidades_Presion_Textil->SetText("[inH2O]");
Unidades_Presion_Placa->SetText("[inH2O]");
Unidades_Gasto->SetText("[cfm]");
Unidades_Permecabilidad->SetText("[cmFR2]");
}
}

```

```

unidades=1;
fUnidades_de_Presion=248.84;
fUnidades_de_Gasto=0.00211888000329;
fUnidades_de_Permabilidad=1.968503937;
}
)

```

```

LRESULT TDialogoIniciarMediccion::EvWMMCommNotify(WPARAM,LPARAM)
{
tagCOMSTAT ComStat;
UINT CommEvent;
UINT intError;
CommEvent=PuertoSerial->GetEventMask(EV_RXFLAG|EV_ERR|EV_TXEMPTY);
intError=PuertoSerial->GetError();
PuertoSerial->ObtenCOMSTAT(&ComStat);
if(intError>0)
    CommEvent=CommEvent|EV_ERR;
if(CommEvent&EV_TXEMPTY)
    switch(Comando)
    {
    case BOTON_INICIAR:
        strcpy(Mensaje.intMsg,"0065",4);
        strcpy(Mensaje.wParam,"0000",4);
        strcpy(Mensaje.lParam,"00000000",8);
        if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
            Error->SetText("Error en Transmision");
        Comando=0;
        byteAccionFlag=ALMACENAR;
        boolIniciarFlag=TRUE;
        BotonUnidades->EnableWindow(WS_DISABLED);
        break;
    case BOTON_PAUSA:
        strcpy(Mensaje.intMsg,"0021",4);
        strcpy(Mensaje.wParam,"0000",4);
        strcpy(Mensaje.lParam,"00000000",8);
        if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
            Error->SetText("Error en Transmision");
        Comando=0;
        byteAccionFlag=PAUSA;
        break;
    case BOTON_CONECTAR:
        strcpy(Mensaje.intMsg,"0020",4);
        strcpy(Mensaje.wParam,"0000",4);
        strcpy(Mensaje.lParam,"00000000",8);
        if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
            Error->SetText("Error en Transmision");
        Comando=0;
        byteAccionFlag=NULL;
        Status->SetText("Recibiendo...");
        break;
    default:
        break;
    }
if((CommEvent&EV_RXFLAG)&&ComStat.cbInQue>=NUMERO_DE_DATOS)
{
    BotonConectar->EnableWindow(WS_DISABLED);
    PuertoSerial->Read(Buffer.Bloque,ComStat.cbInQue);
    Buffer.PorDato.Dato1[3]=0x0;
    Buffer.PorDato.Dato2[3]=0x0;
    Buffer.PorDato.NumdeMuestras[3]=0x0;
    fPresion_Textil=atoi(Buffer.PorDato.Dato1)*248.84/(fUnidades_de_Presion*255); /*Presion en unidades*/
    fPresion_Placa=atoi(Buffer.PorDato.Dato2)*3732.6/255; /*Presion en Pa */
    fGasto=96.32*sqrt(2*fPresion_Placa/1.29); /*Gasto en cm3/s*/
    fPermeabilidad=fUnidades_de_Permabilidad*fGasto/40.5;
    fPresion_Placa=fPresion_Placa/fUnidades_de_Presion;
    fGasto=fUnidades_de_Gasto*fGasto;
    strcpy(&strCadena[4],ccvt(fPresion_Textil,6,&intdec,&intsign),7);
    Presion_Textil->SetText(strFormat(&strCadena[4],7,intdec,intsign));
    strcpy(&strCadena[11],ccvt(fPresion_Placa,6,&intdec,&intsign),7);
    Presion_Placa->SetText(strFormat(&strCadena[11],7,intdec,intsign));
    strcpy(&strCadena[18],ccvt(fGasto,6,&intdec,&intsign),7);
}
}

```

```

Gasto->SetText(strformat(&strCadena[18],7,intdec,insign));
strcpy(&strCadena[25],cvt(Permeabilidad,6,&intdec,&insign),7);
Permeabilidad->SetText(strformat(&strCadena[25],7,intdec,insign));
strcpy(strCadena,Buffer.ForDato.NumdeMuestra,3);
Bloque->SetText(Buffer.ForDato.NumdeMuestra);
Error->SetText("");
if(boolIniciarFlag==TRUE)
    if(strlen(cmp"001",Buffer.ForDato.NumdeMuestra,3)!=0)
        else
            Error->SetText("No se recibieron ceros");
    else
        boolIniciarFlag=FALSE;
if(byteAccionFlag==ALMACENAR&&boolIniciarFlag==FALSE)
    {
    strCadena[3]=0x20;
    strCadena[10]=0x20;
    strCadena[17]=0x20;
    strCadena[24]=0x20;
    strCadena[31]=0xd;
    strCadena[32]=0xa;
    _write(HArchivo,strCadena,33);
    }
}
else
    {
    if(CommEvent&EV_ERR)
        {
        Error->SetText("Error 2: EV_ERR");
        PuertoSerial->Flush(1);
        PuertoSerial->Flush(0);
        }
    else
        PuertoSerial->Flush(1);
        PuertoSerial->Flush(0);
    }
}

return 0;
}

void TDialogoIniciarMedicion::CmCerrar()
{
int intRespuesta;
if(byteAccionFlag==ALMACENAR||byteAccionFlag==PAUSA)
    {
    intRespuesta=MessageBox("Guardar los datos?","Almacenaje",
    MB_YESNOCANCEL|MB_ICONQUESTION);
    switch(intRespuesta)
        {
        case IDNO:
            OpenFile("results.txt",&tagOFSTRUCTArchivo,OF_DELETE);
            if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
                Error->SetText("Error en transmision");
                strcpy(Mensaje,intMsg,"0021",4);
                strcpy(Mensaje.wParam,"0000",4);
                strcpy(Mensaje.lParam,"00000000",8);
                if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
                    Error->SetText("Error en Transmision");
                    PuertoSerial->SetEventMask(0);
                    PuertoSerial->EnableNotification(0,0,0);
                    CmCancel();
                case IDYES:
                    _fclose(HArchivo);
                    if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
                        Error->SetText("Error en transmision");
                        strcpy(Mensaje.intMsg,"0021",4);
                        strcpy(Mensaje.wParam,"0000",4);
                        strcpy(Mensaje.lParam,"00000000",8);
                        if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
                            Error->SetText("Error en Transmision");

```

```

        PuertoSerial->SetEventMask(0);
        PuertoSerial->EnableNotification(0,0,0);
        CmCancel();
        break;
    case IDCANCEL:
        return;
    default:
        break;
    }
}

if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
    Error->SetText("Error en transmission");strcpy(Mensaje.intMsg,"0021",4);
strcpy(Mensaje.wParam,"0000",4);
strcpy(Mensaje.lParam,"0000000",8);
if((PuertoSerial->Write(Mensaje.intMsg,16))!=16)
    Error->SetText("Error en Transmission");
PuertoSerial->SetEventMask(0);
PuertoSerial->EnableNotification(0,0,0);
CmCancel();
}

void TDialogoIniciarMediccion::CmConectar()
{
    Comando=BOTON_CONECTAR;
    PuertoSerial->GetError();
    PuertoSerial->SetEventMask(EV_RNFLAG|EV_ERR|EV_TXEMPTY);
    PuertoSerial->EnableNotification(HWindow,1,1);
    PuertoSerial->GetError();
    PuertoSerial->Flush(1);
    if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
        Error->SetText("Error en transmission");
    Status->SetText("Conectando...");
    Error->SetText("");
}

void TDialogoIniciarMediccion::CmPausa()
{
    Comando=BOTON_PAUSA;
    if((PuertoSerial->TransmitChar(ESC_CHAR))!=0)
        Error->SetText("Error en transmission");
    Status->SetText("Pausa");
}

void TDialogoIniciarMediccion::CmIniciar()
{
    Comando=BOTON_INICIAR;
    if(PuertoSerial->TransmitChar(ESC_CHAR)!=0)
        Error->SetText("Error en transmission");
    else
    {
        Status->SetText("Almacenando lecturas.");
        HArchivo=OpenFile("resulta.txt",&tagOFSTRUCTArchivo,OF_CREATE|OF_WRITE);
        if(HArchivo==HFILE_ERROR)
            MessageBox("Error en archivo temporal");
    }
}
}

```

Apéndice H Programas de control.

H.1 Control proporcional (P).

TESIS.EXE	fcc	"TESIS.EXE"	
	fdb	\$00aa	
TES.Descriptor	fcc	"reloc"	
TES.CompAdr	fdb	#TESIS.EXE	
TES.Entrypoint	fdb	#TES.handler-#TES.Descriptor	
TES.Stacksize	fdb	\$0000	
TES.ProgramSize	fdb	#TES.fin-#TESIS.EXE	
TES.RelocAddr	fdb	#TES.handler+1	
	fdb	#TESa1+1	
	fdb	#TESa2+2	
	fdb	#TESa3+1	
	fdb	#TESa4+1	
	fdb	#TESa5+1	
	fdb	#TESa6+1	
	fdb	#TESa7+1	
	fdb	#TESa8+1	
	fdb	#TESa9+2	
	fdb	#TESaa+1	
	fdb	#TESab+1	
	fdb	#TESac+2	
	fdb	#TESad+1	
	fdb	#TESae+1	
	fdb	#TESaf+1	
	fdb	#TESa10+1	
	fdb	#TESa11+1	
	fdb	#TESa12+1	
	fdb	#TESa13+1	
	fdb	#TESa14+1	
	fdb	#TESa15+2	
	fdb	#TESa16+1	
	fdb	#TESa17+1	
	fdb	#TESa18+1	
	fdb	#TESa19+1	
	fdb	#TESa1a-1	
	fdb	#TESa1b+1	
	fdb	#TESa1c+1	
	fdb	#TESa1d+1	
	fdb	#TESa1f-1	
	fdb	#TESa20+1	
	fdb	#TESa21+1	
	fdb	#TES.Pausa+1	
	fdb	#TES.Exit+1	
	fdb	#TES.trans+1	
	fdb	#ev_RTI.err2+1	
	fdb	#TES.Iniciar+1	
	fdb	\$0000	
TES.FDEM	fdb	\$55aa	;File Descriptor End Mark
TES.tabla	fdb	#TES.ev_RTI	;funcion respuesta a evento RTI
	fdb	#TES.Iniciar	
	fdb	#TES.Pausa	
TES.tabla.fin	fdb	#TES.Exit	
tesis.hwnd	fdb	\$00	
tesis.handler	fdb	0	
tesis.padre	fdb	\$00	
tesis.posic	fdb	\$0	
tesis.puntero	fdb	\$00	
tesis.buffer	fcc	"1234567890123456"	

```

fcc "1234567890123456"

TES.CommId      fdb $0
byteAccion     fdb $0
TES.SCI        fcc "SCI"

dato           fcc "P:"
               fcc 1234567890123456
dato1          fcc "000 [ini]"
               fcc "K:"
dato2          fcc "000 [cfm]"
               fdb #EOB
num_bloque     fcb 0
bloque        fcb #EVT_CHAR
               fcc "000"
               fcb $10
bloque.0       fcc "000"
               fcb $10
bloque.1       fcc "000"
               fcb $10
               fcb #EOT_CHAR
TES.defaultmsg fcc "Tesis default msg"
               fdb #EOB
TES.err1       fcc "No se abrio Comm"
               fdb #EOB
TES.err2       fcc "Error en la transmision"
               fdb #EOB

TES.handler    idx #TES.tabla
               cmpb #MSG
TESa1          beq TES.msgs
               idx #TES.msgdefault
               jsr display
               stop
               rts

TES.msgs       ldd SYS.msg
               cpd #CM_INIT
               bne TES.msg2
               bar TES.init
               cli
               bra TES.Idle
TES.msg2       cmpd #EV_RT1
               bne TES.msg3
               idx 0,x
               jsr 0,x
               bra TES.Idle
TES.msg3       cmpd #CM_INICIAR
               bne TES.msg4
               idx 2,x
               jsr 0,x
               bra TES.Idle
TES.msg4       cmpd #CM_PAUSA
               bne TES.msg5
               idx 4,x
               jsr 0,x
               bra TES.Idle
TES.msg5       cmpd #CM_EXIT
               bne TES.msg6
               idx 6,x
               jsr 0,x
               bra TES.Idle
TES.msg6       cmpd #CM_STARTUP
               bne TES.msg7
               idx 2,x
               jsr 0,x
               bra TES.Idle
TES.msg7       cmpd #CM_RESUME
               bne TES.msgdefault
               idx SYS.wparam
               ldd SYS.lparaml

```

```

                jmp          0,x

TES.msgdefault  ldx          #TES.defaultmsg
                jsr          display
                ldx          SYS.msg
                jsr          showIX
                stop

TES.Idle
TESa2          nop
                ldy          #tesis.hwnd
                clr          #posic.y
                ldd          #CM_XGTASK
                std          SYS.msg
                ldd          #S0 ;cambio no forzado
                std          SYS.wparam
                ldaa         #SYS
                ldab         #MSG
                swi
                cli
                wai
                bra          TES.Idle

TES.init       sei
                ldx          #TES.tabla-#TES.Descriptor ;desplazamiento para tabla
                pahx         ;crea variable temporal.
                tay
                ldd          SYS.lparam1 ;se lee una referencia la direccion de instalacion que es el entry pont
                subd         #TES.handler-#TES.Descriptor ;se obtiene la direccion real de instalacion
                std          SYS.lparam1
                addd         0,y ;se obtiene la direccion real
                xgdx         #TES.tabla.fin-#TES.tabla ;se pasa a IX la direccion de tabla
                ldy          0,x
                ldd          #TES.Descriptor
                subd         SYS.lparam1
                addd         0,x
                dey
                cmpy         #S0
                beq          TESinit.cont1
                inx
                inx
                bra          TESinit.c1
                pulx         ;destruye variable temporal
                ldx          #TES.RelocAddr-#TES.Descriptor
                xgdx         SYS.lparam1
                addd         0,x
                ldy          #S0000
                cmpy         TESinit.cont2
                beq          TESinit.c2
                xgdy         #TES.Descriptor
                subd         SYS.lparam1
                addd         0,y
                pshy
                ldy          0,y
                xgdy         #TES.Descriptor
                subd         SYS.lparam1
                addd         0,y
                puly
                std          0,y
                inx
                inx
                bra          TESinit.c2
                ldd          #WM_CREATE ;crea una ventana para esta apli
                std          SYS.msg ;cacion
                ldd          #tesis.hwnd
                std          SYS.wparam

```

```

TESa4      ldd      #TES.handler
           std      SYS.lparam1
           ldaa    #LCD
           ldab    #MSG
           swi
           ldaa    #S3                ;se ajusta el intervalo de
           psha    ;interrupcion del RTI
           tax
           stx     SYS.wparam
           ldaa    #RTI
           ldab    #RTI_setRTIrate
           swi
           pula

TESa10     idx      #TES.SCI
           pshx
           tax
           stx     SYS.wparam
           ldaa    #SCI
           ldab    #SCI_OpenComm
           swi
           pulx
           ldd     intRegreso
           cmpa    #SCI
           bne     TESinit.err1
           cmpb    #S0
           bnc     TESinit.err1
TESa11     std      TES.CommId
TESa5      idx      #TES.handler    ;se habilita la notificacion del
           pshx    ;evento RTI.
           tax
           stx     SYS.wparam
           ldaa    #RTI
           ldab    #RTI_enablenotif
           swi
           ldaa    #ADC
           ldab    #ADC_enablenotif
           swi
           ldaa    #SCI
           ldab    #SCI_EnableNotif
           swi
           pulx
           ldaa    #S30                ;multiple y continuo
           psha
           tax
           stx     SYS.wparam
           ldaa    #ADC
           ldab    #ADC_setmode        ;se inicializa el convertidor A/D
           swi
           pula
           cira
           staa    $R000
           rts

TESinit.err1  idx      #TES.err1
           jsr     display

TES.Iniciar  clr      num_bloque
TESa1d      clr      byteAccion

TES.Kc      fcb      #131
TES.KcMN    fdb      #SC5F3
TES.ERROR   fdb      #B600
TES.ERRORM  fcb      #128
TES.ERRORMN fdb      #0
TES.ERRORMNSGN fdb    #0
TES.V       fcb      #128
TES.VMN     fdb      #0
TES.VMNSGN  fdb      #0

```



```

TES.AREA1      fcb      #121      ;exponente de PI
TES.AREA1MN    fdb      #S84B5
TES.AREA1MNSGN fdb      #S5400
TES.AREA2      fcb      #116
TES.AREA2MN    fdb      #S$A7C5
TES.AREA2MNSGN fdb      #SAC00
TES.DENS      fcb      #129
TES.DENSMN    fdb      #S$A1E
TES.DENSMNSGN fdb      #S$B800
TES.FACTOR     fcb      #128
TES.FACTORMN   fdb      #S$9A1C
TES.FACTORMNSGN fdb      #SAC00
TES.PASCAL     fcb      #136
TES.PASCALMN   fdb      #S$F8D7
TES.PASCALMNSGN fdb      #S$0A00
TES.FLTASC     rmb      14
TES.FPACCEX    rmb      1
TES.FPACCMN    rmb      3
TES.FPACCSGN   rmb      1
TES.cuenta     fcb      $0

TES.ajuste     pahx
               pahy
               ldaa      #7
TESajuste.c1   ldaa      0,x
               cmpa     #0
               beq      TESajuste.fin
               staa     0,y
               inx
               iny
               decb
TESajuste.fin   bne      TESajuste.c1
               pulv
               rta

TES.ev_RTI     equ      *

TES.ControlP   ldaa      #ADC
               ldab     #ADC_IsReady
               svi
               ldd      intRegreso
               cpd      #FALSE
               beq      TES.ControlP
               ldd      #255
               std      FPACC1MN+1
               jsr      UINH2FLT
               jsr      TFR1TO2
               ldaa     ADR1
               paha
               cira
               paha
               tax      #128
               ldab     0,x
               subd     ;Obtiene el error acutal.
               std      FPACC1MN+1
               jsr      SINT2FLT
               pulx     ;Se convierte el error a FP
               jsr      FLTDIV
               ldx      #TES.Kc
               jsr      FP2ACC2
               jsr      FLTMUL
               jsr      TFR1TO2
               ldx      #TES.V
               pahx
               jsr      FP2ACC1
               jsr      FLTADD
               pulx
               jsr      ACC1MEM

```

```

ldaa MANTSGN1
cmpa #0
bcq PROP.cont1
ldab #S1
bra PROP.salida
jrd FLT2INT
jra FPACC1MN+1
cmpa #0
bcq PROP.salida
ldab #SfT
stab $8000

TESa7 inc TES.cuenta
TESa8 ldaa TES.cuenta
cmpa #Sa
bhi TESev_RTI.cont1
rts

TESev_RTI.cont1 ldd #255 ;:se convierte 255 a float
std FPACC1MN+1
jrd UINT2FLT
TESa13 #TES.FPACCEN
jrd PUTFPAC1 ;:se pasa el dato a FPACC2
jrd GETFPAC2
ldx #ADR1 ;:se convierte el dato de la
pslx ;:conversion A/D a float
cfrd 0,x
ldab FPACC1MN+1
std UINT2FLT
jrd FLTDIV ;:se obtiene el valor de presion:
;:Lectura/255;
TESa14 #TES.FLTASC
jrd FLTASC ;:convierte el resultado de la
;:division a caracteres ASCII
TESa9 #dato1
TESa20 jrd TES.ajuste
pulk
inx
cfrd 0,x
ldab FPACC1MN+1
std UINT2FLT
jrd FLTDIV
jrd TFR1TO2
ldd #15
std FPACC1MN+1 ;:convierte la lectura del CAD a
jrd UINT2FLT ;:flotante
jrd FLTMUL

TESa17 ldx #TES.PASCAL
bsr TES.FP2ACC2
jrd FLTMUL ;:convierte InH2O a Pascales.
jrd TFR1TO2 ;:realiza la operacion para
ldd #2 ;:obtener gasto.
std FPACC1MN+1
jrd UINT2FLT
jrd FLTMUL
TESa18 #TES.DENS
ldx TES.FP2ACC2
bsr FLTDIV
jrd FLTSQR
ldx #TES.AREA2
bsr TES.FP2ACC2
jrd FLTMUL
TESa19 #TES.FACTOR
ldx TES.FP2ACC2
bsr FLTMUL
jrd TES.AREA1 ;:gasto en m3/s
TESa1a #TES.FP2ACC2
ldx TES.FP2ACC2
bsr FLTDIV ;:permeabilidad en m/s
jrd TFR1TO2

```

```

ldd      #100
std      FPACC1MN+1
jnr      UNT2FLT
jnr      FLT MUL      ;permeabilidad en cm/s

TESa16   ldx      #TES.FLTASC
         jsr      FLTASC
TESa15   ldy      #dato2
TESa1b   jar      TES.ajuste
         bra      TESa1c

TES.FP2ACC1  pshx
             ldy      #FPACC1EX
             ldab     #5
TESFP2ACC1.c1 ldaa     0,x
             staa     0,y
             inx
             iny
             decb
             bnc
             pulx
             rts
             TESFP2ACC1.c1

TES.FP2ACC2  pshx
             ldy      #FPACC2EX
             ldab     #5
TESFP2ACC2.c1 ldaa     0,x
             staa     0,y
             inx
             iny
             decb
             bne
             pulx
             rts
             TESFP2ACC2.c1

TESa1c   ldaa     byteAccion
TESa1c   bnc
TESa1c   inc
TESa1c   ldx      num_bloque
TESa1c   ldy      #num_bloque
TESa1c   pshy
TESa1c   iny
TESa1c   bar
TESa1c   ldx      #ADR1
TESa1c   xgdy
TESa1c   addd     #bloque.1-#bloque.0
TESa1c   xgdy
TESa1c   bsr
TESa1c   inx
TESa1c   xgdy
TESa1c   addd     #bloque.1-#bloque.0
TESa1c   xgdy
TESa1c   bsr
TESa1c   pulx
TESa1c   bsr
TESa1c   TESconversion ;salva puntero al buffer de datos
TESa1c   #ADR1         ;la siguiente secuela de instrucciones
TESa1c   #bloque.1-#bloque.0 ;es para convertir los numeros de los
TESa1c   TESconversion ;registros del ADC en caracteres y
TESa1c   ;ADR2         ;enviarlos por el puerto de comunicaciones
TESa1c   #bloque.1-#bloque.0
TESa1c   TESconversion ;recupera puntero al buffer de datos
TESa1c   TES.trans

TESad   clr      TES.cuenta
TESae   ldx      #tesis.hwnd
TESae   clr      _posic,x
TESa6   ldx      #tesis.hwnd
TESaf   pshx
TESaf   ldx      #dato
TESaf   pshx
TESaf   tsx
TESaf   stx
TESaf   ldaa     SYS.wparam
TESaf   ldab     #LCD
TESaf   swi      #LCD_outtext
TESaf   pulx
TESaf   pulx
TESaf   rts
TESev_RTI.fin

```

TESconversion	<pre> pahx ldaa paha paha tax clr clr clr cmpa beq clra cirtb addb cmpb bhi cmpb beq inc ldaa cmpa blo inc clr bra subb stab ldaa suba staa ldaa adda staa ldaa adda staa adda staa pula pula pulx rts </pre>	<pre> 0,x 0,y 1,y 2,y #0 TESconver.fin2 #S0a 0,x TESconver.fin1 #S4 TESconver.fin1 1,y 1,y #S0a TESconver.c1 0,y 1,y TESconver.c1 #Sa 0,x 1,x 0,x 2,y 2,y #S30 2,y 1,y #S30 1,y 0,y #S30 0,y </pre>	<pre> ; carga el dato a convertir y lo guarda en pila ; crea variable temporal con el numero a convertir ; unidades ; decenas ; centenas ; verifica si el dato fuente es cero ; si es cero no hace nada ; esta convirtiendo un 0xFF ; incrementa decenas ; verifica se se ha generado acarreo ; en centenas. ; incrementa centenas. ; obtiene el numero a convertir </pre>
TESconver.c1	<pre> addb cmpb bhi cmpb beq inc ldaa cmpa blo inc clr bra subb stab ldaa suba staa ldaa adda staa ldaa adda staa adda staa pula pula pulx rts </pre>	<pre> #S0a 0,x TESconver.fin1 #S4 TESconver.fin1 1,y 1,y #S0a TESconver.c1 0,y 1,y TESconver.c1 #Sa 0,x 1,x 0,x 2,y 2,y #S30 2,y 1,y #S30 1,y 0,y #S30 0,y </pre>	<pre> ; esta convirtiendo un 0xFF ; incrementa decenas ; verifica se se ha generado acarreo ; en centenas. ; incrementa centenas. ; obtiene el numero a convertir </pre>
TESconver.fin1	<pre> bra subb stab ldaa suba staa ldaa adda staa ldaa adda staa adda staa pula pula pulx rts </pre>	<pre> TESconver.c1 #Sa 0,x 1,x 0,x 2,y 2,y #S30 2,y 1,y #S30 1,y 0,y #S30 0,y </pre>	<pre> ; obtiene el numero a convertir </pre>
TESconver.fin2	<pre> adda staa ldaa adda staa adda staa pula pula pulx rts </pre>	<pre> #S30 2,y 1,y #S30 1,y 0,y #S30 0,y </pre>	
TES.trans	<pre> ldx pahx pahy ldaa paha tax stx ldaa ldab swi pula puly pulx ldd cpd bnc rts ldx jsr rts </pre>	<pre> TES.CommId #14 SYS.wparam #SCI #SCI_WriteComm intRegreso #S0 ev_RTI.err2 #TES.err2 display </pre>	
ev_RTI.err2	<pre> ldx jsr rts </pre>	<pre> #TES.err2 display </pre>	
TES.Pausa	<pre> ldx ldaa cmpa beq </pre>	<pre> #byteAccion 0,x #CM_PAUSA TESPausa.1 </pre>	

	ldaa	#CM_PAUSA
	staa	0,x
	rts	
TESPausa.1	ldaa	#0
	staa	0,x
	rts	
TES.Exit	ldx	#TES.SCI
	pshx	
	tax	
	stx	SYS.wparam
	ldaa	#SCI
	ldab	#SCI_CloseComm
	swi	
TESa12	pulx	
	ldx	#tesis.hwnd
	stx	SYS.wparam
	ldd	#WM_QUIT
	std	SYS.msg
	ldaa	#LCD
	ldab	#MSG
	swi	
	ldx	#\$0
	pshx	
	tax	
	stx	SYS.wparam
	ldaa	#RTI
	ldab	#RTI_enablenotif
	swi	
	ldaa	#ADC
	ldab	#ADC_enablenotif
	swi	
	pulx	
	ldaa	#SYS
	ldab	#SYS_terminate
	swi	
TES.fin	ftb	#EOB

H.2 Control proporcional-integral (PI).

TESIS.ENE	fcc	"TESIS.ENE"	
	fdb	\$00aa	
TES.Descriptor	fcc	"reloc"	
TES.CompAddr	fdb	#TESIS.ENE	
TES.Emrtpoint	fdb	#TES.handler-#TES.Descriptor	
TES.Stacksize	fdb	\$0000	
TES.ProgramSize	fdb	#TES.fin-#TESIS.ENE	
TES.RelocAddr	fdb	#TES.handler+1	
	fdb	#TESa1+1	
	fdb	#TESa2+2	
	fdb	#TESa3+1	
	fdb	#TESa4+1	
	fdb	#TESa5+1	
	fdb	#TESa6+1	
	fdb	#TESa7+1	
	fdb	#TESa8+1	
	fdb	#TESa9+2	
	fdb	#TESa+1	
	fdb	#TESab+1	
	fdb	#TESac+2	
	fdb	#TESad+1	
	fdb	#TESae+1	
	fdb	#TESaf+1	
	fdb	#TESa10+1	
	fdb	#TESa11+1	
	fdb	#TESa12+1	
	fdb	#TESa13+1	
	fdb	#TESa14+1	
	fdb	#TESa15+2	
	fdb	#TESa16+1	
	fdb	#TESa17+1	
	fdb	#TESa18+1	
	fdb	#TESa19+1	
	fdb	#TESa1a+1	
	fdb	#TESa1b+1	
	fdb	#TESa1c+1	
	fdb	#TESa1d+1	
	fdb	#TESa1e+1	
	fdb	#TESa1f+1	
	fdb	#TESa20+1	
	fdb	#TESa21+1	
	fdb	#TESa22+1	
	fdb	#TESa23+1	
	fdb	#TESa24+1	
	fdb	#TESa25+1	
	fdb	#TES.Pausa+1	
	fdb	#TES.Excit+1	
	fdb	#TES.trans+1	
	fdb	#ev.RTI.err2+1	
	fdb	#TES.Iniciar+1	
	fdb	\$0000	
TES.FDEM	fdb	\$55aa	:File Descriptor End Mark
TES.tabla	fdb	#TES.ev_RTI	:funcion respuesta a evento RTI
	fdb	#TES.f_iniciar	
	fdb	#TES.f_pausa	
TES.tabla.fin	fdb	#TES.Excit	
tesis.hwmd	fdb	\$00	
tesis.handler	fdb	0	
tesis.padre	fdb	\$00	
tesis.posic	fcc	\$0	
tesis.puntero	fdb	\$00	
tesis.buffer	fcc	"1234567890123456"	
	fcc	"1234567890123456"	
TES.Commid	fdb	\$0	
byteAccion	fdb	\$0	

```

TES.SCI          fcc      "SCI"
dato            fcc      "p.:"
*              fcc      1234567890123456
dato1          fcc      "000 [InH]"
               fcc      "K:"
dato2          fcc      "000 [cfm]"
               fdb      #EOB
num_bloque     fcb      #EVT_CHAR
bloque        fcc      "000"
               fcb      $10
bloque.0      fcc      "000"
               fcb      $10
bloque.1      fcc      "000"
               fcb      $10
               fcb      #EOT_CHAR
TES.defaultmsg fcc      "Tes default msg"
               fdb      #EOB
TES.err1       fcc      "No se abrio Comun"
               fdb      #EOB
TES.err2       fcc      "Error en la transmision"
               fdb      #EOB

TES.handler    ldx      #TES.tabla
               cmpb    #MSG
               beq     TES.msgs
TES.a1         ldx      #TES.msgdefault
               jsr     display
               stop
               rts

TES.msgs       ldd      SYS.msg
               cpd     #CM_INIT
               bne    TES.msg2
               bsr    TES.init
               cli
               bra    TES.Idle
TES.msg2       cmpd    #EV_RT1
               bne    TES.msg3
               ldx    0,x
               jsr    0,x
               bra    TES.Idle
TES.msg3       cmpd    #CM_INICIAR
               bne    TES.msg4
               ldx    2,x
               jsr    0,x
               bra    TES.Idle
TES.msg4       cmpd    #CM_PAUSA
               bne    TES.msg5
               ldx    4,x
               jsr    0,x
               bra    TES.Idle
TES.msg5       cmpd    #CM_EXIT
               bne    TES.msg6
               ldx    6,x
               jsr    0,x
               bra    TES.Idle
TES.msg6       cmpd    #CM_STARTUP
               bne    TES.msg7
               ldx    2,x
               jsr    0,x
               bra    TES.Idle
TES.msg7       cmpd    #CM_RESUME
               bne    TES.msgdefault
               ldx    SYS.sparam
               ldd    SYS.iparam1
               jmp    0,x

TES.msgdefault ldx      #TES.defaultmsg
               jsr     display

```

```

idx      SYS.msg
jst      showIX
stop

TES.Idle
TESa2    nop
        ldy      #tesis.hwnd
        clr
        ldd      #CM_XGTASK
        std      SYS.msg
        ldd      #S0 ;cambio no forzado
        std      SYS.wparam
        ldaa     #SYS
        ldab     #MSG
        swi
        cli
        wai
        bra      TES.Idle

TES.init    sei
           idx      #TES.tabla-#TES.Descriptor ;desplazamiento para tabla
           pahx     ;crea variable temporal.
           tay
           ldd      SYS.lparam1 ;se lee una referencia la direccion de instalacion que es el entry pont
           subd     #TES.handler-#TES.Descriptor ;se obtiene la direccion real de instalacion
           std      SYS.lparam1
           addd     0,y ;se obtiene la direccion real
           xgdx     ;se pasa a IX la direccion de tabla
           ldy      #TES.tabla.fin-TES.tabla
           ldd      0,x
           subd     #TES.Descriptor
           addd     SYS.lparam1
           std      0,x
           dey
           cmpy     #S0
           beq      TESinit.cont1

TESinit.c1    inx
           bra      TESinit.c1
           pulx     ;destruye variable temporal
           idx      #TES.RelocAddr-#TES.Descriptor
           xgdx
           addd     SYS.lparam1

TESinit.c2    xgdx
           ldy      0,x
           cmpy     #S0000
           beq      TESinit.cont2
           xgdy
           subd     #TES.Descriptor
           addd     SYS.lparam1
           xgdy
           psdy
           ldy      0,y
           xgdy
           subd     #TES.Descriptor
           addd     SYS.lparam1
           puty
           std      0,y
           inx
           inx
           bra      TESinit.c2

TESinit.cont2    ldd      #WM_CREATE ;crea una ventana para esta apli
                 std      SYS.msg ;cacion
TESa3            ldd      #tesis.hwnd
                 std      SYS.wparam
TESa4            ldd      #TES.handler
                 std      SYS.lparam1
                 ldaa     #LCD
                 ldab     #MSG

```



```

swi
ldaa #S3 ;se ajusta el intervalo de
paha ;interrupcion del RTI
tax
tax SYS.wparam
ldaa #RTI
ldab #RTI_actRTIrate
swi
pula

TESa10 ldx #TES.SCI
pahx
tax
tax SYS.wparam
ldaa #SCI
ldab #SCI_OpenCorum
swi
pulk
ldd intRegreso
cmpa #SCI
bne TESinit.err1
cmpb #S0
TESa11 bne TESinit.err1
TESa5 stc TES.CommId
ldx #TES.handler ;se habilita la notificacion del
pahx ;evento RTI.
tax
tax SYS.wparam
ldaa #RTI
ldab #RTI_enablenotif
swi
ldaa #ADC
ldab #ADC_enablenotif
swi
ldaa #SCI
ldab #SCI_EnableNotif
swi
pulk
ldaa #S30 ;multiple y continuo
paha
tax
tax SYS.wparam
ldaa #ADC
ldab #ADC_setmode ;se inicializa el convertidor A/D
swi
pula
cra
staa $8000
rs

TESinit.err1 ldx #TES.err1
jst display

TES.Iniciar chr num_bloque
TESa1d clr byteAccion
rs
TES.Kc fcb #131
TES.KcMN fdb #SC5F3
TES.KcMNSGN fdb #SB600
TES.MUESTREO fcb #124
TES.MUESTREOMN fdb #S8312
TES.MUESTREOMNSGN fdb #S6E00
TES.TAU fcb #128
TES.TAUMN fdb #SC51E
TES.TAUMNSGN fdb #SB800
TES.ERROR fcb #128
TES.ERRORMN fdb #0
TES.ERRORMNSGN fdb #0
TES.ERR fcb #128
TES.ERRMN fdb #0

```

```

swi
ldaa #S3 ;se ajusta el intervalo de
psha ;interrupcion del RTI
tax
stx SYS.wparam
ldaa #RTI
ldab #RTI_setRTIrate
swi
pula

TESa10 ldx #TES.SCI
pshx
tax
stx SYS.wparam
ldaa #SCI
ldab #SCI_OpenComm
swi
puls
ldd intRegreso
cmpa #SCI
bnc TESinit.err1
cmpb #S0
bnc TESinit.err1

TESa11 TESa5 ldx #TES.handler ;se habilita la notificacion del
pshx ;evento RTI.
tax
stx SYS.wparam
ldaa #RTI
ldab #RTI_enablenotif
swi
ldaa #ADC
ldab #ADC_enablenotif
swi
ldaa #SCI
ldab #SCI_EnableNotif
swi
puls
ldaa #S30 ;multiple y continuo
psha
tax
stx SYS.wparam
ldaa #ADC
ldab #ADC_setmode ;se inicializa el convertidor A/D
swi
pula
clra
staa
rts

TESinit.err1 ldx #TES.err1
jsr display
stop

TES.Iniciar clr
TESa1d clr num_bloque
rts byteAccion

TES.Kc fcb #131
TES.KcMn fdb #SC5F3
TES.KcMNSGN fdb #S600
TES.MUESTREO fcb #124
TES.MUESTREOMN fdb #S8312
TES.MUESTREOMNSGN fdb #S6E00
TES.TAU fcb #128
TES.TAUMN fdb #SC51E
TES.TAUMNSGN fdb #S1800
TES.ERROR fcb #128
TES.ERRORMN fdb #0
TES.ERRORMNSGN fdb #0
TES.ERR fcb #128
TES.ERRMN fdb #0

```

TES.ERRMNSGN	fdb	#0	
TES.V	fdb	#128	
TES.VMNSGN	fdb	#0	
TES.VMNSGN	fdb	#0	
TES.AREA1	fdb	#121	;exponente de PI
TES.AREA1MN	fdb	#84B5	
TES.AREA1MNSGN	fdb	#55400	
TES.AREA2	fdb	#116	
TES.AREA2MN	fdb	#5A7C5	
TES.AREA2MNSGN	fdb	#5AC00	
TES.DENS	fdb	#129	
TES.DENSMN	fdb	#5A51E	
TES.DENSMNSGN	fdb	#5B800	
TES.FACTOR	fdb	#128	
TES.FACTORMN	fdb	#59A1C	
TES.FACTORMNSGN	fdb	#5AC00	
TES.PASCAL	fdb	#136	
TES.PASCALMN	fdb	#5F8D7	
TES.PASCALMNSGN	fdb	#50A00	
TES.FLTASC	rmb	14	
TES.FPACCEX	rmb	1	
TES.FPACCMN	rmb	3	
TES.FPACCSGN	rmb	1	
TES.cuenta	fdb	50	
TES.ajuste	pbhx		
	pbhy		
	ldab	#7	
TESajuste.c1	ldaa	0,x	
	cmpa	#0	
	beq	TESajuste.fin	
	staa	0,y	
	inx		
	iny		
	decb		
	bne	TESajuste.c1	
TESajuste.fin	pulx		
	pulx		
	rsa		
TES.ev_RTI	equ	*	
TES.ControlPI	ldaa	#ADC	
	ldab	#ADC_IsReady	
	swi		
	ldd	intRegreso	
	cpd	#FALSE	
	beq	TES.ControlPI	
	ldd	#255	
	std	FPACC1MN+1	
	jsr	UINT2FLT	
	jsr	TFR1T02	
	ldaa	ADR1	
	psha		
	cira		
	psha		
	tsx		
	ldab	#128	
	subd	0,x	;Obtiene el error actual.
	std	FPACC1MN+1	
	jsr	SINT2FLT	;Se convierte el error a FP
	pulx		
	jsr	FLTDIV	
TESa1c	ldx	#TES.ERROR	;Error en INH20
	pbhx		
	jsr	ACC1MEM	;Se guarda el error actual
	ldd	#52	
	std	FPACC1MN+1	
	jsr	UINT2FLT	

TESa1f	jar ldx jar jar jar pulx jar jar jar jar	TFR1TO2 #TES.ERR FP2ACC1 FLTDIV TFR1TO2 FP2ACC1 FLTSUB TFR1TO2	
TESa20	ldx jar jar ldx	#TES.MUESTREO FP2ACC1 FLTMUL	
TESa24	ldx jar jar ldaa paha ldx paha ldx paha ldx paha ldx jar ldx jar jar jar pulx six pulx six pula sta jar jar ldx jar jar jar ldx paha jar jar pulx jar ldaa cmpa beq clrb bra jar ldd cmpa beq ldab stab	FPACC1EX FPACC1MN FPACC1MN+2 #TES.ERROR FP2ACC1 #TES.ERR FP2ACC2 ACCIMEM FLTSUB FPACC2MN+2 FPACC2MN FPACC2EX FLTADD TFR1TO2 #TES.Kc FP2ACC1 FLTMUL TFR1TO2 #TES.V FP2ACC1 FLTADD ACCIMEM MANTSGN1 #0 PI.cont1 PI.salida FLT2INT FPACC1MN+1 #0 PI.salida #S7 \$8000	
TESa21	ldx jar ldx jar jar jar pulx six pulx six pula sta jar jar ldx jar jar jar ldx paha jar jar pulx jar ldaa cmpa beq clrb bra jar ldd cmpa beq ldab stab	FPACC1MN FPACC1MN+2 #TES.ERROR FP2ACC1 #TES.ERR FP2ACC2 ACCIMEM FLTSUB FPACC2MN+2 FPACC2MN FPACC2EX FLTADD TFR1TO2 #TES.Kc FP2ACC1 FLTMUL TFR1TO2 #TES.V FP2ACC1 FLTADD ACCIMEM MANTSGN1 #0 PI.cont1 PI.salida FLT2INT FPACC1MN+1 #0 PI.salida #S7 \$8000	
TESa22	ldx jar jar jar ldx paha jar jar pulx jar ldaa cmpa beq clrb bra jar ldd cmpa beq ldab stab	FPACC1EX FLTADD TFR1TO2 #TES.Kc FP2ACC1 FLTMUL TFR1TO2 #TES.V FP2ACC1 FLTADD ACCIMEM MANTSGN1 #0 PI.cont1 PI.salida FLT2INT FPACC1MN+1 #0 PI.salida #S7 \$8000	
TESa25	inc ldaa cmpa bhi rts	TES.cuenta TES.cuenta #S4 TESev_RTI.cont1	
TESev_RTI.cont1	ldd std jar ldx jar jar ldx paha	#255 FPACC1MN+1 UIN72FLT #TES.FPACC2EX PUTFPAC1 GETFPAC2 #ADR1	;se convierte 255 a float
TESa13	ldx jar jar ldx paha	FPACC1EX FLTADD TFR1TO2 #TES.Kc FP2ACC1 FLTMUL TFR1TO2 #TES.V FP2ACC1 FLTADD ACCIMEM MANTSGN1 #0 PI.cont1 PI.salida FLT2INT FPACC1MN+1 #0 PI.salida #S7 \$8000	;se pasa el dato a FPACC2 ;se convierte el dato de la

```

cira
ldab      0,x
std       FPACC1MN+1
jsr       UNT2FLT
jsr       FLTDIV
;conversion A/D a float

TESa14    ldx       #TES.FLTASC
jsr       FLTASC
TESa9     ldy       #dato1
TESa23    jsr       TES.ajuste
pulsx
inx
cira
ldab      0,x
std       FPACC1MN+1
jsr       UNT2FLT
jsr       FLTDIV
jsr       TFR1TO2
ldd       #15
std       FPACC1MN+1
jsr       UNT2FLT
jsr       FLTMUL
;convierte la lectura del CAD a
;flotante

TESa17    ldx       #TES.PASCAL
bar       TES.FP2ACC2
jsr       FLTMUL
jsr       TFR1TO2
ldd       #2
std       FPACC1MN+1
jsr       UNT2FLT
jsr       FLTMUL
;convierte InH2O a Pascales.
;realiza la operacion para
;obtener gasto.

TESa18    ldx       #TES.DENS
bar       TES.FP2ACC2
jsr       FLTDIV
jsr       FLTSQR
ldx       #TES.AREA2
bar       TES.FP2ACC2
jsr       FLTMUL

TESa19    ldx       #TES.FACTOR
bar       TES.FP2ACC2
jsr       FLTMUL
;gasto en m3/s

TESa1a    ldx       #TES.AREA1
bar       TES.FP2ACC2
jsr       FLTDIV
jsr       TFR1TO2
ldd       #100
std       FPACC1MN+1
jsr       UNT2FLT
jsr       FLTMUL
;permeabilidad en m/s
;permeabilidad en cm/s

TESa16    ldx       #TES.FLTASC
jsr       FLTASC
TESa15    ldy       #dato2
TESa1b    jsr       TES.ajuste
bra       TESa1c

TES.FP2ACC1  pshx
ldy       #FPACC1EX
ldab      #5
TESFP2ACC1.c1 ldaa      0,x
sta       0,y
inx
iny
decb
bnc
pulsx
ris

TES.FP2ACC2  pshx
ldy       #FPACC2EX
ldab      #5

```

TESFP2ACC2.c1	ldaa staa inx iny decb bne puls rts	0,x 0,y	
TESa1c	ldaa	byteAccion	
TESaa	bse	TESad	
TESab	inc	num_bloque	
TESac	ldx ldy pshy iny bar ldx xgdy addd xgdy bar inx xgdy addd xgdy bar puly bar	#num_bloque #bloque	:salva puntero al buffer de datos
		TESconversion #ADR1	:la siguiente actuella de instrucciones :es para convertir los numeros de los :registros del ADC en caracteres y :enviarlos por el puerto de comunicaciones
		TESconversion #bloque.1-#bloque.0	
		TESconversion #ADR2	
		TESconversion #bloque.1-#bloque.0	
		TESconversion #recupera puntero al buffer de datos	
		TES.trans	
TESad	clr	TES.cuenta	
TESae	ldx	#tesis.hwnd	
	clr	.posic,x	
TESa6	ldx	#tesis.hwnd	
TESaf	pshx ldx pshx tsx six ldaa ldab swi puls puls rts	#dato	
		SYS.wparam #LCD #LCD_outtext	
TESev_RTI.fin			
TESconversion	pshx ldaa paha psha tsx clr clr clr cmpa beq clra clrb	0,x 0,y 1,y 2,y #0 TESconver.fin2	:carga el dato a convertir y lo guarda en pila :crea variable temporal con el numero a convertir :unidades :decenas :centenas :verifica si el dato fuente es cero :si es cero no hace nada
TESconver.c1	addb cmpb bhi cmpb beq inc ldaa cmpa blo inc clr	#S0a 0,x TESconver.fin1 #S4 TESconver.fin1 1,y 1,y #S0a TESconver.c1 0,y 1,y	:esta convirtiendo un 0xFF :incrementa decenas :verifica se se ha generado acarreo :en centenas. :incrementa centenas.

```

TESconver.fin1      bra      TESconver.c1
                   subb     #5a
                   stab     0,x
                   ldaa     1,x           :obtiene el numero a convertir
                   suba     0,x
TESconver.fin2      staa     2,y
                   ldaa     2,y
                   adda     #530
                   staa     2,y
                   ldaa     1,y
                   adda     #530
                   staa     1,y
                   ldaa     0,y
                   adda     #530
                   staa     0,y
                   pula
                   pula
                   pulx
                   rts

TES.trans           idx      TES.CommId
                   pshx
                   pshy
                   ldaa     #14
                   psha
                   tax
                   stx
                   ldaa     SYS.wparam
                   ldab     #SCI
                   swi      #SCI_WriteComm
                   pula
                   puly
                   pulx
                   ldd      intRegreso
                   cpd      #50
                   bnc     cv_RTL.err2
                   rts
cv_RTL.err2        idx      #TES.err2
                   jar
                   rts

TES.Pausa          idx      #byteAccion
                   ldaa     0,x
                   cmpa     #CM_PAUSA
                   beq      TESFausa.1
                   ldaa     #CM_PAUSA
                   staa     0,x
                   rts
TESFausa.1         ldaa     #0
                   staa     0,x
                   rts

TES.Exit           idx      #TES.SCI
                   pshx
                   tax
                   stx
                   ldaa     SYS.wparam
                   ldab     #SCI
                   swi      #SCI_CloseComm
                   pula
                   pulx
TESa12             ldx      #tesis.hwnd
                   stx      SYS.wparam
                   ldd      #WM_QUIT
                   std      SYS.mag
                   ldaa     #LCD
                   ldab     #MSG
                   swi
                   ldx
                   pshx
                   tax
                   stx      SYS.wparam

```

```
ldaa #RTI
ldab #RTI_enablenotif
swi
ldaa #ADC
ldab #ADC_enablenotif
swi
puls
ldaa #SYS
ldab #SYS_terminate
swi
fdb #EOB
```

TES.fin

Fe de erratas.

Capítulo 3, página 57: dice "El controlador TDA1085 necesita estar referenciado...."; debe decir: "El controlador TDA1085 necesita estar referido....".

Capítulo 3, página 52: dice "El circuito TDA1085 monitorea señales de voltaje y corriente; por lo que deben estar referenciadas a la alimentación del motor"; debe decir: "El circuito TDA1085 monitorea señales de voltaje y corriente; por lo que debe estar referido a la alimentación del motor."
