

24. 87

**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

CONTROLADOR
PARA
ACTUADORES

TESIS PROFESIONAL

Que para obtener el Título de
INGENIERO MECANICO ELECTRICISTA

Presenta

Héctor Palma Moreno

Director de Tesis: M. I. Luis Alvarez-Icaza Longoria

MEXICO, D. F.

1988



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 está 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.

INDICE

RESUMEN	1
1. INTRODUCCION	2
Antecedentes	2
Contenido	4
2. DESCRIPCION GENERAL	5
Consideraciones preliminares	5
Definición y funcionamiento de los elementos	8
3. DESCRIPCION FUNCIONAL	10
Modos de operación	10
Criterio y metodología de programación	13
Funcionamiento general del programa	14
4. DESCRIPCION FISICA	24
Microcomputadora	26
Tarjeta analógica	33
Tarjeta digital	36
Tarjeta de pruebas	37
5. COMUNICACION CON EL PUESTO CENTRAL	42
Protocolo general	42
Lista de códigos de comunicación	44

6. CONCLUSIONES	46
7. REFERENCIAS	47
ANEXOS	
A. DESCRIPCION ENTRADA-PROCESO-SALIDA	49
B. LISTADO DEL PROGRAMA DE LA MICROCOMPUTADORA	66
C. DIAGRAMAS ELECTRONICOS, DE DISPOSICION Y LISTAS DE COMPONENTES	116
D. INSTRUCTIVO DE OPERACION	143

RESUMEN

Se describe un dispositivo orientado al manejo de actuadores en el control de procesos. Mediante el mismo se pueden manejar actuadores analógicos y digitales a través de un esquema jerárquico de tres niveles: local, remoto y automático,

El equipo se basa en una microcomputadora especializada que maneja, a través de un número variable de módulos, los distintos actuadores que se le conectan. Se consigue así un dispositivo que se adapta a procesos de muy distinta magnitud.

El sistema desarrollado forma parte de un sistema de control de procesos en el que existe un órgano de toma de decisiones, para comunicarse con el mismo está dotado de un enlace serie convencional.

El equipo puede realizar operaciones complementarias para proteger modificaciones en el estado de actuadores que se encuentren fuera de servicio o en mantenimiento, y está dotado de un acoplamiento de pruebas que permite verificar el funcionamiento tanto del dispositivo como de los actuadores.

1. INTRODUCCION

ANTECEDENTES

Existen diversas formas para abordar el problema de diseño de sistemas para control de un proceso cualquiera. La tendencia actual, sin embargo, se dirige a la realización de sistemas en que las tareas se encuentren distribuidas entre varios equipos, de tal forma que pueda existir flexibilidad en las características de los mismos y no se tenga excesiva dependencia de un sólo equipo central, cuya falla ocasione la del sistema completo.

En el Instituto de Ingeniería de la UNAM se diseñó un sistema para control de procesos que distribuye las tareas fundamentales en un conjunto de dispositivos especializados para realizar alguna de las funciones básicas del control de procesos. La filosofía de diseño se basa en el esquema que se muestra en la Fig. 1.1. Como puede notarse, se considera que las tareas de gestión para el control de un proceso se pueden dividir en adquisición de datos, toma de decisiones y acción sobre el proceso. El objeto del presente escrito es describir un dispositivo que se especializa en el manejo de los actuadores que se encuentran típicamente en un proceso cualquiera, al que se denominó controlador de actuadores. Los dispositivos para el manejo de los otros bloques se describen en las Refs. 1, 2 y 3.

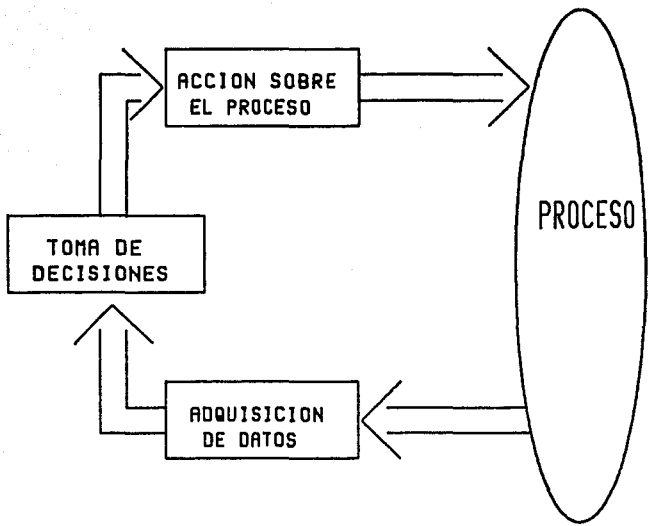


FIG. 1.1 ESQUEMA PARA EL DISEÑO DE SISTEMAS DE CONTROL

El sistema se construyó alrededor de una micromcomputadora especializada, a la que se dotó de los periféricos necesarios para manejar actuadores analógicos y digitales.

El controlador de actuadores se desarrolló como parte del sistema para control de una planta solar experimental que se construyó en el campus de Ciudad Universitaria.

CONTENIDO

En los siguientes capítulos y anexos se describe la estructura, funcionamiento y operación del Controlador de Actuadores.

El capítulo 2 contiene algunas consideraciones que se tomaron en cuenta para el diseño y define de manera formal los elementos que se manejan en el dispositivo. En el capítulo 3 se explica el funcionamiento del programa, la forma como opera el dispositivo en sus distintos modos de operación y las tareas principales que desempeña.

En el capítulo 4 se detalla la construcción de los diferentes componentes del controlador de actuadores y la forma en que estos se relacionan.

El capítulo 5 se refiere a la comunicación con el puesto central, en él se especifica el protocolo en que están basadas dichas comunicaciones y se mencionan los códigos en particular a los que responderá el controlador de actuadores.

Se incluyen, finalmente, dos capítulos con las conclusiones y referencias, respectivamente.

La información técnica se distribuye en cuatro anexos al final del trabajo; estos contienen la descripción entrada-proceso-salida de las principales tareas, el listado del programa que ejecuta el dispositivo, los diagramas electrónicos y de disposición de componentes, y un instructivo de operación.

2. DESCRIPCION GENERAL

Este capítulo contiene algunas consideraciones tomadas en cuenta para el diseño y las definiciones necesarias para situar al controlador de actuadores en el contexto de los sistemas para control de procesos. En particular se refiere a la estructura jerárquica del funcionamiento de los actuadores de un sistema de control y a la estructura de información que se debe asociar con cada actuador.

CONSIDERACIONES PRELIMINARES

El objeto primordial de este controlador es conseguir que los actuadores de un proceso se puedan manejar en forma automática. Se consideró además como hipótesis básica que todos los actuadores responden a señales analógicas o digitales.

Los sistemas de control de procesos tienen una estructura jerárquica con diversos números de niveles. Esta estructura tiene por objeto que las órdenes provenientes de niveles de más jerarquía puedan excluir a las

que llegan de los niveles de menos importancia. Para el caso del controlador de actuadores que nos ocupa se adoptó una estructura a tres niveles como la que se muestra en la Fig. 2.1.

El nivel local es el de mayor prioridad y se diseña para permitir la operación del actuador a través de controles manuales situados in situ junto al mismo. Las funciones de este nivel son múltiples: permite constatar el funcionamiento apropiado, sacar de funcionamiento equipos dañados o que funcionan incorrectamente o aislar alguna parte del proceso en que se realizan labores de mantenimiento, entre otras.

El segundo nivel en importancia es el remoto. Se considera asociado a controles manuales situados en un panel de control que permiten la acción directa del operador del proceso sobre el estado de cada actuador. Este panel se localiza normalmente en el puesto de supervisión del proceso. La intención de las órdenes provenientes de este nivel es la de permitir que el operador tome las acciones correctivas necesarias por la presencia de una condición de emergencia.

Finalmente, el nivel de menor jerarquía es el automático. Se considera que las órdenes correspondientes a este nivel son las que provienen del controlador de actuadores que nos ocupa en este escrito. Para un proceso automatizado es el nivel normal en que funciona el sistema de control.

Una ventaja adicional de la implantación de sistemas de control jerárquicos, es la de permitir el funcionamiento del proceso independientemente de la presencia del órgano de toma de decisiones.

Como ya se ha mencionado, se decidió desarrollar el dispositivo alrededor de una microcomputadora y que el enlace de esta con el resto de los equipos se realizara a través de una canal serie estándar.

Dado que el número y tipo de los actuadores que se emplean en los procesos es muy variable, también se consideró apropiado que el controlador de actuadores pudiera crecer modularmente para adaptarse, lo mejor posible, a la configuración específica de algún proceso en particular. Para los efectos de este escrito se entenderá como configuración el número de módulos, analógicos o digitales, que integran este dispositivo.

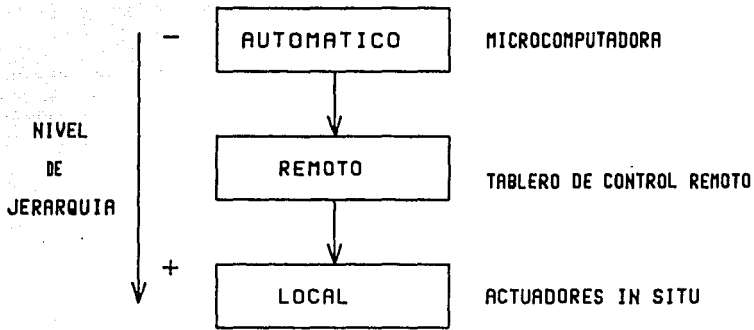


FIG. 2.1 ESTRUCTURA DEL CONTROL JERARQUIZADO

DEFINICION Y FUNCIONAMIENTO DE LOS ELEMENTOS

Para manejar apropiadamente los actuadores, tanto analógicos como digitales, se requiere definir estructuras de información asociadas a cada uno. A partir de dichas estructuras se realizan operaciones o transformaciones mediante las cuales se determinan las órdenes a los actuadores.

Para los fines de este trabajo se define como canal a la unidad mínima sobre la cual el controlador puede realizar modificaciones. El canal será analógico cuando maneja un actuador de este tipo, y digital en caso contrario. Se entenderá como estado al valor que tiene la orden enviada por el controlador hacia un actuador en particular.

Como el controlador de actuadores se basa en una microcomputadora, internamente sólo trata con valores digitales. De aquí que las señales analógicas se manejen como números enteros representados por variables de ocho bits de precisión.

El estado de los canales digitales se representa mediante palabras de un sólo bit de extensión. Esto no impide agrupar funcionalmente los canales digitales, en los casos en que un actuador requiera más de un bit para funcionar apropiadamente.

En muchas ocasiones resulta de utilidad impedir que se modifique el estado de algún actuador, bien porque se encuentre defectuoso, se estén efectuando pruebas, o porque la parte del proceso en que se ubica esté fuera de servicio. Para conseguirlo, sin tener que recurrir a los niveles local y remoto del control jerárquico, el controlador de actuadores maneja una opción para inmovilizar el estado de cualquier actuador, a la que se conoce como candado. Cuando se establece un candado se asocia al actuador un valor prefijado. El valor del candado y su estado asociado se determinan por omisión al conectar el dispositivo, según valores que se almacenan en la memoria permanente de la microcomputadora. Sin embargo, en cualquier momento el órgano de toma de decisiones, o puesto central, puede mandar una nueva lista de candados y estados asociados a ellos.

El candado se representa mediante una variable de un bit, sin importar el tipo de actuador, mientras que el estado asociado a la presencia de candado debe tener la misma representación que corresponda al actuador en cuestión.

De acuerdo con lo anterior el estado del i ésimo actuador se obtiene como el resultado de la siguiente operación:

$$E_i = C_i * EC_i + \sim C_i * I_i \quad (2.1)$$

donde E_i =estado del actuador; C_i =candado; EC_i =Estado en candado; I_i =Información proveniente del puesto central. Los operadores + y * indican disjunción y conjunción, respectivamente.

La operación tiene por objeto tomar la información del puesto central si no se ha definido el candado, o el estado asociado a este, en caso contrario.

Los valores correspondientes a C_i , EC_i , I_i , E_i para todos los actuadores se manejan en tablas en las que el número de actuador actúa como índice.

3. DESCRIPCION FUNCIONAL

Este capítulo se refiere al funcionamiento del controlador de actuadores, para ello describe los distintos modos de operación permitidos, menciona los criterios y métodos de programación, y explica el funcionamiento del programa que ejecuta la microcomputadora.

MODOS DE OPERACION

Se establecieron tres formas o modos para el funcionamiento del controlador de actuadores: normal, congelamiento y pruebas. A continuación se describe cada uno de ellos.

Modo normal

Este modo, como su nombre lo indica, corresponde al conjunto de tareas que normalmente ejecutará el controlador de actuadores. Las principales

de ellas consisten en mantener la comunicación con el puesto central a través del canal serie de enlace y calcular el estado correspondiente a cada actuador según la información y comandos correspondientes.

Modo de congelamiento

En el modo de congelamiento se mantienen sin modificación los estados de todos los actuadores y sólo se supervisa que no exista de parte del operador del controlador de actuadores, una petición para pasar al modo normal o de pruebas. Respecto a las comunicaciones, sólo se transmiten los datos pendientes, pero se inhibe totalmente la recepción. Este modo se diseñó como un auxiliar para verificar el estado del equipo.

Modo de pruebas

La operación en este modo tiene por objeto realizar pruebas sobre el funcionamiento de cualquier actuador que se maneje mediante este controlador. Las pruebas se pueden efectuar de dos maneras: con o sin acoplamiento de pruebas.

Para el primer caso se utiliza un acoplamiento de entrada salida que contiene un teclado y un despliegue de seis dígitos. En el otro se ocupa una triplete de interruptores para controlar la secuencia de las pruebas.

Al igual que en el modo anterior, las comunicaciones se suspenden salvo para terminar de transmitir datos solicitados con antelación.

TAREAS PRINCIPALES

A continuación se proporciona una descripción somera de las tareas principales que ejecuta el controlador de actuadores.

Recepción de mensajes

Esta tarea administra la comunicación del puesto central hacia el controlador de actuadores. Los mensajes que se reciben están formados por bloques de caracteres ASCII, cuya estructura se describe

con más detalle en el capítulo 5.

La función de esta tarea es la de integrar los caracteres recibidos y revisar que formen un bloque con la estructura prescrita. De ser así, se analiza el contenido del bloque y se procede a colocar la información recibida en el lugar apropiado.

Para manejar los bloques de caracteres recibidos se cuenta con dos espacios de memoria cuya magnitud se ha calculado en función de la longitud máxima que se puede esperar en un mensaje, más un margen de seguridad. La tarea de recepción maneja un sistema de banderas para decidir en que almacén colocar la información y para indicar que alguno de ellos contiene información que ya ha sido validada.

La recepción y análisis primario de los caracteres se realiza por interrupción.

Transmisión de mensajes

El objeto de esta tarea es el de manejar el tránsito de mensajes entre el controlador de actuadores y el órgano de toma de decisiones. Este intercambio de información también se realiza a través de bloques de caracteres. Para manejar el envío se implantó una cola de tareas pendientes de transmitir. En esta cola se almacena un indicador por cada bloque a transmitir.

La tarea se divide a su vez en dos subtareas. La primera se encarga de revisar la cola de tareas pendientes, si encuentra alguna debe integrar el bloque de caracteres a transmitir, lo colocará en un área de memoria reservada para ello e iniciará su transmisión. Después de esto actualizará la colas de tareas de transmisión pendientes.

La segunda subtaska se encarga de enviar los caracteres restantes del bloque cuya transmisión se haya iniciado.

Cálculo del estado de los actuadores

La finalidad de esta tarea es aplicar la ecuación (2.1) a todos los actuadores que maneje el controlador. Esta tarea se activa de manera periódica, pero el cálculo de la tabla se sujeta a la presencia de nueva información proveniente del puesto central que provoque cambios en el

estado de los actuadores.

Modificación de estado

Esta tarea toma el último estado calculado y lo envía a todos los actuadores del sistema.

Pruebas con acoplamiento

En esta rutina se manejan el teclado y el despliegue. El objeto de las pruebas es permitir que el usuario del controlador de actuadores pueda modificar arbitrariamente el estado de cualquiera de las salidas del dispositivo. En este caso se deja al operador el manejo de la secuencia de inicio, realización y fin de pruebas.

Pruebas sin acoplamiento

Mediante esta tarea es posible colocar los actuadores en estados predefinidos (por ejemplo en mínima y máxima escala). El usuario cuenta con tres interruptores para manejar la secuencia de pruebas, que indican tipo de prueba, siguiente del tipo y siguiente valor, respectivamente.

CRITERIO Y METODOLOGIA DE PROGRAMACION

El programa se codificó mediante el ensamblador de una microcomputadora APPLE II Plus (Consultar Refs. 4 y 5 para más detalles).

Para elaborarlo se partió de los modos de operación mencionados y se definió el conjunto de tareas que se deberían realizar en cada uno.

Posteriormente se tomaron estas tareas principales y se dividieron en subtareas. Para estas últimas se especificó con detalle la entrada, proceso y salida. Las subtareas se determinaron de forma tal que fuesen comunes para el mayor número posible de tareas principales. Finalmente se propuso la articulación de dichas subtareas mediante un programa principal y otro de atención a interrupciones.

La subtareas se escribieron en lenguaje ensamblador y se probaron por separado. Después se agruparon en las tareas correspondientes mismas

que también se comprobaron de manera aislada. Finalmente, una vez que se obtuvieron resultados satisfactorios se integró el programa principal y el de atención a interrupciones, que incluyen de manera ordenada todas las tareas anteriores.

Todo el programa fué grabado en una memoria de lectura (ROM) y se procedió a las pruebas finales.

FUNCIONAMIENTO GENERAL DEL PROGRAMA

El programa con el que la microcomputadora maneja los actuadores se divide en dos partes. La primera, o programa principal, se encarga de ejecutar un ciclo, cuya función es la de atender las tareas relativas al modo de operación elegido. La segunda parte, o programa de atención a interrupciones, tiene como tarea fundamental manejar los eventos asíncronos o espontáneos, que por dicha naturaleza resultan difíciles de atender en el programa principal, sin sobrecargar el ciclo mencionado.

En este inciso se proporciona un algoritmo del programa que ejecuta la microcomputadora, tanto en la parte referente al programa principal, como en la que corresponde a la atención de interrupciones.

Programa principal

El algoritmo básico que ejecuta este programa está dado por los siguientes pasos:

- 0- Iniciar y preparar apuntadores y variables.
- 1- Leer del modo de operación
- 2- Si cambia el modo reprogramar los periféricos
- 3- Verificar la bandera de modificación de estado.
- 4- Si (3) se cumple, modificar el estado.
- 5- Revisar la cola de transmisión, si hay algo pendiente, transmitir.
- 6- Verificar si se puede trasladar el bloque recibido a la zona de memorias, hacerlo si es posible.
- 7- Verificar si se va a trabajar en el modo de pruebas, en caso afirmativo las realiza; de lo contrario se va al paso 1.

Paso 0.

La preparación inicial se realiza en dos etapas: la primera consiste en generar un retraso que permite que se establezcan las señales de todos los periféricos, especialmente el despliegue de cristal líquido.

En la segunda etapa se da valor inicial a las banderas y apuntadores que se usarán en el programa. Se prepara la cola de transmisión y se definen los primeros valores de los candados y estados asociados con ellos, a partir de los valores que contiene la memoria permanente. Por último, se configura el puerto paralelo para que se pueda leer el modo de operación elegido. El modo de operación se inicializa con una primera lectura al puerto paralelo, lo mismo ocurre con la configuración. Se guarda registro de dicho modo y configuración. En seguida se va al paso 3.

Paso 1.

En este paso se lee la información del puerto paralelo. De esta lectura se obtienen el modo de operación y la configuración del controlador de actuadores. Esta última se compara con los límites máximos admisibles, para determinar su validez. En caso necesario realiza ajustes para no violar dichos límites.

Paso 2.

El modo de operación se compara con el modo anterior, si se detecta cambio, se programan los periféricos de la microcomputadora según el nuevo modo. El modo de operación leído se salva como el modo anterior, con el fin de tener un patrón de comparación para detectar el siguiente cambio de modo.

Paso 3.

Verifica la bandera de refresco, si esta se encuentra abajo irá al paso 5, en caso contrario continuará. La bandera de refresco puede ser levantada por el programa de atención a interrupciones que se describe más adelante.

Paso 4.

Consulta la bandera de recálculo, si está levantada pregunta si existen módulos analógicos. De ser así calculará el nuevo estado analógico, y levantará la bandera de vaciado. Hace lo propio con los módulos digitales.

Si la bandera de vaciado está arriba efectúa la modificación de estado. Ello se consigue realizando escrituras ordenadas en las tarjetas analógicas y digitales.

Finalmente bajará las banderas de recálculo, vaciado y refresco, y continuará.

Paso 5.

Pregunta si la cola de tareas de transmisión está vacía. En caso afirmativo continúa con el paso 6.

Si dicha cola no está vacía, verifica que no se encuentre transmitiendo un bloque de caracteres. Esto se realiza con la bandera de primer carácter enviado que se levanta cuando este es enviado y se baja cuando se termina de transmitir el bloque.

Si no hay ningún bloque en proceso toma el primer elemento de la cola. Según su valor formará el bloque de caracteres correspondiente. Toma el primer elemento de dicho bloque y lo pone en el registro de transmisión del puerto serie. En seguida configura dicho puerto para que atienda interrupciones por transmisión.

Las interrupciones por recepción del puerto serie se habilitan o deshabilitan según el modo elegido. En este paso el programa respeta la forma elegida.

El resto de los caracteres del bloque se transmiten en el programa de atención a interrupciones. Este último baja la bandera de primer carácter al iniciar la transmisión del último elemento del bloque.

Paso 6.

Consulta la bandera de guardado que indica que se ha recibido sin problemas un bloque de caracteres completo. Si la bandera está baja irá al paso 7.

Cuando está alta, consulta si el bloque recibido tiene información a guardar o si es una orden para el controlador de actuadores transmita el estado de los actuadores hacia el puesto central. En este último caso coloca la petición como un indicador en la cola de transmisión, baja la bandera de guardado y va al paso 7.

Cuando se trata de información para guardar, revisa que el número de caracteres que se recibió coincida con los que se esperan. Si esto se cumple traslada los datos hacia las tablas correspondientes. Baja la bandera de guardado y levanta la de recálculo. En el caso en que no coincidan los números mencionados salta directamente al paso 1.

Paso 7.

Verifica el modo de operación, si no se trata del del pruebas se va al paso 1. En caso contrario, leerá el puerto paralelo para saber si el controlador está dotado con el acoplamiento de pruebas.

Sin el acoplamiento de pruebas, las pruebas se hacen en forma secuencial fija. Los canales se recorren uno a uno hasta concluir, para pasar de nueva cuenta al primer canal.

El manejo se efectúa mediante por tres interruptores, conectados al puerto paralelo. El primero indica si serán pruebas analógicas o digitales; en cualquier momento de la prueba, puede cambiar de analógico a digital o viceversa. El segundo indica cambio al siguiente canal de la secuencia.

El último interruptor permite manejar los valores; en el caso de pruebas analógicas entrega la secuencia de valores 00_H , $7F_H$, FF_H ; en el caso de las digitales uno o cero lógico (0V ó 5V).

Las pruebas sin acoplamiento se pueden suspender en cualquier momento con la sola modificación del modo de operación que forzará la ejecución del paso 2.

Para las pruebas con acoplamiento, dado que estas se realizan en el programa de atención a interrupciones, el programa permanece en un ciclo de lectura del modo de operación. Este termina cuando se elige un modo diferente al de pruebas, en cuyo caso se va al paso 2.

Lineamientos generales de la atención de interrupciones

El proceso extraordinario en las computadoras permite un adecuado tratamiento para eventos asíncronos. En el controlador de actuadores una parte importante de la operación está determinada por este tipo de eventos. Debido a que el microprocesador elegido sólo maneja una interrupción enmascarable, todo el proceso extraordinario se canalizó por vía de esta señal.

El manejo de los eventos por interrupción proporciona dos ventajas fundamentales para la programación. La primera estriba en que eventos externos se pueden atender sin necesidad de contar con un ciclo de interrogación que sobrecargue inútilmente al microprocesador. La otra ventaja radica en que al utilizar la posibilidad de inhibir en algún periférico la generación de interrupciones, es posible bloquear a discreción la ocurrencia de ciertos eventos, según convenga a la ejecución de un modo de operación específico. Bajo este enfoque, por ejemplo, para cambiar del modo normal al de congelamiento, todo lo que se requiere es inhibir las interrupciones por recepción. El resto del programa continúa igual.

La atención a interrupciones se realiza por el método de encuesta. Este consiste en interrogar uno a uno a todos los dispositivos que pueden generar una señal de interrupción, hasta encontrar a aquel que la produjo. El orden de interrogación determina la prioridad que se asigna a los diferentes dispositivos que puedan generar interrupciones.

Una vez que se encuentra el dispositivo que generó la interrupción se ejecuta la rutina correspondiente para darle servicio. Una parte medular de esta consiste en garantizar que dicho dispositivo no siga produciendo la señal de interrupción.

El programa no acepta interrupciones concurrentes, esto es, atiende completamente una, para después pasar a la siguiente.

Programa de atención de interrupciones.

En el controlador de actuadores existen cinco fuentes posibles para la señal de interrupción, que son:

1. Reloj1 -para el autoreset-
2. Por recepción I
3. Por transmisión I^{rx}
4. Reloj2 -consulta la bandera de recálculo-
5. Teclado -del acoplamiento de pruebas-

El orden de la lista corresponde con el que se sigue para determinar la fuente de la interrupción.

Los relojes son contadores especiales que se programan para producir señales de interrupción con una frecuencia fija. Para que alguno de ellos la produzca se requieren dos condiciones:

1. Que concluya su cuenta, al ocurrir esto, siempre se enciende su bandera de interrupción.
2. Que esté habilitado para generar una interrupción al ocurrir la primera condición, esto se consigue mediante una escritura en el registro de control del reloj.

Cuando ocurre una interrupción el microprocesador salva automáticamente el registro de estados y el contador de programa, pero no salva ninguno de los otros registros de trabajo. Para garantizar total independencia entre el programa principal y el de atención a interrupciones, cada vez que se inicia un ciclo de servicio a la señal de interrupción se salvan los valores de dichos registros en variables auxiliares. Al finalizar el servicio correspondiente se recupera el valor de esas variables y se coloca en los registros apropiados.

En todos los casos, el acceso al dispositivo que interrumpe por medio de la rutina de servicio apropiada, garantiza que este deja de producir la señal de interrupción.

Interrupción del Relojl

Con esta interrupción se implementa un vigía (watch-dog) que tiene por objeto garantizar que no se pierda el control en la ejecución del programa que ejecuta el microprocesador. El vigía se implementa combinando el Relojl con un par de monoestables redispables (74LS123). La señal de salida del Relojl se utiliza como entrada de los monoestables conectados en cascada. La salida del segundo monoestable está conectada a la pata de RESET del microprocesador. Si no se redispara el primer monoestable con una frecuencia mínima, la salida del segundo producirá un ciclo de inicialización.

La frecuencia mínima de redisparo se consigue de la siguiente forma: el Relojl se programa para contar un único periodo de tiempo. Al vencer este se produce una señal de interrupción y se modifica el estado de la salida del Relojl. Al atender la interrupción se reprograma el Relojl para que cuente de nuevo el periodo mencionado. La reprogramación produce además, como efecto lateral, un cambio en el nivel de la salida del reloj, que equivale a producir un nuevo disparo para los monoestables.

Si el periodo de tiempo mencionado es menor al que se produce según la red RC de los monoestables, el redisparo ocasionará que nunca se produzca un ciclo de inicialización.

La hipótesis detrás de este vigía es que mientras el microprocesador atiende la interrupción del Relojl apropiadamente, retiene el control; si por el contrario, no puede dar servicio a dicha interrupción se producirá un ciclo de inicialización que reprogramará todos los dispositivos y le hará ganar de nueva cuenta el control del programa.

Interrupción por recepción

Quando el puerto serie recibe un carácter completo y el dispositivo trabaja en modo normal, se produce un señal de interrupción. Al dar servicio a esta lo primero que se revisa es que el carácter en cuestión se haya recibido sin problemas de paridad, formación o superposición. Si este no es el caso se levantará una bandera que indica carácter inválido.

Quando el carácter recibido fue correcto y la última bandera mencionada no está alta, se almacena el dato en una zona de memoria temporal a fin de empezar a formar el bloque de recepción. En seguida se analiza la

estructura de dicho bloque para conocer si cumple con todas las reglas de estructura. Cuando se detecta el fin del bloque y no hay problemas, se levanta la bandera de guardado y se concluye con el servicio a la interrupción.

La bandera de carácter inválido se baja cuando se recibe sin problemas el carácter que indica el principio de un nuevo bloque.

Interrupción por transmisión

Al atender esta interrupción se toma el siguiente elemento de la tabla de caracteres a transmitir y lo envía por el puerto serie. Se pregunta si se trata del último. Cuando esto ocurre baja la bandera de primer carácter, programa el puerto serie para no generar más interrupciones por transmisión y actualiza la cola de tareas por transmitir.

La actualización de la cola consiste correr todos los elementos de la cola un elemento hacia adelante y en decrementar una unidad el apuntador al último elemento no nulo de la cola. El primer elemento, que correspondía con la tarea recién finalizada, se pierde.

Interrupción del Reloj²

Esta interrupción tiene como objetivo proporcionar un patrón de tiempo con base en el cual decidir el recálculo del estado de los actuadores. La interrupción se presenta porque el Reloj² se programa para interrumpir periódica y repetitivamente al microprocesador.

El servicio a la interrupción consiste en levantar la bandera de refresco cuando la de recálculo esté también levantada.

Interrupción por teclado

La interrupción por teclado tiene por objeto manejar la secuencia de pruebas, en el caso en que se cuenta con el acoplamiento respectivo. Para verificar tanto los módulos analógicos como los digitales se trabaja en tres niveles.

Primer nivel: determina el tipo de módulos bajo prueba, analógicos o digitales. Por ello sólo se aceptan los caracteres

A ó D, cualesquiera otro será ignorado. La letra elegida se coloca en el despliegue D_6 (el que está situado más a la izquierda).

Segundo nivel: determina el número de canal bajo prueba, este se mostrará en los despliegues D_5 y D_4 , que funcionan como registros de corrimiento, esto es, cada vez que se oprime una tecla esta se muestra en D_4 y la anterior se corre a D_5 . Si se proporciona un número inapropiado según la configuración, no es posible pasar al siguiente nivel.

Tercer nivel: permite modificar el valor del canal a prueba. El valor elegido se muestra en los despliegues D_2 y D_1 , que también funcionan en este caso como registros de corrimiento. Para conseguir que el valor se modifique en el actuador correspondiente se debe oprimir la tecla Exc.

El cambio al nivel inferior se consigue al oprimir la tecla Exc, y el regreso al nivel inmediato superior si se oprimen simultáneamente Shift y E.

El servicio a la interrupción del teclado depende del nivel en que esté situada la secuencia de pruebas, mismo que se maneja a través de una bandera. Las únicas operaciones comunes a todos los niveles son la recuperación de código que corresponde a la tecla oprimida y en el caso en que esta corresponda a Shift-E, la modificación apropiada de la bandera.

Cada vez que ocurre una interrupción se ejecuta una rutina asociada con el nivel presente. Existen en total cuatro rutinas que se describen a continuación.

Rutina 1.- se usa en el primer nivel; identifica si se oprimieron las teclas A, D o Exc, cualesquiera otra es ignorada terminando el servicio. Si la tecla fué A ó D, la despliega en el 6° lugar de la pantalla. Si se oprimió Exc consulta la configuración del sistema para ver si hay tarjetas analógicas o digitales, según la tecla oprimida. Si este es el caso levantará una bandera para conseguir el cambio nivel en la siguiente interrupción. En caso de que no haya tarjetas del tipo solicitado permanece en el mismo nivel y termina el servicio a la interrupción.

Rutina 2.- está asociada con el segundo nivel; la primera vez que se ejecuta inicializa el valor del número de canal solicitado y del carácter anterior. Si el carácter recibido no es Exc, lo coloca en D_4 , y el anterior situado ahí lo recorre a D_5 . Si el carácter fue Exc, consulta si el número definido en D_5 y D_4 corresponde a un canal permitido; los requisitos para esto último son que el número en cuestión sea mayor que cero y menor o igual que el límite que se define en la siguiente tabla, según la cantidad de módulos de cada tipo:

	No. módulos	1	2
Tipo de módulo			
Analógico		10 _H	20 _H
Digital		40 _H	80 _H

En caso de que el número sea válido, indica el cambio al tercer nivel por medio de la bandera respectiva.

Rutina 3.- se usa en el tercer nivel de pruebas de canales analógicos; la primera vez que se ejecuta borra los despliegues D_2 y D_1 , e inicializa el valor del carácter anterior. Para el caso en que se trate de un carácter diferente de Exc, funciona igual que la parte anterior, salvo que trabaja con D_2 y D_1 . Cuando se oprime Exc toma el valor definido por D_2 y D_1 y lo transfiere al módulo analógico apropiado.

Rutina 4.- permite manejar el tercer nivel para la prueba de actuadores digitales. Se acepta que se opriman las teclas 0, 1 y Exc. Cuando se trata de las dos primeras, la despliega en D_1 . En el caso de Exc procede a trasladar el valor de D_1 al actuador digital indicado en la parte 2.

4. DESCRIPCION FISICA

El controlador de actuadores tiene una estructura como la que se muestra en la Fig. 4.1. En ella se puede notar que el dispositivo está constituido por una microcomputadora, acoplamiento para manejar los actuadores analógicos y digitales, y una tarjeta para pruebas y despliegue local.

La microcomputadora y los acoplamiento para actuadores analógicos y digitales se alojan en un chasis y se interconectan a través de un conjunto de conectores colocados sobre una tarjeta madre, ésta permite el intercambio de las señales de control, direcciones y datos.

En el caso del acoplamiento de pruebas y despliegue local, la conexión a la microcomputadora se hace mediante una tarjeta de extensión que le envía las señales mencionadas. La conexión de las señales de salida de las diferentes tarjetas se realiza a través de conectores apropiados.

En los siguientes incisos se describen con detalle cada uno de los elementos mencionados.

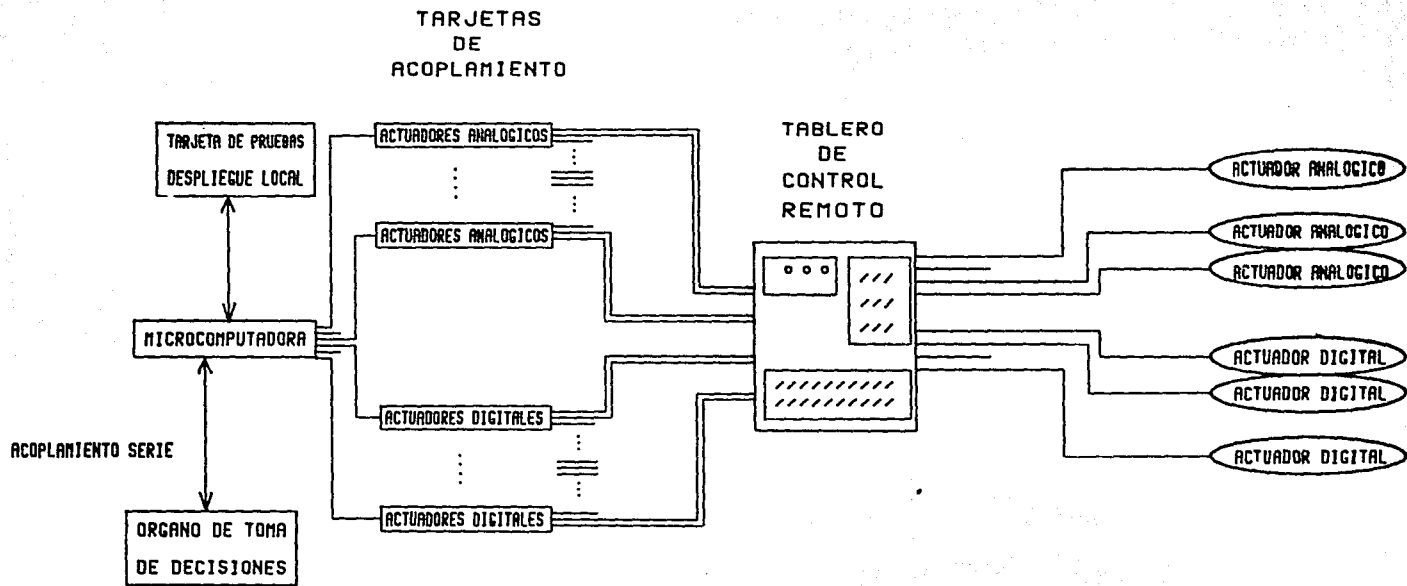


FIG. 4.1 ESTRUCTURA DEL CONTROLADOR DE ACTUADORES

MICROCOMPUTADORA

La microcomputadora que se emplea para el controlador de actuadores está basada en el procesador Rockwell 6502 de ocho bits (Ref. 6). Está provista de 2 Kbytes de memoria de lectura y escritura y 4 Kbytes de memoria de lectura (RAM y ROM, respectivamente). Como periféricos soporta un acoplamiento serie (ACIA), otro paralelo (PIA) y un conjunto de tres relojes (PTM) que producen señales para manejar la secuencia de ejecución de las distintas tareas.

Las señales de sincronía para el procesador, el acoplamiento serie y el conjunto de relojes provienen de un generador de señales de sincronía. (véase Fig. 4.2)

Microprocesador

El funcionamiento del microprocesador se basa principalmente en las siguientes líneas:

D0-D7 -- Líneas de datos. La transferencia de todas las instrucciones y datos ocurre por estas líneas bidireccionales.

A0-A15 -- Líneas de dirección, indican la parte del sistema en que se manejan los datos. Son exclusivamente de salida y pueden direccionar hasta $(FFFF)_{16}$ ó 65536 palabras de memoria.

R/~W -- Lectura/Escritura. Esta línea le permite al procesador controlar la dirección de la transferencia de datos a los demás elementos; si está alta efectúa lectura, en caso contrario, escritura.

Los tiempos para la transferencia de datos a través de las líneas descritas están controlados por un sistema de dos señales de reloj que marcan la sincronía del sistema. Estas señales son simétricas una respecto a la otra, con ciclo de trabajo al 50%. Durante el pulso positivo de la señal de fase 1 las líneas de dirección cambian y comienzan a estabilizarse. Durante la parte positiva de la fase 2 ocurren las transferencias de datos.

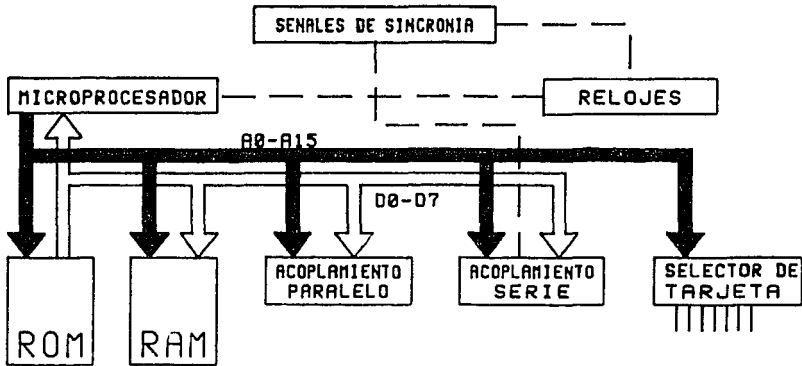


FIG. 4.2 ARQUITECTURA DE LA MICROCOMPUTADORA

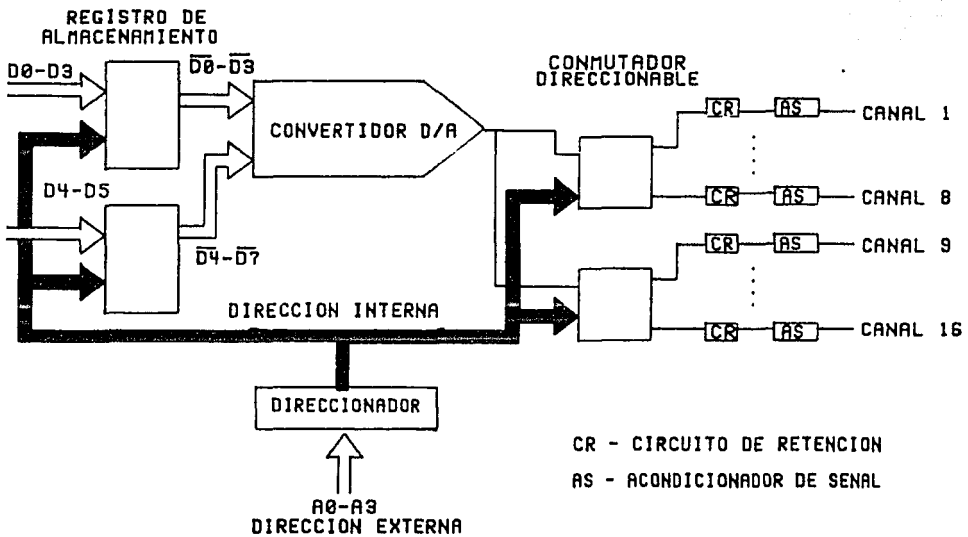


FIG. 4.3 ACOPLAMIENTOS PARA ACTUADORES ANALOGICOS

Los periféricos en general se habilitan y seleccionan mediante la fase 2, que garantiza tanto datos como direcciones válidas.

~IRQ -- Interrupción enmascarable. Es una línea de entrada al microprocesador que le indica que algún periférico requiere atención. Para que una interrupción sea reconocida se requiere que el voltaje en esta línea corresponda a un cero lógico (0.4V) y que la máscara del registro de estado del microprocesador que habilita interrupciones esté baja (bit 2 del registro de estado).

La línea de interrupción funciona en colector abierto, lo que ofrece la posibilidad de colocar más de una señal de interrupción en la misma entrada. Basta que un periférico solicite servicio por interrupción para que el procesador pueda atenderlo. En caso de que más de uno soliciten dicho servicio de manera simultánea, el procesador atenderá la solicitud del periférico con mayor jerarquía.

~RES -- Restablecer. Es una línea de entrada que fuerza un ciclo de reiniciación de la microcomputadora. Cuando su valor es bajo el procesador permanece parado. Una vez que toma valor alto, pasarán 6 ciclos de la fase 2 y en seguida se ejecutará la rutina cuya dirección esté contenida en las localidades $FFFC_H$ y $FFFD_H$ de la memoria.

La microcomputadora construida prevee dos formas para producir un ciclo de reiniciación: a través de un botón, o mediante el vigía que se describió en el capítulo anterior.

Se debe hacer notar que esta línea también está conectada al PIA y PTM, de tal forma que en ellos también se producirá una reiniciación cada vez ~RES tome valor bajo.

Memoria de lectura/escritura RAM (Ref. 7)

Como su nombre lo indica se trata de un dispositivo que puede interactuar con el microprocesador para lectura y escritura de datos e instrucciones. En el sistema se empleó la memoria 6116 que posee 2048 localidades de memoria de 8 bits cada una.

La memoria se conecta al microprocesador a través de las líneas de datos, las 11 primeras líneas de direcciones, la línea de ~R/W y una

línea de selección. Esta última proviene de un decodificador de direcciones que asigna a la memoria RAM las direcciones $0000_H - FFFF_H$.

Memoria de lectura ROM (Ref. 8)

El controlador de actuadores tiene dos memorias de 4096 localidades cada de 8 bits, cada una, de uso exclusivo para lectura. Se utilizó la memoria TMS2532. La conexión al procesador es similar a la descrita en el párrafo anterior.

Las memorias ocupan las zonas de memoria $E000_H - EFFF_H$ y $F000_H - FFFF_H$, respectivamente.

Puerto paralelo PIA (MC6821, Ref. 9)

Se le conoce como PIA (peripheral interface adapter). Tiene dos puertos bidireccionales A y B, cada uno de ocho bits y con dos líneas de control. Los puertos se manejan a través de tres registros: control (CRA y CRB), dirección de datos (DDRA y DDRB) y datos (DRA y DRB). El primero sirve para definir la forma de operación de las líneas de control del puerto y para permitirle que genere interrupciones al procesador. El registro de dirección de datos determina, para cada uno de los ocho bits del puerto, la forma como será operado: entrada o salida. Finalmente el registro de datos permite realizar lecturas o escrituras del puerto, según se haya programado en el registro de dirección.

En el programa desarrollado no se contempla que el puerto paralelo opere por interrupciones, ni se da uso a las líneas de control de los puertos. Todas las líneas de ambos puertos se definieron como entradas.

El PIA está mapeado en las siguientes localidades de memoria:

6000 DDRA ó DRA*
 6001 CRA
 6002 DDRB ó DRB*
 6003 CRB

* Determinado por el bit 2 de CR, 0=DDR, 1=DR

Puerto serie ACIA (MC6850, Ref. 9)

Se le denomina como ACIA (asynchronous communications interface adapter). El ACIA permite que los datos puedan ser recibidos y transmitidos en un formato serie, es decir, funciona como convertidor serie-paralelo y paralelo-serie, respectivamente.

Cuenta con cuatro registros, dos de escritura y los otros de lectura. Los primeros corresponden con el registro de transmisión (TDR), en que se escriben los datos a transmitir por el canal serie, y con el registro de control (CR) que determina los formatos de transmisión y el manejo de las señales de interrupción al procesador.

Los dos registros de lectura corresponden al registro de recepción de datos (RDR), del que se pueden recuperar los datos recibidos por el enlace serie, y el registro de estado (SR), el cual permite conocer la situación de los registros de datos.

Para mayor seguridad e inmunidad de las comunicaciones frente a variaciones en la frecuencia del reloj que marca la sincronía para el ACIA, se eligió el modo de división entre 16 de dicha frecuencia, que opera tanto en transmisión como en recepción. Como la comunicación entre el controlador de actuadores y el puesto central se estableció a 4800 bauds, la frecuencia de alimentación requerida es de 76.8 KHz y se obtiene del generador de señales de sincronía.

La línea de comunicación estará ligada directamente al órgano de toma de decisiones con una línea exclusivamente para transmisión y otra para recepción (full-duplex). Los niveles de voltaje en dichas líneas obedecen al estándar RS232 (Ref. 10).

Los registros del ACIA se accesan en las siguientes direcciones:

4000 SR línea de R/~W=1
 4000 CR línea de R/~W=0
 4001 TDR línea de R/~W=0
 4001 RDR línea de R/~W=1

Relojes programables PTM (MC6840, Ref. 9)

Se trata de un módulo provisto de tres contadores programables de 16 bits cada uno. Su función básica consiste en que el valor de cada

contador se decreta una unidad cuando ocurre un periodo de una señal de alimentación. Las operaciones de escritura sobre cada contador se manejan a través de un registro de control (CR) y de un alimentador que permite prefijar el valor de contador. El registro de control determina las distintas variantes que maneja el contador para los formatos de cuenta, señales de entrada y de salida (que incluyen la señal de interrupción), y procedencia de la señal de cuenta.

Las lecturas se realizan a través de un único registro de estado (SR) para los tres contadores, que permite reconocer que contador generó interrupción, y de tres registros de lectura que permiten recuperar el valor instantáneo de los contadores.

En el controlador, de los tres contadores se utilizan dos: uno para implementar el vigía y el otro para controlar el refresco del estado de los actuadores. El primero de ellos funciona en formato de cuenta simple y el otro en cuenta repetitiva. En ambos casos el decremento del contador se realiza con señales provenientes del generador de señales de sincronía a 1200 Hz y 2400 Hz, respectivamente. Se programan para generar una señal de interrupción al procesador en el instante en que el valor de algún contador llegue a cero.

Las direcciones que corresponden a los diferentes registros en el mapa de memoria son:

Dirección	operación	
	R/~W=0 escritura	R/~W=1 lectura
5000	CR3 si CR20=0 CR1 si CR20=1	no operación no operación
5001	registro de control 2 (CR2)	registro de estado (SR)
5002	BMS alimentador 1	BMS contador 1
5003	BPS alimentador 1	BPS contador 1
5004	BMS alimentador 2	BMS contador 2
5005	BPS alimentador 2	BPS contador 2
5006	BMS alimentador 3	BMS contador 3
5007	BPS alimentador 3	BPS contador 3

CR registro de control (Control Register)
 CR20 bit 0 del registro de control 2.
 BMS bits más significativos
 BPS bits poco significativo

Generador de señales de sincronía (MC14411, Ref. 11)

Este dispositivo es un divisor de frecuencias de 19 etapas, 2 de ellas programables y las restantes fijas. Está diseñado para que se conecte un cristal directamente e incluye la circuitería necesaria para implementar el oscilador que proporcione la frecuencia básica a dividir. Si se alimenta con un cristal de 1.8432 MHz, genera todas las frecuencias que se emplean como estándares para la transmisión de caracteres en formato serie. La primera etapa de división proporciona una señal de 921.6 KHz que determina la frecuencia de las fases 1 y 2 del microprocesador. De las otras 16 salidas fijas, 2 se dirigen al PTM, 1 al ACIA y 2 más al conector de la microcomputadora, de donde se hacen llegar a través de la tarjeta madre al resto de los componentes del controlador de actuadores.

Multivibrador monoestable (74LS123, Ref. 12)

Constituye la otra parte del vigía implementado, junto con el reloj programable. Los dos monoestables que contiene la componente se conectan en cascada. La entrada del primero está directamente conectada a la línea de salida del contador 1 del reloj programable, mientras que la salida del segundo monoestable se conecta a la línea ~RES del procesador.

Los monoestables, como es sabido, son circuitos que responden a una transición de la señal de entrada con un cambio en el voltaje de su salida. El voltaje original se restituye después de un periodo determinado por un circuito RC conectado también al monoestable.

A partir de la programación del contador mencionado se genera un pulso que se presenta al primer monoestable con una frecuencia mayor que la que corresponde a la de su red RC. Mientras se mantenga esta condición, no se presentará ningún pulso de disparo para el segundo monoestable, y por ende tampoco habrá señal de restablecimiento para el procesador.

La hipótesis de diseño del vigía consistió en asumir que, cuando la operación de la microcomputadora es normal, el procesador podrá provocar el disparo del primer monoestable de manera regular. En caso de que existan problemas no podrá hacerlo y con ello se generará un ciclo de restablecimiento.

Para calcular el ancho del pulso (T) se emplea la fórmula:

$$T = 0.45 RC \quad (4.1)$$

En el caso del primer multivibrador $R=33K$ y $C=22 \times 10^{-6} F$ lo que da un ancho de 0.327 s. Para el segundo $R=33K$ y $C=1 \times 10^{-6} F$, que dan un tiempo de 0.015 s.

Circuitos de apoyo

Los circuitos de apoyo más importantes consisten en un conjunto de amplificadores de corriente que permiten que las líneas de direcciones y datos del microprocesador manejen todas las compuertas a que están conectadas sin exceder su capacidad.

Adicionalmente se cuenta con dos decodificadores conectados a las líneas $A_{11} - A_5$ que generan señales para seleccionar 32 páginas de memoria de 2¹¹ Kbytes cada una. Algunas de estas señales se usan directamente en la microcomputadora y otras se envían a través de su conector para poder seleccionar los elementos restantes del controlador de actuadores.

TARJETA ANALOGICA

Los módulos para el manejo de los actuadores analógicos tienen una estructura como la que se muestra en la Fig. 4.3. Estos acoplamientos manejan las señales analógicas a través de 16 registros que almacenan el valor binario que equivale al valor analógico, bajo la precisión usada (ocho bits en este caso). Los valores binarios se presentan a un convertidor digital-analógico que dirige su salida a un circuito conmutador direccionable. En este último circuito, según sea el número de registro que se presentó al conversor, se permite la salida de la señal convertida hacia un circuito de retención. Existe un ciclo de refrescamiento, o barrido, de tal suerte que periódicamente se presenta al convertidor al valor de cada registro.

El acoplamiento puede manejar hasta dieciseis canales analógicos de salida a partir de un sólo convertidor. El diagrama correspondiente a cada canal se muestra en la Fig. 4.4. La constante de tiempo del circuito de retención se calculó de tal forma que el valor de voltaje no perdiese más del 0.5% del valor nominal entre cada ciclo de refrescamiento.

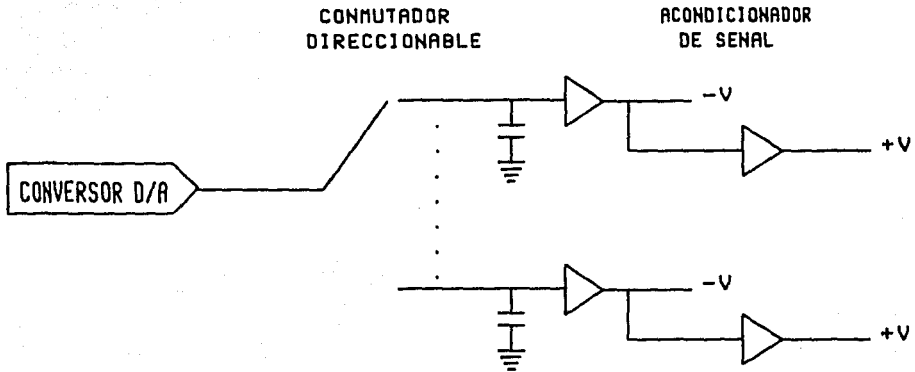


FIG 4.4 CIRCUITOS BASICOS DE RETENCION

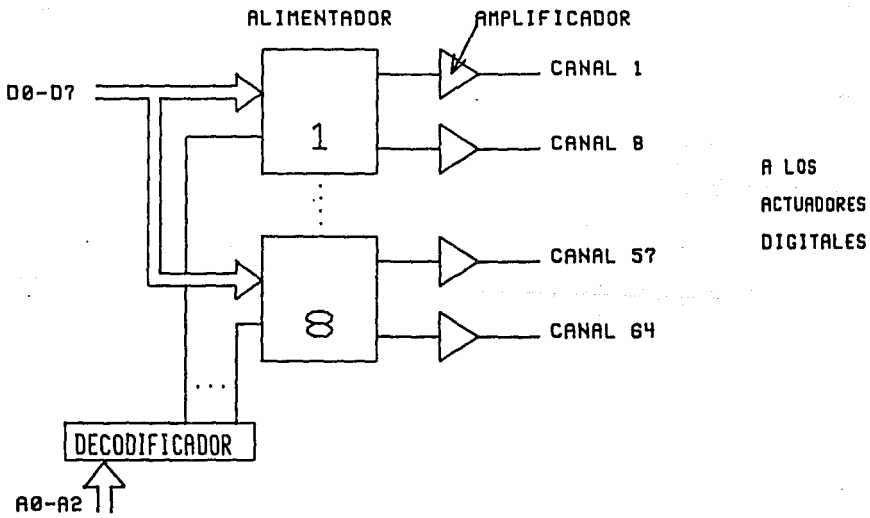


FIG. 4.5 DIAGRAMA DE BLOQUES DEL ACOPLAMIENTO PARA ACTUADORES DIGITALES

A la salida de cada canal analógico se entrega un voltaje en la escala de 0 a 5 voltios y su simétrico (de 0 a -5 voltios). Se emplea esta configuración a fin de poder manejar la presencia de ruido en las señales.

La tarjeta se conecta a la microcomputadora a través de cinco líneas de direcciones A_0-A_4 , las ocho de datos D_0-D_7 , un selector de tarjeta (ST_1 ó ST_2 , según el número de módulo), R/\bar{W} , una salida del generador de señales de sincronía para marcar la frecuencia de refrescamiento (F_3) y la línea de restablecimiento.

El circuito de refrescamiento se basa en un contador ascendente de 4 bits (74LS193). Cada vez que ocurre un periodo en la señal F_3 , se incrementa en uno el valor del contador, lo que se refleja en las líneas de salida de este último. Estas líneas de salida se conectan como direcciones a una memoria de 16 localidades de ocho bits (2 X 74LS89) y a un conmutador direccionable con el mismo número de salidas (2 X CD4051). En cada salida de conmutador se encuentra un retén de orden cero. La transición positiva de la señal F_3 inicia la conversión en el convertidor digital-analógico (DAC0830), de tal forma que el valor situado en la localidad de memoria cuyo número corresponde a las salidas del contador se convierte y presenta al retén de orden cero que tenga el mismo número.

El párrafo anterior describe el ciclo de refrescamiento, falta detallar la forma como se modifica el valor de una canal analógico cualquiera, lo que se hace a continuación. Las cuatro primeras líneas de direcciones (A_0-A_3) se usan para determinar el número de canal a modificar, y están conectadas a las entradas de carga paralela del contador, las líneas de datos están conectadas directamente a las entradas de la memoria. Cuando la línea R/\bar{W} indica escritura, el valor del contador se modifica según las líneas de direcciones dadas y la memoria se carga con el contenido de las líneas de datos. Al no realizarse ninguna otra escritura, el ciclo de refrescamiento continúa a partir de la localidad modificada.

La última línea de dirección (A_4) se usa para bloquear ($A_4=0$) y desbloquear ($A_4=1$) todas las salidas analógicas a la vez. Si se aplica la señal de $\sim RES$ las salidas de las memorias valen cero y los conmutadores están deshabilitados.

A continuación se resume el efecto de los distintos valores de las señales de control:

R/~W	ST	A ₄	
0	0	0	Escribe. Además se emplean A ₀ -A ₃ para las 16 localidades.
X	1	X	Refresca
X	0	1	Habilita

Las direcciones que corresponden a los módulos analógicos, según la decodificación adoptada son:

7000-700F	escribe en tarjeta 1
701X	habilita tarjeta 1
8000-800F	escribe en tarjeta 2
801X	habilita tarjeta 2

Una aclaración importante es que la salida de los datos de las memorias va negada respecto a sus respectivas entradas, por lo que es necesario complementar los datos antes de escribirlos en dichas memorias y así obtener el voltaje deseado a la salida.

TARJETA DIGITAL

La tarjeta digital tiene una estructura como la que se muestra en la Fig. 4.5. Se compone de 8 memorias de ocho bits cada una, ocho grupos de amplificadores de corriente a colector abierto y un circuito direccionador para cada una de las memorias. Cada bit de las memorias representa un canal digital.

Las líneas de acoplamiento con la microcomputadora son las ocho de datos, tres de direcciones, la de restablecimiento y una de selección (ST₅ ó ST₆, según el módulo).

La operación es muy sencilla pues consiste en direccionar cada una de las ocho memorias (por medio de A₀-A₂) y escribir sobre ellas un byte. La selección de las memorias se consigue a través de un decodificador de 3 a 8 (74LS138) y la línea de selección de tarjeta. La señal de ~RES produce como efecto que todas las salidas queden en nivel lógico bajo.

El colector abierto en las salidas tiene por objeto proteger al

dispositivo y permitir mayor longitud en el cableado hasta los actuadores digitales.

En total cada acoplamiento puede manejar hasta 64 actuadores digitales de un bit cada uno. En el caso de que un actuador requiera mas de un bit se requiere combinar el número de salidas que sea necesario.

El programa la direcciona desde la B000-B008 ó C000-C008 para la tarjeta digital 1 y digital 2 respectivamente.

TARJETA DE PRUEBAS

La idea de este dispositivo es dotar al controlador de actuadores con un medio para realizar fácilmente pruebas del funcionamiento de los actuadores analógicos y digitales.

El acoplamiento está compuesto por los elementos que se muestran en la Fig. 4.6. Como puede verse está centrado en un despliegue de cristal líquido de seis dígitos y un teclado de 18 botones. Alrededor de estos se cuenta con circuitos que permiten su manejo y comunicación con la microcomputadora.

Teclado

La operación del teclado se realiza por medio de un ciclo de barrido rápido, controlado por un contador, que polariza periódicamente un borne de cada tecla. Se lleva registro del número de tecla que se está polarizando. Mientras no se oprima ninguna tecla el contador continuará realizando el ciclo de barrido.

Cada vez que se oprime una tecla del grupo principal de dieciseis, se interrumpe la cuenta del contador y su valor se transfiere a un registro de recepción. La interrupción de la cuenta se realiza inhibiendo su señal de cuenta, a partir del voltaje de polarización mencionado. El valor del contador está directamente relacionado con la tecla oprimida.

Además de registrar el valor numérico asignado a la tecla del grupo principal, se registra el valor de las otras dos teclas (Shift y Exc),

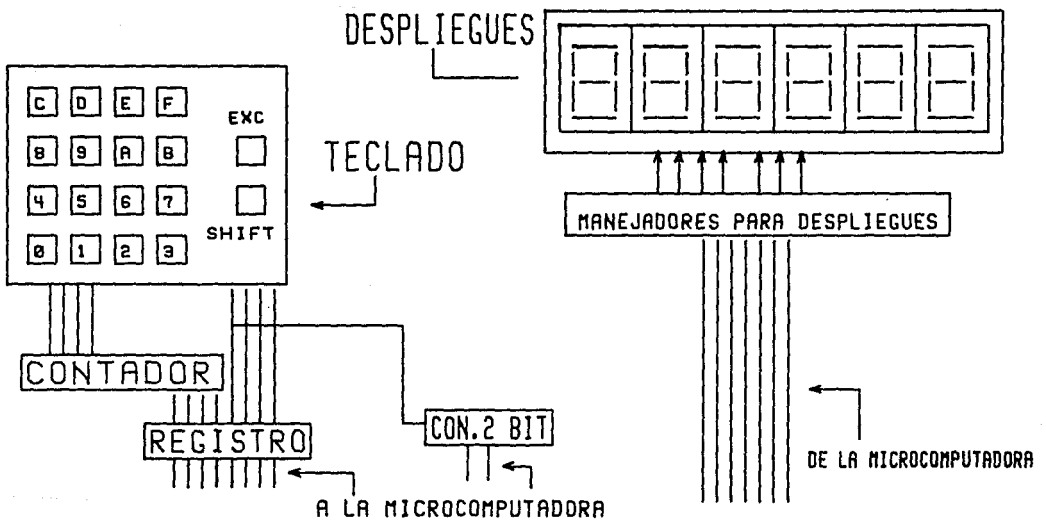


FIG. 4.6 COMPONENTES DE LA TARJETA DE PRUEBAS

cuya acción es independiente de las anteriores y tiene por objeto producir alguna acción específica relacionada con el valor del despliegue mencionado.

Las dieciseis teclas y Exc podrán generar una señal de interrupción al procesador, según se describe a continuación. Al oprimir una tecla se produce un pulso que opera un biestable de tipo D (74LS74) y pone su salida en nivel bajo. La salida del biestable se conecta a la línea de interrupción del microprocesador a través de un alimentador. Este último puede filtrar la salida del biestable, según el valor de una línea de control.

El manejo de esta línea de control y la restitución del valor original del voltaje en la salida del biestable se consiguen a través de un registro de escritura de dos bits. El valor del primero permite restituir la línea de interrupción y del segundo determina si se filtra la señal de interrupción en el alimentador.

Existe un contador de dos bits, cuyo valor también se encuentra en el registro de recepción que asociado al valor de las teclas oprimidas permite diferenciar si se ha dejado de leer el teclado o si existen repeticiones intencionales de alguna tecla.

El valor de las teclas oprimidas se maneja a través de un registro de lectura de ocho bits, que se interpretan como sigue:

b ₀ -b ₃	código de la tecla oprimida.
b ₄	detector de tecla Exc (activo alto)
b ₅	" " " Shift " "
b ₆ -b ₇	contadores de teclas presionadas

Para tomar la información de este registro se realiza una operación de lectura.

Despliegue

Los despliegues de cristal líquido se manejan a través de circuitos especializados para el efecto, a los que se debe de presentar únicamente el valor hexadecimal del dígito que se desea desplegar y su posición dentro del conjunto de los seis despliegues. Esta presentación se logra

por medio de dos memorias, de ocho bits cada una, que comunican el canal de datos con estos circuitos especializados, el paso de información se consigue con operaciones de escritura.

La microcomputadora se interconecta con esta tarjeta a través de las líneas de datos, una de selección, tres de direcciones A_0-A_2 , la de $\sim RES$ y la de interrupción $\sim IRQ$.

La línea de $\sim RES$ borra las memorias que se comunican a los circuitos que manejan el despliegue.

La línea de selección y las de direcciones se conectan a un decodificador, cuyas salidas son las que activan los circuitos manejados por el programa. Las direcciones que corresponden a la tarjeta de prueba y su efecto sobre la misma se resumen en la siguiente tabla.

Localidad	Comando
9000	Manda el código a desplegar del canal de datos al circuito manejador del despliegue.
9001	Manda la posición del dígito a desplegar del canal de datos al manejador del despliegue.
9002	Pasa el código de la tecla oprimida al canal de datos.
9003	Limpia contadores de presiones de teclas.
9004	Habilita interrupción por presión de teclas.
9005	No genera interrupción por presión de teclas.
9006	Limpia solicitud de interrupción.

MAPA DE MEMORIA

A continuación se muestra el mapa de memoria general para todo el dispositivo:

Localidad	Elemento
0000-07FF	RAM
4XX0-4XX1	ACIA
5XX0-5XX7	PTM
6XX0-6XX3	PIA
7XX0-7XXF	Selector de tarjeta analógica1
8XX0-8XXF	" " " analógica2
9XX0-9XX	" " " de pruebas
BXX0-BXX7	" " " digital1
CXX0-CXX7	" " " digital2
E000-EFFF	ROM1, utilizada para guardar variables tabulares
F000-FFFF	ROM2, alojamiento del programa y apuntadores

5. COMUNICACION CON EL PUESTO CENTRAL

Para comunicarse con el órgano de toma de decisiones, se fijó de antemano un protocolo general de comunicación, el cual se debe aplicar en todos los mensajes que transiten entre el puesto central y el controlador de actuadores.

Este capítulo contiene una descripción general de dicho protocolo y su particularización al caso del controlador de actuadores.

PROTOCOLO GENERAL

La información entre el puesto central y el controlador de controladores se realiza a través de un enlace serie que funciona bajo la norma RS232 (Ref. 10). Se considera que la unidad mínima de intercambio de información serán caracteres en formato ASCII (American Standar Code for Information Interchange).

Los mensajes se compondrán de conjuntos o bloques de caracteres que deberán responder a la estructura que se muestra en la Fig. 5.1. Cada

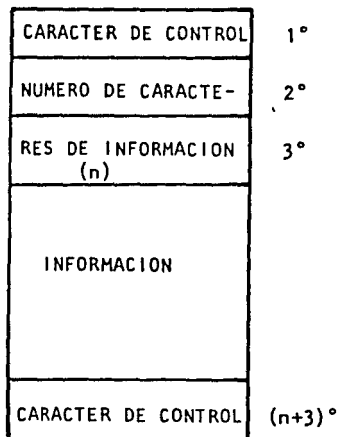
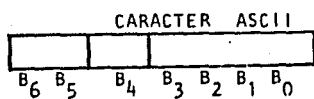


Fig 5.1 Estructura a bloques de los mensajes



B₅ a B₆-identificación del receptor
 B₄ - caracter de control o dato
 B₀ a B₃-caracter hexadecimal

Fig. 5.2 Interpretación de cada caracter del bloque.

carácter de un bloque se interpretará como se muestra en la Fig. 5.2.

Los caracteres de inicio y fin del bloque se consideran caracteres de control. El primero sirve para indicar, de manera general, el contenido del bloque; el otro indica al receptor que el bloque ha concluido. Los demás caracteres del bloque se consideran de datos.

Después del carácter de inicio de bloque seguirán obligatoriamente otros dos que especificarán el número de caracteres de datos que contiene el bloque. Como cada carácter contiene sólo cuatro bits de información útil, según el formato mostrado en la Fig. 5.2, es posible transmitir bloques de entre 0 y 255 caracteres de datos.

El primero y segundo carácter especifican los dígitos hexadecimales menos y más significativos de la cantidad de datos, respectivamente.

Según lo anterior los bloques se componen de entre 4 y 259 caracteres en total. Como regla general se especificó que cuando un mensaje no cumpla con este formato, la información que contenga será ignorada.

LISTA DE CODIGOS DE COMUNICACION

La aplicación particular del protocolo arriba descrito para el controlador de actuadores implica especificar la interpretación de los bits de destino de cada carácter y el significado de todos los posibles caracteres de control, que guardan relación con el control de la planta solar experimental para la que se desarrolló originalmente el dispositivo.

Los bits de destino se establecieron de la siguiente forma:

B ₅	B ₄	Destino
1	0	Controlador de actuadores
1	1	Puesto central

La selección se realizó de tal forma que los caracteres que se intercambien se puedan desplegar directamente en una terminal de video.

La interpretación de los caracteres de control obedece a la siguiente tabla:

CARACTER	ACCION POR REALIZAR	RECEPTOR	TRANSMISOR
54	Información bloque digital	CA	PC
55	" bloque analógico	"	"
56	" válvulas binarias	"	"
57	" válvulas analógicas	"	"
58	" estado espejos	"	"
59	" estado bombas	"	"
5A	" estado dispositivos auxiliares	"	"
5B	" candados y parámetros	"	"
5C	Petición de candados y parámetros	"	"
5D	Petición de estado de válvulas digitales	"	"
5E	Petición de estado de válvulas analógicas	"	"
5F	Fin de transmisión-recepción	"	"
76	Recepción de Estado de válvulas binarias	PC	CA
77	Recepción de Estado de válvulas regulables	"	"
7C	Recepción de candados y parámetros	"	"
7F	Fin de recepción-transmisión	"	"

PC - puesto central; CA - controlador de actuadores

En el caso de las órdenes de petición (5C a 5E), el bloque tendrá solo los cuatro bytes obligatorios.

Adicionalmente, con excepción de los caracteres de fin de bloque (5F y 7F), para el resto de los caracteres de control la longitud de los mensajes queda establecida de manera fija, según la siguiente tabla:

CARACTER	LONGITUD PREFIJADA	
	Decimal	Hexadecimal
54	33	21
55	65	41
56 y 58	08	08
57	06	06
59	01	01
5A	02	02
5B	34	22
76	08	08
77	06	06
7C	34	22

6. CONCLUSIONES

Se ha descrito un dispositivo que permite simplificar el manejo de los actuadores de un proceso. El equipo presenta características que lo hacen ideal para emplearse en procesos que se controlen mediante dispositivos digitales.

Sus mayores ventajas estriban en su flexibilidad para usarse en procesos de muy distinta magnitud, dada su modularidad. La estrategia con que se conecta al proceso a través de un sistema jerarquizado constituye una ventaja para la supervisión y mantenimiento del sistema de control y coincide, además, con la tendencia actual en la implantación de este tipo de sistema.

El sistema se ha empleado con buen éxito en la planta solar experimental del Instituto de Ingeniería.

7. REFERENCIAS

1. L. Alvarez-Icaza. Controlador para tableros mímicos. Memorias del IX Congreso de la Academia Nacional de Ingeniería. Cd. Obregón, Son. Septiembre, 1984. pp. 223-228.
2. L. Alvarez-Icaza, G. Fernández y R. Straulino. Sistema operativo para el control de procesos. Memorias del X Congreso de la Academia Nacional de Ingeniería. San Luis Potosí, SLP. Septiembre, 1985. pp. 262-266.
3. L. Alvarez-Icaza y A. Noriega. Adquisidor de datos para el control de procesos. Memorias del X Congreso de la Academia Nacional de Ingeniería. San Luis Potosí, SLP. Septiembre, 1985. pp. 182-186.
4. Apple Computer Inc. APPLE PASCAL, Operating System Manual. Apple Computer, Inc. Cupertino, USA. 1981.
5. Apple Computer Inc. APPLE PASCAL, Language Reference Manual. Apple Computer, Inc. Cupertino, USA. 1981.

6. MOS Technology, Inc. MCS6500 Microcomputer Family Hardware Manual. MOS Technology, Inc. Norriston, USA. 1976 pp.1-151.
7. Hitachi. The MOS Memory Data Book. Hitachi. San Jose, USA, 1982, pp. 1-6.
8. Texas Instruments Inc. The MOS Memory Data Book. Texas Inst. Inc. Houston, USA. 1979. pp. 117-124.
9. Motorola Inc. The Complete Microcomputer Data Library. Technical Information Center, Motorola Inc. USA. 1978, pp. 1-97 a 1-100, 1-107 a 1-118, 1-211 a 1-218.
10. H. Folts and H. Karp. Compilation of Data Communications Standards. Mc Graw-Hill's. NY, USA. 1978. pp. 699-726.
11. Motorola Inc. The CMOS Data Book. Technical Information Center, Motorola Inc. US. 1978. pp. 7.221-7.223.
12. P. Kawakami and R. Mc Carthy. Signetics Logic-TTL Data Manual. Uniplan. San Francisco, USA. 1978. pp. 1-759.

ANEXO A

DESCRIPCION ENTRADA-PROCESO-SALIDA PARA
LAS SUBTAREAS QUE EJECUTA EL CONTROLADOR
PARA ACTUADORES

LISTA DE LAS SUBTAREAS

1. INICIAR1
2. INICIA2
3. PROGPER -programar periféricos-
4. VALIDEZ
5. MULTIP -multiplicar-
6. RELOJ2
7. RELOJ1
8. CALECA -cálculo ecuación analógica-
9. CALECA2
10. CALECD -cálculo ecuación digital-
11. ESANALOG -estado analógico-
12. ECDIG1 -ecuación digital, 1a. parte-
13. ECDIG2 - " " , 2a. parte-
14. SALTARAN -salida a tarjetas analógicas-
15. SALTARD -salida a tarjetas digitales-
16. GUARDAB4 -guarda en buffer4-
17. APUNDES -apuntador destino-
18. PONBUFF4 -pone en buffer4-
19. DESCOMP -descompacta-
20. ACONDIC -acondiciona-
21. ABRETT -análisis del bloque recibido, encolar tareas de transmisión-
22. GUARDAR
23. CASOS
24. COMPACT -compacta-

- 25. DESECHAR
- 26. CASOS3
- 27. TARPRUE1 -tarjeta de pruebas-
- 28. PONMUMS -pone números-
- 29. PONCAR -pone carácter-
- 30. CAMBIO
- 31. POSD -posiciona digital-
- 32. VALBIT -válido bit-
- 33. SINACPR1 -sin acoplamiento de pruebas-
- 34. SINACPR2
- 35. SALIDAA -salida analógica-
- 36. SALIDAD -salida digital-
- 37. RECDTBM -recepción y detección de bloques malos-
- 38. RALMACEN -recepción almacen-
- 39. COMPACTR -compactar para recepción-
- 40. APCAMINO -apuntador camino-
- 41. PROCESAT -procesa tecla-
- 42. PREPCOMP -prepara compactación-
- 43. LEEPIAA

DESCRIPCION DE ENTRADA-PROCESO-SALIDA DE LAS SUBTAREAS

1. INICIARI

Entrada.- PRINBLOQ, DATOCOMP, NUMDATO1, NUMDATO2, ALMCONT, CONT2, MALBLOQ, RECHAZO, NUMDATO.

Proceso.- Asignarles el valor de cero a todas las variables.

Salida.- Las mismas de la entrada.

2. INICIA2

Entrada.- ROMCANAN, ROMCAND, EDOCANAN, EDOCANDD (son variables ROM).

Proceso.- Definir los candados y estados en RAM a partir de los definidos en ROM.

Salida.- ACANDADO, DCANDADO, EANALOG, EDODIG (variables RAM).

3. PROGPER. Usa las subrutinas: Validez, Multip, Reloj2, Reloj1.

Entrada.- PUERTOA, VALIDO, acumulador (de Validez, Multip).

Proceso.- Recibe la información del puerto de entrada A (PIA), donde se le definió el modo de operación, la configuración de tarjetas analógicas y digitales. Analiza si no se excede de 2 tarjetas de cada tipo. Si se excede no se modifica la bandera VALIDO, si no, si la modifica. Según el modo definido configura periféricos y variables.

Salida.- NATARAN, NTARDIG, MUANDO, MUADOR, REGANA, REGDIG, VALIDO, MODO, CR, PTM+1, MODO2, STATUSP, NOINTEC, RECALCUL, VACIADO, CRB, DDRB.

4. VALIDEZ

Entrada.- NTARDIG, NTARAN.

Proceso.- Compara que el número de tarjetas digitales y analógicas sea menor o igual que dos. A través del acumulador se comunica el resultado.

Salida.- Acumulador.

5. MULTIP

Entrada.- MUANDO, MUADOR.

Proceso.- Multiplica dos bytes uno es el multiplicador y otro el multiplicando, el resultado como máximo es de dos bytes (TOTAL1, TOTAL1+1). En este caso sólo interesa el menos significativo.

Salida.- TOTAL1.

6. RELOJ2

Entrada.- #OC1(parámetro que configura al contador2), #5DC0(valor inicial del contador).

Proceso.- Se configura al reloj2 para que trabaje con las siguientes características:

- que se pueda escribir sobre el contador1.
- usar reloj externo, contador de 16 bits, modo continuo.
- el arranque del contador es cuando ocurre:~G +W+R.
~G =transición negativa de la entrada ~Gate.
W=escritura sobre los contadores.
R=reset del reloj (CR1₀=1 o ~reset externo=0).
- bandera habilitada para encenderse cuando ocurre una ~IRQ.
- salida habilitada.

Salida.- PTM, PTM+4, PTM+5.

7. RELOJ1.

Entrada.- #OE0 (parámetro que configura al contador1), #OFFF(valor inicial del contador).

Proceso.- Se configura al contador1 para que trabaje con las siguientes características:

- T3 ~clock no preescalado.
- uso de reloj externo, contador de 16 bits, disparo simple (necesita configurarse para que vuelva a disparar).
- el arranque del contador es:~G +W+R.
- bandera de interrupción habilitada al ocurrir una ~IRQ.
- salida habilitada.

Salida.- PTM, PTM+2, PTM+3.

8.CALECA. Usa la subrutina: Esanalog

Entrada.- #08, #10 y a través del acumulador:ACANDADO, ACANDADO+1.

Proceso.- Calcula los 16 primeros canales analógicos. Se calculan por grupos de ocho, por lo que se invoca dos veces a la rutina Esanalog.

Salida.- LIMANA, LIMANA2.

9.CALECA2. Usa la subrutina: Esanalog

Entrada.- #18,#20, y a través del acumulador:ACANDADO+2, ACANDADO+3.

Proceso.- Se calculan los siguientes 16 canales por grupos de ocho, se invoca dos veces la rutina Esanalog.

Salida.- LIMANA, LIMANA2.

10.CALECD. Usa las subrutinas: Ecdig1, Ecdig2.

Entrada.- Regdig, el resultado de Ecdig por el acumulador, tablas de: IVALDIG, IESPEJOS, IBOMDISA.

Proceso.- Toma el resultado proporcionado por Ecdig1, lo procesa y el resultado lo pasa por el acumulador a Ecdig2.

Salida.- Ecdig2 por el acumulador.

11.ESANALOG

Entrada.- Acumulador, LIMANA, LIMANA2, tablas de datos: AINFORM y EDOCANAN.

Proceso.- Cada bit de temp2 define un candado analógico. Se va preguntando bit por bit, si vale 1 iguala a AINFORM con ENALOG, si vale cero entonces lo hará con EDOCANAN y EANALOG.

Salida.- EANALOG.

Trabajo.- TEMP2.

12.ECDIG1

Entrada.- Tablas de DCANDADO, EDOCANDD, DCANDADO.

Proceso.- Se realiza la operación AND entre DCANDADO y EDOCANDD y EOR entre DCANDADO y #OFF.

Salida.- TEMP2 y acumulador.

13.ECDIG2

Entrada.- Acumulador, TEMP2.

Proceso.- Ocurre una suma entre el acumulador y TEMP2, almacenándose el resultado en EDODIG.

Salida.- Tablas de EDODIG.

14.SALTARAN

Entrada.- VACIADO, REGANA, Tablas de EANALOG.

Proceso.- Verifica que esté encendida la bandera de vaciado y que haya registros analógicos. La salida la hace por grupos de 16 (son dos grupos si la configuración es de dos tarjetas analógicas). Primero se habilita una tarjeta, se toman las variables ENALOG, se niegan y se mandan a la tarjeta analógica. Se hace lo mismo para el siguiente grupo.

Salida.- ABTARAN1, ABTARAN2, variables tabulares: TANALOG1, TANALOG2.

15.SALTARD

Entrada.- VACIADO, REGDIG, tablas de EDODIG, MASCARAD

Proceso.- Verifica que este encendida la bandera de vaciado y que haya registros digitales. Se toma EDODIG, se efectúa EOR con MASCARAD con el objeto de negar algunos bits a la salida de EDODIG. Estos bits los define MASCARAD. El resultado sale por TDIG1 y TDIG2.

Salida.- TDIG1 y TDIG2.

16.GUARDAB4. Invoca a Apuntdes, Ponbuff4, Acondic.

Entrada.- Acumulador, los valores:#0C, #0E, #06, #08, #0A para Apuntdes.

Proceso.- Se compone todo el bloque antes de transmitirlo; se forman primero los tres primeros bytes. Con el índice X y la subrutina Apuntdes se especifica de donde va a ser tomada la información, la variable LIM indica el volumen de esta. Con estas condiciones previas se invoca a Ponbuff4 para formar el resto del bloque.

Se podrán mandar tres tipos de bloques: válvulas binarias, válvulas regulables (3), y candados y parámetros.

Para el caso de las válvulas digitales, después de formados los 3 primeros bits, se invoca a Acondic, y se continúa con el proceso arriba descrito.

Para los candados y parámetros, por cada tipo de candado a mandar, se realiza lo descrito en el primer párrafo, hasta formar todo el bloque. Se usa INDX2 como registro de trabajo.

Al final se indica de cuantos caracteres es el bloque en la variable COTSB4 y el valor 1 a INDX2.

Salida.- LIM, COTSB4, INDX2, BUFFER4, BUFFER4+1, BUFFER4+2.

17.APUNTTDES

Entrada.- Índice X, APUNT que es una tabla con la ubicación de: BUFFER3, AINFORM, IESPEJOS, PARRELOJ, ACANDADO, DCANDADO, EANALOG, EDODIGT, IVALDIG.

Proceso.- según el valor del índice X apunt tendrá el domicilio, de una variable de entrada, este domicilio lo almacena en INDIR e INDIR+1.

Salida.- INDIR, INDIR+1.

18.PONBUFF4. Usa a Descomp.

Entrada.- TNIBBLE1, TNIBBLE2, LIM y variables de tablas definidas por °INDIR.

Proceso.- Toma cada variable tabular y a través del acumulador la pasa a Descomp, la cual entrega dos bytes listos para ser transmitidos. Estos son almacenados a partir de la primera posición de BUFFER4+3, para generar una tabla. El final de la tabla lo determina LIM, cuando se llega a su valor, se forma el último caracter que es #7F.

Salida.- Acumulador (para Descomp), una tabla a partir de BUFFER4+3.

19.DESCOMP

Entrada.- Acumulador.

Proceso.- El byte recibido a través del acumulador, se debe convertir a

dos bytes, donde el nibble más significativo de cada uno debe valer 4 y el menos significativo es un nibble del byte recibido. TNIBBLE1 contendrá la mitad menos significativa y TNIBBLE2 la más significativa del byte recibido.

Salida.- TNIBBLE1, TNIBBLE2.

20. ACONDIC

Entrada.- EDODIG, EDODIG+1, EDODIG+2.

Proceso.- Se tomarán ocho bits de EDODIG, EDODIG+1 y los 4 bits menos significativos de EDODIG+2 y se formarán cuatro registros EDODIGT. Se igualarán algunos bits de esta tabla con los de EDODIG y sucesivos. Los bits no igualados valdrán uno. Se igualan de la siguiente forma:

EDODIGT : $E_7 = EDODIG_3$, $E_6 = EDODIG_2$
 $E_1 = EDODIG_1$, $E_0 = EDODIG_0$

EDODIGT+1: $E_7 = EDODIG_7$, $E_4 = EDODIG_6$
 $E_3 = EDODIG_5$, $E_2 = EDODIG_4$

EDODIGT+2: $E_7 = EDODIG+1_5$, $E_6 = EDODIG+1_4$
 $E_5 = EDODIG+1_3$, $E_4 = EDODIG+1_2$
 $E_1 = EDODIG+1_1$, $E_0 = EDODIG+1_0$

EDODIGT+3: $E_7 = EDODIG+2_4$, $E_6 = EDODIG+2_3$
 $E_3 = EDODIG+2_1$, $E_2 = EDODIG+2_0$
 $E_1 = EDODIG+1_7$, $E_0 = EDODIG+1_0$

Salida.- Tabla EDODIGT

21. ABRETT

Entrada.- ALMACEN, BUFFER1+44, BUFFER2+44, APCOLA, ENCOLA (puede ser modificada).

Proceso.- La bandera ALMACEN indica de que buffer tomará la información.

Identifica si recibió un byte de petición o de información. Si es de petición procederá a colocarlo en una cola, por medio de un apuntador APCOLA que señala a una serie de registros en que se almacenan estos bytes. La petición puede ser de candados y parámetros, estado de válvulas digitales, y estado de válvulas analógicas.

La cola está formada por diez elementos, si llega uno más se coloca en último lugar, después de almacenado un elemento de petición el apuntador se incrementa en uno. En la rutina de transmisión este apuntador se decrementa después de transmitir todo el bloque.

Al tratarse de información, según el tipo, compara el número de datos del bloque recibido, con el número correcto. Si son iguales considera el bloque válido.

Salida.- Acumulador, ENCOLA.

22.GUARDAR. Invoca a Apuntdes, Casos, Desechar, Casos3, Compact.

Entrada.- ALMACEN, REGDIG, REGANA, APUNT2 es el apuntador que tiene acceso a BUFFER1 y BUFFER2.

Proceso.- La bandera de ALMACEN le indica de que buffer tomará los datos. Esta rutina señala a través de Apuntdes en donde se guardarán los datos; especifica por el índice X el límite superior del bloque de información recibido. Una vez hecho esto llama a Casos.

Para guardar candados y parámetros se indica, según su tipo, el lugar del bloque a partir de la cual se tomarán los datos. Con el registro INDY2 se apunta al inicio del sitio en donde se guardarán los datos. Se usa para ello la segunda parte de Casos, que es Casos2.

Para las válvulas binarias se tomarán 20 bits de los 32, esto se hace con la subrutina Desechar.

Al tratarse de los dispositivos auxiliares, se considerarán los dos bits menos significativos de IBOMDISA.

Para los dispositivos auxiliares serían los 6 bits más significativos de IBOMDISA.

Los bloques analógicos y digital se trabajan íntegramente con Casos3.

Salida.- Registro X para Apuntdes y Casos, registro Y para Casos2; IBOMDISA, TEMP5, INDYD, NIBBLE1, NIBBLE2, INDIR2, INDIR2+1, INDIR2B, INDIR2B+1.

23. CASOS

Entrada.- Registro X para APLIM, INDYD para Casos2(segunda parte de Casos), APLIM, BUFFER1+3 ó BUFFER2+3 a través de °INDIR2B.

Proceso.- Se le indica el buffer de donde debe tomar la información, toma los dos bytes y los compacta mediante Compact; el resultado se almacena en °INDIR. Se finaliza el proceso comparando el índice Y del buffer con una cota (APLIM)

Salida.- NIBBLE1, NIBBLE2, INDYO, INDYD, °INDIR.

24. COMPACT

Entrada.- NIBBLE1, NIBBLE2

Proceso.- Borra la mitad más significativa de las variables de entrada y con lo que resta forma un byte cuya parte menos significativa procede de NIBBLE1 y la más significativa de NIBBLE2. El resultado se almacena en el acumulador.

Salida.- Acumulador.

25. DESECHAR. Usa a Apuntdes.

Entrada.- La tabla de BUFFER3 por medio de °INDIR.

Proceso.- De 32 bits que contiene BUFFER3 (B3), se desechan 12 bits, los bits restantes se almacenarán en la tabla IVALDIG (IVD). Los bits de IVALDIG se igualan con los BUFFER3 de la siguiente forma:

$$\begin{array}{l} \text{IVALDIG} : \text{IVD}_7 = \text{B3} + 1_5, \quad \text{IVD}_6 = \text{B3} + 1_4 \\ \text{IVD}_5 = \text{B3} + 1_3, \quad \text{IVD}_4 = \text{B3} + 1_2 \\ \text{IVD}_3 = \text{B3}_7, \quad \text{IVD}_2 = \text{B3}_6 \\ \text{IVD}_1 = \text{B3}_1, \quad \text{IVD}_0 = \text{B3}_0 \end{array}$$

$$\begin{array}{l} \text{IVALDIG} + 1 : \text{IVD}_7 = \text{B3} + 3, \quad \text{IVD}_6 = \text{B3} + 3_0 \\ \text{IVD}_5 = \text{B3} + 2_7, \quad \text{IVD}_4 = \text{B3} + 2_6 \\ \text{IVD}_3 = \text{B3} + 2_5, \quad \text{IVD}_2 = \text{B3} + 2_4 \\ \text{IVD}_1 = \text{B3} + 2_1, \quad \text{IVD}_0 = \text{B3} + 2_0 \end{array}$$

IVALDIG+2:IVD₇=IVD₆=IVD₅=IVD₄=0
 IVD₃=B3+3₇, IVD₂=B3+3₆
 IVD₁=B3+3₃, IVD₀=B3+3₂

Salida.- Tabla IVALDIG.

26.CASOS3. Invoca a Compact.

Entrada.- INDYD(para la segunda parte de casos3), BUFFER1+3 ó BUFFER2+3 a través de °INDIR2B, NUMDATO3.

Proceso.- Igual que Casos, sólo que el fin de ejecución está dado por NUMDATO3

Salida.- NIBBLE1, NIBBLE2, INDYO, INDYD, °INDIR.

27.TARPRUE1. Emplea a: Ponnuns, Cambio, Saltaran, Salidad, Posd.

Entrada.- Acumulador, CAMBACPR, AVANCE, ENTDAT1, CAMBNIV, NUCAN, VACANA

Proceso.- Esta rutina se usa en el modo de pruebas y es la que coordina todo el modo de operación con el acoplamiento de pruebas.

Inicialmente pone en ceros el despliegue, usando Ponnuns. Recibe valores entregados vía atención de interrupciones. Cuando no atiende interrupciones, queda en un lazo en el que invoca a CAMBIO. Esta toma información de la parte A y B del puerto paralelo PIA y con ello revisa si no hubo un cambio en la configuración, ya sea para cambiar de modo, para pasar al mismo modo de pruebas pero sin emplear el acoplamiento de pruebas ó para retroceder de nivel de operación.

Puede probar canales analógicos y digitales, en ambos casos se trabaja en tres niveles.

Primero se ubica en el nivel de comandos, y espera a que se le especifique si serán pruebas analógicas o digitales.

En las pruebas analógicas, primero prepara a la bandera CAMINO para indicar como procesar la tecla oprimida en la atención de interrupciones y COTSUP que indica el número de canal máximo permitido. Espera la presión de una tecla, si fuera Shift E, irá al nivel de comandos, si fué una tecla numérica saldrá en la pantalla (consultar en atención de interrupciones) y espera la presión de otra tecla, conservándose en ese nivel todavía. El número de canal se especifica vía dos dígitos, si es un

número permitido al oprimir Exc avanzará al siguiente nivel; si no lo fué aunque se oprima Exc se conservará en ese nivel.

El último nivel es el de del canal, en el se prepara CAMINO, para procesar la tecla oprimida en atención de interrupciones y se espera a que se oprima una tecla. Si fue Shift E, retrocede al nivel de especificación del número de canal; en el caso de una presión de tecla numérica saldrá en la pantalla (ver atención de interrupciones). El valor del canal se termina de especificar a la siguiente presión de tecla numérica.

Sólo saldrá a la tarjeta el último valor definido antes de la presión de Exc con que se toma el número de canal definido, el valor de este, y se deposita en la variable E₁ correspondiente y se invoca a SALTARAN para que salga a la tarjeta. Concluyendo esto, comienza de nuevo en este nivel.

Pruebas digitales. En este nivel de número de canal digital ocurre lo mismo que en el nivel analógico. En el nivel de valor de canal digital, se prepara CAMINO para atender interrupción espera presión de tecla, si es Shift E, ira al segundo nivel, si es de tecla numérica sólo acepta el valor 1 o 0 (ver interrupciones), si es Exc se invoca a Posd que modificará el bit del registro correspondiente de la tabla E, según el número de canal dado y valor definido.

Modificado un bit de E, se invoca a SALTARD para su salida a tarjetas, después se inicia un nuevo ciclo en este nivel.

Salida.- CAMINO, COTSUP, NUCANA, VACIADO, DIG1, DIG2 y un elemento de EANALOG.

28.PONNUMS. Usa a Poncar.

Entrada.- No hay.

Proceso.- Pone un mismo número en toda la pantalla. Inicialmente se le da el valor de 1 a SELDIG1, y se invoca a Poncar, se recorre un lugar a la izquierda los bits de SELDIG1 y se vuelve a llamar a Poncar, al tener SELDIG1 el valor de 40 el proceso termina.

Salida.- SELDIG1.

29.PONCAR

Entrada.- ENTDAT1, SELDIG1.

Proceso.- Pondrá un caracter en pantalla a partir de las variables de entrada, cuyos valores definen el lugar (SELDIG) que va a ocupar en la pantalla y el valor que se dará al despliegue este despliegue (ENTDAT). Primero se borra SELDIG, después se le da un valor y posteriormente se borra otra vez.

Salida.- ENTDAT, SELDIG.

30.CAMBIO. Invoca a Sinacprl.

Entrada.- PDRA, DETCAMMO, PDRB

Proceso.- Al comparar la información dada por el puerto A con DETCAMMO, se sabe si se pidió cambio en el modo de operación. Si sucedió tal, se le da al acumulador el valor de 1 y concluye la rutina.

Si no lo hubo, toma la información del puerto B y consulta si se pidió cambiar a trabajar sin acoplamiento de pruebas, en caso afirmativo, invoca a la rutina Sinacprl, y da el valor de 1 a CAMBACPR, en caso contrario le da el valor de cero. Al no haber petición en el cambio de modo de operación, se le asigna con valor 0 al acumulador.

Salida.- Acumulador, CAMBACPR.

31.POSD

Entrada.- NUCANDIG, Tabla de EDODIG, acumulador.

Proceso.- A partir del número de canal, la rutina define que registro y bit que se va a modificar. Con esta información se toma el registro a modificar, se invoca a Valbit y el registro ya modificado se almacena en uno de los registros de EDODIG.

Salida.- Tabla de EDODIG modificada, RESIDUO, COCIENTE, MODVALD.

32. VALBIT

Entrada.- COCIENTE, RESIDUO, VACAND, MODVALD.

Proceso.- Esta rutina enciende o apaga un bit de MODVALD, la posición del bit está indicada por RESIDUO y su valor por VACAND, el resultado queda en el acumulador.

Salida.- Acumulador, RESIDUO1, SUSTRA, RESULT.

33. SINACPR1

Entrada.- PUERTO B.

Proceso.- Pone en cero las variables de salida y el valor del PUERTO B lo almacena en REFERENC.

Salida.- REFERENC, BANDERA5, COCIENTE, RESIDUO, INDX3.

34. SINACPR2. Llama a: Salidaa, Salidad, Valbit.

Entrada.- INDX3, REFERENC, PUERTO B, PDRA, REGANA, DETCAMMO, RESIDUO, COCIENTE, REGDIG, REGANA, BANDERA5.

Proceso.- Esta rutina usa los tres bits menos significativos del puerto B. El bit 2 define directamente si las pruebas serán analógicas o digitales. El resto de la rutina trabaja con base en detectar cambios de valores del bit 0 y del bit 1 del PUERTO B.

Con el bit 0 se define el cambio de valor en el canal y con el bit 1 el cambio al siguiente canal. Se sigue una secuencia tanto en la salida de valores como en el recorrido de los canales. Al terminar la secuencia de valores, se regresa al primer valor, y al terminar toda la secuencia de recorrido de canales, regresa al primero. En cualquier momento puede pasar de variables analógicas a digitales, pues el puerto B se revisa continuamente.

Para el caso analógico la secuencia de aparición de los valores de salida serán: #00, #7F y #OFF. Y en los canales digitales se define por el valor del bit 0 del puerto B.

Al finalizar cada ciclo, el programa pregunta a través de PDRA si hubo cambio en el modo de operación, si lo hubo, se sale de esta rutina. Si no lo hubo, pregunta si sigue probando sin

acoplamiento de pruebas o comienza a probar con el acoplamiento. Si no hay cambios de nivel en el puerto B, no hace nada y sigue encuestando.

Salida.- REFERENC (con el nuevo valor de PUERTOB), PUERTOA, RESIDUO, COCIENTE, BANDERA5 (modificados).

Trabajo.- TEMP4.

35.SALIDAA. Invoca a Saltarán.

Entrada.- Registro X, acumulador.

Proceso.- Se define que elemento de la tabla va a ser modificado, se modifica este elemento y sale a la tarjeta analógica, a través de Saltaran.

Salida.- Un elemento de la tabla EANALOG, VACIADO, INDX3.

36.SALIDAD. Usa a SALTARD.

Entrada.- Registro X.

Proceso.- Se descontamina el registro X, se pone 1 en la bandera de VACIADO antes de invocar a Saltard y al final se da el valor 0 a VACIADO.

Salida.- INDX3, el valor a la entrada del registro X.

37.RECDTBM. Llama a Ralmacén, Compactr, Iniciar1.

Entrada.- ERRORAC, CHARACTER, acumulador.

Proceso.- Las variables de trabajo, ERRORAC son pueden modificar según lo acontecido y darán información para lo que se debe hacer al ocurrir el siguiente evento. La rutina operará bajo las siguientes premisas:

- Si no hay ningún byte recibido, tiene que ser un carácter de inicio de bloque el que active la rutina.
- Si hay N caracteres de control de inicio de bloque, sólo le hace caso al último recibido.
- Si se define mal el número de datos, el programa rechazará

en lo sucesivo los siguientes bytes que le lleguen, sólo aceptará un carácter de inicio de bloque. Se define mal el número cuando se excede el tamaño del buffer o hubo un error en la recepción.

- Si el número de datos definido, es superado por los datos que le van llegando, el programa rechaza los sucesivos.
- Todo carácter que no tenga su nibble de identificación, se rechaza, con todo el bloque que le acompaña.
- Un carácter de control mal definido provoca que se ignoren los siguientes de datos, hasta que llegue otro de control de inicio de bloque.
- Si se recibe otro carácter de control, sea de inicio o fin de bloque, fuera de lugar, se asume como bueno y el bloque en proceso se marca como malo.

Salida.- Acumulador, NIBBLE1R, NIBBLE2R.

Trabajo.- PRINBLOQ, DATOCOMP, NUMDATO1, NUMDATO2, ALMCONT, MALBLOQ, RECHAZO, NUMDATO.

38.RALMACEN

Entrada.- ALMACEN, CHARACTER

Proceso.- Según lo indicado por ALMACEN, con lo recibido en CHARACTER, se formará la tabla BUFFER1 ó BUFFER2.

Salida.- Tabla BUFFER1 ó BUFFER2.

39.COMPACTR

Entrada.- NIBBLE1R, NIBBLE2R.

Proceso.- Esta es igual a la #24, solo cambian los títulos de las variables, se duplicó porque esta rutina se usa al atender una interrupción y la programación no es concurrente.

Salida.- Acumulador.

40.APCAMINO.

Entrada.- Índice X, tabla CAMINOS que permite acceder una de las siguientes direcciones: CAMINO1, CAMINO2, CAMINO3, CAMINO4.

Proceso.- Con el índice X, toma la dirección de la tabla CAMINOS.

Salida.- INDIR3, INDIR3+1.

41.PROCESAT. Invoca a Poncar, Prepcomp.

Entrada.- TECLA1, DIG1

Proceso.- Según la tecla oprimida o combinación de teclas, efectúa ciertas tareas.

Si detectó EXC, compacta los dos dígitos que especifican el # de canal, en uno sólo y prepara al programa para que pregunte si puede ser aceptado este nuevo número.

Si se oprimió Shift E, se sale de esta rutina retrocediendo un nivel anterior en el programa invocante.

Al no ser ninguna de estas opciones, se trata de un carácter a desplegar, que se coloca en el lugar menos significativo. Lo que estaba antes ahí, se recorre al lugar mas significativo, perdiéndose lo anterior.

Salida.- GAMBIV, RAMACAM2, AVANCE, DIG1, DIG2, ENTDAT1.

42.PREPCOMP. Llama a Compactr.

Entrada.- DIG1, DIG2, acumulador (via Compactr).

Proceso.- Se reciben dos bytes (DIG1 y DIG2) que sus valores pasan a las variables de entrada de Compactr.

Salida.- NIBBLE2R, NIBBLE1R.

43.LEEPIA

Entrada.- PDRA.

Proceso.- Toma la información del puerto A, borra bits 7 y 6 y almacena en PUERTOA.

Salida.- PUERTOA.

ANEXO B

LISTADO DEL PROGRAMA DE LA MICROCOMPUTADORA

RUTINAS EMPLEADAS EN EL PROGRAMA PRINCIPAL

Primer nivel	Segundo nivel	Tercer nivel
INICIAR1		
INICIA2		
LEEPIAA		
PROGPER	VALIDEZ MULTIP RELOJ1 RELOJ2	
CALECA	ESANALOG	
CALECA2	ESANALOG	
CALECD	ECDIG1 ECDIG2	
SALTARAN		
SALTARD		
GUARDAB4	APUNDES PONBUFF4 ACONDIC	DESCOMP
ABRETT		
GUARDAR	APUNDES CASOS DESECHAR CASOS3 COMPACT	COMPACT APUNDES COMPACT
TARPRUE1	PONNUMS CAMBIO SINACPR2 SALTARAN SALTARD POSD	PONCAR SINACPR1 VALBIT
SINACPRI		

SINACPR2

SINACPR1
SALIDAA
SALIDAD
VALBITSALTARAN
SALTARD

RUTINAS EMPLEADAS PARA ATENCION DE INTERRUPCIONES

Primer nivel

Segundo nivel

Tercer nivel

RELOJ1

RECDTBM

RALMACEN
COMPACTR
INICIARI

INICIARI

APCAMINO

PONCAR

PROCESAT

PONCAR
PREPCOMP

COMPACTR


```

.ABSOLUTE
:FILE CON
:FECHA DE TERMINACION:30-NOV-87
:*****
:
:
:PROGRAMA CONTROLADOR DE ACTUADORES
:PARA LA PLANTA SOLAR
:PROY 5136
:INSTITUTO DE INGENIERIA
:AUTOR:HECTOR PALMA MORENO
:ASESOR:M. I. LUIS ALVAREZ-ICAZA LONGORIA
:
:
:*****
:ROUTINA DE RESET.
:DEFINICION DE VARIABLES
ALMACEN .EQU 0A ;A QUE BUFFER SE VAN A DEPOSITAR LOS DATOS RECIBIDOS DEL
;ACTA, DE QUE BUFFER SE TOMARAN PARA FUTUROS PAGOS.
GUARDA .EQU ALMACEN+1 ;INDICA SI PUEDE EFECTUAR EL PASO 6
PRIMCAR .EQU GUARDA+1 ;POSIBILITA LA INTERRUPCION POR TRANSMISION
RECALCUL .EQU PRIMCAR+1 ;BANDERA QUE AL ENCENDERSE SE AUTORIZA EL RECALCULO
VACIADO .EQU RECALCUL+1 ;BANDERA QUE AL ENCENDERSE POSIBILITA
;EL VACIADO A LAS TARJETAS
PUERTOA .EQU VACIADO+1 ;CONTENIDO DEL PUERTO A DEL PIA
DETCAMMO .EQU PUERTOA+1 ;EL ULTIMO MODO REGISTRADO.
;ANTES DE RECIBIR EL NUEVO MODO
VALIDO .EQU DETCAMMO+1 ;BANDERA QUE INDICA SI FUE VALIDA LA CONFIGURACION
REFRESCO .EQU VALIDO+1 ;PARA ACTUALIZAR LOS VALORES DE LOS ESTADOS
ENCOLA .EQU REFRESCO+1 ;INDICA SI DEBE O NO GUARDAR
COLA .EQU ENCOLA+1 ;PIA DE DATOS A PARTIR DE LOS CUALES.
;QUE TRANSMITIR
BUFFER4 .EQU COLA+0A ;BLOQUE DE DATOS A MANDAR
MODO .EQU BUFFER4+45 ;CODIGO DEL MODO DE OPERACION
REGDIG .EQU MODO+1 ;= AL # DE TARJETAS DIGITALES POR OCHO.
REGANA .EQU REGDIG+1 ;= AL # DE TARJETAS ANALOGICAS POR OCHO.
APCOLA .EQU REGANA+1 ;A DONDE SE VA A DEPOSITAR LA
;ULTIMA PETICION DE ESTADO RECIBIDA.
PUERTOB .EQU APCOLA+1 ;EL CONTENIDO DEL PUERTO B DEL PIA.
CR .EQU 4000 ;REGISTRO DE CONTROL DEL ACTA
TDR .EQU 4001 ;ACTA REGISTRO DE TRANSMISION
CRA .EQU 6001 ;PIA REGISTRO DE CONTROL PARTE A
DDRA .EQU 6000 ;PIA REGISTRO DE DIRECCION DE DATOS PARTE A
PDRA .EQU 6000 ;PIA REGISTRO DE DATOS PERIFERICOS PARTE A
PDRB .EQU 6002 ;PIA REGISTRO DE DATOS PERIFERICOS PARTE B
NOINTTEC .EQU 9005 ;NO ACEPTA INT. POR TECLA OPRIMIDA

:*****
:VARIABLES DE ATENCION DE INTERRUPCIONES
:SOLO APARECEN LAS NO DEFINIDAS ANTERIORMENTE
SR .EQU 4000 ;REGISTRO DE ESTADO DEL ACTA
RDR .EQU 4001 ;REGISTRO DE RECEPCION DE DATOS DEL ACTA
PTM .EQU 5000 ;REGISTRO DE CONTROL DE LOS RELOJES 1 Y 3 DEL PTM
STATUSP .EQU PUERTOB+1 ;MASCARILLA QUE VALIDA LAS BANDERAS DE INTERRUPCION
;ENCENDIDAS POR RELOJ1,RELOJ2 O SOLO RELOJ1
;EN EL REGISTRO DE ESTATUS DEL PTM
ERRORAC .EQU STATUSP+1 ;FALLA EN LA RECEPCION DEL CARACTER.
CARACTER .EQU ERRORAC+1 ;CONTIENE LO RECIBIDO POR EL ACTA
INDX2 .EQU CARACTER+1 ;LA POSICION DEL APUNTAOR DEL BUFFER DE RECEPCION
COTSB4 .EQU INDX2+1 ;NUMERO DE BYTES DEL BLOQUE A TRANSMITIR

```

```

ENTDAT1 .EGU COTSB4+1 ;LA TECLA A DESPLEGAR
AVANCE .EGU ENTDAT1+1 ;SI DEBE AVANZAR UN NIVEL
CAMBRIV .EGU AVANCE+1 ;BANDERA PARA RETROCEDER DE NIVEL
CAMINO .EGU CAMBRIV+1 ;EL SEGMENTO A EJECUTAR DE LA INTERRUPCION
TECLA1 .EGU CAMINO+1 ;ALMACENA EL CODIGO DE LA TECLA OPRIMIDA
COTSUP .EGU TECLA1+1 ;EL NUMERO MAXIMO-1 DE CANALES POSIBLES
NUCAN .EGU COTSUP+1 ;TIENE EL NUMERO DE CANAL DEFINIDO POR EL OPERADOR
SELDIG1 .EGU NUCAN+1 ;INDICA A PONCAR QUE DIGITO1
;SE ACTIVARA EN LA PANTALLA
NTARAN .EGU SELDIG1+1 ;NUMERO DE TARJETAS ANALOGICAS QUE MANEJARA EL CA
NTARDIG .EGU NTARAN+1 ; " " " " DIGITALES " " " "
RAMACAM2 .EGU NTARDIG+1 ;INDICA SI DETECTA EXC,SHIFT E. TECLA OPRIMIDA
VACANA .EGU RAMACAM2+1 ;VALOR CANAL ANALOGICO A SALIR POR LA TARJETA RESPEC
VACAND .EGU VACANA+1 ; " " " " DIGITAL " " " "
MOD02 .EGU VACAND+1 ;MASCARILLA QUE CONFIGURA EL RELOJ,RELOJ2 O SOLO EL 1
INDIR3 .EGU MOD02 ;TIENE EL DOMICILIO DEL SEGMENTO A EJECUTAR EN LA INTERRUP
TECLA .EGU INDIR3 ;RECIBE EL CODIGO DE LA
;TECLA OPRIMIDA
SOLINT .EGU TECLA ;LIMPIA SOLICITUD DE INTERRUPCION

```

```

;*****

```

```

;SUB #1 INICIAR1
PRINBLOQ .EGU MOD02+1 ;INDICA QUE YA RECIBIO EL PRIMER BYTE DEL BLOQUE
DATOCONP .EGU PRINBLOQ+1 ;RECIBIO EL NUMERO DE DATOS ESPERADO
NUMDATO1 .EGU DATOCONP+1 ;RECIBIO EL PRIMER BYTE DEL NUMERO DE DATOS
NUMDATO2 .EGU NUMDATO1+1 ; " " SEGUNDO " " " "
ALMCONT .EGU NUMDATO2+1 ;CUENTA LOS BYTES QUE ESTA LLEGANDO AL BUFFER
CONT2 .EGU ALMCONT+1 ;REGISTRA SOLO LOS DATOS QUE LE VAN LLEGANDO
;EXCLUYE BYTE DE INICIO,FIN Y LOS # DE DATOS
MALBLOQ .EGU CONT2+1 ;AL MENOS UN BYTE DEL BLOQUE ESTA MAL
RECHAZO .EGU MALBLOQ+1 ;SOLO DEBE ACEPTAR UN BYTE DE INICIO DE BLOQUE
NUMDATO .EGU RECHAZO+1 ;RESULTADO DE LA COMPACTACION DE LOS DOS
;BYTES QUE CONTIENEN EL NUMERO DE DATOS

```

```

;*****

```

```

;SUB #2 INICIA2
ROMCANAN .EGU OE000 ;ROM CANDADO ANALOGICOS
ROMCAND .EGU ROMCANAN+4 ;ROM CANDADO DIGITAL
ACANDADO .EGU ROMCAND+4 ;CANDADO ANALOGICO
DCANDADO .EGU ACANDADO+4 ;CANDADO DIGITAL

```

```

;*****

```

```

;SUB #3 PROPPER
MUAND0 .EGU DCANDADO+10 ;MULTIPLICANDO
MUADOR .EGU MUAND0+1 ;MULTIPLICADOR, USADAS AMBAS PARA LA SUBROUTINA MULTIP
DDBR .EGU MUADOR+1 ;DIRECCION DEL DATO DEL PIA, LADO B
CRB .EGU DDBR+1 ;REGISTRO DE CONTROL DEL PIA

```

```

;*****

```

```

;SUB #4 VALIDES
TEMP1 .EGU CRB ;DE TRABAJO

```

```

;*****

```

```

;SUB #5 MULTIP
TOTAL1 .EGU TEMP1+1 ;LSB DEL RESULTADO DE MULTIP
RANDO1 .EGU TOTAL1+2 ;LSB DEL MULTIPLICANDO ROTADO
RANDO2 .EGU RANDO1+1 ;MSB " " "

```

```

;*****

```

```

;SUB #8 CALECA
LIMANA .EQU RANDO2+1 ;LIMITE DE LOS OCHO PRIMEROS CANALES ANALOGICOS
LIMANA2 .EQU LIMANA+1 ;LIMITE DE LOS SIGUIENTES OCHO CANALES

;*****
;SUB #10 CALECD
IVALDIG .EQU O210 ;INFORMACION VALVULAS DIGITALES
IESPEJOS .EQU IVALDIG+3 ;INFORMACION ESPEJOS
IBOMDISA .EQU IESPEJOS+4 ;INFORMACION BOMBAS Y DISPOSITIVOS AUXILIARES

;*****
;SUB #11 ESANALOG
TEMP2 .EQU IVALDIG+10 ;DE TRABAJO
AINFORM .EQU TEMP2+1 ;INFORMACION ANALOGICA
EANALOG .EQU AINFORM+20 ;ESTADO ANALOGICO
EDOCANAN .EQU ROMCAND+10 ;ROM ESTADO EN CANDADO ANALOGICO

;*****
;SUB #12 Y #13 ECDIG1,ECDIG2
EDODIG .EQU EANALOG+20 ;ESTADO DIGITAL
EDOCANDD .EQU EDOCANAN+20;ROM ESTADO EN CANDADO DIGITAL

;*****
;SUB #14 Y #15 SALTARAN,SALTARD
TANALOG1 .EQU 7000 ;PRIMERA TARJETA ANALOGICA
ABTARAN1 .EQU 7010 ;HABILITA A LA TARJETA ANALOGICA
TANALOG2 .EQU 8000 ;SEGUNDA TARJETA ANALOGICA
ABTARAN2 .EQU 8010 ;HABILITA LA SEGUNDA TARJETA ANALOGICA
TDIG1 .EQU 0B000 ;PRIMERA TARJETA DIGITAL
TDIG2 .EQU 0C000 ;SEGUNDA TARJETA DIGITAL
MASCARAD .EQU EDOCANDD+10 ;ROM ESTA MASCARA DETERMINA QUE BITS SERAN NEGADOS
;A LA SALIDA DE LAS TARJETAS DIGITALES

;*****
;SUB #16 GUARDAB4
LIM .EQU EDODIG+10 ;EL NUMERO DE DATOS A MANDAR

;*****
;SUB #17 APUNTDES
PARRELOJ .EQU LIM+1 ;TIENE LAS CUENTAS DE LOS RELOJES
BUFFER3 .EQU PARRELOJ+7 ;EN DONDE SON RECIBIDAS LAS 16 VALVULAS DIGITALES
;APUNT .EQU OFF00 ROM APUNTADOR
INDIR .EQU 0002 ;APUNT

;*****
;SUB #18 PONBUFF4
TNIBBLE1 .EQU BUFFER3+4 ;CONTIENE EL BYTE A TRANSMITIR
TNIBBLE2 .EQU TNIBBLE1+1 ;EL SIGUIENTE BYTE A TRANSMITIR

;*****
;SUB #19 DESCOMP
TEMP3 .EQU TNIBBLE2+1 ;DE TRABAJO

;*****
;SUB #20 ACONDIC
EDODIGT .EQU TEMP3+1 ;TABLA DE VALVULAS DIGITALES A TRANSMITIR
TEMP4 .EQU EDODIGT+4 ;DE TRABAJO

;*****
;SUB #21 ABRETT

```

```

BUFFER1 .EQU TEMP4+1 ;DONDE ESTA EL BLOQUE RECIBIDO
BUFFER2 .EQU BUFFER1+44 ;SEGUNDO DEPOSITO DE BLOQUES RECIBIDOS
NUMDAT03 .EQU BUFFER2+44 ;NUMERO DE DATOS DEPOSITADOS EN EL BUFFER RECEPTOR
INFORM .EQU NUMDAT03+1 ;DE TRABAJO

```

```

;*****

```

```

;SUB #22 GUARDAR
INDIR2 .EQU 0004 ;ORIGEN:APUNT2
INDIR2B .EQU 0006 ;ORIGEN+3
NIBBLE1 .EQU INFORM+1 ;PRIMER BYTE A COMPACTAR
NIBBLE2 .EQU NIBBLE1+1 ;SEGUNDO BYTE A COMPACTAR
TEMP5 .EQU NIBBLE2+1 ;DE TRABAJO
INDYD .EQU TEMP5+1 ;DESCONTAMINA EL INDICE Y DEL DESTINO
;APUNT2 .EQU 0FF00 ROM APUNTADOR

```

```

;*****

```

```

;SUB #23 CASOS
;APLIM .EQU INDYD+1 ROM APUNTADOR
INDYD .EQU INDYD+1 ;DESCONTAMINAR EL INDICE EL INDICE Y DEL ORIGEN

```

```

;*****

```

```

;SUB #27 TARPRUE1
CONTTEC .EQU 9003 ;CANTIDAD DE PRESIONES AL TECLADO
;EN EL ACOPLAMIENTO DE PRUEBAS
INTEC .EQU 9004 ;ACEPTA INT POR PRESION DE TECLAS
CAMBACPR .EQU INDYD+1 ;BANDERA PARA CAMBIAR A PRUEBAS
;SIN ACOPLAMIENTO DE PRUEBAS
DIG1 .EQU CAMBACPR+1;CARACTER MENOS SIGNIFICATIVO A DESPLEGAR EN PANTALLA
DIG2 .EQU DIG1+1 ;CARACTER MAS SIGNIFICATIVO A DESPLEGAR EN PANTALLA
NUCANA .EQU DIG2+1 ;NUMERO DE CANAL ANALOGICO
NUCANDIG .EQU NUCANA+1 ;NUMERO DE CANAL DIGITAL

```

```

;*****

```

```

;SUB #29 PONCAR
ENTDAT .EQU 9000 ;MANDA CODIGO A DRIVER
SELDIG .EQU 9001 ; * EL # DE DIGITO SELECCIONADO

```

```

;*****

```

```

;SUB #31 POSD
RESIDUO .EQU NUCANDIG+1 ;BIT A MODIFICAR DEL REGISTRO
COCIENTE .EQU RESIDUO+1 ;REGISTRO A MODIFICAR
MODVALD .EQU COCIENTE+1 ;EL REGISTRO A MODIFICARLE UN BIT

```

```

;*****

```

```

;SUB #32 VALBIT
RESIDUO1 .EQU MODVALD+1 ;DE TRABAJO
RESULT .EQU RESIDUO1+1 ;INDICA QUE BIT SE MODIFICARA DE MODVALD
SUSTRA .EQU RESULT+1 ;SOLO TIENE UN BIT ENCENDIDO,
;ES EL CANAL DIGITAL A MODIFICAR

```

```

;*****

```

```

;SUB #33 SINACPR1
INDX3 .EQU SUSTRA+1 ;DESCOTAMINA AL REGISTRO X ANTES DE SEGUIR
REFERENC .EQU INDX3+1 ;VALOR ANTERIOR DEL PUERTO PARALELO
BANDERAS .EQU REFERENC+1 ;INDICA QUE VALOR SE DARA A LA SALIDA

```

```

;*****

```

```

;SUB #37 RECDTBH
NIBBLE1R .EQU BANDERAS+1 ;PRIMER BYTE TOMADO DEL BUFFER DE RECEPCION
NIBBLE2R .EQU NIBBLE1R+1 ;SEGUNDO BYTE TOMADO DEL BUFFER DE RECEPCION

```

```
                                ;AMBOS SERAN COMPACTADOSS  
INDX4 .EQU NIBBLE2R+1 ;SE ALMACENA EL INDICE X DEL BUFFER1  
INDY4 .EQU INDX4+1   ;SE ALMACENA EL INDICE Y DEL BUFFER2
```

```
*****  
;SUB #40 APCAMINO  
;CAMINOS .EQU OFF00 ROM APUNTADOR  
  
*****  
.INCLUDE CON2  
.INCLUDE CON3  
.INCLUDE CON4  
.INCLUDE CON5  
.INCLUDE CON6  
*****  
;  
;  
;CONTINUARA EN CON2  
;  
;  
*****
```

```

;*****
;
;
;ARCHIVO CON2
;SEGMENTO ANTERIOR:CON
;
;*****
;PROGRAMA PRINCIPAL
.PROC ACTUADOR
.ORG 0F000
RESET LDY #OFF ;EL SEGMENTO RESET PONDRÁ UN CICLO DE DEMORA
      $01 LDX #OFF ;DE FE01 O 65025 CICLOS
      $02 DEX
          BNE $02
          DEY
          BNE $01
          LDX #OFF
          TXS ;EL VALOR FF LO MANDA AL STACK POINTER
          STA NOINTTEC ;IMPIDE GENERACION DE INTERRUPCIONES
                    ;POR EL ACOPLAMIENTO DE PRUEBAS
INICIAL LDA #00 ;PONDRÁ EN CERO LAS VARIABLES MENCIONADAS ABAJO
        JSR ALMACEN ;PARA LA RUTINA DE RECEPCION
        STA GUARDA ;TRANSLADAR EL BLOQUE RX
        STA PRIMCAR ;TRANSMISION
        STA APCOLA ;ENCOLAR TAREAS DE TRANSMISION
        STA RECALCUL
        STA VACIADO
        STA REFRESCO
        STA ENCOLA
        LDX #00

      $01 STA COLA.X ;PONDRÁ EN CEROS LA COLA DE PETICION
                    ;DE TRANSMISION, SON 10 ELEMENTOS

        INX
        CPX #0A
        BEC $01 ;X<A BRINCA A $01
        JSR INICIA2 ;DEFINE LOS PRIMEROS VALORES DE CANDADOS Y ESTADOS

;CONFIGURAR PIA ANTES DE LEER MODO
      LDA #00
      STA CRA ;PARA ACCESAR AL DDRA
      STA DDRA ;EL DDRA SERA ENTRADA
      LDA #04
      STA CRA ;HABILITA AL PDRA

INICIO JSR LEEPIAA ;LEE EL MODO DE OPERACION EN EL LADO A DEL PIA
      STA DETCAMMO ;PRIMERA CONFIGURACION Y MODO QUE DEFINE
      LDA #00
      STA VALIDO ;SALIDA USADA EN PROGPER
      JSR PROGPER ;PROGRAMACION DE PERIFERICOS SEGUN EL MODO
      LDA VALIDO ;SI FUE VALIDO
      BNE PASO3 ;CONTINUA EN EL PASO3
      JMP INICIO ;SI NO BRINCAR A INICIO

PASO1 JSR LEEPIAA ;PARA LA N LECTURA

;PASO2 SI CAMBIA DE MODO, REPROGRAMAR PERIFERICOS

```

PASO2 CMP DETCAMMO ;LO COMPARA CON LA N-1 LECTURA
 BEQ PASO3 ;BRINCA SI NO HUBO CAMBIO DE MODO O CONFIGURACION

PASO2A LDA #00
 STA VALIDO ;PODRA SER ENCENDIDA EN PROPER
 JSR PROPER ;REPROGRAMA LOS PERIFERICOS Y VARIABLES DE CONFIGURACION

LDA VALIDO ;SI ES VALIDO ADELANTE
 BNE PASO3 ;CON EL PASO3
 JMP PASO1 ;SI NO ES VALIDO HARA UNA N+1 LECTURA

PASO3 CLI ;PREPARA AL SISTEMA PARA RECIBIR INTERRUPCIONES
 LDA PUERTO A ;TOMA LA N LECTURA PARA QUE
 STA DETCAMMO ;SEA LA REFERENCIA PARA LA N+1 LECTURA
 LDA REFRESCO ;PROPORCIONADA POR LA RUTINA PTM2
 BEQ PASOS ;SI NO HAY ORDEN DE REFRESCO IRSE AL PASO 5
 ;PASO 4 SI ESTA ENCENDIDA LA BANDERA REFRESCA

PASO4 LDA RECALCUL ;CONSULTA SI ESTA AUTORIZADO PARA RECALCULAR
 BEQ #03 ;SI NO HAY AUTORIZACION IRSE A #03

LDA REGANA ;SI NO HAY REGISTROS PARA CANALES ANALOGICOS
 BEQ #01 ;PREGUNTARA SI HAY PARA DIGITALES

JSR CALECA ;CALCULA ECUACION ANALOGICA
 LDA #01 ;HABILITA PARA QUE
 STA VACIADO ;PUEDA VACIAR SALTARAN

LDA REGANA ;PREGUNTA SI HAY OTRA TARJETA ANALOGICA
 CMP #10 ;SI HAY 16 CANALES ANALOGICOS
 BEQ #01
 JSR CALECA2 ;CALCULA ESTADOS PARA LA SEGUNDA TARJETA

#01 LDA REGDIG ;SI NO HAY REGISTRO DIGITALES
 BEQ #03 ;BRINCA #03

JSR CALECD ;CALCULA LA ECUACION DIGITAL
 LDA #01
 STA VACIADO ;HABILITARA LA SALIDA A TARJETAS DIGITALES Y ANALOGICAS
 JSR SALTARAN ;EFECTUA LA SALIDA DE VOLTAJES ANALOGICOS
 JSR SALTARD ;EFECTUA LA SALIDA DE VOLTAJES DIGITALES

#03 LDA #00 ;SE APAGARAN LAS BANDERAS PARA OTRA RONDA DE VALORES NUEVOS
 STA RECALCUL ;APAGA RECALCULO
 STA VACIADO ;APAGA VACIADO
 STA REFRESCO ;APAGA REFRESCO

PASOS LDX #00 ;ATENCION COLA TRANSMISION
 LDA COLA, X ;TOMA EL PRIMER ELEMENTO DE LA COLA
 BNE #01 ;SI NO ESTA VACIA IRA A #01
 JMP PASO6 ;BRINCA SI ESTA VACIA LA COLA

#01 LDA PRIMCAR ;SI SE MANDO EL PRIMER CARACTER
 BNE PASO6 ;SE IRA AL PASO 6

LDA COLA, X ;TOMA EL PRIMER ELEMENTO
 JSR GUARDABA ;SE PONEN TODOS LOS CARACTERES EN EL BUFFER

LDX #00 ;PONE EN CERO EL INDICE
 LDA MODO

```

CMP #03      ;SI NO ES MODO NORMAL
BNE #02      ;IRA A #02

LDA BUFFER4,X ;TOMA EL PRIMER ELEMENTO A MANDAR
STA TDR      ;SE MANDA EL PRIMER CARACTER LA IRO SE PODRA

LDA #0A1     ;GENERAR DESPUES DE QUE SE MANDO EL PRIMER
STA CR       ;CARACTER, PUEDE RECIBIR INTERRUPCIONES POR TX,RX
JMP #03

#02 LDA BUFFER4,X ;TOMA EL PRIMER CARACTER
STA TDR      ;LO MANDA AL REGISTRO DE TRANSMISION

LDA #21
STA CR       ;RECIBIRA INTERRUPCIONES POR TRANSMISION

#03 LDA #01   ;ENCIENDE PRIMCAR
STA PRIMCAR  ;INDICANDO QUE YA SE MANDO EL PRIMER CARACTER

PASO6 LDA GUARDA ;SE ENCIENDE SI EL BLOQUE FUE VALIDO POR RECEPDTBM
BNE #01      ;SI NO HAY QUE GUARDAR IRA AL PASO 7
JMP PASO7

#01 LDA #00   ;PARA PODER GUARDA OTRA
STA GUARDA   ;RONDA DE DATOS
JSR ABRETT   ;DEVOLVERA EL ACUMULADOR A
BNE #02      ;SI FUE VALIDO EL ANALISIS PROCEDERA A
              ;VERIFICAR SIGUIENTE CONDICION
JMP PASO1    ;SI NO, TOMARA OTRA LECTURA

#02 LDA ENCOLA ;CONSULTA SI EL BYTE FUE DE PETICION DE ESTADOS
BEQ #03      ;PROCEDERA A GUARDAR

LDA #00      ;LO APAGA PARA EL SIGUIENTE BYTE DE
STA ENCOLA   ;DE CONTROL QUE LLEGUE
JMP PASO7    ;NO GUARDAR E IRSE A PASO 7

#03 JSR GUARDAR ;GUARDA
LDA #01
STA RECALCUL ;CON DATOS NUEVOS SE POSIBILITA EL RECALCULO

PASO7 LDA MODO ;CONSULTA EL MODO
CMP #01      ;SI FUE DE PRUEBAS TENDRA
BEQ #01      ;EL CODIGO 01 O 02, CONTINUARA EL PASO 7

CMP #02
BEQ #01
JMP PASO1    ;SI NO ES MODO DE PRUEBAS COMENZARA DE NUEVO

#01 LDA PORB  ;EN EL LADO B DEL PIA SE ESPECIFICAN
              ;LAS CONDICIONES GENERALES DE PRUEBA
STA PUERTOB
AND #80      ;TOMA EL BIT 7
BEQ #02      ;SI VALE CERO ES PRUEBA SIN ACOPLAMIENTO
JMP TARPRUE1 ;SI FUE UNO ES PRUEBA CON ACOPLAMIENTO

#02 JSR SINACPR1 ;INICIALIZA PRUEBA SIN ACOPLAMIENTO
JMP SINACPR2 ;CONTINUA EN PRUEBAS SIN ACOPLAMIENTO

```

```

;*****

```


; ATENCION DE INTERRUPCIONES NEG(IRQ)

; HARA LA SECUENCIA DE ENCUESTAMIENTO

.ORG 0F189

```

SALVA PHA ; SALVA EL ACUMULADOR
      TXA
      PHA ; SALVA EL REGISTRO X
      TYA
      PHA ; SALVA EL REGISTRO Y
ENCPM1 LDA PTM+1 ; TOMA EL REGISTRO DE ESTATUS DEL PTM
      AND STATUSP ; DEFINIDO EN PROGPER, TOMA LOS BITS
      CMP #81 ; AUTORIZADOS PARA GENERAR UNA INTERRUPCION
      BNE ENCRECTR ; INTERRUMPIO EL PTM Y DE ESTE EL RELOJ 1
      ; SI NO, HUBO INTERRUPCION POR RELOJ2

```

; ATIENDE INTERRUPCION POR PTM1

```

LDA MODO2 ; SE CONFIGURA RELOJ2 Y ACCESA RELOJ1,
          ; O SOLO SE ACCESA RELOJ 1
STA PTM+1 ; REGISTRO DE CONTROL RELOJ 2
JSR RELOJ1 ; RECONFIGURA EL PTM1
LDA PTM+1 ; LEE STATUS
LDA PTM+2 ; Y TIMER #1 COUNTER PARA APAGAR LA BANDERA, DEL RELOJ 1
JMP PONE

```

; ENCUESTAMIENTO Y DESPACHO DE LA RUTINA DE RECEPCION/TRANSMISION

```

ENCRECTR LDA SR ; TOMA EL REGISTRO DE ESTADO DEL ACIA
          AND #80 ; BORRA DEJANDO EL BIT 7
          CMP #80 ; INVESTIVA AL BIT DE INTERRUPCION
          BNE ENCPM2 ; ENCUESTA PTM2

LDA SR ; TOMA EL REGISTRO DE ESTADO DEL ACIA
AND #01 ; BORRA DEJANDO EL BIT 1
CMP #01 ; INVESTIGA AL BIT DE REGISTRO DE RECEPCION LLENA DEL ACIA
BEQ ATIENDER ; SI ESTA ENCENDIDO ATIENDE RECEPCION
JMP ATIENDET ; SI NO LA INTERRUPCION ES DE TRANSMISION

```

; ATIENDE LA RECEPCION:

```

ATIENDER LDA RDR ; TOMA LO DEL REGISTRO DE RECEPCION DEL ACIA
          STA CHARACTER ; PARA ENTRADA DE LA RUTINA RECOTEM

LDA SR ; REGISTRO DE ESTADO DEL ACIA
AND #70 ; BORRA DEJANDO BITS INDICADORES DE ERROR DE RECEPCION
BEQ #01 ; BRINCA SI NO HUBO ERROR DE RECEPCION

LDA #01
STA ERRORAC ; INDICA SI HUBO ERROR DE RECEPCION
JMP #02

$01 LDA #00
    STA ERRORAC ; MARCA SI NO HUBO ERROR

$02 JSR RECOTEM ; LO GUARDA Y ANALIZA EL BLOQUE QUE SE ESTA FORMANDO
    CMP #01 ; LA RUTINA RECOTEM ENTREGA UN VALOR EN EL ACUMULADOR
    BEQ #03 ; EL BLOQUE FUE BUENO
    JMP PONE ; VA A FINALIZAR ATENCION DE INTERRUPCIONES

$03 JSR INICIAR1 ; PARA OTRA RONDA DE RECEPCION
    LDA #01 ; BANDERA PARA EL PROGRAMA PRINCIPAL, LE INDICA
    STA GUARDA ; QUE PUEDE PROCESAR LOS DATOS RECIBIDOS
    JMP PONE ; VA A FINALIZAR ATENCION DE INTERRUPCIONES

```

; ATIENDE TRANSMISION

```

ATIENDET LDA INDX2 ;DESCONTAMINA AL REGISTRO INDICE X
LDA BUFFER4,X ;TOMA UN ELEMENTO DEL BLOQUE A TRANSMITIR
STA TDR ;LO DEPOSITA EN EL REGISTRO DE TRANSMISION DEL ACIA

INC INDX2 ;INCREMENTA PARA PODER TOMAR EL SIGUIENTE DATO
INX ; " " " " " " " "
CPX COTSB4 ;PROPORCIONADO POR PONBUFF4, COMPARA CON LIMITE SUPERIOR
BEQ ACTCOLA ;SI EL CONTENIDO DEL REGISTRO X = COTSB4 ACTUALIZA COLA
JMP PONE ;VA A FINALIZAR INTERRUPCION

```

;ACTUALIZA COLA DE PETICION DE ESTADOS

```

ACTCOLA LDA #01 ;TOMARA A PARTIR DEL SEGUNDO ELEMENTO DE LA COLA
#01 LDA COLA,X ;TOMA EL N ELEMENTO

DEX ;DECREMENTA EL INDICE
STA COLA,X ;LO ALMACENA EN EL LUGAR QUE OCUPABA EL ELEMENTO N-1
INX ;INCREMENTA DOS VECES
INX ;PARA TOMAR EL ELEMENTO N+1
CPX #0A ;COMPARA EL INDICE CON EL ELEMENTO MAXIMO POSIBLE QUE ES 10
BCC #01 ;SI X<=A CONTINUA ACTUALIZANDO

LDA #00 ;PONE EN CERO
STA COLA+9 ;EL ULTIMO ELEMENTO DE LA COLA

LDA APCOLA ;SI EL APUNTADOR DE COLA
BEQ RECONFIG ;VALE CERO, BRINCA A RECONFIG
DEC APCOLA ;SI NO, DECREMENTAR APUNTADOR DE COLA

```

;RECONFIGURA EL ACIA

```

RECONFIG LDA MODO ;TOMA EL MODO DE OPERACION
CMP #03 ;LO COMPARA CON EL CODIGO DE OPERACION NORMAL
BNE #01 ;SI NO ES OPERACION NORMAL BRINCA A #01

LDA #81 ;EL CODIGO DE CONFIGURACION PARA EL MODO NORMAL
STA CR ;PUEDE RECIBIR INTERRUPCIONES POR RX
JMP #02

#01 LDA #03 ;CODIGO DE CONFIG. PARA EL MODO DE CONGELAMIENTO Y PRUEBAS
STA CR ;NO PUEDE RX NI TX

#02 LDA #00 ;FIN DE TRANSMISION INDICA QUE SE PODRA
STA PRIMCAR ;TRANSMITIR EL PRIMER CARACTER DEL SIGUIENTE BLOKE
JMP PONE ;VA A FINALIZAR INTERRUPCION

```

;ENCUESTAMIENTO DEL PTM2

```

ENCPTM2 LDA PTM+1 ;TOMA EL REGISTRO DE ESTADO DEL PTM
AND STATUSP ;BORRA LOS BITS NO AUTORIZADOS PARA INTERRUPTIR
CMP #82 ;SI ESTA ENCENDIDO EL BIT DE IRQ Y EL BIT 2 LO ENCENDIO
BEQ #01 ;SI SE CUMPLE LO ANTERIOR BRINCA A #01
JMP INTETEC ;SI NO, BRINCA A ATENDER LA INTERRUPCION POR TECLADO

#01 LDA PTM+1 ;CARGA EL ESTATUS DEL PTM Y
LDA PTM+4 ;EL TIMER #2 COUNTER PARA APAGAR LA BANDERA DEL RELOJ 2

LDA RECALCUL
CMP #01 ;SI ESTA ENCENDIDA LA BANDERA DE RECALCULO
BEQ #02 ;BRINCA A #02

LDA #00 ;ESTO IMPLICA QUE NO SE PUEDE RECALCULAR, NI

```

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

```

STA REFRESCO ;VACIAR, NO SE HARA ESTO EN EL PASO 2
JMP PONE ;FINALIZARA INTERRUPCIONES

$02 LDA #01 ;AUTORIZARA A RECALCULAR Y
STA REFRESCO ;VACIAR EN EL PASO 3
JMP PONE ;FINALIZARA INTERRUPCION

INTETEC LDA TECLA ;TOMA DEL TECLADO EL CODIGO DE LA TECLA OPRIMIDA
STA TECLA1 ;LO ALMACENA EN MEMORIA DE TRABAJO

LDA CAMINO ;EL VALOR DE CAMINO LO PROPORCIONA LA RUTINA TARPRUE1
TAX ;LO PONE EN EL REGISTRO INDICE
JSR APCAMINO ;DA UN VALOR DE DIRECCION
JMP @INDIR3 ;SEGUN EL NIVEL EN QUE SE ENCUENTRE TARPRUE1
;SE ATENDERA UNA DE LOS CUATRO CAMINOS

;PARA EL NIVEL DE COMANDOS
CAMINO1 LDA TECLA1 ;TECLA OPRIMIDA
AND #0F ;BORRA LOS 4 BITS MAS SIGNIFICATIVOS
CMP #0A ;IDENTIFICA SI SE OPRIMIO LA TECLA "A"
BEQ $01 ;SI FUE ASI BRINCA A $01, LA DESPLEGARA

CMP #0D ;IDENTIFICA SI FUE "D" LA TECLA OPRIMIDA
BEQ $01 ;LA DESPLEGARA

LDA TECLA1 ;TOMA EL CODIGO DE LA TECLA OPRIMIDA
AND #10 ;BORRA TODOS LOS BITS MENOS D4
CMP #10 ;COMPARA SI SE OPRIMIO EXC
BEQ $02 ;SI SE OPRIMIO "EXEC" BRINCA A $02
JMP $06

$01 STA ENTDAT1 ;SE DESPLEGARA EL CARACTER A O D

LDA #20 ;OCUPARA EL LUGAR SEXTO DE LA PANTALLA
STA SELDIG1 ;MEMORIA DE TRANSLADO

JSR PONCAR ;LO DESPLIEGA EN PANTALLA
JMP $06

$02 LDA ENTDAT1 ;TIENE EL DATO PREVIO AL EXEC
CMP #0A ;COMPARA SI FUE "A"
BEQ $03 ;SI LO FUE BRINCAR A $03

CMP #0D ;SI FUE "D"
BEQ $04 ;BRINCA
JMP $06 ;BRINCA A $06, NO FUE NI "A" NI "D"

$03 LDA NTARAN ;CHECA SI EXISTEN TARJETAS ANALOGICAS
BEQ $06 ;AVANCE=0, NO EXISTEN TARJETAS ANALOGICAS
JMP $05 ;AVANCE=1, SI EXISTEN

$04 LDA NTARDIG ;CHECA SI EXISTEN TARJETAS DIGITALES
BEQ $06 ;AVANCE=0, SE QUEDA EN EL NIVEL

$05 LDA #01
STA AVANCE ;AVANZARA DE NIVEL

$06 STA SOLINT ;HABILITA OTRA VEZ AL TECLADO PARA QUE VUELVA A INTERRUMPIR
JMP PONE ;FINALIZARA LA INTERRUPCION

```

; CANALES ANALOGICOS O DIGITALES (SEGUN EL CASO)

CANINO2 LDA #08
 STA SELDIG1 ; PARA ENCENDER EL CUARTO DIGITO DE LA PANTALLA
 JSR PROCESAT ; PROCESARA TECLA OPRINIDA

LDA RAMACAM2 ; VALOR DEFINIDO EN PROCESAT
 CMP #02 ; SI NO ES DOS BRINCA A #01
 BNE #01

LDA #10 ; COLOCA OTRO CARACTER EN PANTALLA
 STA SELDIG1 ; PARA PONERLO EN QUINTO LUGAR DE LA PANTALLA
 JSR PONCAR ; LO COLOCA EN PANTALLA
 JMP #05

#01 CMP #03 ; SI NO ES 03 BRINCA A #02
 BNE #05
 LDA NUCAN ; ENTREGADO POR PROCESAT
 BNE #03 ; NO ACEPTA #CANAL =0, BRINCA A #03
 ; SI CONTINUA CON LA ACEPTACION
 JMP #05

#03 CMP COTSUP ; NUMERO DE CANAL MAXIMO POSIBLE-1
 BCC #04 ; NUCAN<COTSUP
 JMP #05

#04 LDA #01 ; ACEPTA EL VALOR DEL CANAL Y
 STA AVANCE ; PASARA AL SIGUIENTE NIVEL

#05 STA SOLINT ; PARA RECIBIR OTRO INTERRUPCION
 JMP PONE ; IRA A FIN DE INTERRUPCION

; VALORES ANALOGICOS

CANINO3 LDA #01 ; PARA PROCESAT, SE DESPLEGARA EN EL
 STA SELDIG1 ; PRIMER LUGAR DE LA PANTALLA

JSR PROCESAT ; PROCESARA TECLA OPRINIDA
 LDA RAMACAM2 ; VALOR DEFINIDO EN PROCESAT
 CMP #02 ; SI NO FUE 2 BRINCA A #01

BNE #01
 LDA #02 ; COLOCARA OTRO CARACTER EN EL SEGUNDO
 STA SELDIG1 ; LUGAR DE LA PANTALLA
 JSR PONCAR ; PONE CARACTER EN PANTALLA
 JMP #02

#01 CMP #03
 BNE #02

LDA NUCAN ; VALOR ENTREGADO POR PROCESAT
 STA VACANA ; VALOR CANAL ANALOGICO

LDA #01
 STA AVANCE ; PARA QUE VACANA PUEDE SALIR A LA TARJETA

#02 STA SOLINT
 JMP PONE

; VALORES DIGITALES

CANINO4 LDA TECLA1 ; INFORMACION DE LA TECLA OPRINIDA
 AND #10 ; DEJA SOLO D4

```

CMP #10      ;BIT DETECTOR DE "EXC"
BEQ #03      ;PARA SALIDA DE INTERRUPCION

LDA TECLA1
AND #2F      ;BORRA D7, D6 Y D4
CMP #2E      ;PARA DETECTAR "SHIFT" "E"
BNE #01      ;BRINCA SI NO ES "EXC", NI "SHIFT" "E"

LDA #01
STA CAMBIV   ;PARA CAMBIAR DE NIVEL
STA AVANCE   ;PARA AVANZAR EN LA RUTINA #27 DE TARJPRUE
JMP #04

$01 CMP #02   ;SI EL VALOR DE LA TECLA OPRIMIDA
    BCC #02   ;ES < DUE 2
    JMP #04

$02 STA ENDDAT1 ;LO ALMACENA PARA FUTURO DESPLIEGUE Y SALIDA A TARJETA
    LDA #01   ;LO DESPLEGARA EN
    STA SELDIG1 ;EL PRIMER DIGITO DE LA PANTALLA
    JSR PONCAR ;LO DESPLEGARA
    JMP #04

$03 LDA ENDDAT1 ;LO TOMA DE $02
    STA VACAND ;SALVA VALOR DIGITAL
    LDA #01
    STA AVANCE ;PARA HABILITAR LA SALIDA A TARJETAS

$04 STA SOLINT ;SE PREPARA PARA ATENDER OTRA INTERRUPCION

PONE PLA      ;RESTABLECE Y
    TAY
    PLA      ;RESTABLECE X
    TAX
    PLA      ;RESTABLECE A
    RTI     ;ORDENA QUE REGRESE AL PROGRAMA PRINCIPAL

```

```

;*****
;
;
;CONTINUARA EN EL ARCHIVO CONS
;
;
;*****

```

```

;*****
;
;
;SEGMENTO CON3
;SEGMENTO ANTERIOR CON2
;
;
;*****
.DRG OF400
;ZONA DE SUBRUTINAS
;*****
;SUB #1
INICIAR1 LDA #00 ;PONE EN CEROS LAS
          STA PRINBLOQ ;SIGUIENTES VARIABLES
          STA DATOCONP
          STA NUMDATO1
          STA NUMDATO2
          STA ALMCONT
          STA CONT2
          STA MALBLOQ
          STA RECHAZO
          STA NUMDATO
          RTS
;*****
;SUB #2 INICIA2
INICIA2 LDX #00
        #01 LDA ROMCANAN,X ;TOMA DEL ROM Y
          STA ACANDADO.X ;LO PASA A RAM
          INX
          CPX #04 ;SON 4 REGISTROS PRIMEROS REGISTROS
          BCC #01
          LDX #00

        #02 LDA ROMCAND,X ;TOMA DEL ROM Y
          STA DCANDADO.X ;PASA AL RAM
          INX
          CPX #20 ;SON 64 REGISTROS
          BCC #02
          LDX #00

        #03 LDA EDOCANAN,X ;TOMA DEL ROM
          STA EANALOG,X ;DEPOSITA EN RAM
          INX
          CPX #20 ;HASTA 64 REGISTROS
          BCC #03
          LDX #00

        #04 LDA EDOCAND,X ;DEL ROM LO
          STA EDODIG,X ;TRANSLADA AL RAM
          INX
          CPX #10 ;SON 32 REGISTROS
          BCC #04
          RTS
;*****
;SUB #3 PROGPER
;VERIFICA QUE SEA VALIDA LA CONFIGURACION
PROGPER LDA PUERTO A ;CONTIENE LA CONFIGURACION
        AND #0C ;CONSIDERA BITS 2 Y 3
        LSR A ;LOS COLOCA EN LOS LUGARES

```

LSR A ;MENOS SIGNIFICATIVOS
STA NTARAN ;SIENDO EL NUMERO DE TARJETAS ANALOGICAS

LDA PUERTOA ;CONTIENE LA CONFIGURACION
AND #30 ;TOMA LOS BITS 4 Y 5
LSR A ;LOS RECORRE A LOS
LSR A ;LUGARES MENOS SIGNIFICATIVOS
LSR A
LSR A
STA NTARDIG ;ES EL NUMERO DE TARJETAS DIGITALES

JSR VALIDEZ ;CONSULTA SI FUE VALIDA LA CONFIGURACION
BNE #01 ;SI ES VALIDO BRINCA
RTS

;CALCULA LOS REGISTROS ANALOG Y DIG NECESARIO

#01 LDA #10 ;CADA TARJETA ANALOGICA
STA MUANDO ;TIENE 16 REGISTROS

LDA NTARAN ;EL NUMERO DE TARJETAS ANALOGICAS
STA MUADOR ;SE DEFINE COMO EL MULTIPLICADOR

JSR MULTIP ;MULTIPLICA NTARAN*10
STA REGANA ;ES EL RESULTADO DE MULTIP

LDA #08 ;OCHO REGISTROS POR TARJETA DIGITAL
STA MUANDO

LDA NTARDIG ;NUMERO DE TARJETAS DIGITALES
STA MUADOR

JSR MULTIP ;NTARDIG*8
STA REGDIG ;RESULTADO DE MULTIP

LDA #01 ;BANDERA DE SALIDA
STA VALIDO ;PARA INICIO2A

;PROSEGUIR LA PROG DE PERIF SEGUN MODO

LDA PUERTOA ;TOMA LOS DOS PRIMEROS BITS
AND #03 ;DE LA CONFIGURACION QUE SON
STA MODO ;DEFINIDOS PARA EL MODO DE OPERACION

CMP #03 ;PREGUNTA SI EL EL MODO NORMAL
BEQ NORMAL ;BRINCANDOSE A ESTA ETIQUETA
JMP SEGIN1 ;SI NO LO ES SIGUE INTERROGANDO

NORMAL LDA #03 ;EFECTUA UN MASTER RESET
STA CR ;AL ACIA

LDA #31
STA CR ;PUEDE RECIBIR BYTES DEL EXTERIOR

LDA #00 ;GARANTIZA LA SECUENCIA CORRECTA
STA PTM+1 ;DE ACCESO A CADA RELOJ DEL PTM

JSR RELOJ2 ;CONFIGURA RELOJ2
JSR RELOJ1 ;CONFIGURA RELOJ1

LDA #0C1 ;MASCARA DE CONFIGURACION QUE
STA MODO2 ;CONFIGURARA EL PTM1 Y PTM2

```

LDA #83      ;CONSIDERA SOLO LOS BITS DE INTERRUPCION DEL RELOJ1 Y
STA STATUSP ;RELOJ2, Y EL BIT 7 DE NEG(IRQ)

STA NOINTEC ;NO PERMITE INTERRUPCION DEL ACOPLAMIENTO DE PRUEBAS
RTS

SEGINT LDA MODO    ;SI VALE CERO ES
BEQ DESPUCEN ;DESHABILITACION DEL PUESTO CENTRAL
JMP PRUEBAS ;SI NO, BRINCA A PRUEBAS

DESPUCEN LDA #00
STA RECALCUL ;NO PERMITE RECALCULO, DE ECUACION
STA VACIADO ;NO PERMITE VACIADO A VACIADO A TARJETAS

LDA #00
STA PTM+1 ;EL RELOJ2 NO GENERARA UNA IRQ

LDA #01 ;PARA ACCESAR
STA PTM+1 ;AL RELOJ 1
JSR RELOJ1 ;CONFIGURA AL RELOJ1

LDA #01
STA MODO2 ;CONFIGURARA SOLO EL PTM1
STA NOINTEC ;NO PERMITE AL TECLADO GENERE UNA INTERRUPCION

LDA #81 ;PARA IDENTIFICAR LA INTERRUPCION DEL RELOJ1
STA STATUSP ;Y SI LA SEBAL NEG(IRQ) FUE ENCENDIDA POR ESTE
RTS

LDA #03 ;NO PERMITE TRANSMISION
STA CR ;NI RECEPCION
RTS

FRUEBAS LDA #00 ;NO PERMITE RECALCULO DE ECUACION
STA RECALCUL
STA VACIADO ;NO PERMITE VACIADO A TARJETA

LDA #01 ;PARA ACCESAR AL RELOJ1
STA PTM+1
JSR RELOJ1 ;CONFIGURA AL RELOJ1

LDA #81 ;PARA IDENTIFICAR LA INTERRUPCION DEL RELOJ1
STA STATUSP ;Y SI LA SEBAL NEG(IRQ) FUE ENCENDIDA POR ESTE

LDA #01 ;PARA CONFIGURAR SOLO EL RELOJ1
STA MODO2 ;EN LA INTERRUPCION GENERADA POR ESTE

LDA #03 ;NO PERMITE TRANSMICION
STA CR ;NI RECEPCION
;CONFIGURA EL PUERTO B COMO ENTRADA
LDA #00 ;PARA ACCESAR AL
STA CRB ;PHERIFERAL DATA REGISTER B
STA DDRB ;EL LADO B SERA ENTRADA DE DATOS

LDA #04 ;SE PODRA TENER ACCESO AL
STA CRB ;PDRB
RTS

```

```

;*****

```



```

;SUB #4 VALIDEZ
VALIDEZ LDA #02
        CMP NTARDIG
        BCC $04      ;SI 2<NTARDIG

        CMP NTARAN
        BCC $04      ;SI 2<NTARAN

        LDA #01      ;VALOR DE SALIDA PARA CONFIGURACION VALIDA
        RTS

$04 LDA #00      ;VALOR DE SALIDA PARA CONFIGURACION NO VALIDA
    RTS

;*****
;SUB #5 MULTIP
MULTIP CLC      ;BORRA EL CARRY
        LDA #00
        STA TOTAL1 ;LIMPIA LOS TOTALES Y RANDO2 YA QUE
        STA TOTAL1+1 ;SERAN SUMADOS POSTERIORMENTE
        STA RANDO2 ;RECIBIRA AL MULTIPLICANDO ROTADO, PARTE MAS SIGNIFICATIVA

        LDA MUANDO ;MULTIPLICANDO
        STA RANDO1 ;PARTE MENOS SIGNIFICATIVA DEL MUANDO ROTADO

        LDA MUADOR ;MULTIPLICADOR
        STA TEMP1 ;DE TRABAJO

$01 LSR TEMP1 ;PARA SUMAR DOS NUMEROS DE 2 BYTES C/U
    BCS $02 ;SI EL CARRY ESTA PRENDIDO
    BNE $03 ;PARA ROTAR MULTIPLICANDO, SI NO VALE CERO

        LDA TOTAL1 ;BYTE MENOS SIGNIFICATIVO DE LA MULTIPLICACION
        RTS

$02 CLC ;BORRA CARRY
    LDA TOTAL1 ;SUMA EL TOTAL ANTERIOR, MENOS SIGNIFICATIVO
    ADC RANDO1 ;CON RANDO1
    STA TOTAL1 ;SE ALMACENA EL RESULTADO

        LDA TOTAL1+1 ;SUMA EL TOTAL
        ADC RANDO2 ;ANTERIOR MAS SIGNIFICATIVO
        STA TOTAL1+1 ;ESTA EL NUEVO RESULTADO

$03 ASL RANDO1 ;ROTAR MULTIPLICANDO, EL
    ROL RANDO2 ;CARRY DE RANDO1 SE PASA A RANDO2
    JMP $01 ;SE REPITE LA OPERACION

;*****
;SUB #6 RELOJ2
RELOJ2 LDA #0C1
        STA PTM+1 ;CONFIGURO PTM2, SE DISPARA AL CARGAR LOS LATCHES

        LDA #5D
        STA PTM+4 ;MSB BUFFER2

        LDA #0C0
        STA PTM+5 ;LSB LATCH #2
        RTS

```

```
*****
```

```
;SUB #7 RELOJ1
RELOJ1 LDA #OE0
      STA PTM      ;PTM1 MODO SIMPLE.SALIDA HABILITADA

      LDA #OFF
      STA PTM+2    ;MSB BUFFER1

      LDA #OFF
      STA PTM+3    ;LSB LATCH#1
      RTS
```

```
*****
```

```
;SUB #8 CALECA
CALECA LDX #00      ;PARA LA PRIMERA TARJETA
      LDA #03
      STA LIMANA    ;CANAL MAXIMO DE LOS OCHO PRIMEROS

      LDA #10       ;16
      STA LIMANA2   ;CANAL MAXIMO PARA LOS SIGUIENTES OCHO, SEGUNDA TARJETA

      LDA ACANDADO  ;CANDADO PARA LOS 8 PRIMEROS REGISTROS
      JSR ESANALOG ;CALCULA ESTADOS ANALOGICOS

      LDA ACANDADO+1 ;CANDADO PARA LOS CANALES 9-16
      JSR ESANALOG ;CALCULA LA EC ANALOGICA
      RTS
```

```
*****
```

```
;SUB #9 CALECA2
CALECA2 LDA #18     ;24
      STA LIMANA    ;CANAL MAXIMO DE LOS 8 PRIMEROS CANALES

      LDA #20       ;32
      STA LIMANA2   ;SIGUIENTES 8 CANALES,SGUNDA TARJETA

      LDA ACANDADO+2 ;CANALES 17-24
      JSR ESANALOG ;CALCULO ESTADO ANALOGICO

      LDA ACANDADO+3 ;CANALES 25-32
      JSR ESANALOG ;CALCULO ESTADOS ANALOGICOS
      RTS
```

```
*****
```

```
;SUB #10 CALECD
CALECD LDX #00      ;PARA TABLA EN GENERAL
      LDY #00      ;PARA TABLA EN PARTICULAR

      #01 JSR ECDIG1 ;CALCULA LA PRIMERA PARTE
          AND IVALDIG,Y ;EL RESULTADO EN EL ACUMULADOR
          JSR ECDIG2 ;CALCULA LA SEGUNDA PARTE
          CPY #03
          BNE #01    ;HASTA EL TERCER ELEMENTO DE IVALDIG

          LDY #00    ;PONE EN CERO PARA IESPEJOS

      #02 JSR ECDIG1
          AND IESPEJOS,Y
          JSR ECDIG2
```

```

CPY #04
BNE #02 ;HASTA EL CUARTO ELEMENTO DE IESPEJOS

JSR ECDIG1
AND IBOMBISA ;CALCULO DE BOMBAS Y DISPOSITIVOS AUXILIARES
JSR ECDIG2

$03 JSR ECDIG1 ;EL INDICE X VALE 8
AND IVALDIG,X ;NOTESE EL CAMBIO DE INDICE
JSR ECDIG2

CPX REGDIG ;LIMITE DEL REGISTRO A CALCULAR
BCC #03 ;X<REGDIG
RTS

;*****
;SUB #11 ESANALOG
;RUTINA QUE CALCULA LA EC ANALOGICA
ESANALOG STA TEMP2 ;ENTRA EL CANDADO ANALOGICO

$01 LSR TEMP2
BCS #02 ;SI HAY CANDADO

LDA AINFORM,X ;NO HAY LA INFORMACION
STA EANALOG,X ;ES IGUAL AL ESTADO ANALOGICO
JMP #03

$02 LDA EDOCANAN,X ;TOMA EL ESTADO EN CANDADO
STA EANALOG,X ;LO IGUAL AL ESTADO ANALOGICO

$03 INX
CPX LIMANA ;COTA PARA LOS 8 PRIMEROS REGISTROS
BCC #01 ;SI NO HA ACABADO
BEQ #04 ;CONCLUYE LA RUTINA PARA LOS 8 PRIMEROS

CPX LIMANA2 ;COTA PARA LOS SIGUIENTES 8 REGISTROS
BCC #01 ;SI NO HA ACABADO
$04 RTS

;*****
;SUB #12 Y SUB #13
;RUTINAS PARA LA EC DIGITAL
ECDIG1 LDA DCANDADO,X ;TOMA LOS CANDADOS DEFINE QUE
AND EDOCANDD,X ;ESTADOS EN CANDADO, SE TOMARAN EN CUENTA
STA TEMP2 ;ALMACENA RESULTADO
LDA DCANDADO,X ;TOMA LOS CANDADOS
EOR #0FF ;LOS NIEGA PARA OPERAR CON INFORMACION I
RTS

ECDIG2 CLC
ADC TEMP2 ;LE SUMA LA PRIMERA PARTE CALCULADA EN ECDIG1
STA EDODIG,X ;EL NUEVO ESTADO DIGITAL
INX ;INCREMENTA REGISTRO DE TABLA GENERAL
INY ;INCREMENTA REGISTRO DE TABLA PARTICULAR
RTS

;*****
;SUB #14 SALTARAN
SALTARAN LDA VACIADO ;PREGUNTA SI PUEDE VACIAR
BEQ #02 ;SI NO PUEDE VACIAR SALDRA DE RUTINA

```

```
LDA REGANA ;PREGUNTA SI EXISTEN REGISTROS ANALOGICOS
BEQ #02 ;SI NO LOS HAY SALDRA DE RUTINA
```

```
LDX #00
STA ABTARAN1 ;HABILITA TARJETA ANALOGICA
```

```
*01 LDA EANALOG,X
EOR #OFF ;NIEGA LOS ESTADOS YA QUE A LA
STA TANALOG1,X ;SALIDA A LA TARJETA SE NIEGAN OTRA VEZ
INX ;INCREMENTA PARA FORMAR EL SIGUIENTE ELEMENTO DE LA
INX ;TABLA, EL REGISTRO 1 ES EL INDICE 0
CPX REGANA
BCC #03
BEQ #03 ;X<=REGANA
```

```
*02 RTS
```

```
*03 DEX ;REGRESANDO A TABLA
CPX #10 ;16 CANALES
BCC #01 ;X<16
LDY #00 ;PARA EL PRIMER ELEMENTO DE LA SEGUNDA TARJETA
STA ABTARAN2 ;HABILITA LA SEGUNDA TARJETA
```

```
*04 LDA EANALOG,X ;TOMA DE LA TABLA Y
EOR #OFF ;NIEGA EL RESULTADO
STA TANALOG2,Y ;PARA SALIR A LA TARJETA
```

```
INX ;INDICE DE LA TARJETA
INX ;INDICE DE LA TABLA
INX
CPX REGANA
BCC #05
BEQ #05 ;X<=REGANA
RTS
```

```
*05 DEX
JMP #04
```

```
;*****
```

```
;SUB #15 SALTARD
```

```
SALTARD LDA VACIADO ;PREGUNTA SI PUEDE VACIAR
BEQ #02 ;SI NO PUEDE SALE DE RUTINA
```

```
LDA REGDIG ;PREGUNTA SI EXISTEN REGISTROS DIGITALES
BEQ #02 ;SI NO HAY SALE DE RUTINA
LDX #00
```

```
*01 LDA E00DIG,X ;TOMA EL ESTADO DIGITAL
EOR MASCARAD,X ;LO NIEGA CON BASE EN UNA MASCARA
STA TDIG1,X
```

```
INX ;INCREMENTA PARA TOMAR EL SIGUIENTE ELEMENTO DE LA
INX ;TABLA, EL REG 1 ES EL INDICE 0 DE LA TABLA
```

```
CPX REGDIG
BCC #03
BEQ #03 ;X<=REGDIG
```

```
*02 RTS
```

```

$03 DEX          ;PARA PASAR A TABLA
    CPX #08      ;LOS PRIMEROS OCHO REGISTROS
    BCC #01      ;X<=8
    LDY #00      ;PRIMER ELEMENTO DE LA PRIMERA TARJETA

$04 LDA EDDIG,X
    EOR MASCARAD,X ;NIEGA CON BASE EN UNA MASCARILLA
    STA TDIG2,Y   ;SALIDA A TARJETA DIGITAL
    INY           ;PARA EL SIGUIENTE ELEMENTO DE TARJETA
    INX           ; " " " " " " TABLA
    INX           ;EL PRIMER REGISTRO ES EL INDICE CERO DE LA TABLA
    CPX REGDIG
    BCC #05
    BEQ #05      ;X<=REGDIG
    RTS

$05 DEX          ;EL VALOR PARA TABLA
    JMP #04
    RTS

```

```

;*****

```

```

;SUB #16 GUARDAB4
GUARDAB4 CMP #5E ;SI NO ES PETICION DE ESTADO DE
    BNE #01      ;VALVULAS ANALOGICAS SIGUE PREGUNTANDO

    LDA #77      ;FORMACION 3 PRIMEROS BYTES, SE PONE RECEPCION DE
    STA BUFFER4 ;ESTADO DE VALVULAS REGULARES EN EL BUFFER DE SALIDA

    LDA #76      ;SE ESPECIFICA EL # DE DATOS
    STA BUFFER4+1 ;A MANDAR, SE MANDARAN 6

    LDA #70
    STA BUFFER4+2

    LDX #0C      ;PARA LOCALIZAR EL DIMICILIO
    JSR AFUNDES ;DE EANALOG

    LDA #06      ;SE LE DICE A PONEFF4 QUE
    STA LIM      ;SON 6 DATOS A MANDAR
    JSR PONEUFF4 ;SE FORMA EL RESTO DEL BLOQUE

    LDA #0A      ;10 BYTES TIENE EL BLOQUE A MANDAR
    STA COTSB4   ;SE USARA EN LA INTERRUPCION POR TRANSMISION
    JMP #03

$01 CMP #5D      ;SI NO ES PETICION DE ESTADO DE VALVULAS DIGITALES
    BNE #02      ;IRSE A $02

    LDA #76      ;RECEPCION DE ESTADO DE VALVULAS BINARIAS
    STA BUFFER4 ;PRIMER BYTE DEL BLOQUE A TRANSMITIR

    LDA #78      ;SE ESPECIFICA EL NUMERO DE DATOS, ES 8
    STA BUFFER4+1 ;SEGUNDO BYTE

    LDA #70      ;ESPECIFICA # DE DATOS, PARTE MAS SIGNIFICATIVA
    STA BUFFER4+2 ;TERCER BYTE
    JSR ACONDIC ;COMO SON 10 VALVULAS EN EDDIG SE MANDARAN
                ;6 ADICIONALES QUE VALDRAN UNOS.
    LDX #0E      ;PARA LOCALIZAR EL DOMICILIO DE

```

JSR APUNTES ;LA TABLA EDDODIGT

LDA #08 ;SE LE DICE A PONBUFF4
 STA LIM ;QUE SERAN 8 DATOS A MANDAR
 JSR PONBUFF4 ;SE FORMA EL RESTO DEL BLOQUE A MANDAR,

LDA #0C ;12 BYTES TIENE EL BLOQUE A
 STA COTSB4 ;SE USARA EN LA INTERRUPCION POR TRANSMICION
 JMP #03

;PARA LOS CANDADOS Y PARAMETROS

\$02 LDA #7C ;INDICA ENVIO DE CANDADOS Y PARAMETROS
 STA BUFFER4

LDA #72 ;SE MANDARAN 22H DATOS=34
 STA BUFFER4+1 ;EL BYTE MENOS SIGNIFICATIVO DEL # DE DATOS

LDA #72
 STA BUFFER4+2 ;BYTE MAS SIGNIFICATIVO DEL # DE DATOS

LDX #06 ;LOCALIZA EL DOMICILIO
 JSR APUNTES ;DE PARRELOJ

LDA #0E ;INDICA EL VALOR MAXIMO (14) DEL INDICE
 STA LIM ;DE BUFFER4+3, PARA EL SEGUNDO SEGMENTO
 JSR PONBUFF4 ;FORMA LOS 14 BYTES
 STX INDX2 ;SALVA EL INDICE DE BUFFER4

LDX #08 ;LOCALIZA EL DOMICILIO
 JSR APUNTES ;DE ACANDADO

LDA #12 ;FORMAR LOS 4 SIGUIENTES BYTES, 18 SERA EL
 STA LIM ;VALOR DEL INDICE PARA EL SIGUIENTE SEGMENTO

LDX INDX2 ;DESCONTAMINA EL REGISTRO INDICE PARA PONB41
 JSR PONB41 ;ES UNA SUBDIVISION DE PONBUFF4 TOMA EL SIGUIENTE ELEMENTO
 STX INDX2 ;Y LO DESCOMPACTA, SALVA EL INDICE DE BUFFER4

LDX #0A ;LOCALIZA EL DOMICILIO
 JSR APUNTES ;DE ACANDADO

LDA #22 ;VALOR MAXIMO DEL INDICE DE BUFFER4+3
 STA LIM ;PARA USARSE EN PONB41
 LDX INDX2 ;DESCONTAMINA EL REGISTRO INDICE X PARA PONB41
 JSR PONB41 ;TOMA LOS SIGUIENTES 16 BYTES

LDA #26 ;26 BYTES TIENE EL BLOQUE A MANDAR
 STA COTSB4 ;SE USARA EN LA INTERRUPCION POR TRANSMICION

\$03 LDA #01
 STA INDX2 ;ATENDER LA INT A PARTIR DEL 2 CARACTER
 RTS

;

;

;CONTINUA EN CON4

;

;

```

;*****
;
;
;SEGMENTO CON4
;SEGMENTO ANTERIOR: CON3
;
;
;*****
;SUB #17 APUNTES
;RUTINA QUE ALMACENA EL DOMICILIO DE APUNT
APUNTES LDA APUNT,X ;TOMA EL BYTE MENOS SIGNIFICATIVO
        STA INDIR ;DOMICILIO MENOS SIGNIFICATIVO
        INX

        LDA APUNT,X ;TOMA EL BYTE MAS SIGNIFICATIVO
        STA INDIR+1 ;DOMICILIO MAS SIGNIFICATIVO
        RTS

;*****
;SUB #18 PONBUFF4
;RUTINA QUE COLOCA LOS ESTADOS INDICADOS EN EL BUFFER4
PONBUFF4 LDX #00 ;PARA EL BUFFER4, ES EL DESTINO
PONE41 LDY #00 ;PARA EL ORIGEN

%01 LDA @INDIR,Y ;(INDIR),Y DE DONDE TOMARA EL NUMERO A DESCOMPACTAR
     JSR DESCOMP ;LO DESCOMPACTA, SALIENDO DOS BYTES PARA TRANSMITIR

        LDA TNIBBLE1 ;CARGA EL PRIMER BYTE DESCOMPACTADO
        STA BUFFER4+3,X ;LO PONDRÁ EN EL BUFFER DE SALIDA
        INX ;PARA PONER EN EL SIGUIENTE LUGAR

        LDA TNIBBLE2 ;TOMA EL SEGUNDO BYTE DESCOMPACTADO
        STA BUFFER4+3,X ;LO PONE EN EL BUFFER DE SALIDA
        INY ;PARA DESCOMPACTAR EL SIGUIENTE BYTE
        INX ;PARA COLOCAR EN EL SIGUIENTE ELEMENTO
        CPX LIM ;DE CADA TIPO DE DATOS EN EL BUFFER4
        BNE %01 ;X NO ES IGUAL DE LIM

        LDA #7F ;ULTIMO CODIGO FIN DE TRANSMISION
        STA BUFFER4+3,X ;COLOCADO EN EL BUFFER DE SALIDA
        RTS

;*****
;SUB #19 DESCOMP
;RUTINA DE DESCOMPACTACION
DESCOMP STA TEMP3 ;A CONVERTIR EN 2 BYTES
        LDA #70 ;BITS DE CONTROL, LOS 4 MAS SIGNIFICATIVOS
        STA TNIBBLE1 ;PARTE MENOS SIGNIFICATIVA DEL BYTE DESCOMPACTADO
        STA TNIBBLE2 ; " MAS " " " "

        LDA TEMP3
        AND #0F ;TOMA LOS 4 BITS MENOS SIGNIFICATIVOS
        CLC
        ADC TNIBBLE1 ;LOS AÑADE A LOS BITS DE CONTROL
        STA TNIBBLE1 ;DEFINIDOS ANTERIORMENTE

        LDA TEMP3 ;TOMA LOS 4 BITS MAS SIGNIFICATIVOS
        LSR A ;DEL BYTE A DESCOMPACTAR
        LSR A ;RECORRIENDOLOS A LOS LUGARES MENOS
        LSR A ;SIGNIFICATIVOS

```

```

LSR A      ;PARA AADIRLOS A LOS
CLC
ADC TNIBBLE2 ;BITS DE CONTROL
STA TNIBBLE2 ;RESULTADO FINAL
RTS

```

```

;*****

```

```

;SUB #20 ACONDIC
;ROUTINA QUE HACE UN PREPARATIVO PREVIO DE ESTADOS VALVULAS DIG
;SE MANDAN UNOS PARA LAS VALVULAS NO DEFINIDAS
ACONDIC LDA EDD01G
        ASL A      ;TOMARA LOS BITS 2 Y 3
        ASL A      ;RECORRIENDOLOS 4 LUGARES
        ASL A      ;A LA IZQUIERDA
        ASL A
        AND #0C0   ;LIMPIANDO TODO LO DEMAS
        STA EDD01G

        LDA EDD01G ;TOMA EN CUENTA LOS BITS
        AND #03    ;0 Y 1
        CLC
        ADC EDD01G ;AADIENDOLOS A LOS DOS BITS CONSIDERADOS ANTES
        STA EDD01G

        LDA #3C    ;LOS BITS RESTANTES (2,3,4,5) DE EDD01G
        ADC EDD01G ;SERAN UNOS, AADIENDOSE A LOS ANTERIORES
        STA EDD01G ;LISTO PRIMER REG A MANDAR

        LDA EDD01G ;DE ESTE REGISTRO
        AND #0F0   ;SE MANDARA EL NIBBLE MAS SIGNIFICATIVO
        LSR A
        LSR A      ;SE RECORREN DOS LUGARES A LA IZQUIERDA
        STA EDD01G+1 ;SE ALMACENAN EN EL REGISTRO A MANDAR

        LDA #0C3   ;PONIENDOSE UNOS EN LOS BITS (0,1,6,7)
        CLC        ;DE ESTE
        ADC EDD01G+1
        STA EDD01G+1 ;LISTO SEGUNDO REG A MANDAR

        LDA EDD01G+1 ;EL SIGUIENTE REGISTRO A DESMENUJAR SE
        STA TEMP3   ;PONDRÁ EN UNA VARIABLE DE TRABAJO

        ASL A
        ASL A      ;RECORRE DOS LUGARES A LA IZQUIERDA
        AND #0F0   ;SE BORRAN LOS 4 BITS MENOS SIGNIFICATIVOS
        STA EDD01G+2 ;SE ALMACENA EN EL REGISTRO A MANDAR

        LDA EDD01G+1 ;SE CONSIDERARAN LOS BITS 0 Y 1
        AND #03    ;DE ESTE REGISTRO
        CLC        ;EL CARRY NO SE SUMA AL REGISTRO A MANDAR
        ADC EDD01G+2 ;SE ADEA A LO DEFINIDO ANTERIORMENTE
        STA EDD01G+2 ;EN ESTE RESULTADO

        LDA #0C    ;SERAN UNOS LOS BITS 2,3
        CLC
        ADC EDD01G+2
        STA EDD01G+2 ;LISTO TERCER REG A MANDAR

        LDA EDD01G+2 ;REGISTRO A MANDAR
        STA TEMP4   ;DE TRABAJO

```



```

ROL TEMP3 ;EL BIT 7 DE EDODIG+1 SE PASA AL CARRY
ROL TEMP4 ;Y EL CONTENIDO DE ESTE SE PASA AL BIT 0

ROL TEMP3 ;SE PASA AL CARRY EL BIT 6 DE EDODIG+1
ROL TEMP4 ;EL CONTENIDO DE CARRY PASA AL BIT 0
LDA TEMP4 ;SE PASARON DOS BITS DE EDODIG+1 A EDODIG+2
AND #0F ;SE TOMA EN CUENTA EL NIBBLE MAS SIGIFICATIVO
STA EDODIG+3 ;PARA ALMACENARLO EN EL REGISTRO A MANDAR

LDA TEMP4 ;SE TOMARAN EN CUENTA LOS BITS 2 Y 3 DE EDODIG+2
ASL TEMP4 ;NOTESE QUE TEMP4 CONTIENE A EDODIG+2, RECORRIDO DOS BITS
ASL TEMP4 ;A LA IZQUIERDA Y SOLO HACE FALTA RECORRERLOS DOS VECES

LDA TEMP4 ;SE TOMA LO RECORRIDO PARA
AND #0C ;TOMAR EN CUENTA LOS BITS 7,6
CLC
ADC EDODIG+3 ;SE ABAEN AL REGISTRO A MANDAR
STA EDODIG+3

LDA #30 ;SERAN UNDS LOS BITS 4,5
CLC
ADC EDODIG+3 ;SE ABAEN AL REGISTRO A MANDAR
STA EDODIG+3 ;SE DEPOSITA EL RESULTADO AL REGISTRO A MANDAR
RTS

;*****
;SUB #21 ABRETT
;ANALISIS DEL BLOQUE RECIBIDO
;ENCOLAR TAREAS DE TRANSMICION
ABRETT LDA ALMACEN ;PARA SABER QUE BUFFER ANALIZAR
BEQ #01 ;BRINCA SI ES EL BUFFER2

LDA BUFFER1+44 ;TOMA EL NUMERO DE DATOS DE ESTE BUFFER
STA NUMDAT03

LDA BUFFER1 ;TOMA EL BYTE DE CONTROL
JMP ANALIZA

#01 LDA BUFFER2+44 ;TOMA EL NUMERO DE DATOS DE ESTE BUFFER
STA NUMDAT03

ANALIZA LDA BUFFER2 ;TOMA EL BYTE DE CONTROL
CMP #5C ;CODIGO DE PETICION
BCS ENCOLAR ;>=5C
STA INFORM
JMP CONTANAL ;CONTINUAR CON EL ANALISIS PARA VER SI SE ENCIENDEN LAS
;BANDERAS DE RECALCULO Y VACIADO
;ENCOLAR (PREVIAMENTE SE INICIALIZO APCOLA A CERO LA PRIMERA VEZ)

ENCOLAR LDX APCOLA ;DESCONTAMINACION DE X
CPX #09 ;PARA EL CASO DEL ULTIMO ELEMENTO DE LA COLA
BCC ADELANTE ;X<09
STA COLA+9 ;PARA EL ULTIMO ELEMENTO, SE ALMACENA EN ULTIMO LUGAR

LDA #09 ;SE IMPIDE QUE EL INDICE
STA APCOLA ;REBASE EL TAMAÑO DE LA COLA
JMP ADELFA

ADELANTE STA COLA,X ;ALMACENA LA PETICION

```

```

      INX          ;APUNTA AL INDICE A ALMACENAR
      STX APCOLA  ;SE CONTAMINARA POSTERIORMENTE

ADELF LDA #01     ;ANALISIS VALIDO
      STA ENCOLA  ;INDICA QUE DEBE IRSE AL PASO 7, NO GUARDAR
      RTS

;CONTINUACION DEL ANALISIS

CONTANAL CMP #57  ;SI NO INFORMACION DE
      BNE $01     ;VALVULAS ANALOGICAS BRINCA
      LDA NUMDAT03 ;SI EL NUMERO DE DATOS QUE LE LLEGARON
      CMP #06     ;ES IGUAL AL NUMERO DE DATOS ESPERADOS
      BEQ ANVALIDO ;IMPLICA QUE EL ANALISIS ES VALIDO
      JMP NOVALIDO ;BRINCA SI NO LO ES

$01 CMP #59      ;INFORMACION ESTADO BOMBAS
      BNE $02
      LDA NUMDAT03
      CMP #01     ;2 BITS MENOS SIGNIFICATIVOS
      BEQ ANVALIDO
      JMP NOVALIDO

$02 CMP #5A      ;INFORMACION ESTADO DISPOSITIVOS AUXILIARES
      BNE $03
      LDA NUMDAT03
      CMP #02     ;6 BITS MAS S
      BEQ ANVALIDO
      JMP NOVALIDO

$03 CMP #5B      ;INFORMACION CANDADOS Y PARAMETROS
      BNE $04
      LDA NUMDAT03
      CMP #22
      BEQ ANVALIDO
      JMP NOVALIDO

$04 CMP #54      ;BLOQUE DIGITAL
      BNE $05
      LDA NUMDAT03
      CMP #21
      BEQ ANVALIDO
      JMP NOVALIDO

$05 CMP #55      ;BLOQUE ANALOGICO
      BNE $06
      LDA NUMDAT03
      CMP #41
      BEQ ANVALIDO
      JMP NOVALIDO

$06 LDA NUMDAT03 ;EL CASO 56 Y 58 IGUALES EN EL NUMDATOS (INFORMACION
      CMP #08     ;VALVULAS BINARIAS E INFORMACION ESPEJOS)
      BEQ ANVALIDO
      JMP NOVALIDO

ANVALIDO LDA #01  ;ANALISIS VALIDO, ACUMULADOR=1
      RTS

NOVALIDO LDA #00  ;ANALISIS NO VALIDO, ACUMULADOR=0

```

;SUB #22 GUARDAR

GUARDAR LDA ALMACEN ;FIJARSE QUE YA ESTA NEGADO,DEBIDO A LA RECEPCION

BEQ #01

LDX #00 ;PARA APUNT2

JMP #02

\$01 LDX #02 ;PARA APUNT2

\$02 CLC

LDA APUNT2,X ;ES EL ACCESO AL ALMACEN

STA INDIR2 ;EL DOMICILIO DE ACCESO BYTE MENOS SIGNIFICATIVO

ADC #03

STA INDIR2B ;TIENE EL DOMICILIO DE ACCESO MENOS SIG. DE BUFFERX+3

INX

LDA APUNT2,X

STA INDIR2+1 ;BYTE MAS SIGNIFICATIVO DE BUFFERX

ADC #00 ;POR SI OCURRIO UN ACARREO AL SUMAR 3

STA INDIR2B+1 ;BYTE MENOS SIGNIFICATIVO DE BUFFERX+3

;PREPARACION DE APUNTADES Y LIMITES SUPERIORES

LIMITES LDY #00

LDA @INDIR2,Y ;(INDIR2),Y, ES EL BYTE DE INICIO DE BLOQUE

CMF #56 ;VALVULAS BINARIAS

BNE #01

LDX #00 ;PARA MANDARLO AL

JSR APUNTES ;BUFFER3

LDX #00 ;PARA APLIM=8

JSR CASOS ;TOMA DEL BUFFERX Y LO MANDA A BUFFER3

JSR DESECHAR ;DE LAS 16 VALVULAS CONSIDERA 10

RTS

\$01 CMF #57 ;VALVULAS ANALOGICAS

BNE #02

LDX #02 ;LO MANDARA A

JSR APUNTES ;AINFORM

LDX #01 ;PARA EL APUNTADES LIMITE=06

JSR CASOS ;TOMA DEL BUFFERX, LO MANDA A AINFORM

RTS

\$02 CMF #58 ;ESPEJOS

BNE #03

LDX #04 ;LO MANDARA A

JSR APUNTES ;IESPEJOS

LDX #02 ;PARA APLIM=8

JSR CASOS ;TOMA DEL BUFFERX, LO MANDA A IESPEJOS

RTS

\$03 CMF #5B ;CANDADOS Y PARAMETROS

BNE #04

LDX #06 ;LO MANDARA A

```

JSR APUNTD5 ;PARRELOJ

LDX #03 ;APLIM=0E
JSR CASOS ;TOMA DEL BUFFERX, LO MANDA A PARRELOJ

LDX #08 ;LO MANDARA A
JSR APUNTD5 ;ACANDADO

LDX #04 ;APLIM=12
LDA #00
STA INDYD ;INDICE DONDE COMIENZA A DEPOSITAR

LDY #0E ;INDICE DE DONDE TOMARA LOS BYTES
JSR CASOS2 ;TOMA DEL BUFFERX, LO PASA A ACANDADO

LDX #0A ;LO MANDARA A
JSR APUNTD5 ;DCANDADO

LDX #05 ;APLIM=22
LDA #00
STA INDYD ;INDICE DONDE COMIENZA A DEPOSITAR
LDY #12 ;INDICE DE DONDE TOMARA LOS BYTES
JSR CASOS2 ;TOMA DEL BUFFERX, LO PASA A DCANDADO
RTS

$04 CMP #59 ;BOMBAS
BNE #05
LDY #00 ;PARA TOMAR EL PRIMER ELEMENTO DE BUFFERX+3

LDA IBOMDISA
AND #0FC
STA IBOMDISA ;BORRA LOS 2 BITS - SIG

LDA @INDIR2B,Y ;(INDIR2B),Y ES EL BUFFERX+3
AND #03 ;2 BITS MENOS SIG.
CLC
ADC IBOMDISA ;LOS ABADE A LO ANTERIOR
STA IBOMDISA ;ALMACENA EL NUEVO RESULTADO
RTS

$05 CMP #5A ;DISP AUX
BNE #06

LDY #00 ;PARA TOMAR EL PRIMER ELEMENTO DE BUFFERX+3
LDA @INDIR2B,Y
AND #0FC ;SE BORRAN BITS 0 Y 1
STA NIBBLE1 ;ALMACENA

LDY #01 ;PARA EL SIGUIENTE ELEMENTO DE BUFFERX+3
LDA @INDIR2B,Y ;LOS SIG 4 BITS, SE TOMARAN EN CUENTA
STA NIBBLE2
JSR COMPACT

STA TEMP5 ;GUARDA LO COMPACTADO
LDA IBOMDISA
AND #03 ;SE BORRAN 6 BITS MAS SIG. SE ALMACENARA
STA IBOMDISA ;EN ESE LUGAR NUEVOS VALORES

LDA TEMP5 ;SE TIENEN LOS NUEVOS VALORES
CLC

```

```

ADC IBOMDISA ;SE ADICIONA A LOS BITS 0,1 NO ALTERADOS
STA IBOMDISA ;SE ALMACENA EL RESULTADO
RTS

%06 CMP #54 ;BLOQUE DIGITAL
   BNE #07

LDA REGDIG ;SI NO EXISTEN REGISTROS DIGITALES
BEQ #08 ;TERMINA LA RUTINA

LDX #10 ;EL BLOQUE DIGITAL SE DEPOSITARA
JSR APUNDES ;APARTIR DE IVALDIG
JSR CASOS3 ;LO TOMA DEL BUFFERX, DEPOSITA EN IVALDIG
RTS

%07 LDA REGANA ;BLOQUE ANALOGICO
   BEQ #08 ;SI NO HAY REGISTROS ANALOGICOS, ENTONCES FIN
LDX #02 ;EL BLOQUE ANALOGICO
JSR APUNDES ;COMIENZA A PARTIR DE AINFORM
JSR CASOS3 ;TOMA DEL BUFFERX, A AINFORM
%08 RTS

;*****
;SUB #23 CASOS
;RUTINA QUE TOMA LOS DATOS DEL BUFFER DE RECEPCION Y LOS COMPACTA Y ALMACENA
;EN SITIOS PRESTABLECIDOS
CASOS LDA #00
      STA INDY0 ;INDICE Y DEL ORIGEN
      STA INDYD ;INDICE Y DEL DESTINO
      LDY #00 ;PARA EL ORIGEN

CASOS2 LDA @INDIR2B,Y ;(INDIR2B),Y
      STA NIBBLE1 ;PRIMER BYTE PARA COMPACTAR

      INY ;PARA TOMAR EL SIGUIENTE BYTE
      LDA @INDIR2B,Y
      STA NIBBLE2 ;SEGUNDO BYTE PARA COMPACTAR
      JSR COMPACT ;SE COMPACTAN

      STY INDY0 ;SE SALVA EL INDICE DEL ORIGEN
      LDY INDYD ;TOMA EL INDICE DE DESTINO
      STA @INDIR,Y ;LO COMPACTADO ES DEPOSITADO EN (INDIR),Y

      INC INDY0 ;PARA EL SIGUIENTE BYTE A DESCOMPACTAR
      INC INDYD ;SIGUIENTE LUGAR A DEPOSITAR

      LDA INDY0 ;PARA COMPARAR CON
      CMP APLIM,X ;NUMERO MAXIMO DE ELEMENTOS A TOMAR
      BEQ #01 ;SI INDY0=APLIM ENTONCES FIN

      LDY INDY0 ;PARA TOMAR EL SIGUIETE BYTE
      JMP CASOS2 ;REGRESARA A CASOS2
%01 RTS

;*****
;SUB #24 COMPACT
;COMPACTA DOS BYTES DE BUFFER DE RECEPCION E UNO PARA ALMACENAR
COMPACT LDA NIBBLE1
      AND #0F ;DEJA LO MENOS SIGNIFICATIVO
      STA NIBBLE1 ;ALMACENANDO EL RESULTADO

```

```

LDA NIBBLE2
AND #0F ;TOMA LO MENOS SIGNIFICATIVO
ASL A ;LO RECORRE A LA IZQUIERDA 4
ASL A ;LUGARES PARA QUE QUEDE EN EL
ASL A ;SITIO MAS SIGNIFICATIVO
ASL A
ADC NIBBLE1 ;LO SUMA, QUEDANDO EL RESULTADO EN EL ACUMULADOR
RTS

```

```

;*****

```

```

;SUB #25 DESECHAR

```

```

;PARA DESECHAR 6 DE LAS 16 VALVULAS

```

```

DESECHAR LDX #00

```

```

JSR APUNTES ;LOCALIZA EL DOMICILIO DE BUFFER3

```

```

LDY #00 ;PARA TOMAR SU PRIMER ELEMENTO
LDA @INDIR,Y ;TOMARA EN CUENTA LAS
AND #0C3 ;DOS PRIMERAS VALVULAS,BITS 0,1 Y 6,7
STA IVALDIG ;QUEDAN EN POSICION DEFINITIVA BITS 0,1

```

```

LSR A ;SE RECORRERAN 4 LUGARES A LA DERECHA
LSR A ;LOS BITS 6 Y 7
LSR A ;PARA QUE OCUPEN LOS LUGARES 2,3
LSR A ;DE IVALDIG
ADC IVALDIG
AND #0F ;SE BORRA EL NIBBLE MAS SIG
STA IVALDIG ;QUEDA 0X, DOS VALVULAS COLOCADAS EN SU SITIO

```

```

INY ;PARA TOMAR EL SIGUIENTE REGISTRO
LDA @INDIR,Y ;BUFFER3+1
AND #3C ;SE TOMA EN CUENTA LAS DOS SIG VALVULAS. BITS 2,3,4,5
ASL A ;SE RECORREN A LA IZQUIERDA DOS LUGARES PARA QUE OCUPEN
ASL A ;LOS BITS 4,5,6,7 DE IVALDIG
ADC IVALDIG ;SE JUNTA CON LO ANTERIOR
STA IVALDIG ;REAGRUPADAS 4 PRIMERAS VALVULAS

```

```

INY ;PARA EL SIGUIENTE BYTE
LDA @INDIR,Y ;BUFFER3+2
AND #03 ;TOMA LOS BITS 0,1
STA IVALDIG+1 ;QUEDARAN EN POSICION 0,1 DE IVALDIG+1

```

```

LDA @INDIR,Y ;BUFFER3+2
AND #0F0 ;CONSIDERA BITS 4,5,6,7
LSR A ;LO RECORRE 2 LUGARES A LA DERECHA
LSR A ;PARA QUE QUEDEN EN LA POSICION 2,3,4,5 DE IVALDIG+1
ADC IVALDIG+1 ;SE JUNTAN CON LOS BITS 0,1 DE IVALDIG+1
ASL A ;SE RECORREN DOS LUGARES A LA IZQUIERDA
ASL A ;CON EL OBJETO DE PREPARARSE PARA
STA IVALDIG+1 ;RECIBIR LOS SIGUIENTES 2 BITS

```

```

INY ;PARA EL SIGUIENTE BYTE
LDA @INDIR,Y ;BUFFER3+3
LSR A ;SE RECORRE UN LUGAR A LA DERECHA,
STA @INDIR,Y ;PASANDO EL BIT 0 AL CARRY, ALMACENANDOSE EN BUFFER3+3

```

```

ROR IVALDIG+1 ;SE ROTA UN LUGAR A LA DERECHA,PASANDO EL CARRY AL BIT 7
LDA @INDIR,Y ;SE TOMA EL BUFFER3+3
LSR A ;SE RECORRE OTRO LUGAR A LA DERECHA,PASANDO EL BIT 0 AL CARRY
STA @INDIR,Y ;SE ALMACENA EN BUFFER3+3

```

ROR IVALDIG+1 ;ROTA A LA DERECHA PASANDO EL CARRY AL BIT 7
AND #33 ;SE CONSIDERAN LOS BITS 0,1,4,5 DE LO ROTADO
STA @INDIR,Y ;ALMACENANDOSE EN BUFFER3+3, LOS BITS RESTANTES SE BORRA

LSR A ;SE RECORREN 2 LUGARES A LA DERECHA PARA
LSR A ;PASAR LOS BITS 4,5 AL 2,3
CLC
ADC @INDIR,Y ;SE ADICIONA AL BUFFER3+3
AND #0F ;PARA QUE SOLO QUEDEN BITS 0,1,2,3
STA IVALDIG+2 ;CONCLUIDAS ULTIMAS 2 VALVULAS
RTS

;
;
;CONTINUARA EN CONS
;
;

```

;*****
;
;
;SEGMENTO CONS
;SEGMENTO ANTERIOR:CON4
;
;
;*****
;SUB #26 CASOS3
;PARA BLOQUE DIGITAL Y ANALOGICO COMPLETOS
CASOS3 LDA #00
      STA INDY0 ;INDICE Y DEL ORIGEN
      STA INDYD ;INDICE Y DEL DESTINO
      LDY #00 ;PARA EL ORIGEN
CASOS4 LDA @INDIR2B,Y
      STA NIBBLE1
      INY
      LDA @INDIR2B,Y
      STA NIBBLE2
      JSR COMPACT
      STY INDY0
      LDY INDYD
      STA @INDIR,Y
      INC INDY0
      INC INDYD
      LDA INDY0
      CMP NUMDAT03
      BEQ #01 ;SI INDY0=NUMDAT03 ENTONCES FIN
      LDY INDYD
      JMP CASOS4
      $01 RTS

;*****
;SUB #27 TARPRUE1
TARPRUE1 NOP
NIVELCOM STA SOLINT ;LIMPIA SOLICITUD DE INTERRUPCION
      STA CONTTEC ;LIMPIA CONTADOR DE PRESIONES DE TECLAS

      LDA #00
      STA ENTDAT1
      STA AVANCE
      STA CAMBNIV
      STA CAMINO ;ATENDERA PARA EL NIVEL DE COMANDOS LA INTERRUPCION

      JSR PONNUMS ;PONDRÁ CEROS EN PANTALLA
      STA INTTEC ;AUTORIZA POSIBLE INTERRUPCION POR TECLADO

$01 JSR CAMBIO ;TOMA INFORMACION DEL PIA, LADO A Y B
      CMP #00
      BEQ #02 ;BRINCA SI NO HUBO CAMBIO DE MODO DE OPERACION
      JMP PASO2A ;BRINCA SI HUBO CAMBIO DE MODO DE OPERACION

$02 LDA CAMBACPR ;DEFINIDA POR CAMBIO, SI NO HUBO CAMBIO
      BEQ #03 ;DE ACOPLAMIENTO DE PRUEBA SIGUE ENCUESTANDO
      JMP SINACPR2 ;SI LO HUBO HARA LAS PRUEBAS SIN ACOPLAMIENTO DE PRUEBAS

$03 LDA AVANCE ;ESPERA LA PRESION
      BEQ #01 ;DE UNA TECLA, EN CUYO CASO, NO BRINCA

      LDA ENTDAT1 ;SE TIENE EL DATO APROBADO POR EXEC

```



```

CMP #0A ;PREGUNTA SI SE LE INDICO TIPO DE PRUEBAS ANALOGICAS
BNE NIVEL2D ;SI NO FUE ASI IRA A TIPO DE PRUEBAS DIGITALES

NIVEL2A LDA #00
STA AVANCE ;SE REDEFINE
STA DIG1 ;SE LIMPIAN PARA
STA DIG2 ;LA RUTINA PROCESAT
STA CAMBNIV

LDA #02 ;PARA LA ATENCION DE LA INTERRUPCION
STA CAMINO ;EN EN NIVEL DE CANAL ANALOGICO

LDA #21
STA COTSUP ;# DE CANAL ANALOGICO+1, MAXIMO PERMITIDO

#01 JSR CAMBIO
CMP #00
BEQ #02
JMP PASO2A

#02 LDA CAMBACPR
BEQ #03
JMP SINACPR2

#03 LDA AVANCE ;ESPERA LA PRESION DE UNA TECLA
BEQ #01 ;EN CUYO CASO NO BRINCA

LDA CAMBNIV ;POSIBILITA IRSE AL NIVEL DE COMANDOS
CMP #01
BEQ NIVELCOM

LDA NUCAN ;VALOR QUE ENTREGA PROCESAT
STA NUCANA ;SE USARA PARA LA SALIDA DE VALORES A TARJETAS

NIVEL3A LDA #00
STA AVANCE
STA DIG1
STA DIG2
STA VACANA

LDA #04
STA CAMINO ;IRA A LA INTERRUPCION REFERIDA A VALOR ANALOGICO

#01 JSR CAMBIO
CMP #00
BEQ #02
JMP PASO2A

#02 LDA CAMBACPR
BEQ #03
JMP SINACPR2

#03 LDA AVANCE
BEQ #01

LDA CAMBNIV ;SI VALE UNO
CMP #01 ;CAMBNIV RETROCEDERA
BEQ NIVEL2A ;A NIVEL DE CANAL ANALOGICO

LDX NUCANA ;EL OPERADOR TRABAJA A PARTIR DEL CANAL 1

```

DEX ;EL SISTEMA TRABAJA A PARTIR DEL CANAL CERO

LDA VACANA ;TOMA EL VALOR DADO POR EL OPERADOR VIA INTERRUPCION
STA EANALOG,X

LDA #01
STA VACIADO ;HABILITA PARA QUE VACIE
JSR SALTARAN ;SALIDA DE ESTE VALOR A TARJETA ANALOGICA

LDA #00
STA VACIADO ;DESHABILITA EL VACIADO A TARJETA
JMP NIVEL3A ;TOMARA OTRO VALOR DEL CANAL

;NIVEL DIGITAL

NIVEL2D LDA #00
STA AVANCE
STA DIG1
STA DIG2
STA CAMBENV

LDA #02 ;ATENDERA INTERRUPCION EN LA ZONA DE CANAL DIGITAL
STA CAMINO

LDA #81 ;MAXIMO NUMERO DE CANAL+1
STA COTSUP ;A ACEPTAR

\$01 JSR CAMBIO
CMP #00
BEQ \$02
JMP PASO2A

\$02 LDA CANSACPR
BEQ \$03
JMP SINACPR2

\$03 LDA AVANCE
BEQ \$01
LDA CAMENV ;SI CAMBIO DE NIVEL VALE 1
CMP #01
BNE NIVEL3D ;IRA AL NIVEL DE VALOR DEL CANAL
JMP NIVELCOM ;SI NO, RETROCEDERA AL NIVEL DE COMANDOS

NIVEL3D LDA NUCAN
STA NUCANDIG ;SALVA EL VALOR OBTENIDO EN PROCESAT

LDA #00
STA AVANCE
STA ENTDAT1
STA VACAND

LDA #06 ;ATENDERA INTERRUPCION EN
STA CAMINO ;EL VALOR DEL CANAL DIGITAL

\$01 JSR CAMBIO
CMP #00
BEQ \$02
JMP PASO2A

\$02 LDA CANSACPR
BEQ \$03

JMP SINACPR2

\$03 LDA AVANCE
BEQ \$01

LDA CAMBIV ;SI VALE 1 IMPLICA QUE
CMP #01 ;RETROCEDERA A
BEQ NIVEL2D ;NIVEL DE NUMERO DE CANAL DIGITAL
JSR POSD ;SE DEFINE EL NUEVO VALOR DE UNO DE LOS BITS DEL REGISTRO

LDA #01
STA VACIADO ;HABILTA A SALIDA DE TARJETAS DIGITALES
JSR SALTARD ;EFECTUA LA SALIDA

LDA #00
STA VACIADO ;DESHABILITA EL VACIADO
JMP NIVEL3D ;VOLVERA A PREGUNTAR POR UN NUEVO VALOR DE CANAL DIGITAL

;SUB #28 PONNUMS

PONNUMS LDA #01

STA SELDIG1 ;EL PRIMER DIGITO DE LA PANTALLA SERA ACTIVADO

PONNUMS1 JSR PONCAR ;PONE EL CARACTER EN PANTALLA
ASL SELDIG1 ;EL SIGUIENTE DIGITO SERA ACTIVADO

LDA SELDIG1 ;COMPARA EL NUMERO DE DIGITO A ACTIVAR
CMP #40 ;CON EL MAXIMO POSIBLE, ES EL SEXTO
BCC PONNUMS1 ;SI NO HA LLEGADO A ESTE REPITE EL CICLO
RTS

;SUB #29 PONCAR

PONCAR LDA ENTDAT1 ;DEFINE EL CARACTER A DESPLEGAR
STA ENTDAT ;ACTIVA EL CARACTER EN LA PANTALLA

LDA #00
STA SELDIG ;BORRA PRIMERO EL DS

LDA SELDIG1
STA SELDIG ;ACTIVA LA POSICION QUE OCUPARA

LDA #00
STA SELDIG ;APAGA LA SEBAL DE POSICIONAMIENTO
RTS

;SUB #30 CAMBIO

CAMBIO JSR LEEPIAA ;TOMA INFORMACION DEL PIA "A"
CMP DETCAMMO ;COMPARA CON EL MODO ANTERIOR DE CONFIGURACION
BEQ \$01 ;BRINCA SI NO HUBO CAMBIO

LDA #01 ;SI HUBO CAMBIO DEVUELVE ESTE VALOR
RTS

\$01 LDA PDRB ;TOMA DEL PIA "B"
AND #87 ;LOS DIGITOS DE CONTROL DE PRUEBAS
STA PUERTOB

AND #30 ;PREGUNTA QUE TIPO DE PRUEBA SE PIDE

```

CMP #80      ;SI ES CON ACOPLAMIENTO
BEQ #02      ;BRINCARA A #02
JSR SINACPR1 ;SI NO, INICIA PRUEBAS SIN ACOPLAMIENTO

LDA #01
STA CABCACPR ;VARIABLE DE SALIDA, INDICA QUE SERA SIN ACOPLAMIENTO
LDA #00      ;VALOR DE RETORNO, NO HUBO CAMBIO DE MODO
RTS

$02 LDA #00      ;NO HUBO CAMBIO DE MODO Y
STA CABCACPR ;SIGUE CON ACOPLAMIENTO DE PRUEBAS
RTS

;*****
;SUB #31 POSD
;RUTINA QUE POSICIONA VARIABLES DIGITALES
POS D SEC      ;LAS SIG LINEAS SON PARA DIVIDIR /8
LDA #00
STA COCIENTE ;REGISTRO A MODIFICAR

LDA NUCANDIG ;CANAL A MODIFICAR
SBC #01      ;EL SISTEMA TRABAJA A PARTIR DEL CANAL 0

$01 STA RESIDUO ;INDICA EL # DE BIT A MODIFICAR (0-7)
SBC #08      ;PARA QUE SOLO QUEDE EL NUMERO DE BIT A MODIFICAR
BCC #02      ;FIN DE RESTA
INC COCIENTE ;INDICA EL REGISTRO A MODIFICAR, EN CADA RESTA
JMP #01

$02 LDX COCIENTE ;POSICION EN LA TABLA DEL REGISTRO HAY QUE MODIFICAR
LDA EDDODIG,X
STA MODVALD ;EL REGISTRO DIG A MODIFICAR

JSR VALBIT ;EL BIT SE ENCENDERA O APAGARA SEGUN VACAND
STA EDDODIG,X ;VALOR DIG MODIFICADO
$04 RTS

;*****
;SUB #32 VALBIT
VALBIT LDA #01
STA SUSTRAS ;ENCIENDE EL BIT 0

LDA RESIDUO
STA RESIDUO1 ;DEFINE EL BIT QUE SE QUEDARA ENCENDIDO EN SUSTRAS
BEQ RESTA ;SI BRINCA QUEDARA ENCENDIDO EL BIT 0 DE SUSTRAS

$01 ASL SUSTRAS ;SE CORRE A LA IZQUIERDA EL BIT ENCENDIDO
DEC RESIDUO1 ;SE DECREENTA EL CONTADOR DE CORRIMIETOS

LDA RESIDUO1
BNE #01 ;PREGUNTA SI VUELVE A HACER EL CORRIMIETOS

RESTA SEC      ;EN LA OPERACION DE RESTA, HAY QUE COMPLEMENTAR
LDA #0FF      ;DE ESTE VALOR, SE APAGARA SOLO UN BIT
SBC SUSTRAS ;SUSTRAS INDICA QUE BIT SE MODIFICARA, EL CARRY ESTA NEG.
STA RESULT ;INDICA QUE BIT SE MODIFICARA DE MODVALD

LDA MODVALD ;SE APAGARA UN BIT
AND RESULT

```

```

STA RESULT ;SE TIENE MODWALD CON EL BIT APAGADO
LDA VACAND ;INDICA SI SE QUEDARA ASI O NO
BEQ #02 ;BRINCA PARA QUEDA APAGADO EL BIT

LDA RESULT ;PARA PRENDER EL BIT
CLC
ADC SUSTRA ;SUSTRA ENCIENDE EL BIT DE RESULT
RTS ;EL RESULTADO ESTA EN EL ACUMULADOR

#02 LDA RESULT
RTS

;*****
;SUB #33 SINACPR1
;SUBROUTINA DE INICIALIZACION PARA CORRERSE LA PRIMERA VEZ O SI HAY
;UN CAMBIO DE NIVEL (A/D O D/A)
SINACPR1 LDA PUERTOB
STA REFERENC

LDA #00
STA BANDERAS
STA COCIENTE
STA RESIDUO

LDX #00
STA INDX3
STA NOINTEC ;NO PERMITE INTERRUPCION POR EL ACOPLAMIENTO DE PRUEBAS
RTS

;*****
;SUB #34 SINACPR2
SINACPR2 LDX INDX3 ;DESCONTAMINA EL REGISTROX
LDA REFERENC ;VALOR ANTERIOR DEL CUAL
AND #04 ;ANALIZARA EL BIT 2 QUE
STA TEMP4 ;ALMACENARA EN LA VARIABLE DE TRABAJO

LDA PUERTOB ;VALOR ACTUAL AL QUE SE APARTA EL BIT 2
AND #04
CMP TEMP4 ;SI EL VALOR ACTUAL ES IGUAL AL ANTERIOR
BEQ #01 ;SI NO HUBO CAMBIOS INVESTIGA SI HUBO CAMBIO DE CANAL
JSR SINACPR1 ;ESTO INFLICA QUE HUBO EN EL TIPO DE CANAL
;A PROBAR, SE INICIA TODO DE NUEVO
#01 LDA PUERTOB ;DEL VALOR ACTUAL
AND #04 ;SE ANALIZARA B2
CMP #04 ;PREGUNTARA SI ES DIGITAL
BNE ANALOG ;SI NO LO ES BRINCARA A ANALOG
JMP DIGITAL ;SI NO IRA A DIGITAL

ANALOG LDA REFERENC ;PARA DETECTAR CAMBIOS DE CANAL
AND #02 ;SE APARTA EL BIT 1 DEL VALOR ANTERIOR
STA TEMP4

LDA PUERTOB ;SE TOMA EL VALOR ACTUAL
AND #02 ;Y SE APARTA EL BIT 1
CMP TEMP4 ;SI EL VALOR ACTUAL ES IGUAL AL ANTERIOR
BEQ #03 ;BRINCARA A DETECTAR INDICACION DE CAMBIO DE VALORES

LDA #00 ;SI NO REINICIA TODO CON
STA BANDERAS ;VALOR INICIAL DE SALIDA DEL CANAL (0)
INX ;INCREMENTA X PARA COMPARARLO CON EL

```

```

CPX REGANA ;REGISTRO ANALOGICO MAXIMO POSIBLE
BNE #02 ;DE SALIDA, SI NO ES IGUAL BRINCA A FINAL, SI ES IGUAL
LDX #00 ;SE INDICA QUE DEBE COMENZAR DE NUEVO POR EL PRIMER REG.

$02 JMP FINAL

$03 LDA REFERENC ;SENSIBLE AL CAMBIOS DE VALOR
AND #01 ;DEL BIT CERO DEL PIA, LADO B
STA TEMP4 ;QUE LE INDICARA QUE DEBERA

LDA PUERTOB ;SALIR OTRO VALOR POR EL CANAL
AND #01 ;COMPARANDO EL VALOR ANTEPENULTIMO CON
CMP TEMP4 ;CON EL ULTIMO, SI SON DIFERENTES
BNE #04
JMP FINAL2 ;NO HACE NADA

$04 LDA BANDERAS ;ESTO DECIDE QUE VALOR DAR A LA SALIDA
CMP #01
BNE #05

LDA #7F ;SI VALE UNO SALDRA EL VALOR 7F
JSR SALIDAA ;LA SALIDA A TARJETAS OJO,SE ALTERA EL REG INDICE

LDA #02 ;PARA INDICAR QUE EN LA SIGUIENTE ITERACION SALDRA
STA BANDERAS ;OTRO VALOR
JMP FINAL

$05 LDA BANDERAS ;SI ES IGUAL A 02
CMP #02 ;SALDRA EL VALOR FF
BNE #06 ;SI NO ES IGUAL ENTONCES SALDRA EL VALOR 0

LDA #OFF
JSR SALIDAA

LDA #00
STA BANDERAS
JMP FINAL

$06 LDA #00
JSR SALIDAA

LDA #01 ;INDICA QUE SALDRA EL VALOR 7F EN LA
STA BANDERAS ;SIGUIENTE ITERACION

FINAL LDA PUERTOB ;EL VALOR ACTUAL SE COLOCA
STA REFERENC ;COMO EL ULTIMO, ESPERA A RECIBIR OTRO

FINAL2 STX INDX3 ;PARA CONSERVAR SU VALOR PARA LA SIG. ITERACION
JSR LEEPIAA ;LEE VALOR ACTUAL DEL PIA "A"
CMP DETCAMMO ;LO COMPARA CON EL MODO ANTERIOR DEFINIDO
BEQ #01 ;SI HUBO CAMBIO INICIA EL PROGRAMA PRINCIPAL
JMP PASO2A

$01 LDA PDRB
AND #87 ;CONSIDERA BITS 7,2,1 Y 0 DEL PIA "B"
STA PUERTOB

AND #80 ;PREGUNTA SI SE LE PIDE PRUEBAS SIN
BEQ #02 ;ACOMPLIMIENTO O CON EL
JSR TARPRUE1

```

```

#02 JMP SINACPR2

DIGITAL LDA REFERENC ;VALOR ANTERIOR
        AND #02      ;EXAMINARA BIT 1
        STA TEMP4

        LDA PUERTOB ;VALOR ACTUAL
        AND #02      ;DEJA SOLO EL BIT 1
        CMP TEMP4    ;SI NO SE INDICO CAMBIO DE CANAL
        BEQ DETCAMVA ;SE IRA A DETCAMVA

        INC RESIDUO  ;INDICA QUE SE MODIFICARA EL SIGUIENTE BIT DEL REGISTRO
        LDA RESIDUO
        CMP #08      ;SI NO SE INDICA QUE ES EL BIT 8
        BNE #01      ;BRINCA A #01

        LDA #00      ;DESPUES DEL BIT 7 SEGUIRA EL 0
        STA RESIDUO  ;DE NUEVO
        INX
        INC COCIENTE ;CONTINUARA CON EL SIGUIENTE REGISTRO

        LDA COCIENTE
        CMP REGDIG   ;LIMITE MAXIMO DE REGISTROS
        BNE #01
        LDX #00      ;INDICE 0 PARA OTRA RONDA

        LDA #00
        STA COCIENTE ;COMIENZA PRIMER REGISTRO DE NUEVO
#01 JMP FINAL

```

```

DETCAMVA LDA REFERENC ;VALOR ANTERIOR
         AND #01
         STA TEMP4    ;EXAMINA EL BIT 0

         LDA PUERTOB ;VALOR ACTUAL DEL
         AND #01      ;QUE SE TOMARA EL BIT 0
         CMP TEMP4    ;SI NO ES IGUAL CAMBIARA EL VALOR
         BNE #01
         JMP FINAL2   ;SI ES IGUAL NO HARA NADA

```

```

;CAMBIA LOS VALORES
#01 LDA PUERTOB
    AND #01
    STA VACAND ;VALOR DE SALIDA, ENTRADA PARA VALBIT

    LDA EDODIG,X
    STA MODVALD ;VALOR QUE SE MODIFICARA, ENTRADA PARA VALBIT
    JSR VALBIT ;SE ENTREGA EL REGISTRO CON EL BIT MODIFICADO

    STA EDODIG,X
    JSR SALIDAD ;SALIDA A TARGETAS DIG OJO;SE ALTERA REG IND
#02 JMP FINAL

```

```

;*****

```

```

;SUB #35 SALIDAA

```

```

SALIDAA STX INDX3 ;INDICA EL REGISTR A MODIFICAR
        STA EANALOG,X ;LO COLOCA

```

```

LDA #01
STA VACIADO ;HABILITA A SALTARAN

```

JSR SALTARAN ;SE EFECTUA LA SALIDA DE DATOS A TARJETAS

LDA #00
STA VACIADO ;DESHABILITA A SALTARAN
LDX INDX3 ;RECUPERA EL VALOR X, YA QUE
RTS ;MODIFICADO POR SALTARAN

;SUB #36 SALIDAD
SALIDAD STX INDX3 ;REGISTRO DE TABLA A MODIFICAR
LDA #01
STA VACIADO ;HABILITA A SALTARD

JSR SALTARD ;SALEN LOS DATOS DIGITALES
LDA #00
STA VACIADO ;DESHABILITA A SALTARD
LDX INDX3 ;SE DESCONTAMINA EL REGISTRO X
RTS

;
;
;CONTINUARA EN:CON6

```

*****
;
;
;SEGMENTO CONS
;SEGMENTO ANTERIOR:CONS
;
;
*****
;SUB #37 RECDTBM
RECDTBM NOF
REGREXY LDX INDY4 ;REGRESA REG IND DE LA ULTIMA ITERACION
        LDY INDY4

        LDA RECHAZO
        BEQ ENTRADA ;SI NO HAY RECHAZO ADELANTE
        JMP RECHAZAR ;ACEPTA SOLO UN CARACTER INICIO DE BLOQUE

ENTRADA LDA ERRORAC
        BEQ $01 ;NO HAY ERRORES DE RECEPCION

        STA MALBLOQ ;HAY ERROR EN RECEPCION
$01 LDA PRINBLOQ
        BNE CARPREV ;YA HUBO CARACTERES PREVIOS

        LDA CARACTER
        CMP #54 ;RECIBE EL CARACTER INICIAL DE CONTROL
        BCC NOINIBLO ;A<54
        CMP #5E
        BEQ ENTRADA1
        BCS NOINIBLO ;A>5E

ENTRADA1 LDA #01
        STA PRINBLOQ ;LLEGO EL PRIMER CARACTER
        JMP ALMACENA

NOINIBLO LDA MALBLOQ
        BNE $01 ;NO DETECTO ERROR EN ACIA
        JMP RECHAZAR ;DETECTO ERROR EN ACIA,RECHAZAR ESTO ESPERANDO
        ;INICIO DE BLOQUE
$01 LDA #01 ;LO ASUME COMO BUENO,ESTE PRIMER CARACTER RECIBIDO
        STA MALBLOQ
        STA PRINBLOQ

ALMACENA LDX #00 ;INDICE BUFFER1
        LDY #00 ;INDICE BUFFER2
        JSR RALMACEN ;ALMACENA EL BYTE DE CONTROL, EN BUFFER1 O BUFFER2
        INC ALMCONT ;INCREMENTA EL CONTADOR DEL ALMACEN
        JMP RETORNO

CARPREV LDA NUMDAT01 ;SI NO EXISTEN NIGUN
        AND NUMDAT02 ;BYTE DE DATO EL RESULTADO ES CERO
        BEQ $01 ;IRA A REGISTRARLO COMO NUMERO DE DATO
        JMP FILTRO1 ;SI NO, ES BYTE DE DATO,FIN DE BLOQUE O INICIO

$01 LDA CARACTER
        CMP #50
        BCC $02 ;A<50
        JMP RECHAZAR ;# DE DATO MAL DEFINIDO

$02 CMP #40

```

```

      BEQ NUMDAT
      BCS NUMDAT ;A>=40 ES NUMERO DE DATO
      JMP RECHAZAR ;# DE DATO MAL DEF.

NUMDAT LDA ERRORAC ;SI NO HAY ERROR DE
      BEQ $01 ;RECEPCION BRINCA
      JMP RECHAZAR ;# DATO INCORRECTO

$01 LDA NUMDAT01 ;SI YA RECIBIO EL PRIMER BYTE
      CMP #01 ;NUMERO DE DATO ENTONCES
      BEQ BYTE2 ;ES EL SEGUNDO BYTE DE DATOS

      LDA #01 ;SI NO, INDICA QUE
      STA NUMDAT01 ;RECIBE EL PRIMER BYTE
      JSR RALMACEN ;Y LO ALMACENA

      INC ALMCONT ;INCREMENTA EL CONTADOR DE ALMACEN
      JMP RETORNO

BYTE2 LDA #01 ;ACUSE DE RECIBO
      STA NUMDAT02 ;DEL SEGUNDO BYTE DE NUMERO DE DATOS
      JSR RALMACEN ;Y LO ALMACENA
      INC ALMCONT ;INCREMENTA EL CONTADOR DE ALMACEN

;PARA COMPACTAR Y GUARDAR EN EL ALMACEN ADECUADO
      LDA ALMACEN ;SI VALE UNO EL INDICADOR DE ALMACEN
      BNE $02 ;LO TOMARA DEL BUFFER2

      LDA BUFFER1+1 ;TOMA EL PRIMER BYTE DEL NUMERO DE DATOS
      STA NIBBLE1R ;Y LO PONE EN VARIABLE DE ENTRADA

      LDA BUFFER1+2 ;TOMA EL SEGUNDO BYTE DEL NUMERO DE DATOS
      STA NIBBLE2R ;LO PONE EN VARIABLE DE ENTRADA
      JSR COMPACTR ;COMPACTA ESTOS 2 BYTES, QUEDANDO
                  ;EN EL ACUMULADOR LO COMPACTADO

      CMP #40
      BEQ $01
      BCC $01 ;SI A<=40
      JMP RECHAZAR ;EXEDE EL BUFFER, RECHAZANDOLO

$01 STA BUFFER1+44 ;SI NO ALMACENA EL #DE DATOS
      STA NUMDAT0
      JMP RETORNO

$02 LDA BUFFER2+1 ;IGUAL QUE LAS LINEAS ANTERIORES
      STA NIBBLE1R ;SOLO CAMBIA BUFFER1 POR BUFFER2

      LDA BUFFER2+2
      STA NIBBLE2R
      JSR COMPACTR
      CMP #40
      BEQ $03
      BCC $03
      JMP RECHAZAR

$03 STA BUFFER2+44
      STA NUMDAT0
      JMP RETORNO

FILTRO1 LDA CARACTER

```

```

AND #0FO      ;BORRA LS 4 BITS MENOS SIGNIFICATIVOS
CMP #50
BEQ CONTROL  ;LO TRATA COMO CONTROL
CMP #40
BEQ DATO     ;LO TRATA COMO DATO
JMP RECHAZAR ;RECHAZA SI NO TIENE BITS DE IDENTIFICACION

CONTROL LDA CARACTER ;PREGUNTA SI ES UN
CMP #5F   ;CARACTER DE FIN DE BLOQUE
BEQ #01   ;SI LO ES, BRINCA A #01

LDA #01   ;SI NO LO ES IMPLICA QUE
STA MALBLOQ ;SE RECIBIO OTRO DE INICIO DE BLOQUE
JMP DATO  ;LO TRATARA COMO CARACTER DE DATO

#01 LDA NUMDATO ;CUANDO SEA UN BLOQUE
BEQ FINRECEP ;SIN DATOS A ESPERAR, BRINCA

LDA DATOCONP ;SI NO, PREGUNTA SI ESTAN Y SON TODOS LOS DATOS
CMP #01
BEQ FINRECEP ;EN CASO QUE ASI SEA BRINCA,

LDA #01   ;SI NO, SE RECIBIO UNO DE FIN
STA MALBLOQ ;DE BLOQUE ANTES DE TIEMPO

DATO LDA NUMDATO ;COMPARA LO QUE SE ESPERA QUE LLEGUE
CMP CONT2 ;CON EL NUMERO DE DATOS QUE LE ESTAN LLEGANDO
BNE #01   ;SI NO SON IGUALES BRINCA A #01
BCS RECHAZAR ;SI SON IGUALES IMPLICA QUE CONT2 SUPERARA A NUMDATO

#01 JSR RALMACEN ;LO ALMACENA EL EL BUFFER
INC ALMCONT ;CONTADOR PARA EL BUFFER
INC CONT2 ;CUENTA LOS DATOS QUE LE LLEGARON

LDA CONT2 ;COMPARA LO QUE LE LLEGO CON
CMP NUMDATO ;LO ESPERADO
BNE #02   ;SI NO SON IGUALES FINALIZARA

LDA #01   ;SI SON IGUALES LO REAL LLEGO A LO ESPERADO
STA DATOCONP ;LEVANTANDO UNA BANDERA

#02 JMP RETORNO

FINRECEP LDA #01 ;PREGUNTA SI HUBO
CMP MALBLOQ ;IRREGULARIDADES
BEQ RECHAZAR
JSR RALMACEN ;SI NO LAS HUBO ALMECENA EL ULTIMO CARACTER RECIBIDO

LDA ALMACEN ;NIEGA LA BANDERA
EOR #01 ;DE ALMACEN PARA
STA ALMACEN ;INDICAR QUE ESTA OCUPADO

LDA #00 ;SE PREPARA PARA OTRA RONDA DE
STA RECHAZO ;RECEPCION

URETORNO LDA #01 ;BLOQUE COMPLETO ACEPTADO
RTS

RETORNO LDA #00 ;BLOQUE SIN CONCLUIR

```

```
SALVAXY STX INDX4 ;SALVA EN VAR AUX SE CONTAMINARAH
STY INDY4
RTS
```

```
RECHAZAR LDA ERRORAC
BEG #01 ;BRINCA SI NO HAY ERROR RECEPCION
JMP #03 ;BRINCA A #03 SI LOS HUBO

#01 LDA CHARACTER
CMP #54
BCC #03 ;SI CHARACTER<54 (MINIMO NUMERO DE ORDEN)
CMP #5E
BEG #02 ;SI CHARACTER=5E (MAXIMO NUMERO DE ORDEN)
BCS #03 ;SI CHARACTER >5E

#02 JSR INICIARI ;INICIALIZA
JMP ENTRADA1 ;PASO LA VARIABLE DE ENTRADA

#03 LDA #01
STA RECHAZO ;SE RECHAZA TODO EL BLOQUE
JMP RETORNO
```

```
;*****
```

```
;SUB #38 RALMACEN
;RUTINA DE ALMACENAJE
RALMACEN LDA ALMACEN ;PREGUNTA A QUE ALMACEN IRAN LOS DATOS
LSR A ;PARA MANDAR EL BIT AL CARRY
BCS #01
```

```
LDA CHARACTER ;TOMA LO RECIBIDO POR EL ACIA
STA BUFFER1,X ;Y LO ALMACENA
INX
RTS
```

```
#01 LDA CHARACTER
STA BUFFER2,Y
INY
RTS
```

```
;*****
```

```
;SUB #39 COMPACTR
;RUTINA DE COMPACTACION
COMPACTR LDA NIBBLE1R
AND #0F
STA NIBBLE1R ;DEJA LO MENOS SIGNIFICATIVO
```

```
LDA NIBBLE2R
AND #0F ;TOMA LO MENOS SIGNIFICATIVO
ASL A ;Y LO RECORRE A LOS LUGARES
ASL A ;MAS SIGNIFICATIVOS
ASL A
ASL A
ADC NIBBLE1R ;LO SUMA, QUEDANDO COMPACTADO
RTS
```

```
;*****
```

```
;SUB #40 APCAMINO
APCAMINO LDA CAMINOS,X
STA INDIR3 ;BYTE MENOS SIGNIFICATIVO DEL DOMICILIO
```

```

INX
LDA CAMINOS,X
STA INDIR3+1 ;BYTE MAS SIGNIFICATIVO
RTS

```

```

;*****

```

```

;SUB #41 PROCESAT
PROCESAT LDA TECLA1 ;DEL CODIGO DE LA TECLA OPRIMIDA
        AND #10 ;CONSIDERA EL BIT 4
        CMP #10 ;DETECTA EXC
        BEQ $02

        LDA TECLA1 ;PARA SEGUIR ENCUESTANDO
        AND #2F ;BORRA LOS BITS 7,6,4
        CMP #2E ;DETECTA SHIFT E
        BNE $01 ;SI NO ES "EXC" NI "SHIFT" "E"
                ;COLOCARA UN CARACTER EN PANTALLA

        LDA #01
        STA CAMBIV ;PARA CAMBIAR DE NIVEL
        STA RAMACAM2 ;PARA LA ATENCION DE INTERRUPCIONES
        STA AVANCE ;SALDRA DEL LOOP
        RTS

```

```

$01 LDA DIG1 ;LO QUE HABIA EN DIG1
    STA DIG2 ;LO PASA A DIG2

    LDA TECLA1 ;PARA COLOCAR EN NUEVO VALOR
    STA DIG1 ;EN DIG1
    STA ENTDAT1 ;VARIABLE DE ENTRADA PARA PONCAR
    JSR PONCAR ;DESPLIEGA LO DE DIG1 EN PANTALLA

    LDA DIG2 ;PARA DESPLEGAR LO DE DIG2
    STA ENTDAT1 ;EN PANTALLA

    LDA #02
    STA RAMACAM2 ;PARA ATENCION DE INTERRUPCIONES
    RTS

```

```

$02 JSR PREPCOMP ;PARA PROCESAR LOS VALORES METIDOS
    STA NUCAN ;NUMERO DE CANAL COMPACTADO

```

```

    LDA #03
    STA RAMACAM2
    RTS

```

```

;*****

```

```

;SUB #42 PREPCOMP
PREPCOMP LDA DIG2 ;CONTIENE EL NIBBLE MAS SIG.
        STA NIBBLE2R

        LDA DIG1 ;TIENE EL NIBBLE MENOS SIG.
        STA NIBBLE1R
        CLC ;PARA COMPACTAR USA LA INSTRUCCION ADC
        JSR COMPACTR ;LOS COMPACTA
        RTS

```

```

;*****

```

```

;SUB #43 LEEPIAA
LEEPIAA LDA PDRA ;LEE EL PERIFERHAL DATA REGISTER A

```

AND #3F ;BORRA LOS BITS 7 Y 6
STA PUERTO4 ;INFORMACION DEL MODO DE OPERACION Y CONFIGURACION
RTS

```
;*****  
.ORG OFFD0  
APUNT .WORD BUFFER3,AIFORM,IESP EJOS,PARRELOJ,ACANDADO  
      .WORD DCANDADO,EANALOG,EDODIGT,IVALIDG  
APUNT2 .WORD BUFFER1,BUFFER2  
APLIM .BYTE 06,06,03,0E,12,22  
CAMINOS .WORD CAMINO1,CAMINO2,CAMINO3,CAMINO4  
.ORG OFFFA  
NMI .WORD RESET  
COMIENSO .WORD RESET  
IRQ .WORD SALVA  
.END
```

:FILE:CONTACT

:TABLA DE VECTORES

.ABSOLUTE

.PROC TABLAS

.ORG 0E000

ROMCANAN .BYTE 00,00,00,00 ;CANDADOS ANALOGICOS, RESERVA 32 BITS,
;CADA BIT ES PARA UN CANAL

ROMCAND .BYTE 00,00,00,00,00,00,00,00 ;RESERVA 16 BYTES. 128 BITS, UN BIT ES
.BYTE 00,00,00,00,00,00,00,00 ;UN CANAL

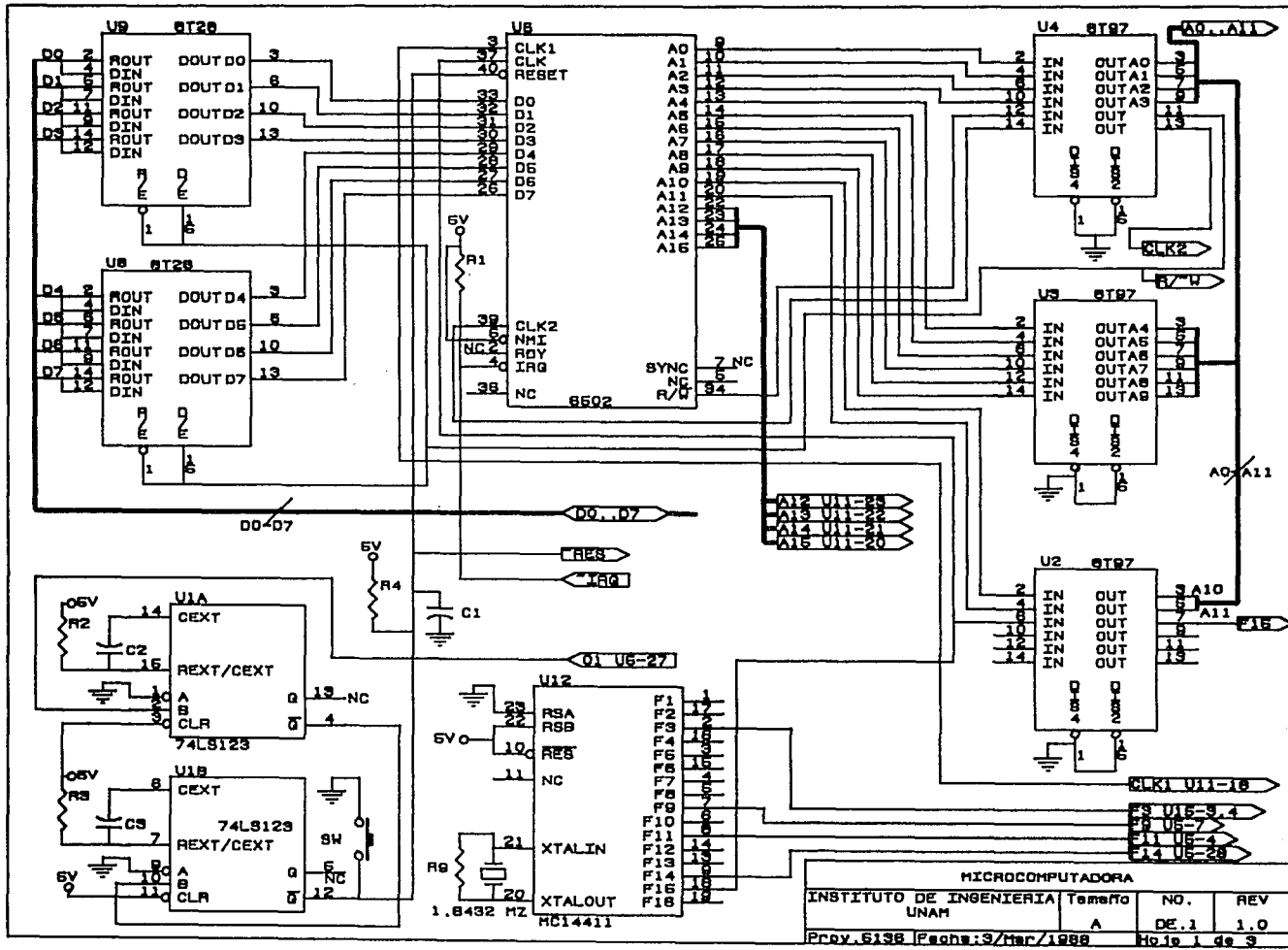
EDOCANAN .BYTE 00,00,00,00,00,00,00,00 ;CADA CANAL ES DE 8 BITS
.BYTE 00,00,00,00,00,00,00,00 ;RESERVA 32 CANALES
.BYTE 00,00,00,00,00,00,00,00
.BYTE 00,00,00,00,00,00,00,00

EDOCANDD .BYTE 00,00,00,00,00,00,00,00 ;16 BYTES
.BYTE 00,00,00,00,00,00,00,00 ;PARA 128 CANALES

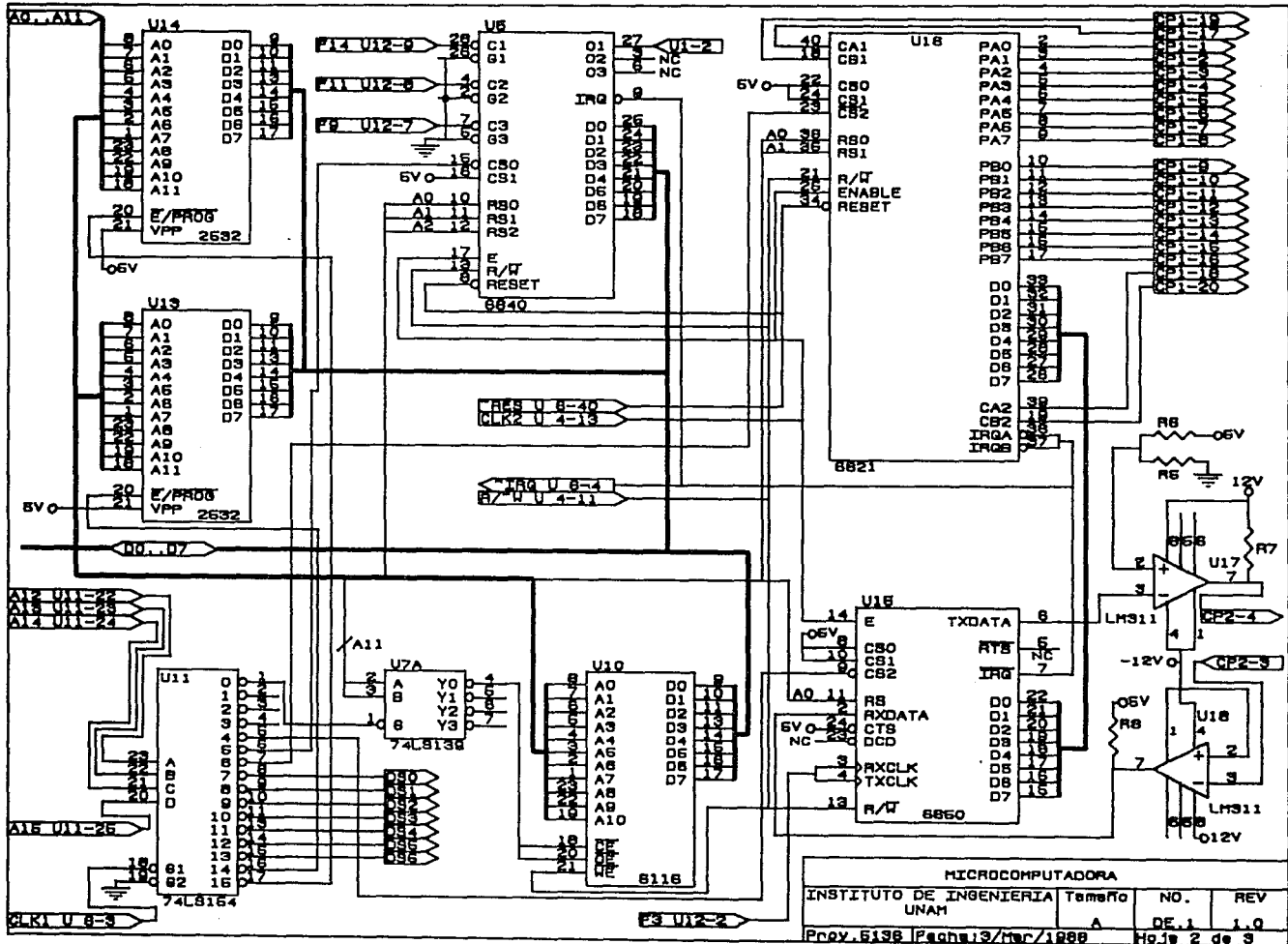
MASCARAD .BYTE 00,00,00,00,00,00,00,0F ;16 BYTES,128 CANALES
.BYTE 00,00,00,00,00,00,00,00

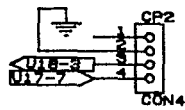
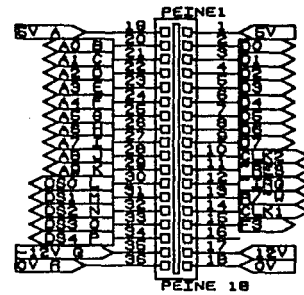
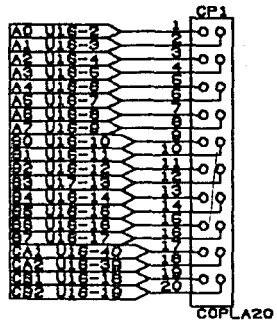
.END

ANEXO C**DIAGRAMAS ELECTRONICOS, DE DISPOSICION
Y LISTAS DE COMPONENTES**

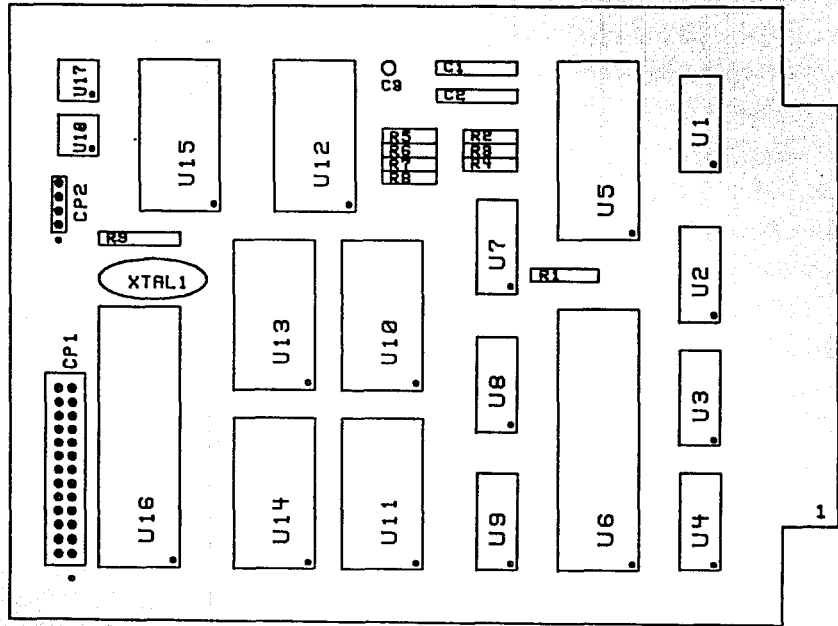


MICROCOMPUTADORA			
INSTITUTO DE INGENIERIA UNAM	Tercero	NO. A	REV 1.0
Proy. 6198 Fecha: 9/Mar/1988		Hoja 1 de 3	





MICROCOMPUTADORA			
INSTITUTO DE INGENIERIA UNAM	Tema 6	NO.	REV
	A	DE.1	1.0
Prov. 515B	Fecha: 9/Mar/1988	Hoja 3 de 3	



MICROCOMPUTADORA			
INSTITUTO DE INGENIERIA UNAM	Tamaño A	NO. DD.1	REV 1.0
Proy. 5136	Fecha: 3/Nov/1988	Ho. 1	de 1

MICROCOMPUTADORA
INSTITUTO DE INGENIERIA, UNAM DE.1

Revisada: 3/marzo/1988
Revisión: 1.0

Lista de materiales

Página 1 de 2

Item	Cantidad	Referencia	Parte
1	1	U6	6502
2	3	U3,U2,U4	8T97
3	2	U9,U8	8T28
4	1	U1	74LS123
5	1	U12	MC14411
6	6	R1,R4,R5,R6,R7,R8	4.7K
7	2	R2,R3	33K
8	1	C1	10uf
9	1	C2	22uf
10	1	C3	1uf
11	1	SW1	RESET
12	1	R9	15M
13	1	XTAL1	1.8432 MZ
14	1	U7	74LS139
15	1	U5	6840
16	2	U13,U14	2532
17	1	U16	6821
18	2	U17,U18	LM311
19	1	U15	6850
20	1	U10	6116
21	1	U11	74LS154

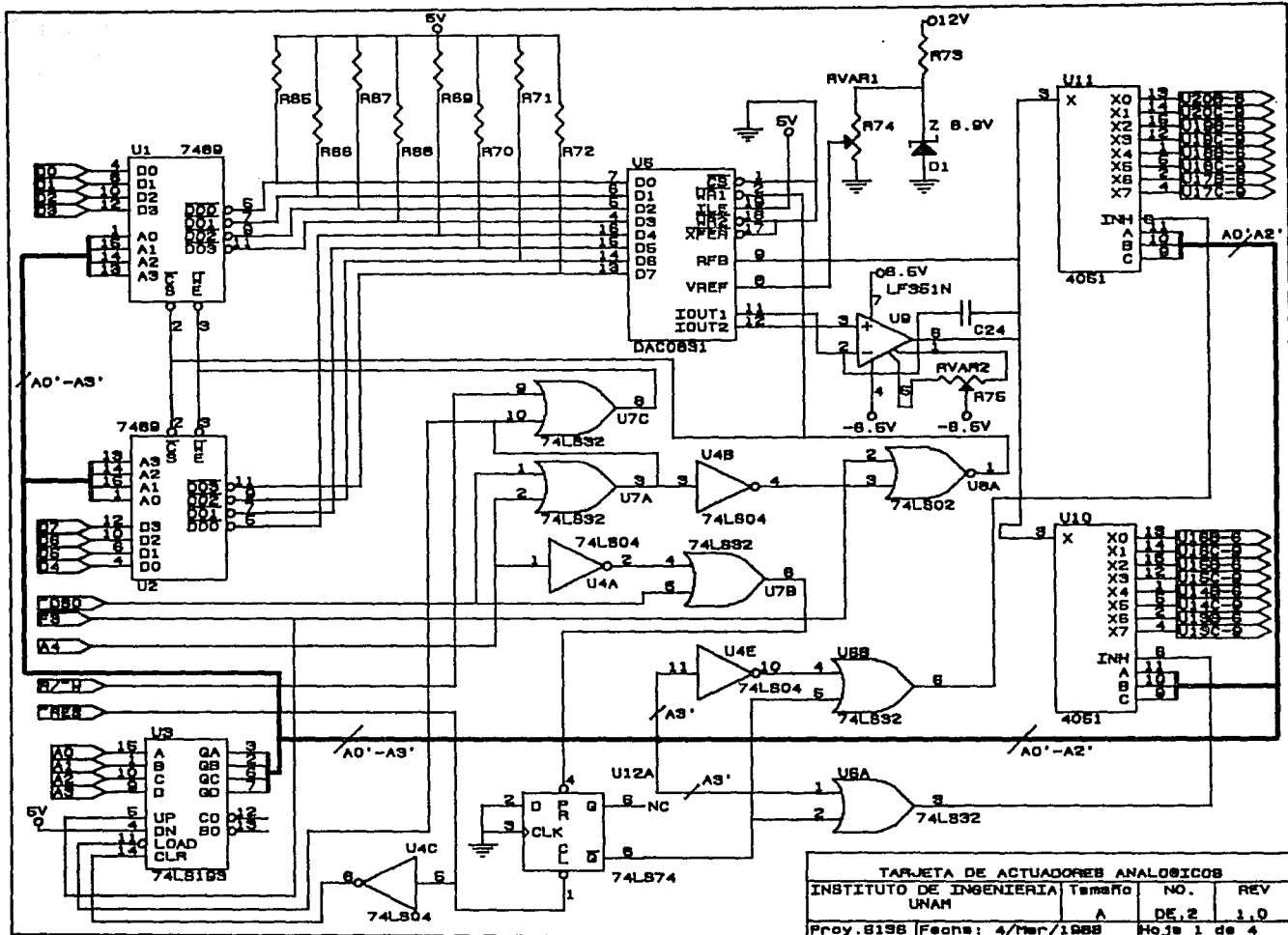
MICROCOMPUTADORA
INSTITUTO DE INGENIERIA, UNAM DE.1

Revisada: 3/marzo/1988
Revisión: 1.0

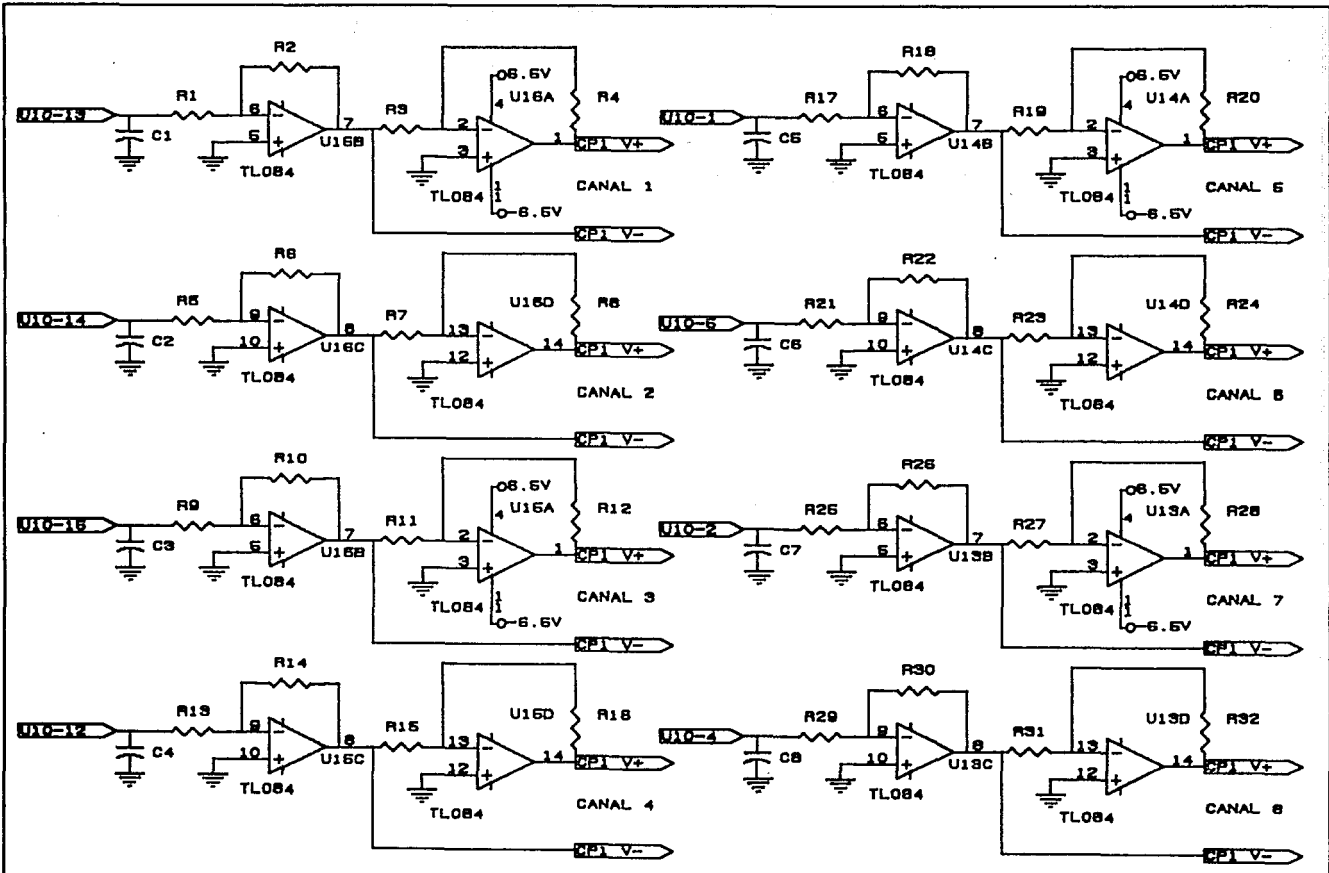
Lista de materiales

Página 2 de 2

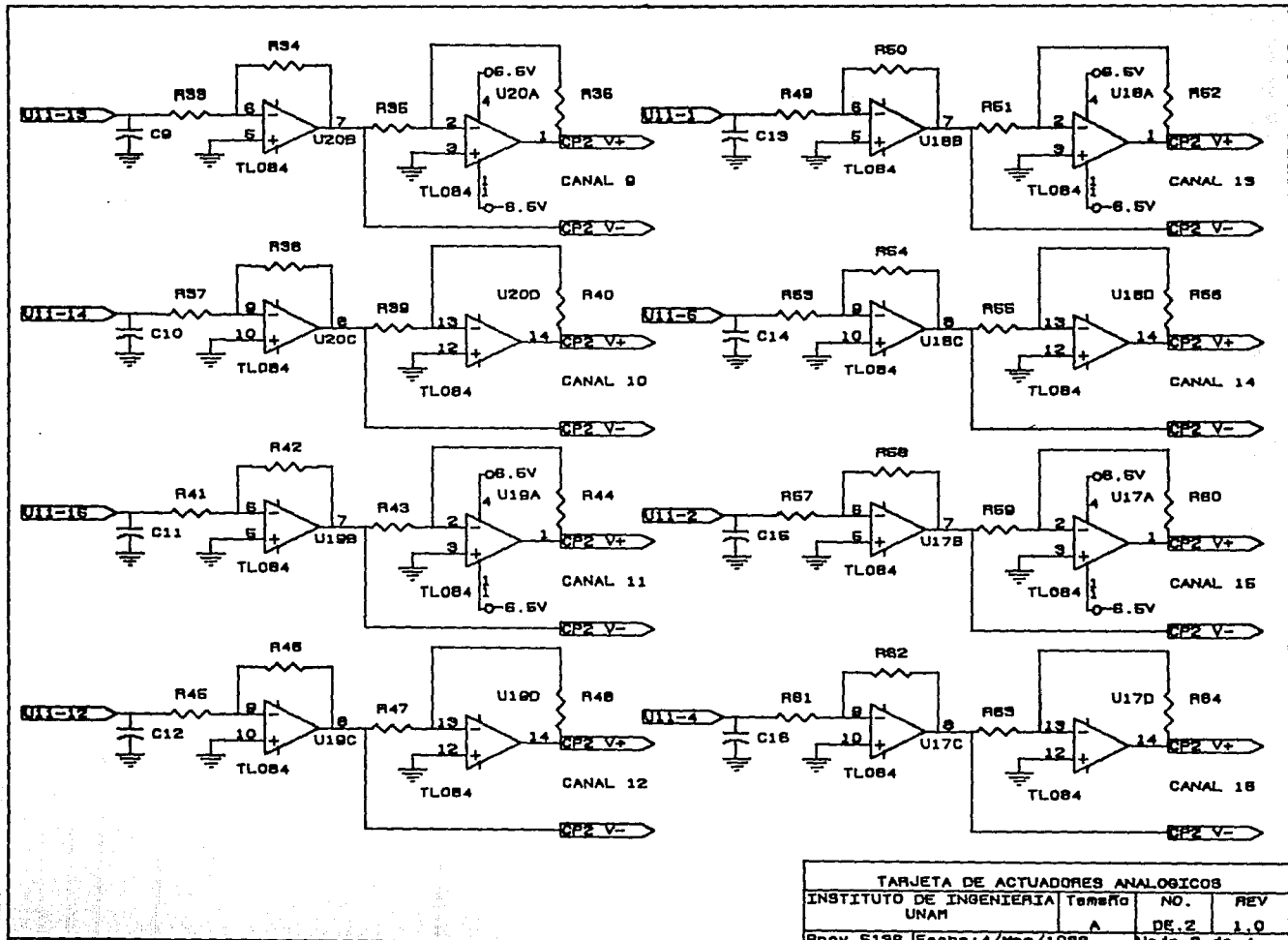
Item	Cantidad	Referencia	Parte
22	1	CP1	COPLA20
23	1	PEINE1	PEINE 18
24	1	CP2	CON4



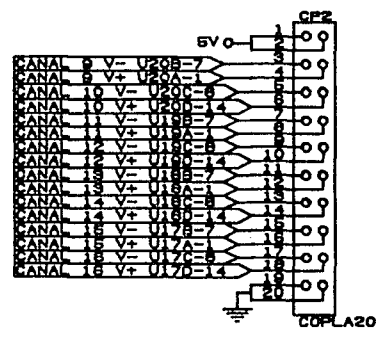
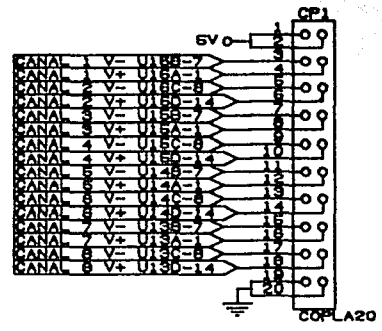
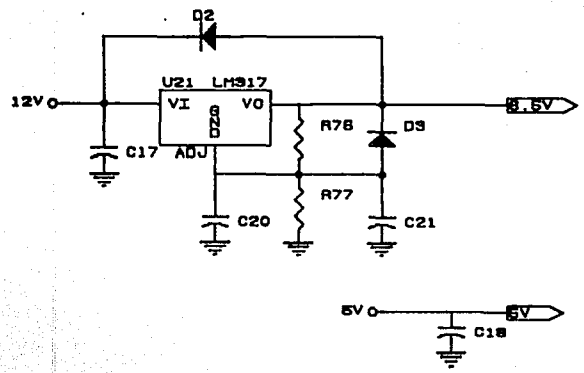
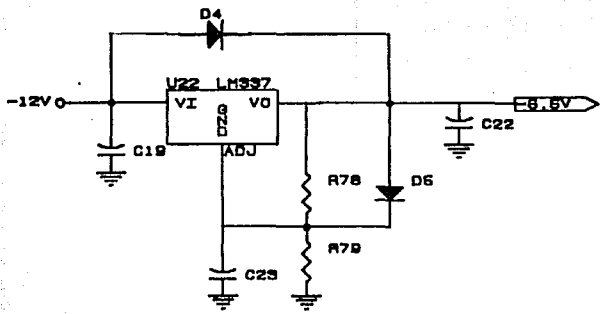
TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA UNAM	Tamaño	NO.	REV
Proy. 8198	Fecha: 4/Mar/1988	A	DE 2 1.0
		Hoja 1 de 4	



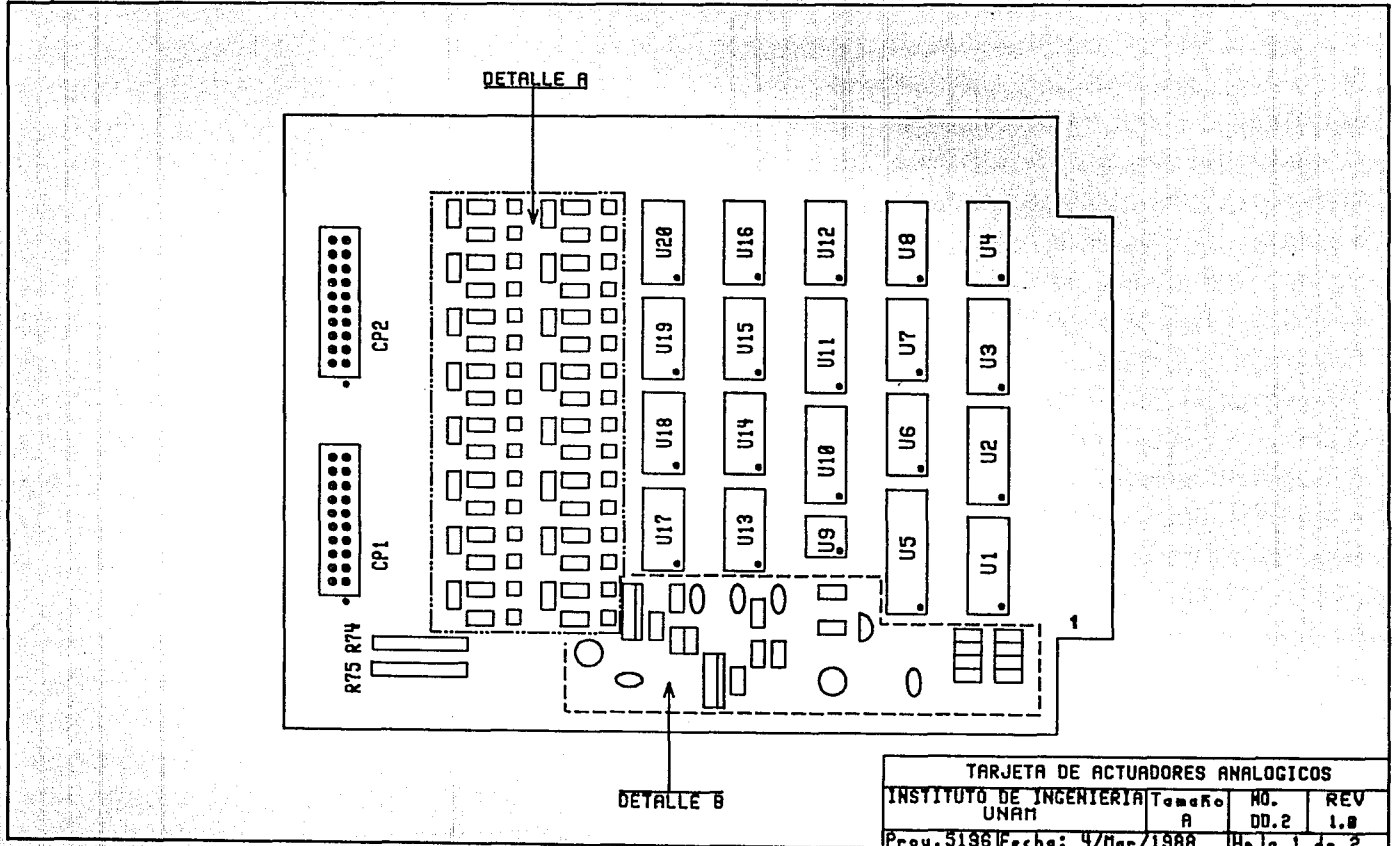
TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA	Tercero	NO.	REV
UNAM	A	DE.2	1.0
Proy. 5138 Fecha: 4/Mar/1988		Hoja 2 de 4	



TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA UNAM	Tomo No	NO.	REV
Proy. 5198 Fecha: 4/Mar/1988	A	DE. 2	1.0
			Hoja 9 de 4



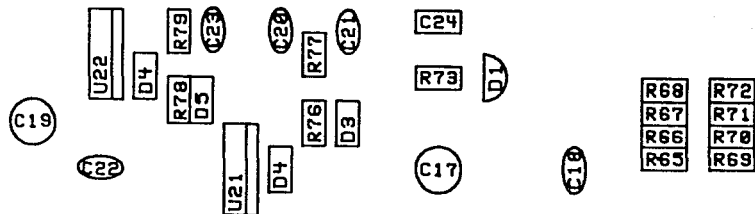
TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA UNAH	Tema No	NO.	REV
Proy. 6136	A	DE 2	1.0
Fecha: 8/Mar/1988	Hoja 4 de 4		



TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA		Tomo	NO.
UNAM		A	DD.2
Proy. 5196		Fecha: 4/Mar/1988	Hoja 1 de 2

C1	R1	R2	C9	R33	R34
	R4	R3		R36	R35
C2	R5	R6	C10	R37	R38
	R8	R7		R40	R39
C3	R9	R10	C11	R41	R42
	R12	R11		R44	R43
C4	R13	R14	C12	R45	R46
	R16	R15		R48	R47
C5	R17	R18	C13	R49	R50
	R20	R19		R52	R51
C6	R21	R22	C14	R53	R54
	R24	R23		R56	R55
C7	R25	R26	C15	R57	R58
	R28	R27		R60	R59
C8	R29	R30	C16	R61	R62
	R32	R31		R64	R63

DETALLE A



DETALLE B

TARJETA DE ACTUADORES ANALOGICOS			
INSTITUTO DE INGENIERIA	Томское	NO.	REV
UNAM	A	00.2	1.0
Proy. 5136	Fecha: 4/Mar/1988	Hoja 2 de 2	

TARJETA DE ACTUADORES ANALOGICOS
 INSTITUTO DE INGENIERIA, UNAM DE.2

Revisada: 4/marzo/1988
 Revisión: 1.0

Lista de materiales

Página 1 de 2

Item	Cantidad	Referencia	Parte
1	2	U1,U2	7489
2	1	U3	74LS193
3	1	U4	74LS04
4	2	U6,U7	74LS32
5	1	U9	LF351N
6	1	U8	74LS02
7	2	U10,U11	4051
8	1	U12	74LS74
9	1	U5	DAC0831
10	9	R65,R66,R67,R68,R69,R70, R71,R72,R75	10K
11	1	R73	1.5K
12	1	R74	20K
13	1	D1	LM329
14	1	C24	22 pf
15	8	U16,U13,U14,U15,U17,U18, U19,U20	TL084
16	16	C1,C2,C3,C4,C5,C6,C7,C8, C9,C10,C11,C12,C13,C14, C15,C16	.47 uf

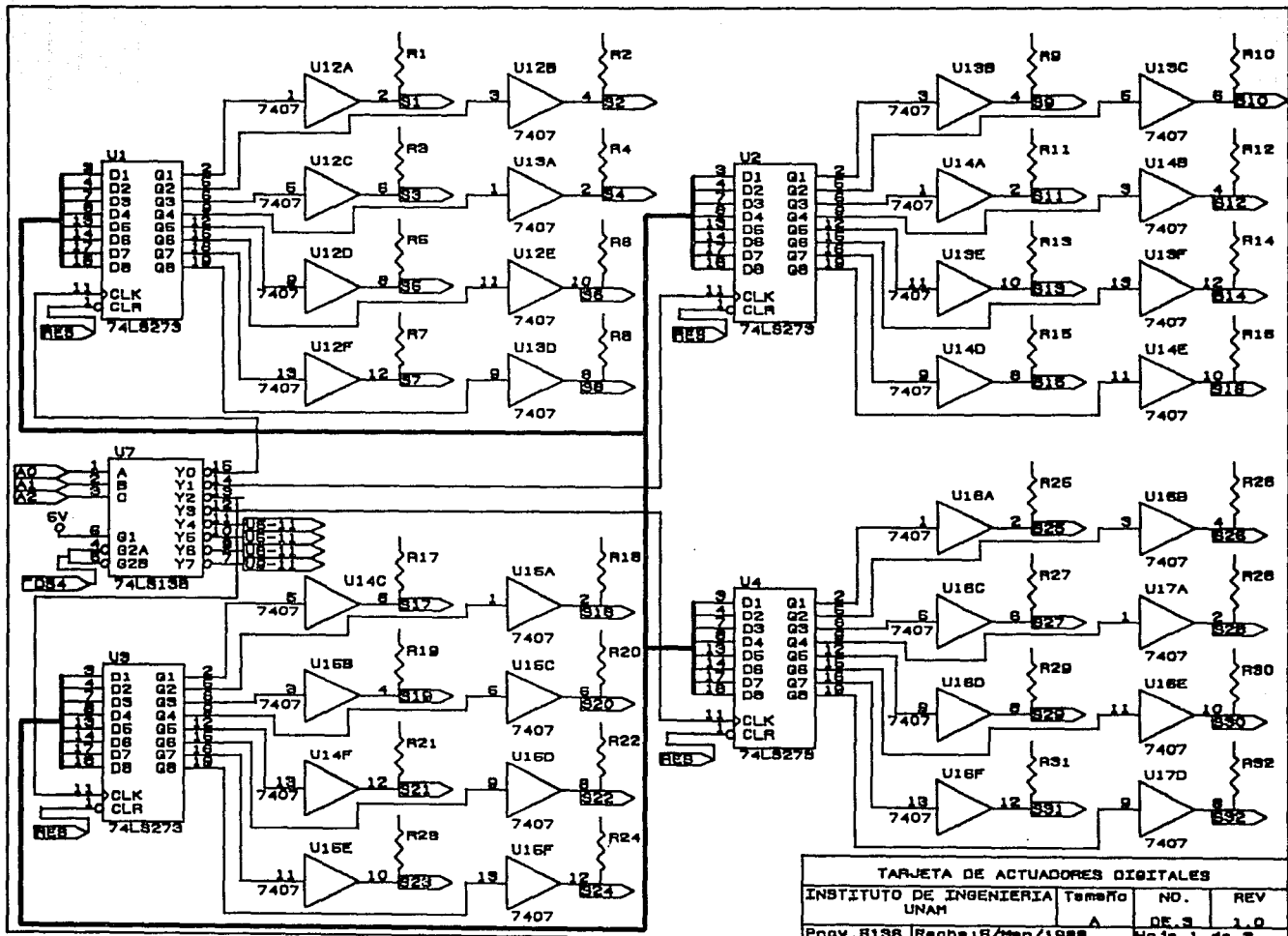
TARJETA DE ACTUADORES ANALOGICOS
 INSTITUTO DE INGENIERIA, UNAM DE.2

Revisada: 1/marzo/1988
 Revisión: 1.0

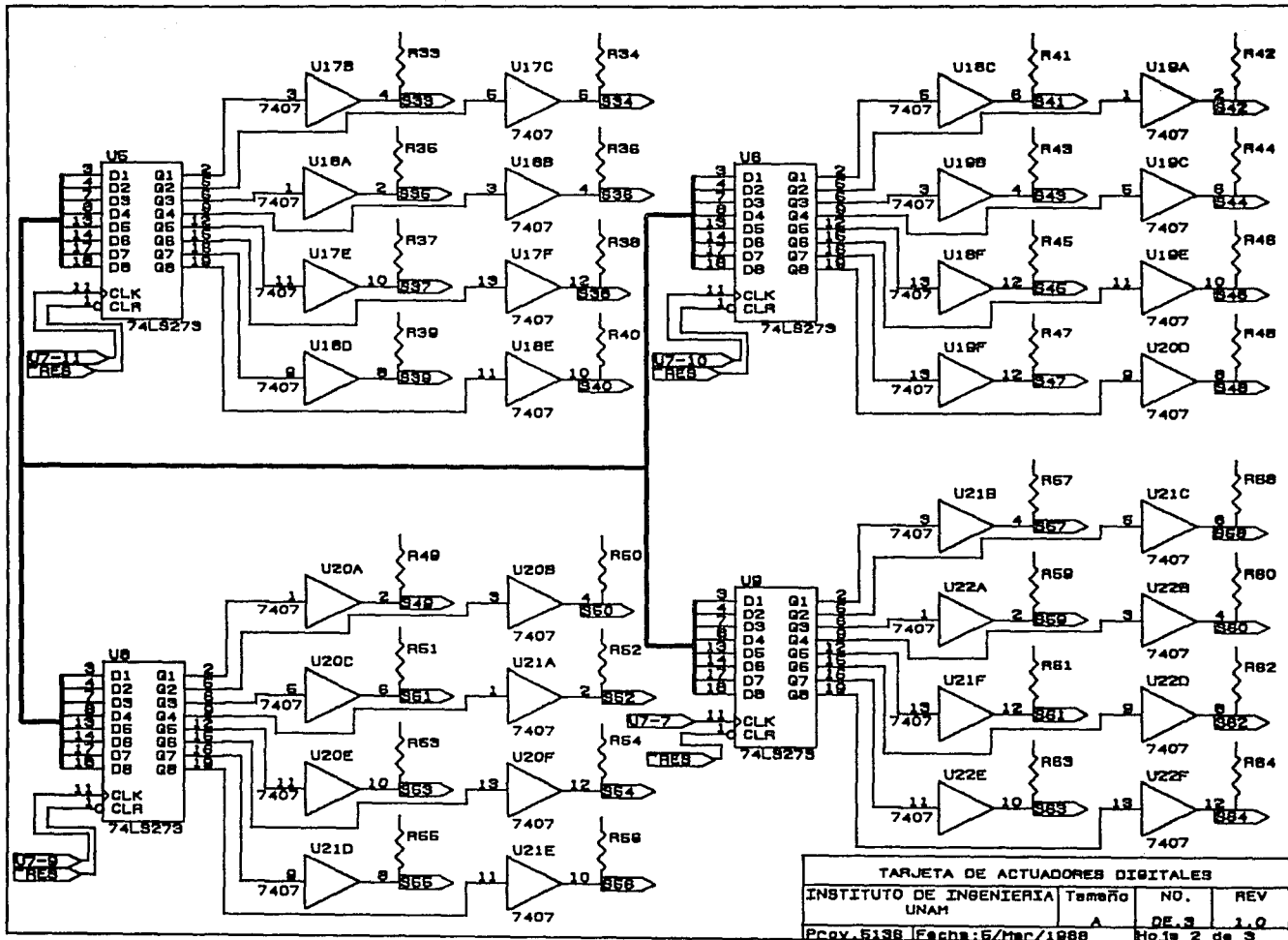
Lista de materiales

Página 2 de 2

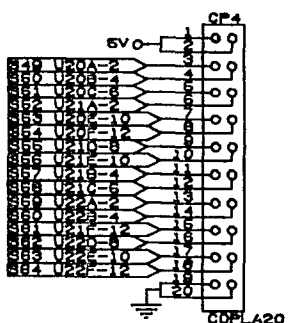
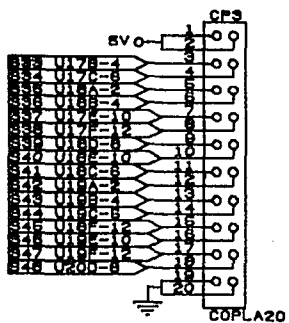
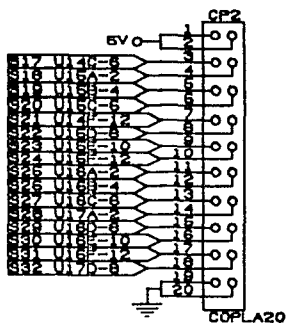
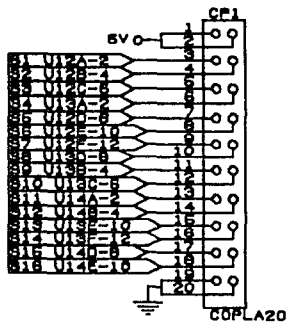
Item	Cantidad	Referencia	Parte
17	64	R1,R2,R3,R4,R5,R6,R7,R8, R9,R10,R11,R12,R13,R14, R15,R16,R17,R18,R19,R20, R21,R22,R23,R24,R25,R26, R27,R28,R29,R30,R31,R32, R33,R34,R35,R36,R37,R38, R39,R40,R41,R42,R43,R44, R45,R46,R47,R48,R49,R50, R51,R52,R53,R54,R55,R56, R57,R58,R59,R60,R61,R62, R63,R64	180K
18	2	CP1,CP2	COPLA20
19	2	C19,C17	10 uf
20	5	C18,C20,C21,C22,C23	4.7 uf
21	1	R78	100
22	1	R79	470
23	1	R76	220
24	1	R77	1K
25	1	U22	LM337
26	1	U21	LM317
27	4	D4,D2,D3,D5	1N4002



TARJETA DE ACTUADORES DIGITALES			
INSTITUTO DE INGENIERIA UNAM	Tercero	NO.	REV
Proy. B138	Requisito	A	DE. 3 1.0
			Hoja 1 de 3



TARJETA DE ACTUADORES DIGITALES			
INSTITUTO DE INGENIERIA UNAH	Tamaño	NO.	REV
	A	DE 3	1.0
Proy. 5138	Fecha: 5/Mar/1988	Hoja 2 de 3	



TARJETA DE ACTUADORES DIGITALES			
INSTITUTO DE INGENIERIA UNAH	Tamaño A	NO. DE.3	REV 1.0
Proy. 6138	Fecha: 5/Mar/1988	Hoja 3 de 3	

R64	R51
R63	R50
R62	R49
R61	R48
R60	R47
R59	R46
R58	R45
R57	R44
R56	R43
R55	R42
R54	R41
R53	R40
R52	R39

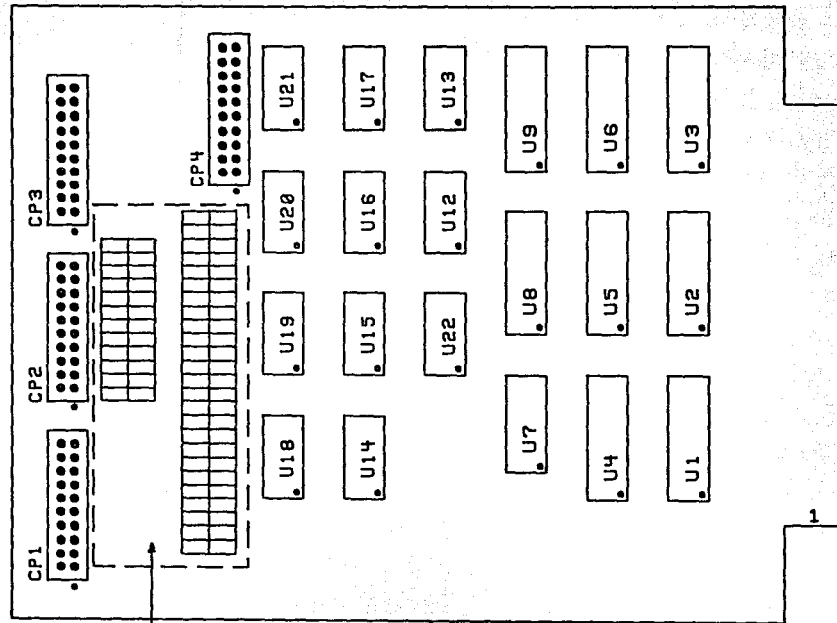
R38	R12
R37	R11
R36	R10
R35	R9
R34	R8
R33	R7
R32	R6
R31	R5
R30	R4
R29	R3
R28	R2
R27	R1
R26	NC
R25	NC
R24	NC
R23	D7
R22	D6
R21	D5
R20	D4
R19	D3
R18	D2
R17	D1
R16	D0
R15	A2
R14	A1
R13	A0

DETALLE A

NC: NO CONECTADO

CADA RESISTENCIA
VA CONECTADA A 5 V
Y A LA LINEA ROTULADA
EN ELLA

TARJETA DE ACTUADORES DIGITALES			
INSTITUTO DE INGENIERIA UNAM	Tamaño	NO.	REV
Proy. 5136	A	DD.3	1.8
Fecha: 5/Mar/1988		Hoja 2	de 2



DETALLE A

TARJETA DE ACTUADORES DIGITALES			
INSTITUTO DE INGENIERIA	Tamaño	NO.	REV
UNAM	A	00.3	1.0
Proy. 5136	Fecha: 5/Abr/1988	Hoja 1	de 2

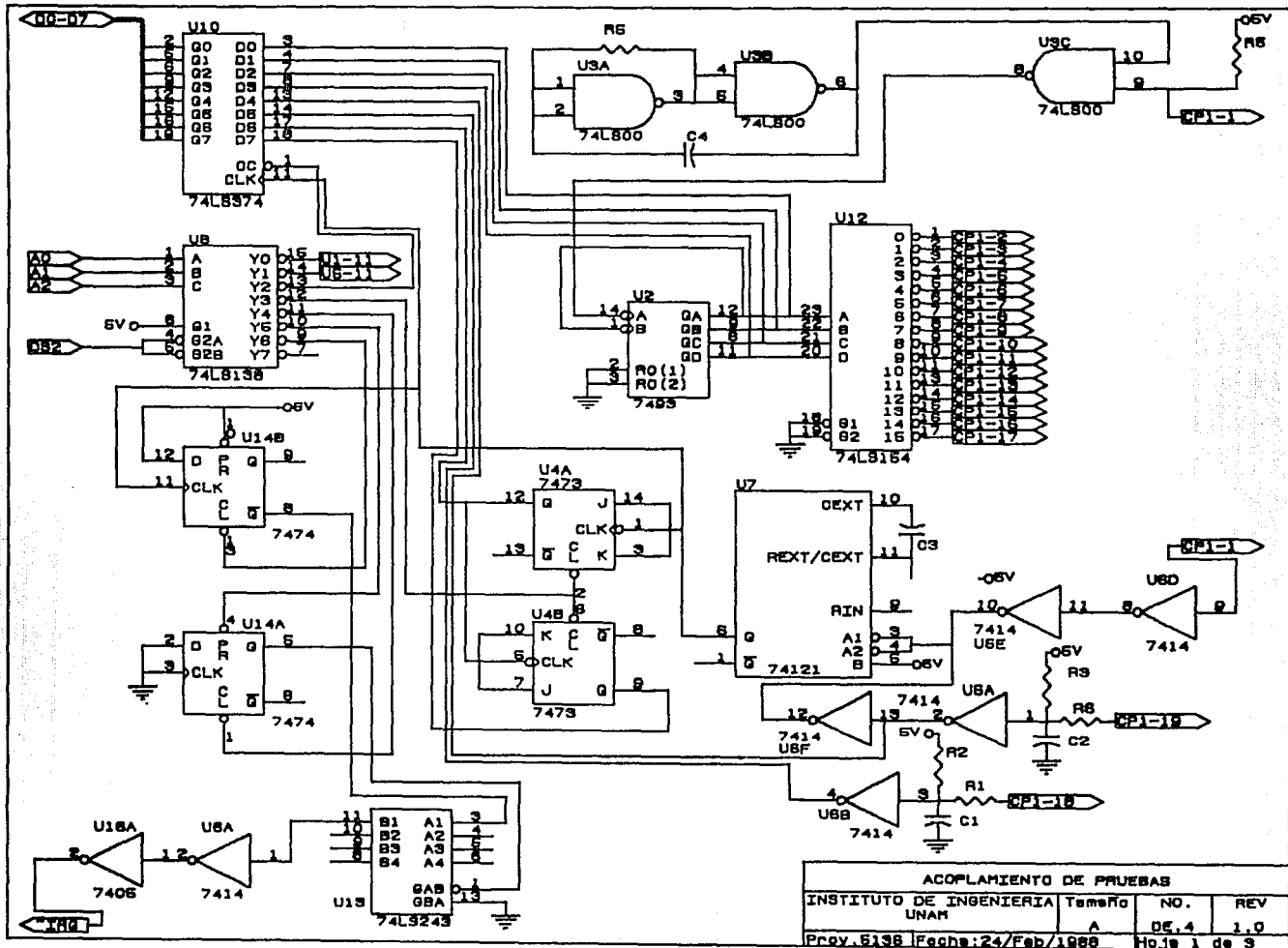
TARJETA DE ACTUADORES DIGITALES
 INSTITUTO DE INGENIERIA, UNAM DE.3

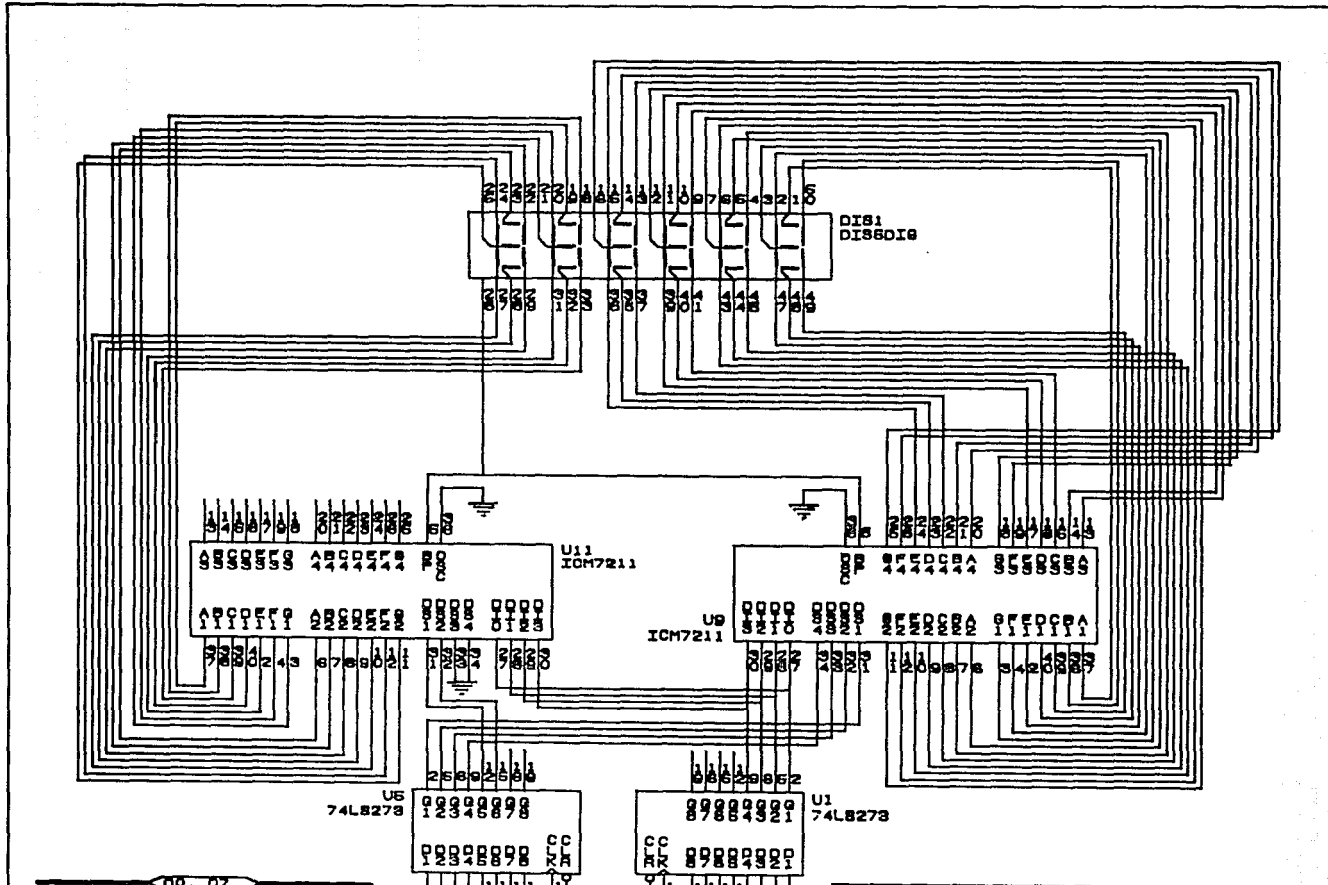
Revisada: 5/Marzo/1988
 Revisión: 1.0

Lista de materiales

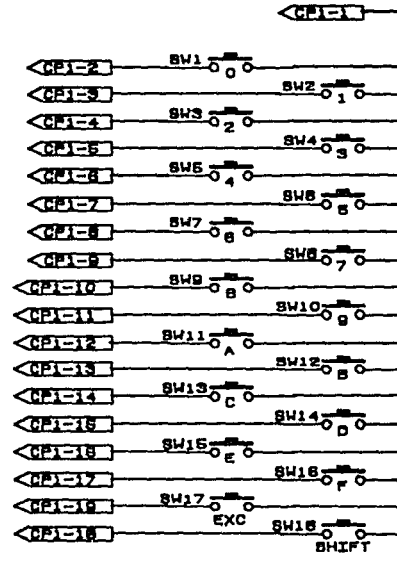
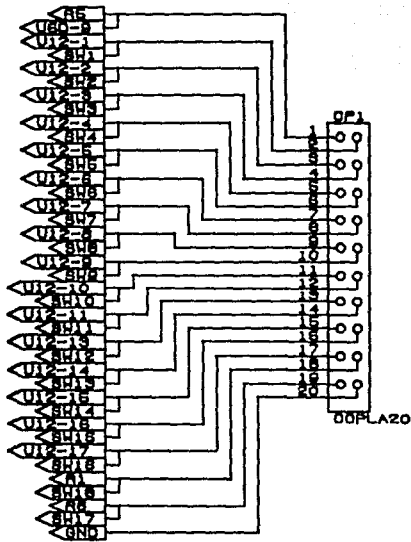
Página 1 de 1

Item	Cantidad	Referencia	Parte
1	8	U1,U2,U3,U4,U5,U6,U8,U9	74LS273
2	1	U7	74LS138
3	11	U12,U13,U14,U15,U16,U17 U18,U19,U20,U21,U22.	7407
4	64	R1,R2,R3,R4,R5,R6,R7,R8, R9,R10,R11,R12,R13,R14, R15,R16,R17,R18,R19,R20, R21,R22,R23,R24,R25,R26, R27,R28,R29,R30,R31,R32, R33,R34,R35,R36,R37,R38, R39,R40,R41,R42,R43,R44, R45,R46,R47,R48,R49,R50, R51,R52,R53,R54,R55,R56, R57,R58,R59,R60,R61,R62, R63,R64	10K
5	4	CP1,CP2,CP3,CP4	COPLA20

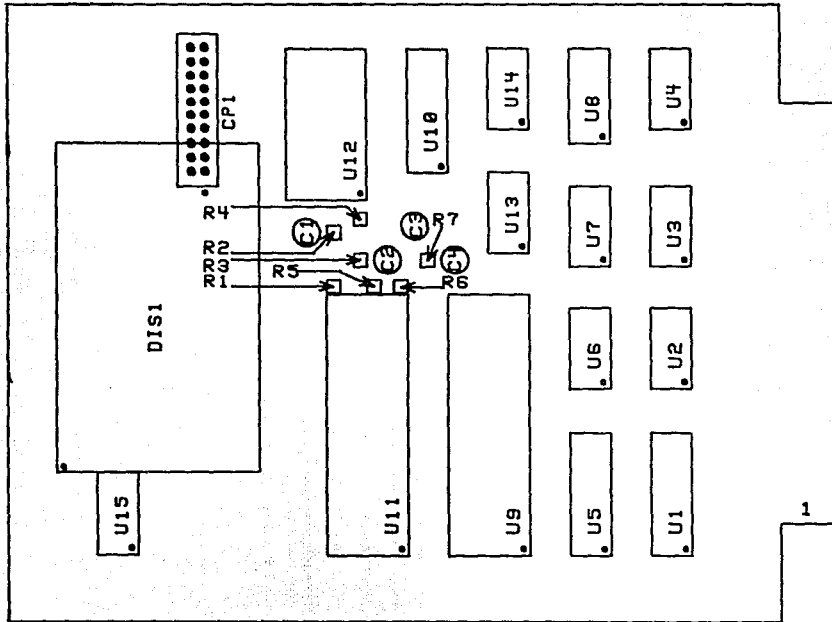




ACOPAMIENTO DE PRUEBAS			
INSTITUTO DE INGENIERIA UNAH	Tercero	NO.	REV
Proy. 8198 / Fecha: 8/Mar/1988	A	08.4	1.0
		Hoja 2 de 3	



ACOPLANIENTO DE PRUEBAS			
INSTITUTO DE INGENIERIA UNAM	Tamaño A	NO. DE.4	REV 1.0
Proy. 5198 Fecha: 6/Mar/1988		Hoja 3 de 3	



ACOPLAMIENTO DE PRUEBAS

INSTITUTO DE INGENIERIA UNAM	Tamaño A	NO. DD.4	REV 1.0
Proy. 5196 Fecha: 24/Feb/1988		Hoja 1 de 1	

ACOPAMIENTO DE PRUEBAS
INSTITUTO DE INGENIERIA, UNAM DE.4

Revisada: 24/Febrero/1988
Revision: 1.0

Lista de materiales

Pagina 1 de 2

Item	Cantidad	Referencia	Parte
1	1	SW1	0
2	1	U3	74LS00
3	1	U2	7493
4	1	U6	7414
5	1	U8	74LS138
6	1	U7	74121
7	1	U4	7473
8	1	SW2	1
9	1	SW3	2
10	1	SW4	3
11	1	SW5	4
12	1	SW7	6
13	1	SW9	8
14	1	SW6	5
15	1	SW8	7
16	1	SW11	A
17	1	SW13	C
18	1	SW15	E
19	1	SW10	9
20	1	SW12	B
21	1	SW14	D

ACOPLAMIENTO DE PRUEBAS
INSTITUTO DE INGENIERIA, UNAM DE.4

Revisada: 24/Febrero/1988
Revision: 1.0

Lista de materiales

Pagina 2 de 2

Item	Cantidad	Referencia	Parte
22	1	SW16	F
23	1	U10	74LS374
24	1	SW17	EXC
25	1	U12	74LS154
26	1	SW18	SHIFT
27	1	U14	7474
28	1	U13	74LS243
29	1	U15	7405
30	3	R1,R6,R7	220
31	1	C1	5uf
32	3	R2,R3,R4	10K
33	1	R5	2.2K
34	1	C2	4.7uf
35	2	C3,C4	1uf
36	2	U1,U5	74LS273
37	1	DIS1	DIS6DIG
38	2	U9,U11	ICM7211
39	1	CP1	COPLA20

ANEXO D

INSTRUCTIVO DE OPERACION

Este instructivo describe la forma en que operará el controlador de actuadores de la planta solar experimental. A este dispositivo corresponde encargarse del nivel automático del sistema de control de los actuadores.

Al energizar el controlador el programa principal del controlador debe de operar automáticamente. Se recomienda de todas manera oprimir el botón de reinicio, situado en la parte superior del gabinete.

El controlador tiene 10 interruptores, los 2 primeros definen el modo de operación del sistema, los 4 siguientes, la configuración de tarjetas y los últimos 4 se utilizan en el modo de pruebas.

INSTRUCCIONES GENERALES

1. Definir el modo de operación de acuerdo con lo siguiente:

S_1	S_0	
0	0	Deshabilitación del puesto central (congelamiento)
0	1	Modo de pruebas
1	0	" " "
1	1	Modo normal.

En cualquier momento de la ejecución del programa se puede cambiar de modo, al hacerlo el programa reconfigurará los periféricos de nuevo por lo tanto habrá que esperar unos instantes para que esté en operación de nuevo.

2. Indicar la configuración de tarjetas analógicas y digitales.

Para las primeras se emplean los interruptores S_2 y S_3 para formar un número de dos bits. El primero es el bit menos significativo y son válidos los números 0, 1 y 2. Si se le da el número 3, la configuración no será válida y ocasionará que no opere el sistema.

Para las tarjetas digitales ocurre lo mismo, sólo que se emplean los interruptores S_4 y S_5 .

3. Cuando se eligió el modo de pruebas en el punto 1, se debe especificar si estas se realizarán con o sin tarjeta de pruebas, con uno y cero en el interruptor S_{10} , respectivamente.

Pruebas sin acoplamiento

Para estas pruebas se emplean los interruptores S_7 - S_9 .

1. Definir si se probarán las tarjetas analógicas o digitales con el interruptor S_9 de la siguiente forma:

S_9	
0	Analógica.
1	Digital.

Cuando se prueba cualquier tipo de tarjetas se podrá pasar a probar el otro, o a pruebas con acoplamiento, en cualquier momento.

2. El programa comienza a probar a partir del canal 1, de tipo elegido. Para cambiar al siguiente se debe mover el interruptor S_8 , al llegar al último, se comenzará de nuevo en el 1.

3. Para cambiar de valor de un canal analógico bajo prueba, se hace lo mismo que en el paso anterior, sólo que se acciona el interruptor S_7 .

El ciclo de valores a que se responde con los movimientos del interruptor mencionado es de 0, 2.5 y 5 V. Se aclara que también los valores simétricos son entregados.

Para el caso digital sólo entrega los valores 0 y 5 voltios, que se definen directamente según la posición del interruptor.

Pruebas con acoplamiento

1. Instalar la tarjeta de pruebas antes de energizar el controlador.
2. Efectuar el paso 1 de instrucciones generales, la pantalla debe de ponerse en ceros, en caso de que esto no ocurra, oprimir el botón rojo de restablecimiento.
3. El sistema se pondrá en el modo de comandos y esperará la presión de la tecla A y D, para conocer si van a realizar pruebas analógicas o digitales. Se desplegará en el sexto lugar de la pantalla la letra oprimida, únicamente serán aceptadas estas teclas.

Una vez definida la letra, oprima el botón Exc (ejecuta) para pasar al paso 4.

4. Indicar el número de canal a probar, el sistema aceptará a partir del canal 1, el valor máximo de aceptación está determinado por la configuración de tarjetas definida. Consultar la siguiente tabla para saber los valores máximos permitidos (el valor se especifica en hexadecimal)