



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

MICROCOMPUTADORA PERSONAL DEDICADA A UN ACELERADOR DE ELECTRONES VAN DER GRAEFF DE 2 MeV

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
(AREA: ELECTRICA Y ELECTRONICA)
P R E S E N T A :
OSVALDO JIMENEZ MEIXUEIRO



DIRECTOR DE TESIS: FIS. FRANCISCO FERNANDEZ ESCOBAR

CIUDAD UNIVERSITARIA

277170
2000



Universidad Nacional
Autónoma de México




UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



Volar no significa
solo mover las alas,
sino quedarse en el aire sin apoyos.

A mis padres y abuela: **Manuel, Griselda, Esperanza,...**
...por su cariño, apoyo y confianza

Cuando vayan mal las cosas como a veces suelen ir
cuando ofrezca tu camino solo cuestas que subir
cuando ya el dolor te agobie y no puedas ya sufrir
descansar acaso debes pero nunca desistir.

A la memoria de: **Francisco, Guillermina, Ramón**, mis abuelos;
y **Justino**, un gran amigo.

A mi director de tesis Fis. Francisco Fernández Escobar; por su asesoría, apoyo y paciencia por el desarrollo de este trabajo.

Agradezco todo el apoyo para la realización de este trabajo; a la M. en C. Esbaide Adem Chaín, al Sr. Margarito Vázquez Rodríguez, al Sr. Antonio Ramírez Arce y la Sra. Martha González Álvarez; además al Instituto de Física, C.U. por la oportunidad de llevarlo a cabo.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 - "ADAPTACIÓN DE SISTEMAS"	
1.1) Definiciones y comentarios	1
1.2) Algo de historia	2
1.3) Tecnología de siglo	3
1.4) Ejemplos de control de sistemas	4
1.5) Presentación del trabajo	4
Referencias del capítulo 1	6
CAPÍTULO 2.- "SISTEMAS ACELERADORES DE PARTÍCULAS"	
2.1) Acelerador Van de Graaff	8
2.2) Aceleradores lineales	11
2.3) Aceleradores circulares	11
Referencias del capítulo 2	12
CAPÍTULO 3 - "TRANSDUCTORES, SEÑALES Y PROCESOS EN EL ACELERADOR"	
3.1) El acelerador Van der Graff del instituto de física de la UNAM	13
3.2) Funcionamiento	15
3.3) Modo de operación	18
3.4) Señales involucradas	20
Referencias del capítulo 3	23
CAPÍTULO 4.- "TEORÍA DE CONTROL. SEÑALES ANALÓGICAS Y DIGITALES"	
4.1) Control de lazo cerrado	25
4.2) Sistemas de control de lazo abierto	26
4.3) Otros conceptos en la teoría de control. Un ejemplo	27
4.4) Señales, analógicas y digitales	31
4.5) Control de dos posiciones o "Si-No"	33
Referencias del capítulo 4	36
CAPÍTULO 5 - "COMPUTADORAS E INTERFACES"	
5.1) Antecedentes	37
5.2) El byte	38
5.3) Interfaz paralela	41
5.4) Interfaz serie	42
5.5) Comunicación simple o simplex	43
5.6) Comunicación semi-duplex	43
5.7) Comunicación duplex	44
5.8) Comunicación asíncrona	44
• Bit de inicio, bit de paro	45
• Bit de paridad	46
• Longitud o duración del bit	47
5.9) Comunicación síncrona	48

5.10) El puerto serie	48
5.11) El UART	50
5.12) Interfaz RS-232	51
Referencias del capítulo 5	54
CAPITULO 6.- "COMPUTACIÓN DEDICADA A LA INSTRUMENTACIÓN"	
6.1) Transmisión	55
6.2) Recepción	60
6.3) Lenguajes de computación	62
• BIOSCOM	64
• INT86	66
Referencias del capítulo 6	68
CAPÍTULO 7 - "INSTRUMENTACIÓN DEDICADA AL ACELERADOR"	
7.1) Autovarías	69
7.2) Señales en el sistema acelerador	71
7.3) Transmisión	72
• Señal de control	72
• Conversión de RS-232C a TTL	73
• La sincronización y el reloj de recepción	74
• Captura y conversión de bits de serie a paralelo	76
• Almacenamiento	77
• Conversión digital-analógica	78
• Direccionamiento	80
7.4) Recepción	83
• Conversión analógica-digital	83
• Entrada de señales digitales	85
• Envío de datos	85
• Conversión TTL a RS-232	87
7.5) Programación	87
7.6) Evaluaciones y conclusiones	90
Referencias capítulo 7	91
APÉNDICE A.- "REGISTROS E INTERRUPTIONES EN EL 8086"	
• Registros del 8086	A-1
• Uniones	A-2
• Otras interrupciones con el INT86	A-3
Referencias del apéndice A	A-4
APÉNDICE B - "DISTRIBUCIÓN Y DIAGRAMAS DE LOS CIRCUITOS"	B-1
APÉNDICE C.- "LISTADO DEL PROGRAMA"	C-1

INTRODUCCIÓN

Este trabajo de tesis consistió en automatizar completamente la operación y el control de un acelerador Van de Graff de 2 MeV., que ha funcionado *manualmente* en el Instituto de Física de la U.N.A.M. desde la década de los 50. Tal automatización consistió en implementar un sistema de control híbrido por actualización, que comprendió el diseño, desarrollo, construcción e instalación de circuitos que permitieron acoplar una microcomputadora personal (tipo AT o superior) dedicada tanto a la operación y el control del acelerador, como a la captura y el manejo de datos de interés en el mismo.

Se tomó como punto de partida al conocimiento más completo de dicho acelerador, tanto en lo referente a su funcionamiento como a su modo de operación, incluyendo todas las señales involucradas en el conjunto (de seguridad, de operación y control, de datos); lo que permitió diseñar el esquema general de dicho control híbrido bajo el enfoque de modernizar el acelerador en adición a lo existente; es decir, preservando también su modo de operación manual.

De acuerdo a lo anterior, la presentación del desarrollo de esta tesis se inicia con algunos comentarios sobre la adaptación de sistemas, haciendo énfasis sobre las diversas formas y tipos de modernizaciones; y cuando éstas son factibles y convenientes. Posteriormente, en los capítulos 2 y 3 se describen respectivamente los sistemas aceleradores de partículas en general; y particularmente el acelerador motivo de este trabajo.

El capítulo 4 se dedica a la teoría de control, cubriendo algunos aspectos sobre los sistemas de lazo cerrado y de lazo abierto, así como tipos de control, que pueden ser analógico, digital o híbrido. En el capítulo 5 se describen las componentes principales de las computadoras personales incluyendo sus interfaces y formas de comunicación, con especial atención en la comunicación asíncrona RS-232C utilizada en esta tesis. El capítulo 6 se ocupa de la computación específicamente dedicada al manejo de instrumentos en general, donde se detallan la transmisión y recepción de datos tanto a nivel "hardware" como de "software", incluyendo algunos ejemplos de comandos del lenguaje C para el protocolo de comunicación RS-232C.

Finalmente, el capítulo 7 describe la implementación de cada una de las etapas que específicamente constituyeron toda la instrumentación dedicada al acelerador mencionado; la cual se dividió en dos grandes grupos: la incluida en el acelerador para su adaptación a la computadora y lo desarrollado para la computadora con el fin de comunicarse con el acelerador, mencionando las partes principales de la programación empleada, cuyo listado completo se presenta en el apéndice C. También en este capítulo se presentan las evaluaciones y conclusiones pertinentes a todo el trabajo.

El apéndice A contiene tópicos suplementarios sobre registros e interrupciones en el microprocesador 8086 de Intel, base de los microprocesadores en las computadoras personales actuales. En el apéndice B se incluyen todos los diagramas de los circuitos desarrollados para la computadora, ya que los correspondientes al acelerador se intercalan a lo largo del texto del mismo capítulo 7.

CAPÍTULO 1

ADAPTACIÓN DE SISTEMAS

1.1) DEFINICIONES Y COMENTARIOS.

El concepto de adaptación puede tener varios significados según el sistema y entorno a los que se aplique. En la biología se refiere a la concordancia entre las cualidades morfológicas y fisiológicas de un organismo con las características físicas y biológicas del medio; en psicología se habla de la acomodación del comportamiento individual humano a los modelos de conducta y a las normas que aseguran la perpetuación y funcionamiento del medio social. En la ingeniería y particularmente en la instrumentación, se refiere a ingenios, dispositivos, máquinas, o sistemas ya construidos, en los que se buscan acciones que cubran nuevas necesidades que hay en el medio de la industria, las ciencias o hasta en el hogar; y por lo mismo exigen características nuevas para continuar con su funcionalidad.

Entre los factores que motivan las adaptaciones, -dentro del campo de la ingeniería electromecánica-, están los cambios que de manera natural demanda la modernización; esto es, los elementos o dispositivos que algún día se adquieren, se deterioran con el paso del tiempo y las refacciones necesarias para su mantenimiento llegan a ser escasas, costosas y hasta discontinuadas, por la aparición de nuevos y mejores equipos. Otro factor importante es la accesibilidad para tener siempre un equipo reciente; prueba de ello es que en algunas publicaciones sobre temas de instrumentación, suelen aparecer artículos sobre el desarrollo de determinado dispositivo, o la adaptación de cierta maquinaria, como de "fácil implementación y bajo costo". En parte, esto surge como necesidad ante la falta de tecnología de punta y/o los medios para adquirirla. En la mayoría de los casos, tales artículos son en revistas o libros de circulación popular, donde aún cuando los autores son profesionales en su área, el tema está más bien dirigido al aficionado o técnico, que al profesional en instrumentación. Desde luego que esto no es criticable ya que tanto profesionales como técnicos juegan un papel importante en el desarrollo de cualquier comunidad; de hecho, por la otra parte, también hay publicaciones especializadas en las que los temas que se divulgan poseen todo el formalismo profesional y hasta científico.

Cualquiera que sea el motivo de una adaptación, ésta debe ser previamente analizada en cuanto a factibilidad, tiempo estimado de realización, costos de partes y mano de obra, etc.; en relación con los beneficios a obtener; ya que el adaptar un sistema "a la ligera" puede generar labores y costos tan excesivos hasta el extremo en que hubiese sido más redituable cambiar totalmente el equipo antiguo por el de modelo reciente. Todo esto puede resumirse en la evaluación previa de la razón beneficio/inversión.

No debe confundirse adaptación con innovación o invención, ya que esta última es un acto de creatividad que da como resultado un dispositivo, proceso o técnica bastante novedosa, que produce un cambio significativo en la tecnología y sus aplicaciones. Invención se refiere a algo no construido, aún cuando ello surja de la combinación o adaptación de dos o más sistemas; por ejemplo, el adaptar una rueda a un brazo de palanca, puede considerarse como la invención

de la carretilla. Realmente no existe una frontera entre adaptaciones e invenciones, ya que éstas últimas demandan adaptaciones constantes que a su vez pueden conducir a nuevas invenciones; lo que sí es un hecho, es que los cambios radicales en la tecnología y sus aplicaciones, repercuten fuertemente en cualquier sociedad.

1.2) ALGO DE HISTORIA.

Entendiendo por máquinas a aquellos dispositivos que ha inventado la humanidad para facilitar la interacción materia-energía y adaptarse más cómodamente a la naturaleza; se dice¹ que hay tres características inherentes al hombre, que desde la prehistoria hasta la actualidad, han inspirado la creación de las máquinas: guerra, diversión y trabajo. Son hechos muy conocidos el que tengamos a nuestro alcance la potencia del viento o de una gran caída de agua, el que la fuerza de varios caballos se encuentre encerrada dentro de un automóvil, o bien aprisionada como la furia destructora de millones de toneladas de dinamita contenidas en una sola bomba. Así como hay múltiples ingenios que aligeran las tareas cotidianas, invitan al descanso y alivian el dolor: también hay por otra parte fusiles, lanzallamas, cañones, etc.; hasta misiles gobernados por computadoras preparadas para calcular las muertes de millones de personas en unos cuantos segundos. Todo esto ha dado lugar a una gran clasificación al respecto: las máquinas son positivas cuando son dedicadas a la vida y al bienestar humano; o negativas, cuando se utilizan para la destrucción.

En 1685 Sir Samuel Morland mencionó que *al ser evaporada el agua por el fuego, los vapores precisan un espacio mucho mayor, unas 2000 veces el ocupado por el agua; y antes que someterse a estar encerrados, harían estallar una pieza de artillería. Sin embargo, bajo control soportan la carga pacíficamente, como buenos caballos, y así pueden ser de gran utilidad para la humanidad.* Estas fueron palabras proféticas para lo que se venía en el futuro con la máquina de vapor; y a consecuencia de ésta, el advenimiento de nuevas máquinas utilizando este principio. La máquina de vapor adquirió muchos usos nuevos en la época, como la propulsión de molinos, torres, trenes, buques y máquinas de hilar por nombrar algunos. Todo esto ocurrió durante la época de la revolución industrial (s. XVIII), que se originó en Gran Bretaña y tuvo una gran repercusión en Europa, pero se produjo a un ritmo más tranquilo en América. Hubo inventores que concibieron máquinas ingeniosas, como Joseph Marie Jacquard, que construyó un telar para tejer diseños complicados en seda fina utilizando tarjetas perforadas para producir los patrones; y de esta forma fue uno de los precursores de la automatización que llegaría 150 años más tarde. Otros más buscaron fuentes de energía alternativas como la turbina de agua y la electricidad; esta última se empezó a utilizar en lugar de las máquinas de vapor en varios procesos dentro de las industrias: aunado a esto se empezaron a buscar y obtener nuevas máquinas y combinaciones de materiales que condujeron al motor de combustión interna a finales del siglo XIX.

Con la electricidad se dio entrada a nuevos inventos como el telégrafo, la bombilla de luz (foco) y el teléfono, además de permitir el nacimiento de la electrónica, -disciplina ya de este siglo-, con la que surgen el radio, la televisión, las computadoras, etc.

Robert O'Brien. Colección científica de Time-Life, Máquinas México 1972 Pp 8-73, 12-166.

1.3) TECNOLOGÍA DE SIGLO.

Sabiendo que la electrónica se encarga de la generación y/o el proceso de señales eléctricas mediante la electricidad, se puede afirmar que tal disciplina nació en 1905 con el descubrimiento del primer bulbo: el diodo de Flemming (rectificador al vacío); posteriormente, en 1908, Lee de Forest al adaptar un tercer electrodo (rejilla) al diodo de Flemming, encontró la primera válvula electrónica: el triodo, con el que se podía controlar el paso de una corriente por medio de un voltaje. Los bulbos, que fueron muy usados hasta hace un par de décadas, tenían las desventajas de eficiencia baja, vida media relativamente corta y requería de un tiempo determinado de espera para llegar a su punto de operación inicial (se requería que alcanzara una cierta temperatura). Al disponer sólo de este tipo de válvula, el desarrollo de la electrónica en la primera mitad del siglo no fue tan acelerado como en la segunda, a raíz del descubrimiento del transistor (1948); válvula diminuta, con mayor eficiencia y respuesta prácticamente "instantánea". A la fecha, los transistores no han logrado sustituir completamente a los bulbos, ya que en algunas aplicaciones como el cinescopio de TV y el magnetrón de un horno de microondas, todavía no existe el equivalente satisfactorio en estado sólido.

Con el estado sólido se desencadenó la investigación y el desarrollo de nuevos equipos y dispositivos tomando como bases de diseño la velocidad de respuesta, el consumo de energía, el tamaño y los costos de producción. Estas directrices se ven de forma práctica en la creación de equipos que antes no eran imaginables: agendas electrónicas, relojes que hacen más que dar la hora, máquinas de escribir electrónicas, televisores de bolsillo y nuevas componentes, que entre otros, permitieron la aparición de las computadoras, que inicialmente estaban contempladas para grandes empresas, ya sea para llevar un control de inventarios, de producción, o bien para el diseño de una gran variedad de artículos, desde un lápiz hasta un automóvil, o una nave espacial. Su uso y aplicaciones se encontraban lejanos para cualquiera que no contara con el capital necesario.

Actualmente los costos al respecto han disminuido a tal grado que ya es posible tener una computadora hasta en el hogar. En las escuelas se han empezado a utilizar con fines didácticos y en los centros de investigación son una poderosa herramienta tanto por el manejo de datos como por el control automatizado de equipos experimentales que operan bajo una misma secuencia. En la industria, además de diseños, inventarios y registros de trabajo, se busca la manera tener el control de procesos de producción en serie en un tiempo menor al que se emplearía realizando el mismo trabajo manualmente.

El control de un proceso o sistema por medio de una computadora, se puede dar de dos formas: por **diseño** y por **actualización**. Cuando es por diseño, antes de instalar la planta se prevén los equipos necesarios para su control, es decir equipos nuevos; se busca que éstos sean compatibles con la computadora o viceversa. En el caso de actualización, ya se cuenta con el equipo a controlar y a veces con la computadora; por lo que deben agregarse los dispositivos y programas necesarios para realizar la integración; es decir, efectuar la adaptación de sistemas *bajo los criterios antes mencionados*.

1.4) EJEMPLOS DE CONTROL DE SISTEMAS.

Con base en lo anterior, se describen brevemente a continuación algunas causas y consecuencias de dos casos de control de sistemas; a saber, dentro de la industria automotriz y de los medios de comunicación escrita

Inicialmente los automóviles se fabricaban pieza por pieza y a mano, donde varios trabajadores estaban encargados de fabricar un solo auto, lo que podía tomar semanas; posteriormente apareció la producción en serie donde a cada trabajador le correspondía una sección específica del auto, con esto se abarataron los costos y el tiempo de producción disminuyó. En la actualidad aun se utiliza esto último pero con la diferencia de que en algunas secciones, que resultaban peligrosas o monótonas, se ha sustituido al hombre por robots programados. Con esto se realiza una producción mayor en menor tiempo y a menor costo

Desde la antigüedad la humanidad ha tratado de documentar lo que hace y observa, al respecto se pueden mencionar a los hombres que en la edad de piedra dibujaban en sus cuevas sus vivencias; los egipcios y los griegos escribían a mano sobre papel; por la edad media apareció la imprenta, con la que se reprodujeron muchos escritos, de los que se considera a la Biblia como uno de los más famosos; ésta era muy difícil de conseguir por el alto precio que tenía cada copia, ya que inicialmente se fabricaban sólo algunos cientos de cada ejemplar. Actualmente se fabrican miles en cada país, mediante imprentas manejadas por computadora, donde se realiza desde el diseño, la impresión, clasificación, etc. Una mayor y más rápida producción de la comunicación escrita se efectúa en periódicos y revistas mediante las grandes rotativas computarizadas. No obstante lo anterior, la imprenta manual, -con insumos actuales-, todavía se sigue ocupando en algunas aplicaciones donde el número de reproducciones no es muy grande (tarjetas de presentación, invitaciones, propaganda sencilla, etc.).

1.5) PRESENTACIÓN DEL TRABAJO.

Actualmente se sabe que la materia, en cualquiera de sus estados, está constituida por moléculas y éstas a su vez por átomos. Desde el punto de vista clásico, el átomo está formado por un núcleo central y electrones girando a su alrededor, a manera de un sistema planetario diminuto. En el núcleo existen protones aglutinados por fuerzas nucleares, que constituyen la carga positiva del átomo, mientras que en los electrones orbitales reside la carga negativa del mismo. Generalmente, la materia se presenta eléctricamente neutra, ya que en promedio, la cantidad de cargas positivas en el núcleo se equilibra con la misma cantidad de electrones orbitales

Al transferir energía de algún tipo (mecánica, térmica, óptica, etc.) a un trozo de materia y por ende a sus átomos neutros, ésta puede ser suficiente para remover electrones de sus órbitas quedando por un lado átomos con exceso de carga positiva (iones +) y por el otro, electrones "libres" (cargas -). Este proceso de separación de cargas se observa en fenómenos de frotamiento, emisión termoiónica, ionización de gases por descargas eléctricas, etc. Una vez que se han separado cargas, éstas pueden reordenarse naturalmente si están en un medio adecuado, volviendo los átomos a su estado neutro; o bien pueden someterse a voltajes externos que las conduzcan a otros lugares dentro de regiones de alto vacío para evitar pérdidas de energía por

colisiones (figura 1.1). Esto último es parte del principio de funcionamiento del diodo de Flemming

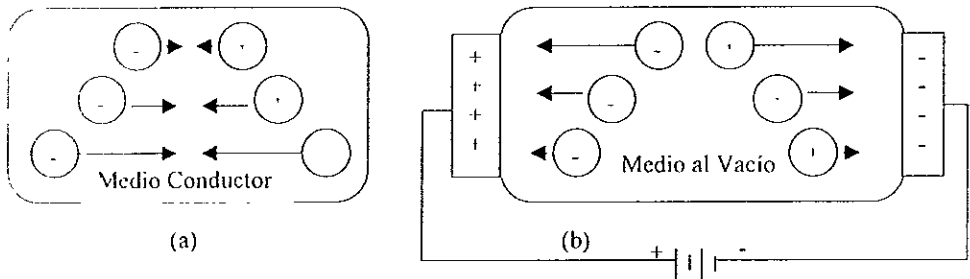


Figura 1.1.- (a) Reordenamiento natural de cargas previamente separadas.
 (b) Movimiento de cargas en el vacío, mediante voltajes externos.

En la física, uno de los campos de estudio es el movimiento de partículas cargadas (electrones, protones) desde el reposo hasta alcanzar velocidades cercanas a la de la luz; lo cual se conoce como **aceleración de partículas** y se utiliza para bombardear materiales ya sea con fines de análisis, o bien para producir cambios diversos en la materia, entre los cuales están las reacciones nucleares y sus ya conocidos usos y efectos. En síntesis, la aceleración de partículas involucra desde la separación de cargas, bombas para producir vacío en un medio determinado, voltajes externos elevados (kV o MV) para alcanzar las velocidades mencionadas, sistemas de seguridad, etc., donde todo el conjunto de equipos necesarios conforman el sistema conocido como **acelerador de partículas**, que se describe con más detalle en el capítulo 2.

A principios de los años 50, el Instituto de Física de la U.N.A.M. adquirió un sistema acelerador de partículas tipo Van der Graaff, de 2 MeV de energía, originalmente diseñado para partículas positivas, pero a los pocos años, -en esa misma década-, se modificó para acelerar electrones debido esencialmente a las áreas de investigación en la dependencia por esa época. Desde entonces y hasta la fecha, tal sistema se ha preservado en buenas condiciones de funcionamiento, ya que siempre ha recibido tanto el riguroso mantenimiento periódico, como algunas innovaciones que de manera natural surgieron por el avance mismo de la tecnología al respecto.

Las modificaciones más recientes surgen en 1994, al proyectar cambios en el sistema que a grandes rasgos contemplaron la actualización de equipos de vacío (bombas turbomoleculares en lugar de difusoras, entre otros) y el rediseño, construcción y montaje de equipos de calibración con electrónica actual. En 1997, ya como fase final del proyecto, se planeó este trabajo de tesis para implementar el **control por actualización** de todo el sistema acelerador; lo que comprendió el diseño, desarrollo, construcción e instalación de circuitos que permitieron acoplar una microcomputadora personal (tipo AT o superior) dedicada tanto a la operación y el control del acelerador, como a la captura y el manejo de datos de interés en el mismo; para lograr en lo posible el funcionamiento del conjunto en modo automático

Para efectos de lo anterior, se tomó como punto de partida al conocimiento más completo posible de dicho acelerador, tanto en lo referente a sus principios de funcionamiento, como a su modo de operación, -entonces sólo manual-, incluyendo todas las señales involucradas en el sistema (de seguridad, de operación/control y de datos). Esto permitió diseñar el esquema general del subsistema de control automático realizado, bajo la filosofía de modernizar el acelerador en adición a lo existente, a fin de preservar el modo de operación manual. El programa de actividades realizadas, que surgió en base a lo anterior, contempló tanto los circuitos electrónicos desarrollados ("hardware") como la programación necesaria ("software"), que permitieron usar a la microcomputadora mencionada para cubrir desde la rutina de encendido del acelerador, la exposición de muestras al bombardeo del haz emitido (con parámetros preestablecidos), capturando datos de interés tales como: voltaje de aceleración, intensidad de haz, tiempos de exposición, etc.; hasta la rutina de apagado del sistema, inspeccionando constantemente las señales de control y de seguridad, previendo que ante la posibilidad de cualquier falla se inhiba inmediatamente la radiación; y dependiendo de la magnitud del problema, se inicie o no el proceso de apagado del equipo ya sea secuencial o abrupto, para proteger primordialmente al usuario además de evitar daños mayores en el sistema mismo. Todo esto, se describe con más detalle en los dos capítulos siguientes.

REFERENCIAS DEL CAPÍTULO I

- 1) Deane Phyllis La primera revolución industrial. Península, Barcelona 1972.
- 2) O'Brien, Robert. Colección Científica de Time-Life, Máquinas. Time-Life, México 1972.
- 3) Purcell, Edward M. Electricity and Magnetism. McGraw-Hill, USA 1965.
- 4) Riveros R., Héctor G. Electricidad y Magnetismo. Trillas, México 1998
- 5) Ryder John D. Electronic Fundamentals and Applications. Prentice-Hall, USA 1959.
- 6) Vázquez M., Ruíz C., Fernández F., Adem E. Manual de operación y mantenimiento del acelerador Van de Graaff para electrones (2 MeV). Rep. int. FE002. I.F.-U.N.A.M. México 1992

CAPÍTULO 2

SISTEMAS ACELERADORES DE PARTÍCULAS

Un sistema acelerador involucra básicamente la separación o "generación" de partículas cargadas, el uso de bombas para producir vacío en el medio donde tales partículas se moverán, y voltajes del orden de kilovolts ó megavolts, que producen el campo eléctrico con la intensidad necesaria para que dichas partículas se aceleren desde casi el reposo, hasta velocidades cercanas a la de la luz

En el proceso de separación de cargas es donde se selecciona el tipo de partícula (+ ó -) por acelerar, y es de ahí de donde el sistema recibe el nombre de **acelerador de partículas positivas** ó bien **acelerador de electrones**. En el primer caso, la etapa que suministra las partículas positivas se conoce como fuente de iones. Un tipo de ésta consiste a grandes rasgos, en una cámara de vidrio que admite un gas a baja presión (hidrógeno, entre otros), el cual es sometido a energía electromagnética proveniente de una fuente externa de radiofrecuencia, que ioniza los átomos de dicho gas, es decir, separa los electrones de sus núcleos. los que quedan con exceso de carga positiva. Una fuente de voltaje de 1 ó 2 kV, con el polo positivo en uno de los extremos de la cámara, se encarga de recoger los electrones liberados y rechazar a los iones positivos hacia el lado opuesto, a la región donde se inicia su aceleración. En la figura 2.1. se ilustra una fuente de iones que proporciona núcleos de hidrógeno (protones).

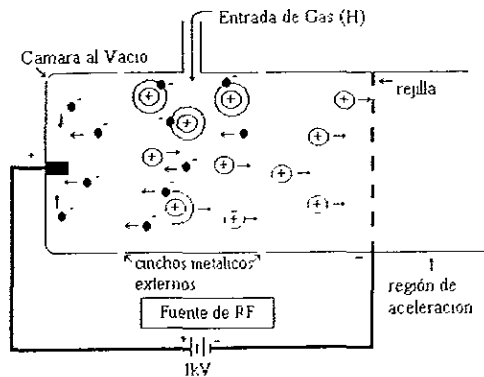


Figura 2.1.- Fuente de iones positivos (núcleos de hidrógeno).

En el acelerador de electrones, éstos se generan por emisión termoiónica, la que ocurre cuando fluye una corriente eléctrica a través de un filamento de algún material con punto de fusión elevado (tungsteno, entre otros), calentándolo a tal grado que emite electrones, además de luz y calor; similarmente a lo que ocurre en una bombilla (foco). Una rejilla cerca del filamento

con voltaje positivo respecto a éste, envía los electrones liberados hacia la región de aceleración. Al igual que en la bombilla, este proceso también es al vacío a fin de evitar que el filamento se quemara por oxidación, además de que los electrones generados no pierdan energía por colisiones con otras partículas. Todo esto también es parte del principio de funcionamiento de los bulbos, particularmente del triodo de Lee de Forest. En la figura 2.2, se muestra este tipo de fuente de electrones.

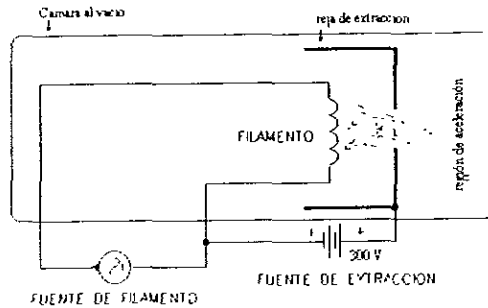


Figura 2.2.- Fuente de electrones.

En general, todos los aceleradores funcionan bajo el mismo principio: a las partículas cargadas se les somete a la acción de un campo eléctrico intenso, -generado por una gran diferencia de potencial-, el cual ejerce sobre ellas una fuerza que es la que las hace acelerarse. Una de las diferencias entre tipos de aceleradores es que la trayectoria que recorren las partículas puede ser recta o bien curva, con lo que el aparato ocupa un espacio menor. Entre los aceleradores del primer tipo, los mas conocidos son el de Van de Graaff y el llamado acelerador lineal, donde las partículas viajan por una serie de tubos rectos y en cada espacio entre un tubo y el siguiente, éstas sufren aceleración por una diferencia de potencial aplicada justamente en el momento preciso en que la partícula entra a cada tubo. Los aceleradores de recorrido curvilíneo, son diseñados de tal manera que mediante campos magnéticos y eléctricos las partículas describen trayectorias en espiral y llegan a determinados puntos en su recorrido en el instante justo para que reciban la aceleración electrostática. A continuación se describe con mayor detalle el acelerador Van de Graaff y posteriormente se hace una descripción breve de los aceleradores lineales y circulares.

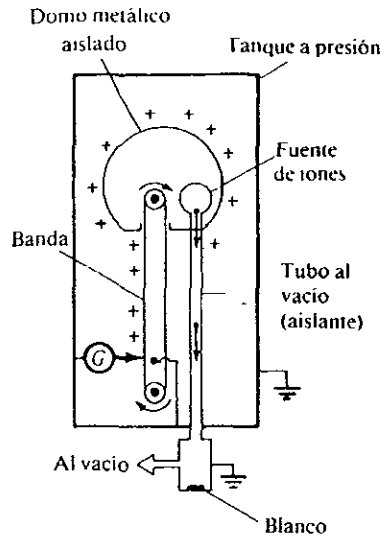
2.1) ACCELERADOR VAN DE GRAAFF

Cuando alguien menciona un "Van de Graaff", en lo que primero en que se piensa es en el efecto del cabello que se para de puntas, debido a que este tipo de acelerador acumula grandes cantidades de carga eléctrica en un domo semiesférico (potencial de aceleración); y cuando una

persona, -aislada de tierra-, toca el domo, las cargas corren por su cuerpo repeliéndose unas a otras, de tal manera que en el pelo, material muy flexible, cada cabello trata de estar lo mas alejado posible de los otros, provocándose el "erizamiento" mostrado en la figura 2.3-a.



(a)



(b)

Figuras 2.3.- (a) Efecto de "erizamiento" del pelo. (b) Esquema básico de un acelerador Van de Graaff.

El acelerador Van de Graaff no ha tenido muchos cambios desde su diseño y construcción por el Dr. Van de Graaff en el año de 1940. Su esquema básico se muestra en la figura 2.3-b, donde se pueden observar una banda transportadora de cargas, sujeta por unos rodillos movidos por un motor, dentro de una columna que sostiene al domo colector de carga

La banda y los rodillos son parte importante del acelerador, ya que dependiendo de su material, -entre otros-, así será la disposición para acumular cargas negativas ó positivas. Al respecto, hay dos formas de funcionamiento: la del tipo autoexcitado y por bombeo de cargas; en ambos casos la banda que gira sobre los rodillos tiene la función de recoger cargas en la parte inferior, transportarlas y depositarlas en el domo colector ubicado en la parte superior. En el tipo autoexcitado se aprovecha la fricción entre banda y escobillas (inferior y superior) para recibir y depositar dichas cargas. En el que opera por bombeo, se utilizan "peines" con puntas aguzadas en lugar de escobillas (a muy corta distancia de la banda, pero sin tocarla), con una fuente externa de relativo alto voltaje conectada al "peine" inferior, con el que se depositan las cargas sobre la banda móvil, aprovechando el efecto de puntas o "viento eléctrico". Un peine similar en la parte superior colecta dichas cargas en el domo mencionado, también llamado terminal de alto voltaje.

Con el alto voltaje así logrado, se dispone del campo eléctrico con la intensidad necesaria para acelerar partículas cargadas hasta velocidades cercanas a la de la luz; lo cual, como ya se dijo, debe hacerse en vacío de modo que la energía perdida por colisiones entre las partículas bajo aceleración con otras del medio donde se mueven, sea despreciable. Por otra parte, no es tan fácil mantener potenciales de aceleración mayores a algunos millones de volts, ya que ocurren efectos de corona o arco voltaico; de hecho en todos los aceleradores que operan con voltajes de cientos de kilovolts en adelante, se utilizan gases aislantes como nitrógeno con bióxido de carbono, para evitar tales efectos. Esto hace necesario que el domo colector o terminal de alto voltaje, banda y rodillos, tubo de aceleración, etc., estén encerrados en un tanque que contenga dicho gas a presión (figura 2.3-b). El límite superior de operación para el alto voltaje de aceleración se alcanza cuando se genera un arco del domo colector hacia el tanque contenedor (conectado a tierra), drenándose toda la carga acumulada. En la práctica, la diferencia de potencial entre la terminal de alto voltaje y el tanque, no puede ser mucho mayor a 10 Megavolts.

En la figura 2.4 se muestra otro esquema de un acelerador Van de Graff de electrones, donde se ilustra la distribución del alto voltaje a través de placas o platos equipotenciales encargados de aplicar el voltaje de aceleración a los electrones justo en el momento en que lo requieren. Nótese que parte de esa distribución de voltajes se utiliza para extracción y enfoque del haz de partículas. Aún cuando todos esos potenciales se simbolizan como fuentes, en la práctica se usan cadenas de resistencias en serie entre la terminal de alto voltaje y tierra como divisor de voltaje equivalente a la serie de potenciales ilustrada

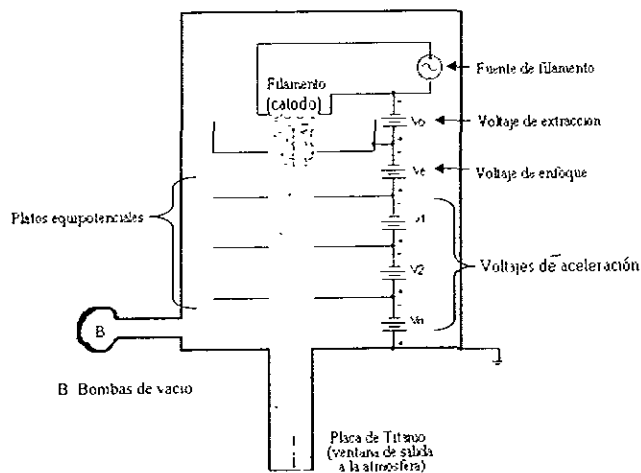


Figura 2.4.- Proceso de aceleración en un Van de Graaff.

Como ya se mencionó, el recorrido de las partículas bajo aceleración debe ser en vacío y la muestra a bombardear o irradiar debe colocarse al final del mismo; es decir cuando ya se tenga el haz de partículas aceleradas. Esto implica colocar dicha muestra dentro de la cámara de vacío o

bien en el ambiente exterior a muy corta distancia (cm), siempre que el haz de electrones pueda salir al "aire" a través de una ventana lo suficientemente delgada para permitir el paso de éstos, pero con el espesor adecuado para mantener el vacío necesario en el tubo de aceleración. Un ejemplo práctico de tal ventana es una placa de titanio de 60 micras de espesor.

La maquina de Van de Graaff es un dispositivo muy útil para acelerar partículas hasta energías de algunos millones de electrón volts. Proporciona un haz de partículas uniforme y bastante intenso con energía constante y conocida. Para magnitudes hasta cerca de 3 MeV es un instrumento valioso y muy usado. Ya que los efectos de arco son un problema serio para diferencias de potencial muy elevadas, el acelerador Van de Graaff no se emplea para energías mayores a 5 MeV, aunque algunos arreglos en forma de cascada (tandem) pueden proporcionar una energía equivalente a 30 MeV.

A continuación se describen muy brevemente otros tipos de aceleradores, en cuanto a trayectoria de haz de partículas se refiere, sin más intención que el mencionar su existencia y principio básico de funcionamiento, omitiendo mayores detalles por quedar éstos fuera del enfoque de este trabajo.

2.2) ACELERADORES LINEALES

Para alcanzar energías mayores que las que son posibles con el acelerador de Van de Graaff, se utilizan otras técnicas. En el acelerador lineal se emplea una numerosa serie de tubos metálicos (en el vacío) como se muestra en la figura 2.5; en el extremo izquierdo del primer tubo se inyecta la partícula ya con una energía inicial, que va a ser sobreacelerada; por lo común se usa un acelerador auxiliar para proporcionar esta energía inicial. Se ajustan entre los tubos siguientes diferencias de voltaje alternantes de modo que la partícula sea acelerada cada vez que pasa de un tubo al inmediato posterior. La energía final de la partícula está determinada por la diferencia de potencial entre los tubos y el número de los mismos en la serie.

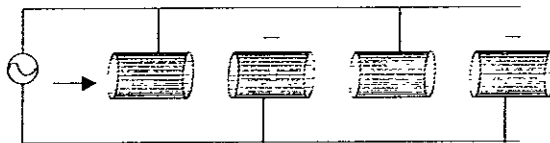


Figura 2.5.- Esquema de un acelerador Lineal.

2.3) ACELERADORES CIRCULARES

Otro modo de lograr energías elevadas consiste en que las partículas se muevan en una trayectoria circular o espiral, pasando a través de la misma diferencia de potencial encargada de proporcionar impulsos de aceleración cada vez que se completa una vuelta. La partículas se mantienen en dicha trayectoria mediante campos magnéticos que sólo proporcionan la fuerza

centrípeta necesaria a tal recorrido, sin cambiar la energía correspondiente a la velocidad tangencial de la partículas. Por este método se pueden acelerar protones a energías cercanas a 500 GeV

Entre los aceleradores circulares se encuentra el ciclotrón, que se ilustra en la figura 2.6, en el cual las partículas recorren un camino en espiral ascendente mediante la variación apropiada de los campos magneticos aplicados en relación a los impulsos de potencial de aceleración.

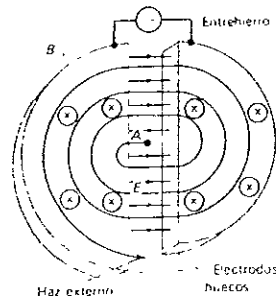


Figura 2.6.- Esquema de un Ciclotrón.

REFERENCIAS DEL CAPÍTULO 2

- 1) Giancoli Douglas, C. Física General (Vol. II). Prentice-Hall Hispanoamericana México 1984
- 2) Livingston M. Stanley. Particle Accelerators: a Brief History. Harvard University Press: Cambridge, Mass. USA 1969.
- 3) Livingston M. Stanley and Blewett John P. Particle Accelerators McGraw-Hill, USA 1962.
- 4) McGrath, Susan. La Física es diversión National Geographic Society México 1993.
- 5) Persico Enrico, Ferrarì Ezio, Segre Sergio E. Principles of Particle Accelerators. W.A. Benjamin, New York 1968.
- 6) Ryder John D. Electronic Fundamentals and Applications. Prentice-Hall, USA 1959.

CAPÍTULO 3

TRANSDUCTORES, SEÑALES Y PROCESOS EN EL ACELERADOR

3.1) EL ACELERADOR VAN DE GRAAFF DEL INSTITUTO DE FÍSICA DE LA UNAM

Fue construido por la compañía "High Voltage Engineering Corp." en Burlington, Mass. U.S.A. e instalado en los laboratorios Van de Graaff del Instituto de Física de la UNAM en 1952 como acelerador de iones positivos; posteriormente, a finales de la misma década, fue transformado en acelerador de electrones, que es como persiste hasta la actualidad.

Ocupa dos grandes espacios, que se pueden llamar sala de control y zona aislada. La sala de control, que se muestra en la figura 3.1, por normas de seguridad está ubicada relativamente lejos de la zona de aceleración y es de aquí desde donde se opera todo el acelerador, vigilando los procesos de irradiación mediante un circuito cerrado de televisión y los detectores y equipos adecuados a los mismos.

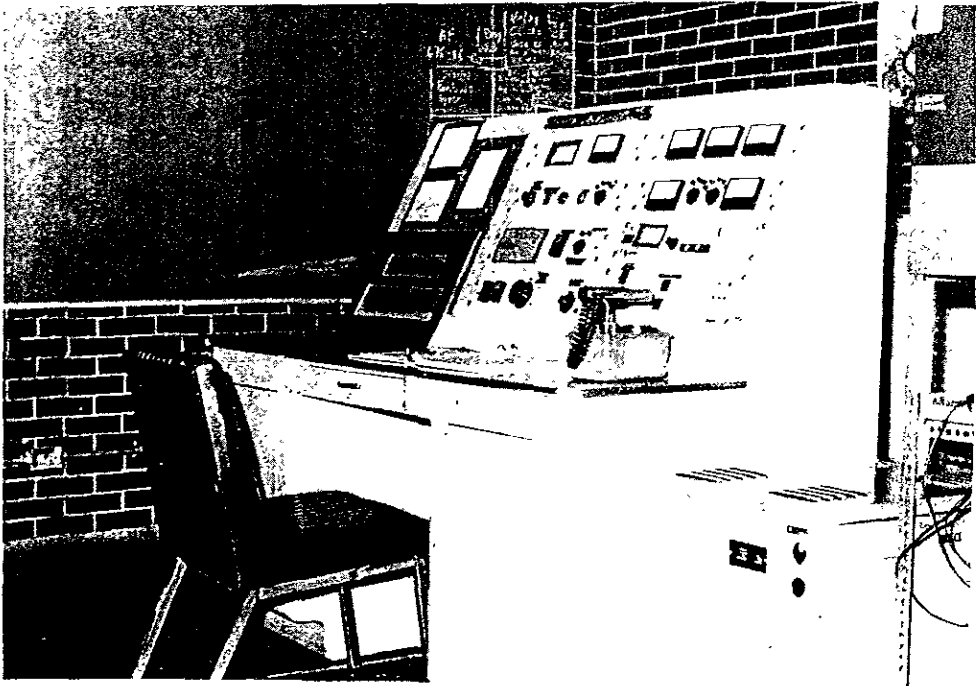
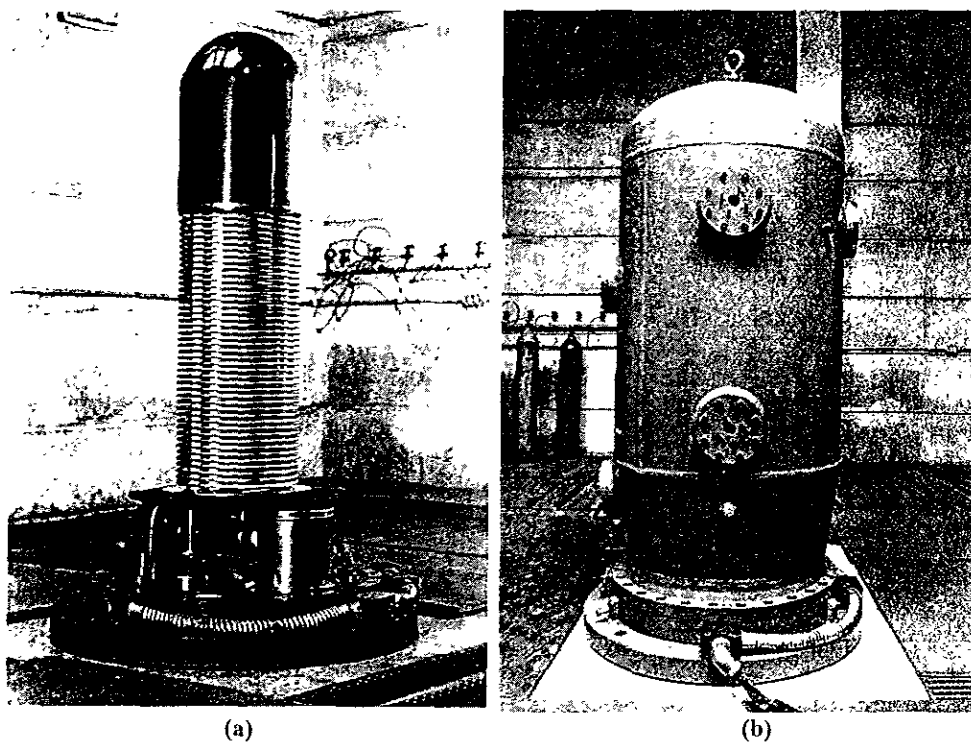


Figura 3.1.- Sala de control del acelerador Van de Graaff de 2 MeV del IFUNAM.

El espacio de la zona aislada es el más grande y está repartido en dos niveles por un mezanine; es ahí donde se encuentra la mayor parte del equipo de aceleración; a saber, en el nivel superior se encuentran: el domo colector de carga o terminal de alto voltaje, los platos equipotenciales, la fuente de electrones, el motor, los rodillos, la banda transportadora de cargas, el tubo de aceleración, etc (figura 3.2-a); todo ello dentro del tanque contenedor (figura 3.2-b) presurizado con gas aislante para evitar los ya mencionados arcos voltaicos indeseables.



Figuras 3.2.- Etapa de aceleración del sistema Van de Graaff de 2 MeV del I.F.U.N.A.M. (nivel superior del mezanine). (a) Al descubierto. (b) Tanque contenedor.

En la parte inferior del mezanine están los sistemas de vacío para el tubo de aceleración (1.3 m de longitud), el cual es continuado por otro tubo, -llamado de extensión-, que termina en una boquilla ancha en forma de sección axial de una campana, que permite deflexiones angulares del haz de partículas aceleradas (para efectos de barrido) antes de salir a la atmósfera por la parte inferior de la misma; esto último dentro de un cuarto de irradiación (figura 3.3), con paredes de plomo, para confinar los rayos X que se producen al irradiar las muestras con electrones de muy alta energía. De hecho toda la zona aislada recibe su nombre por la seguridad que debe mantenerse en ella.

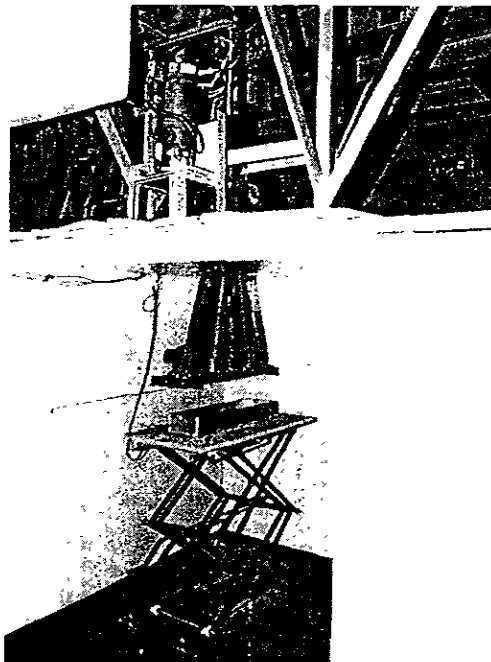


Figura 3.3.- Cuarto de irradiación del acelerador Van de Graaff de 2 MeV.

3.2) FUNCIONAMIENTO

La figura 3.4 corresponde a un diagrama de la etapa de aceleración del sistema (figuras 3.2 anteriores), donde se ilustran sus componentes internos principales; posteriormente, en la figura 3.5 se presenta otro esquema que ilustra a grandes rasgos el conjunto de aceleración ubicado en la ya mencionada zona aislada (ambos niveles del mezanine). Con referencia a estos diagramas, se describe a continuación el funcionamiento de este acelerador en particular.

Con base en lo descrito en el capítulo anterior, este acelerador es del tipo de bombeo de cargas eléctricas, las que son depositadas a la banda por medio de la fuente de alimentación, a través de un peine que forma un campo electrostático contra la polea del motor (1); la banda (2) acarrea mecánicamente las cargas a la terminal de alto voltaje (3). En la parte alta, las cargas son transferidas de la banda a la terminal por medio de una escobilla (4), las que por repulsión se desplazan a la superficie del domo colector, estableciéndose así un alto potencial con respecto a la base del acelerador. La terminal de alto voltaje (domo colector) está sostenida por medio de una columna de platos equipotenciales y acrílico kovar, que se aíslan del tanque (5) por una atmósfera de nitrógeno y bióxido de carbono (6), en relación 4:1, comprimidos a una presión entre 250 y 350 psi, para evitar el arqueo de la terminal al tanque.

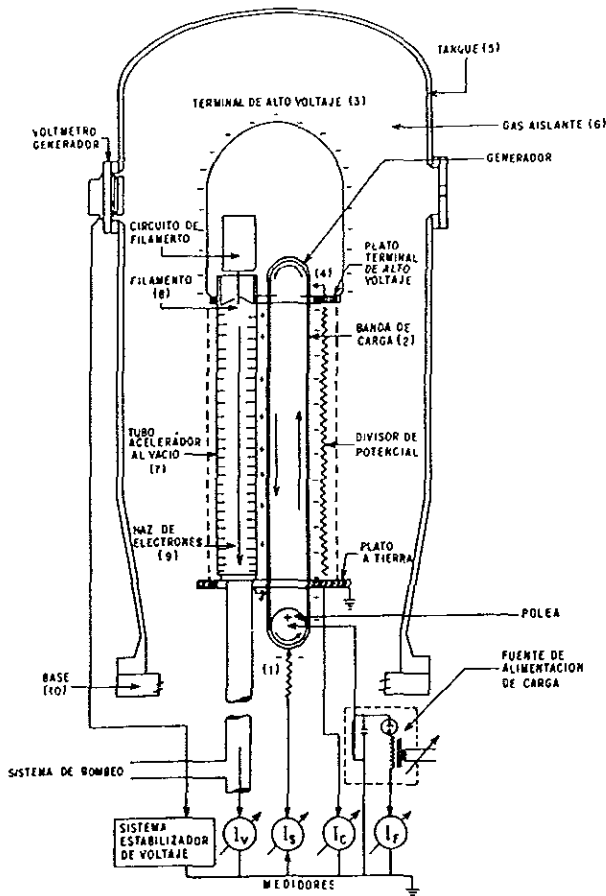


Figura 3.4.- Corte transversal del tanque contenedor de la etapa de aceleración.

El tubo acelerador (7), también de acrílico kovar y con los platos equipotenciales distribuidos a lo largo del mismo, con espacios de 2.5 cm, mantiene su interior a un "alto vacío" (10^{-6} a 10^{-7} Torr) para proveer una salida libre a los electrones que se producen en un filamento incandescente (8) dentro de la terminal, cuyo enfoque se logra mediante la diferencia de potencial entre el electrodo envolvente del filamento y el primer plato equipotencial del tubo. Los electrones son acelerados (9) a velocidades altas por el campo electrostático creado por la *diferencia de potencial* entre la terminal de alto voltaje y la base del acelerador (10), formando un haz de alta energía que sale del acelerador a través de una ventana de titanio (11) de 60 μm de espesor, que aísla el vacío de la atmósfera (ver figura 3.5). La zona de extracción del haz es muy peligrosa, por la generación de rayos X y de electrones difusos, por ello el acceso a la zona del acelerador está restringido y más aún al cuarto de irradiación con el acelerador funcionando.

En determinados usos se desea que el haz sea barrido sobre alguna(s) muestra(s); para ello se emplea un sistema de bobinas deflectoras (12) que producen un campo magnético oscilante y variable en amplitud y frecuencia. Así, las deflexiones angulares del haz (13) se traducen, -al salir a la atmósfera-, en barridos longitudinales a diversas frecuencias entre 1 y 200 Hz; y amplitudes variables en el intervalo de 2 a 70 cm. Tales bobinas se operan mediante un sistema electrónico desde la sala de control.

El "alto vacío" en un volumen de 120 litros correspondientes al tubo acelerador, extensión y región de barrido; se logra mediante un sistema de 2 bombas (14): una mecánica para pre-vacío (10^{-1} Torr) y una turbomolecular con la que se alcanza del orden de 10^{-7} Torr.

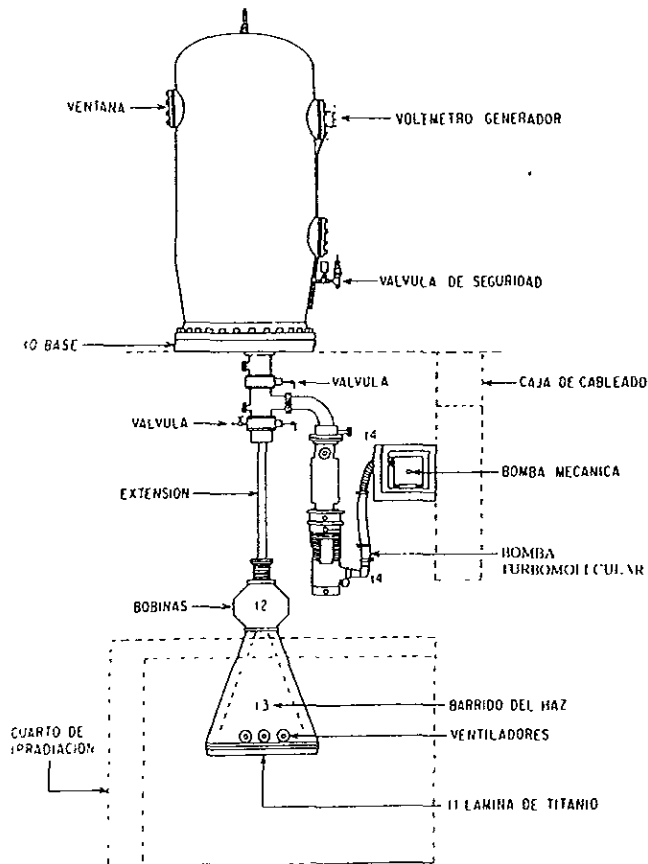


Figura 3.5.- Diagrama del conjunto de aceleración en la zona aislada.

También se tiene un sistema de enfriamiento necesario para la operación del acelerador, mediante un circuito de agua recirculada que fluye por radiadores internos y con la opción de utilizar el intercambiador de calor colectivo de la dependencia; por ello es importante que las bombas encargadas de mover tal refrigerante y los ventiladores de sus radiadores estén encendidos antes, durante y después de la operación del acelerador.

El alto voltaje de aceleración normalmente se opera en el intervalo de 0.5 a 1.5 MV, pudiendo alcanzar los 2 MV en casos de irradiaciones específicas. Cuenta con una etapa analógica estabilizadora de alto voltaje, instalada de fábrica, que trabaja continuamente a lazo cerrado, sensando el alto voltaje de la terminal mediante un voltímetro generador ubicado en la parte superior del tanque contenedor (figuras 3.4 y 3.5). El circuito estabilizador de alto voltaje, que se encuentra en la consola de controles, también permite establecer la magnitud del alto voltaje deseado mediante el giro de un helipot, que internamente es un divisor de voltaje variable en el intervalo de 0 a 7.5 Vcd.

3.3) MODO DE OPERACIÓN

Antes de iniciar la operación del acelerador, se deben cubrir las siguientes reglas de seguridad, llenando el formato correspondiente de Seguridad Radiológica.

- i. Conectar el medidor de ionización de alto vacío, comprobando lecturas entre 5×10^{-6} y 10^{-7} Torr. Opcionalmente, conectar el sistema de barrido del haz y confirmar la señal correspondiente en el osciloscopio.
- ii. Conectar los ventiladores de enfriamiento de la ventana de titanio en el cuarto de irradiación, confirmando su funcionamiento.
- iii. Conectar el extractor del cuarto de irradiación y los extractores generales de la zona aislada, verificando el funcionamiento de todos ellos. Colocar la(s) muestra(s) a irradiar.
- iv. Revisar que la presión del gas aislante en el tanque contenedor de la etapa de aceleración, este entre 250 y 350 psi.
- v. Cerrar con llave las puertas de acceso a la zona aislada, así como las puertas de la sala de control y verificar que todo el personal tenga colocado su monitor de radiación (dosímetro). Prender los contadores Geiger y de Centelleo (en escala 0.25 mr/h).
- vi. Anunciar por el sonido local, por lo menos tres veces, que se iniciará el proceso de irradiación, indicando el peligro a posibles personas encerradas en la zona aislada.

La operación del acelerador se describe en los puntos siguientes:

1. Se debe activar el encendido general de la consola de controles mediante un interruptor de 1P-1T, solo accesible mediante una llave en la cerradura correspondiente.
2. Se activa el motor de la banda de carga mediante otro interruptor de 1P-1T, el cual se encuentra en serie con otros "interlock's" ubicados en las puertas de acceso a la zona aislada y en el sensor de temperatura del agua de refrigeración ; de tal forma que si alguna de dichas puertas está abierta o la temperatura del refrigerante excede 80 °C, dicho motor no arranca.

3. Ya con el motor encendido, hay que esperar unos cuantos segundos hasta que se presenta el aviso de "radiación lista" en la consola de controles, indicando que todos los sistemas del acelerador funcionan correctamente hasta el punto anterior. En ese momento se debe accionar un interruptor de 1P-2T que apaga el aviso anterior y enciende otro de "radiación apagada", señal necesaria para poder suministrar la energía que requiere la etapa de aceleración.
4. El primer paso de dicho suministro de energía, consiste en alimentar el primario de la fuente de la alimentación de cargas a la banda del acelerador, mediante el giro suave y uniforme de un variac ubicado en la consola de controles que entrega un voltaje de c.a. de 83 Vrms a partir de cero.
5. Posteriormente, se debe girar también suave y uniformemente, un helipot que proporciona un voltaje de c.d. variable desde 0 hasta 5.6 Vcd, el cual mediante otros circuitos contenidos en la consola, permite que en la terminal de alto voltaje del acelerador se puedan alcanzar hasta 2 MV. Entre esos circuitos hay detectores de nivel tales que al tener 1.2 Vcd en dicho helipot, automáticamente el estado del acelerador cambia a "radiación lista" con un potencial de aceleración mínimo de 0.5 MeV. Así, el giro en el helipot debe hacerse hasta que éste proporcione algún voltaje de c.d. entre 1.2 y 5.6 Vcd a fin de establecer el alto voltaje deseado dentro del intervalo de 0.5 a 2 MV.
6. Con el potencial de aceleración establecido en la terminal de alto voltaje y con el aviso de "radiación lista", se procede a prender el filamento de la fuente de electrones hasta obtener un máximo de corriente del haz ya en la atmósfera, para después reducirlo a un valor nominal de operación, que es el que se utiliza en toda la sesión de irradiación. La detección y medida de dicho haz es con una caja de Faraday ubicada bajo la campana de barrido, que colecta el haz de electrones acelerados justo después de haber salido a la atmósfera y envía una corriente eléctrica a la consola, proporcional al mismo, que es desplegada al operador en un microamperímetro analógico. Lo anterior antiguamente se lograba mediante varios giros en la perilla de un sincromotor "selsyn" de la consola, cuya pareja (otro sincromotor) se encontraba en la base del acelerador, cerca de la polea inferior de la banda; y transmitía el giro mecánico mediante una barra de lucita hasta la terminal de alto voltaje (atravesando todos los platos equipotenciales), donde se encuentran los potenciómetros que limitan las corrientes de filamento y de emisión del haz, mediante engranes. Recientemente tales sincromotores fueron sustituidos por un solo motor a pasos (en la base del acelerador) e interruptores en la consola de controles que permiten el avance o retroceso del motor a dos velocidades distintas; conociendo en todo momento la posición de su eje (en cuanto a giro) por una etapa que regresa la señal correspondiente hacia la consola de controles.

En este momento, ya se tiene al acelerador emitiendo un haz de electrones de alta energía que bombardean la(s) muestra(s) colocada(s) bajo la campana de barrido, junto a la caja de Faraday, con los parámetros antes establecidos (alto voltaje de aceleración, corriente de filamento y emisión de haz). Todos los pasos anteriores se conocen como rutina de encendido, la que se puede resumir en: encendido general de la consola, activación del motor de la banda, apagado de radiación, alimentar la fuente de cargas, establecer el alto voltaje de aceleración y finalmente prender el filamento y establecer la corriente del haz. Después de esto viene el tiempo de exposición de las muestras a la radiación, con el barrido del haz en forma opcional, hasta que transcurrido el mismo (medido con un cronómetro externo) debe de efectuarse la rutina de

apagado del sistema, que es igual a la de encendido pero efectuada en orden inverso; es decir, bajar corriente del haz y apagar filamento, reducir a cero el potencial de aceleración, quitar la alimentación a la fuente de cargas, dejar el acelerador en "radiación lista", apagar el motor de la banda y finalmente cortar la alimentación de la consola de controles mediante la llave mencionada

3.4) SEÑALES INVOLUCRADAS

Para dedicar una microcomputadora a la operación y el control del acelerador, se plantearon las funciones a realizar por ésta para reproducir los pasos de su operador humano, pero siempre bajo supervisión del mismo; las que básicamente son: rutina de encendido, tiempo de exposición y rutina de apagado, que se resumen en el diagrama de flujo de la figura 3.6.

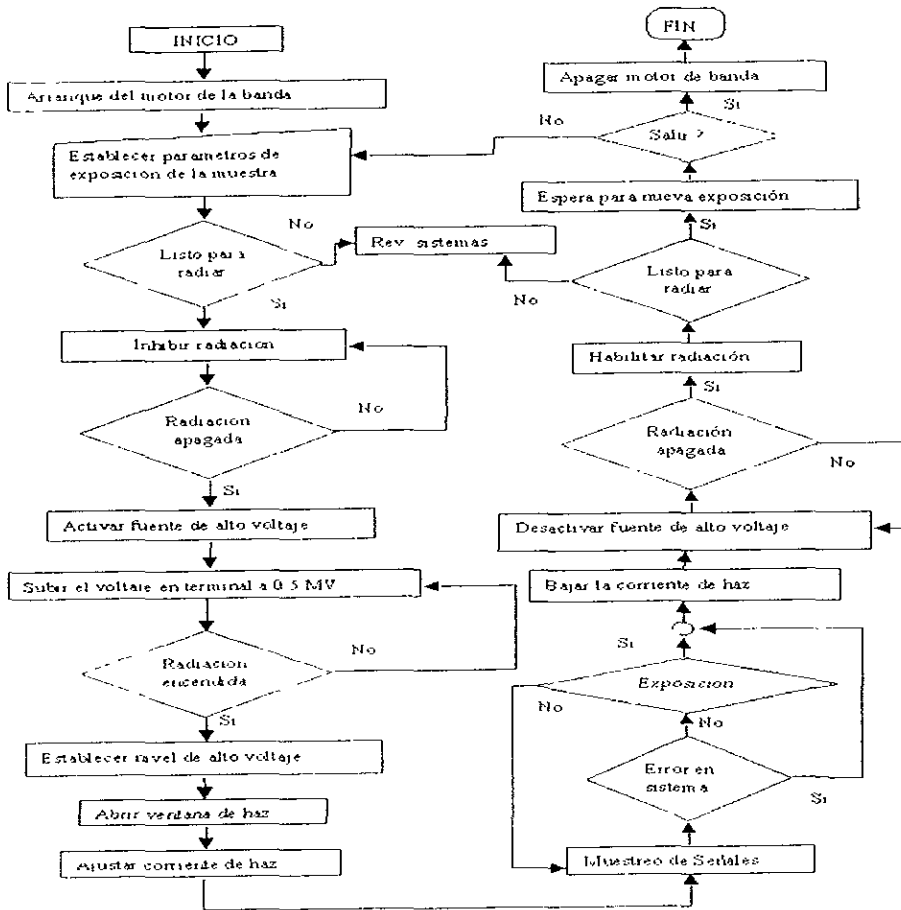


Figura 3.6.-Diagrama de flujo del funcionamiento del acelerador.

Lo anterior y el conocimiento del funcionamiento del acelerador, permitió contemplar las señales tanto analógicas (A) como digitales (D) involucradas en el proceso de operación antes descrito. llamando señales de salida (S) a las que se envían al acelerador (a través de la consola) por parte del operador, y señales de entrada (E) a todos los avisos y lecturas que el acelerador devuelve a dicho operador. Así, una señal analógica de entrada se identifica como "EA" (entrada analógica), una digital de salida como "SD" (salida digital), etc. A continuación se mencionan todas esas señales con referencia a los puntos listados para la operación del acelerador. Esto fue necesario para planear las características del módulo controlador diseñado y su interacción con la microcomputadora dedicada a todo el conjunto.

El encendido general de la consola, por seguridad se decidió dejarlo exclusivamente por intervención humana. Una señal SD1 debe activar el motor de la banda y otra ED1 debe confirmar el aviso de radiación lista. Mediante una segunda señal de salida digital SD2 debe inhibirse la radiación, confirmándose con la correspondiente ED2. Para el suministro de energía de la fuente de alimentación de cargas (punto 4 anterior), primero se pensó en un servomotor que activado por la microcomputadora moviese suavemente y lo necesario al variac mencionado; sin embargo bajo la filosofía de que en la adaptación de un sistema de este tipo no solo se debe acercar el controlador a la planta, sino también la planta al controlador; se diseñó, construyó e instaló en la consola de controles del acelerador, un circuito electrónico analógico que al recibir una señal digital reproduce la salida del variac cuando este se mueve en la forma descrita. Tal circuito se describe con detalle en el capítulo 7; por el momento basta mencionar la necesidad de una tercera señal SD3 para tal efecto. En lo correspondiente al punto 5 del modo de operación, fue necesario un convertidor digital-analógico (SA) para proporcionar los voltajes de c.d. que establecen el potencial de aceleración; además de otra señal ED3 para confirmar el estado de radiación lista en el acelerador cuando en éste se tienen cuando menos 0.5 MV de alto voltaje. Como la lectura de este último debe vigilarse continuamente, también fue necesario contemplar un convertidor analógico-digital (EA1). Para prender el filamento de la fuente de electrones y obtener corriente de haz, de acuerdo a lo mencionado en el punto 6 del modo de operación, se contemplaron otras 4 señales de salida digital (SD4, SD5, SD6 y SD7) para el encendido del motor a pasos, selección de avance o retroceso del mismo y escoger velocidad de giro rápida o lenta, respectivamente; además de otra conversión analógica-digital (EA2) destinada al monitoreo de la corriente de haz obtenida ya en la atmósfera, mediante la caja de Faraday mencionada

En resumen, el diseño del módulo controlador implementado requirió las etapas que a continuación se listan.

- a) Un convertidor digital-analógico, con resolución de 12 bits, exclusivo para establecer el alto voltaje de aceleración, inicialmente en 500 KV de arranque y posteriormente en cualquier valor superior hasta 2 MV, durante los tiempos de exposición de muestras
- b) Un convertidor analógico-digital, también en 12 bits, para capturar datos de hasta cuatro señales de entrada, de las que dos son reservadas para confirmar el alto voltaje de aceleración y la intensidad del haz emitido; y las dos restantes para posibles usos futuros.
- c) Diez señales digitales de salida. Dos de ellas permiten seleccionar la señal analógica deseada para el convertidor A/D del punto anterior; y ocho para activar o desactivar

subsistemas del acelerador, incluyendo encendido gradual de filamento y emisión de haz

- d) Siete señales digitales de entrada, que en su mayoría vigilan señales de seguridad durante todo el tiempo en que el equipo está operando.

El tener en el acelerador sesiones de funcionamiento relativamente largas (horas), con intercambios de información lentos (segundos), permitió decidir por comunicación tipo serie, asíncrona y en protocolo RS-232C; esto implicó incluir en el controlador la captura y conversión de datos tanto serie a paralelo como paralelo a serie, además del almacenamiento temporal de los mismos. En la figura 3.7 se muestran las etapas que se planearon para dicho módulo controlador, donde se incluye el número de bits utilizados en cada "bus" de interconexión.

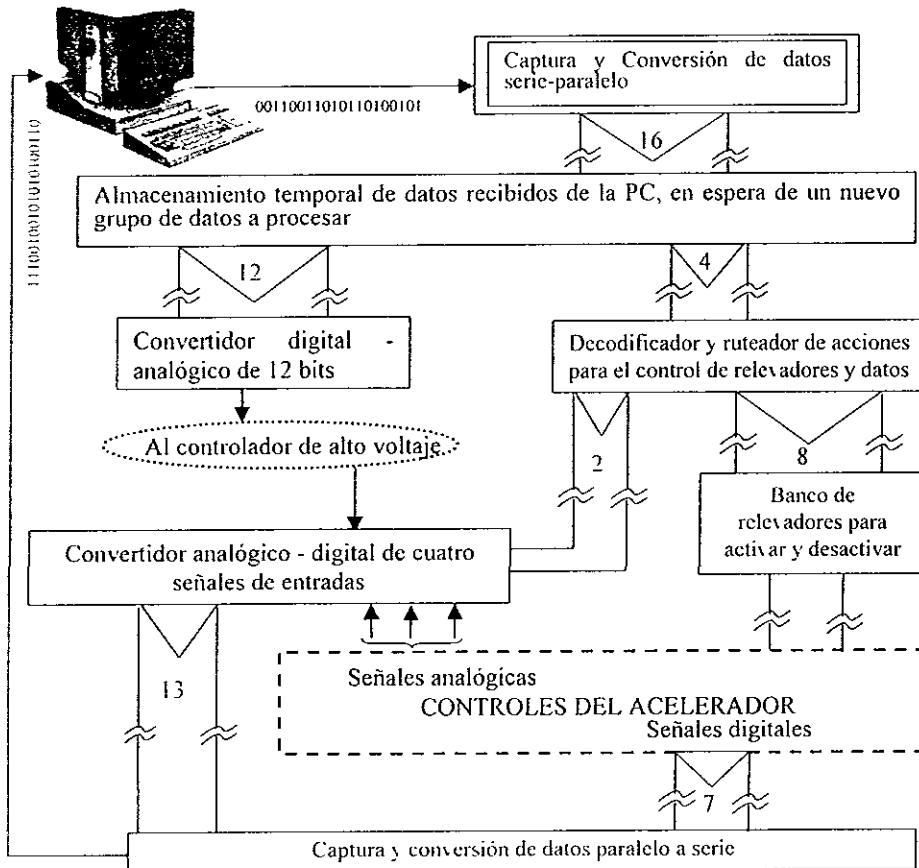


Figura 3.7.-Diagrama a bloques del módulo controlador que acopla la consola de mandos del acelerador a una microcomputadora personal tipo AT286 o superior.

Para llevar a cabo lo anterior, fueron necesarios algunos conocimientos de la teoría de control, para el proceso adecuado de señales analógicas y digitales en el sistema híbrido acelerador-microcomputadora; lo que a su vez requirió del conocimiento de interfaces, computación dedicada a la instrumentación y el manejo de datos a través del lenguaje apropiado. Todo esto se desglosa en los tres capítulos siguientes, para posteriormente concluir este trabajo con el detalle de todas las etapas y circuitos desarrollados, sus evaluaciones parciales y finales; y todas las observaciones que hicieron posible la automatización de dicho acelerador; así como las conclusiones pertinentes al respecto.

REFERENCIAS DEL CAPÍTULO 3

- 1) Bygrave William D., Treado Paul A., Lambert James M. Accelerator Nuclear Physics. Fundamental Experiments with a Van de Graaff Accelerator. High Voltage Engineering Corp., USA 1970.
- 2) Vázquez M., Ruíz C., Fernández F., Adem E.. Manual de operación y mantenimiento del acelerador Van de Graaff para electrones (2 MeV). Rep. int. FE002. I.F.-U.N A.M. México 1992.

CAPÍTULO 4

TEORÍA DE CONTROL. SEÑALES ANALÓGICAS Y DIGITALES

Un sistema, visto como una combinación de componentes que actúan conjuntamente para proporcionar determinada(s) respuesta(s) o salida(s), requiere necesariamente del estímulo o entrada correspondiente a las mismas. Lamentablemente, todo sistema es susceptible a perturbaciones, señales indeseables que tienden a afectar adversamente su salida. Si la perturbación se genera dentro del sistema se le denomina interna, pero si ésta ocurre externamente entonces constituye una entrada indeseable.

Generalmente los sistemas tienen algún tipo de control en su funcionamiento. Hay sistemas planeados con respuesta variable, en los que el controlador primordialmente permite modificar sus características de salida mediante los cambios apropiados en los parámetros de entrada. En otros, cuya respuesta se desea constante o invariante en el tiempo, el controlador se incluye para efectuar cambios de corrección en sus estímulos, que garantizan la estabilidad de dicha respuesta. Un tercer tipo de sistemas en cuanto a control se refiere, es el de respuesta variable y estabilidad alta; en éste, el controlador permite el ajuste a la respuesta deseada y garantiza la estabilidad de la misma mientras no se efectúe un nuevo ajuste.

El concepto de retroalimentación o realimentación, es muy importante en el diseño de los sistemas de control, ya que con ésta se pueden modificar sustancialmente las características de funcionamiento de un sistema inicialmente no retroalimentado. En la figura 4.1 se muestra un sistema con función de transferencia "A", -como sistema aislado-, que es retroalimentado con otro cuya transferencia está dada por "B"; es fácil demostrar que la función de transferencia "A_R" de todo el conjunto es:

$$A_R = \frac{A}{1 - AB} \quad \dots (4-1)$$

así, la magnitud de A_R puede ser mayor, igual o menor que la de A, dependiendo del valor del producto AB. La presencia o no de la retroalimentación, permite clasificar el control de los sistemas en dos grandes grupos: a lazo cerrado y a lazo abierto, que se describen a continuación.

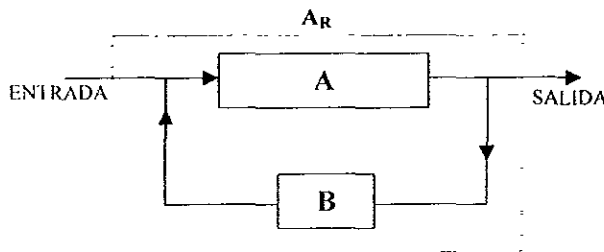


Figura 4.1.- Esquema básico de la retroalimentación.

4.1) CONTROL DE LAZO CERRADO

Un sistema de control por lazo cerrado es aquel en donde la señal de salida tiene efecto directo sobre la acción de control. A este tipo de control se le conoce también como un sistema de control realimentado y funciona de la forma siguiente: dentro de los elementos de medición, parte de la señal de salida es captada por un sensor de muestra que entrega una señal, -proporcional a la misma-, a un comparador que también es alimentado por otra señal que puede ser de entrada o de referencia: la salida del comparador, conocida como señal de error actuante (diferencia entre la señal de muestra y la de entrada), es enviada al detector o controlador de manera que éste realiza las modificaciones necesarias para reducir el error y llevar la salida del sistema al valor estable deseado. En la figura 4.2 se muestra un diagrama a bloques de este tipo de control. En la figura 4.3 se ilustran dos formas de controlar a lazo cerrado.

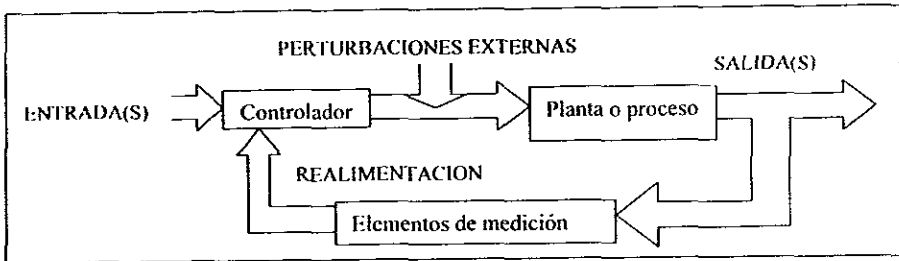


Figura 4.2.- Diagrama a bloques de un control a lazo cerrado.

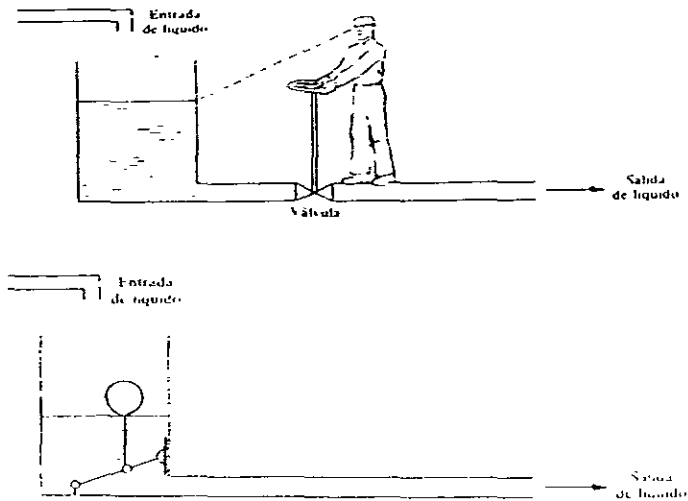


Figura 4.3.- Ejemplo de un control a lazo cerrado manual y automático.

4.2) SISTEMAS DE CONTROL DE LAZO ABIERTO

En los sistemas de control de lazo abierto, la salida no tiene efecto alguno sobre la acción de control; esto es, el sistema está controlado únicamente con la entrada sin realimentación alguna para comparar la salida. Al no existir retroalimentación ni comparación, la operación es fijada únicamente por la entrada; por ello la exactitud del sistema depende fuertemente de la calibración de sus componentes y entrada(s); y por consiguiente así será la respuesta que se desea obtener. Si hay perturbaciones, ya no se cumple con el control de estabilidad porque cualquier señal no prevista e indeseable, produce también una salida no esperada. Por esto, entre otros, los controles a lazo abierto sólo se utilizan en sistemas donde existen relaciones de entrada/salida conocidas y no existen perturbaciones internas ni externas considerables. En la figura 4.4 se muestra el diagrama a bloques de este tipo de control. En la figura 4.5 se ilustran un sistema de freno automotriz y una prensa hidráulica como ejemplos de control a lazo abierto.

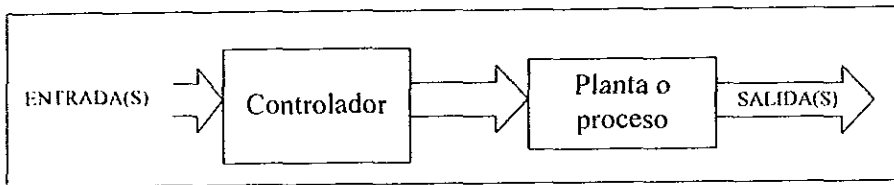


Figura 4.4.- Diagrama a bloques de un control a lazo abierto.

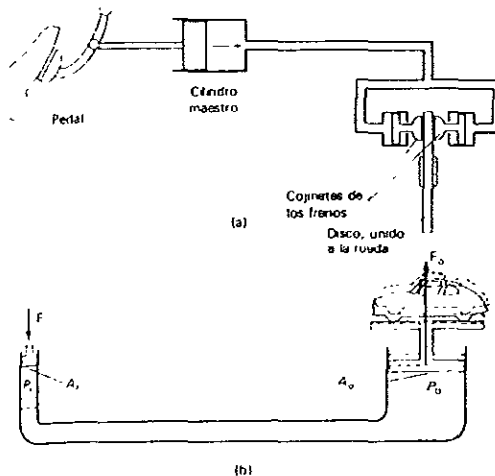


Figura 4.5.- a) Sistema de frenado automotriz. b) Prensa hidráulica.

4.3) OTROS CONCEPTOS EN LA TEORÍA DE CONTROL. UN EJEMPLO.

De analizar las características generales de los dos grandes grupos de formas de control mencionados, surgen las consideraciones siguientes.

El control por lazo cerrado tiene como gran ventaja que por el uso de la retroalimentación, la respuesta del sistema al que se aplica es prácticamente insensible a perturbaciones externas y puede también corregir algunas variaciones internas no específicamente contempladas de diseño, tales como cambios en el funcionamiento de partes por desgaste natural, generación de ruidos, etc. Esto permite en ocasiones disponer de componentes económicas y relativamente imprecisas para lograr una exactitud aceptable en el control.

En cuanto a tiempo para alcanzar la salida deseada, los sistemas a lazo abierto son mucho más rápidos y se estabilizan (de momento) más fácilmente. En los de lazo cerrado, el alcanzar la respuesta requerida puede ser relativamente más tardado y con el riesgo de que se puede llegar a estados inestables, contrariamente a lo que se pretende, con las consecuentes oscilaciones en la respuesta.

No se puede afirmar que para un sistema complejo el mejor tipo de control sea de lazo cerrado ó de lazo abierto; ello depende esencialmente de las características propias del sistema, incluyendo la exactitud y estabilidad permitidas en su(s) respuesta(s); en muchos casos se puede tener a todo el conjunto funcionando satisfactoriamente si se opta por una combinación adecuada de controles a lazo cerrado y a lazo abierto.

Dado el grado de dificultad que presentan los controles a lazo cerrado con respecto a los de lazo abierto, se mencionan a continuación otros aspectos de los primeros.

En los sistemas retroalimentados que operan en modo automático, se debe detectar la señal de error actuante que habitualmente se encuentra a un nivel de potencia bajo; esto hace necesario un amplificador ya que la señal de control automático comúnmente actúa sobre un dispositivo de potencia como puede ser un motor ó válvula, que en general se conoce como controlador o accionador (ver figura 4.6).

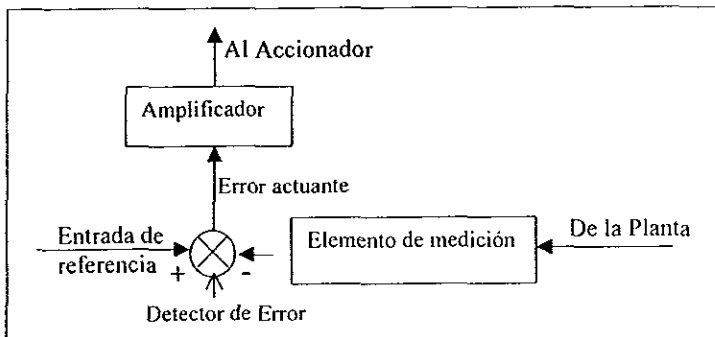


Figura 4.6.- Necesidad de amplificadores en controles retroalimentados.

Ya se menciona que un control retroalimentado consiste en un sensor de muestra o elemento de medición, un detector de error, un amplificador y un accionador o controlador propiamente dicho. El elemento de medición es algún dispositivo que convierte la variable de salida en otra variable adecuada que pueda usarse para la comparación con otra señal de entrada como referencia. Se debe tener en cuenta el punto de ajuste o regulación del control con la entrada de referencia, bajo las mismas unidades que la señal de realimentación proveniente del sensor de muestra, a fin de que el comparador o detector de error proporcione la señal de corrección que escalada adecuadamente por el amplificador de error, permita el manejo adecuado del accionador completando así el lazo cerrado en el control del sistema. El accionador, valga la redundancia, es un elemento que altera la entrada de la planta o proceso de acuerdo con la señal de control, de manera que se pueda poner en correspondencia la señal de realimentación con la señal de referencia de entrada, es decir, después de realizar la comparación entre el valor real de la salida del sistema con la entrada de referencia (valor deseado), se determina el error y se produce la señal de control o de acción que reduce dicho error a cero o a un valor muy pequeño. Todo esto puede comprenderse mejor con uno de los ejemplos clásicos de los controles retroalimentados: el control de temperatura, que se describe a continuación.

Supóngase como planta a controlar a una cámara de calefacción (espacio cerrado con paredes adiabáticas) cuya fuente de calor es por efecto Joule mediante una resistencia de nicromel de 6.25Ω , calculada para soportar un máximo de 25 Vcd y en consecuencia 4 Amperes de corriente (figura 4.7).

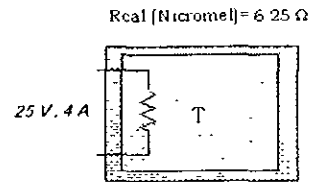


Fig. 4.7.- Cámara de calefacción

Sí tal cámara no está caracterizada; es decir no se conoce la relación entre la temperatura de su interior y el voltaje de alimentación; el primer paso consiste en obtener experimentalmente dicho comportamiento, ya que una estimación teórica al respecto puede ser laboriosa por la cantidad de variables involucradas tales como el grado de aislamiento térmico de sus paredes (o fuga de calor a través de las mismas), el volumen de la región interior, la temperatura ambiental exterior, etc. Para esa evaluación es necesario un sistema de control a lazo abierto que permita suministrar voltajes de alimentación al calefactor en el intervalo de 0 a 25 Vcd: por ejemplo: 5, 10, 15, 20 y 25 Volts, midiendo periódicamente la temperatura de interés correspondiente a cada uno de esos valores, hasta que se establezca el equilibrio térmico en el interior de la cámara. Tal sistema de control se muestra en la figura 4.8, donde una fuente de 30 Vcd con disponibilidad de corriente de hasta 4 Amperes, alimenta tanto al divisor variable formado por la resistencia de 68Ω y el potenciómetro de 500Ω , como al colector del transistor TIP142 (Darlington, $\beta=1000$); así, en la base de éste se pueden tener voltajes entre 0 y 26.4 Vcd, que por la caída base-emisor del mismo se obtiene en la carga una variación de 0 a 25 Vcd.

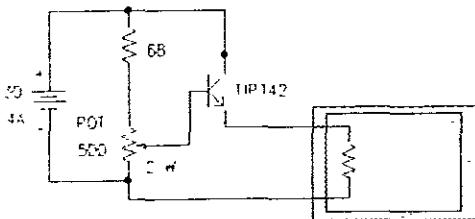
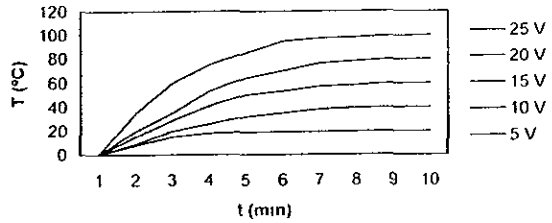


Fig. 4.8.- Sistema de control de calefacción.

Si la caracterización mencionada de la cámara (hecha en la madrugada de un día de invierno) dio como resultado las curvas mostradas en la figura 4.9, se puede afirmar que la relación entre voltaje de alimentación del calefactor y la temperatura alcanzada en el interior de la cámara está dada por:



$T[^{\circ}C] = 4 [^{\circ}C \text{ Volts}] V[\text{Volts}] \dots(4-2)$ **Figura 4.9.- Respuesta de la cámara de calefacción.**

teniendo tiempos de estabilizado de a lo más 9 minutos. Si sólo se desea un control a lazo abierto: es decir si no son relevantes las posibles fluctuaciones alrededor de una temperatura determinada, causadas por agentes externos como la temperatura ambiental, desgaste de partes, etc.; el mismo circuito de la figura 4.8 es útil con el escalamiento o graduación correspondiente en la perilla del potenciómetro que controla el voltaje de calefacción, mediante la ecuación (4-2).

Si se requiere un control más estable o automático, debe emplearse un circuito con retroalimentación como el que se ilustra en la figura 4.10.

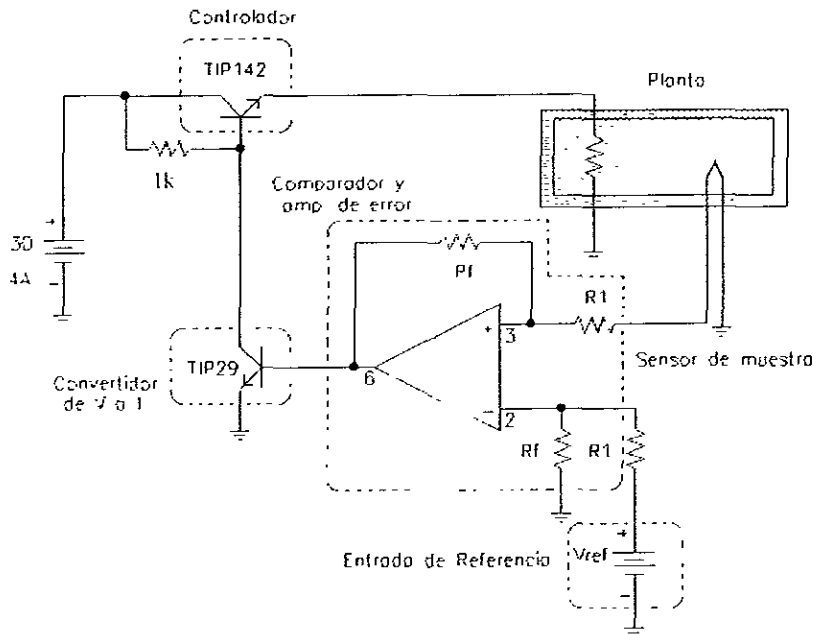


Figura 4.10.- Control automático de calefacción.

Nótese que en el circuito de la figura anterior se utiliza como controlador al mismo transistor TIP142 polarizado en base con una resistencia de $1\text{ k}\Omega$, a fin de no exceder demasiado la corriente máxima de suministro a la planta (con el transistor TIP29 en estado de corte). Dentro de la cámara se encuentra un termopar cuyo voltaje proporcional a la temperatura es en todo momento comparado contra un voltaje de referencia (V_{ref} : entrada de voltaje correspondiente a la temperatura a estabilizar) mediante un amplificador de diferencias usado precisamente como comparador y amplificador de error. La salida de éste satura o corta al transistor TIP29, dependiendo si el voltaje del termopar es mayor o menor que el de referencia. Dicho transistor es usado como convertidor de voltaje a corriente, de tal forma que cortado habilita el paso de corriente por el TIP142 hacia la planta; y en saturación desvía la corriente del controlador hacia tierra, inhibiendo el suministro de energía.

Cabe hacer notar que para mejores resultados al controlar un sistema, es deseable medir directamente la o las variables involucradas en el proceso para garantizar cada uno de los estados del mismo. Al efectuar algún proceso se puede optar por medir y controlar la calidad de determinado producto mediante variables secundarias involucradas, como pueden ser la temperatura o la presión; y relacionarlas para obtener dicha calidad en el producto; esto se conoce como control indirecto de la planta o proceso, que puede ser no muy eficaz. Por ello siempre es recomendable hacer el control a través de la variable primaria más directa al proceso del sistema en lo que sea posible.

Por otra parte, las características dinámicas en la mayoría de los sistemas de control no son constantes debido al deterioro de sus componentes con el uso y el paso del tiempo, o por cambios en la materia prima, o por fluctuaciones en el medio ambiente, etc. Aunque en los sistemas realimentados los efectos de cambios pequeños en tales características dinámicas se atenúan, si las modificaciones en los parámetros del sistema y en el medio son significativas, puede ser que se requiera de otro tipo de control.

Un sistema que se califique de duradero y confiable debe de tener la capacidad de autoadaptación; es decir, la característica de autoajustarse o automodificarse de acuerdo con cambios impredecibles en el medio o estructura. Los sistemas de control que tienen algún grado de esta capacidad, se denominan sistemas de control adaptados o adaptivos; y sus características dinámicas deben de estar identificadas en todo momento, de manera que los parámetros de control o de detección puedan ajustarse para mantener el funcionamiento óptimo de la planta. Un sistema de control adaptivo, además de acomodarse a los cambios ambientales, también lo hace ante errores moderados de proyección de incertidumbres y puede compensar fallas eventuales en sus componentes menores, aumentando así su confiabilidad y durabilidad.

Otros sistemas de control que aparentemente son de lazo abierto pueden considerarse como de lazo cerrado si se toma en cuenta una retroalimentación humana que continuamente detecta y compara la(s) salida(s) con la(s) entrada(s) y además realiza las acciones correctivas necesarias basadas en la diferencias resultantes o errores. Sin embargo, al intentar analizar sistemas de control a lazo cerrado mediante operación humana, surge el difícil problema de establecer ecuaciones que describan el comportamiento del ser humano. Uno de los muchos factores que complican el análisis, es la capacidad de aprendizaje del humano; conforme éste va adquiriendo experiencia, se convierte en un mejor elemento de control, pero más complicado de

describir y no se diga de duplicar En general, los sistemas de control con capacidad de aprender reciben el nombre de sistemas con aprendizaje.

4.4) SEÑALES, ANALÓGICAS Y DIGITALES.

En las secciones anteriores se mencionaron algunos aspectos generales de los sistemas y sus formas de control; las palabras *estímulo*, *respuesta* y *señal* fueron necesarias a la descripción. Esta última, dentro de la física y particularmente en la ingeniería electromecánica tiene un significado específico; a saber, una *señal* es una cantidad medible, variable en el tiempo y que como resultado de esas variaciones contiene información útil, incluyendo la referente a la fuente de donde procede. Como ejemplos de señales físicas se pueden mencionar los cambios de calor en una habitación, las variaciones de intensidad luminosa en determinada zona, los cambios de posición de un objeto, las variaciones de la cantidad de carga eléctrica en un conductor, etc. Dependiendo de su naturaleza, las señales se pueden clasificar según la rama de la física que las estudia; así, cada uno de los ejemplos anteriores se tipifica respectivamente como una señal térmica, óptica, mecánica y eléctrica.

Las señales por lo general no siempre se presentan en forma directamente aprovechable, por lo que es necesario cambiar su naturaleza a otra más manejable, pero preservando la información contenida en la misma; el dispositivo que se encarga de esto se conoce como *transductor*. Por ejemplo, el aumento de calor en la frente de un semejante sugiere que éste podría estar enfermo, pero no se puede estar seguro de ello hasta no *medir* tal cambio con ayuda de un termómetro, dispositivo que en su forma más conocida es un transductor que convierte los cambios de calor corporales en variaciones de longitud en una columna de mercurio previamente graduada en alguna escala de temperaturas, aprovechando el alto coeficiente de dilatación de éste, así, un termómetro clínico de mercurio puede clasificarse como transductor termomecánico.

Lamentablemente, en la mayoría de las transducciones se pierde bastante energía de la señal original debido, entre otros, tanto a la muy baja eficiencia de los transductores en general, como al hecho de que éstos en la mayoría de los casos sólo captan una parte pequeña de la señal que transducen; imagínese un micrófono piezoeléctrico (transductor mecánico-eléctrico) al cual se le conectase una bocina directamente en sus cables de salida; obviamente ésta ni se enteraría de la presencia de la señal eléctrica análoga a la señal mecánica (sonido) recibida por el micrófono. Esto justifica que casi siempre es necesario *procesar* una señal transducida para su mejor aprovechamiento; es decir, para obtener esa información útil pero con características superlativas con respecto a su naturaleza original; así, la señal recibida y transducida por el micrófono del ejemplo anterior, procesada adecuadamente, puede ser escuchada por otras personas a distancias tan lejanas donde la voz humana aún a toda su intensidad no alcanzaría. Entre los muchos procesos que se efectúan sobre las señales se pueden mencionar *la selección, la amplificación, la rectificación, la detección, el filtraje, la modulación, la reconstrucción, la comparación, la presentación o despliegue, etc.*; y muy especialmente *la digitalización*, sobre la que se apoya gran parte de este trabajo.

Naturalmente, todas las señales cuando varían en el tiempo lo hacen en forma continua, a veces de manera abrupta pero no deja de existir continuidad en dicha variación; aún después de

transducidas, las señales resultantes siguen presentando continuidad en el tiempo. Cuando para extraer la información contenida en la señal interesa precisamente esa continuidad en la variación temporal de la misma, se dice entonces que se tiene una *señal analógica*; es decir, en tiempo real; a diferencia de una *señal digital* en la que la variación de la cantidad de interés, en instantes sucesivos de tiempo, se expresa por un conjunto de pulsos codificados, similarmente al código morse usado en telegrafía. La conversión de una señal analógica a digital tiene como primer paso el dividir el rango total de valores posibles de la señal analógica en un número de niveles discretos. El valor de la señal en algún instante es luego expresado aproximadamente por el nivel en el que la señal permanece en dicho instante; esto se conoce como cuantificación y se ilustra en la figura 4.11

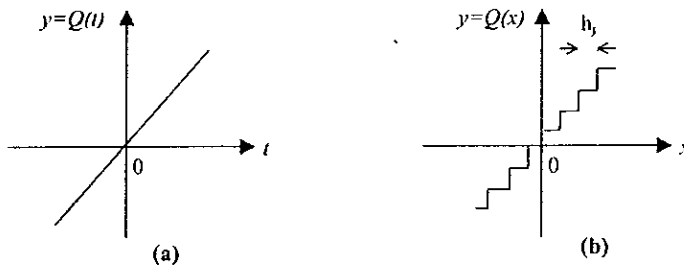


Figura 4.11.-a) Señal Analógica, b) Señal Cuantificada.

Nótese que en la cuantificación, todos los valores que quedan dentro de cada segmento son igualados a un único valor dentro de su intervalo (h_i). Este único valor es la aproximación digital a los distintos valores de la señal analógica. Así, si x es la entrada analógica, la salida digital está dada por $y = Q(x)$. El segundo paso consiste en muestrear la señal en instantes sucesivos de tiempo (figura 4.12) con un circuito electrónico mediante el que se determina el nivel de la señal en cada muestra y se genera para cada uno de ellos el conjunto apropiado de pulsos codificados. Esto último, la codificación, se ve con más detalle en el capítulo 5.

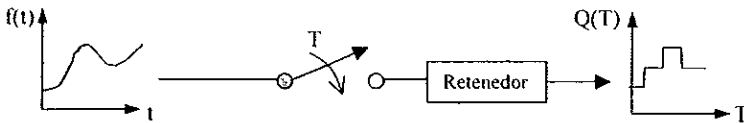


Figura 4.12.- Proceso de Muestreo (cuantificado).

Hay que tener en cuenta que la conversión de una señal analógica en la correspondiente señal digital (número binario) es una aproximación, ya que la señal analógica puede tomar una

cantidad infinita de valores, mientras la variedad de distintos números que pueden formarse con un juego finito de dígitos, es limitada. Obviamente, también existe la conversión de una señal digital a tipo analógico; proceso que se describe en el capítulo 6.

El poder digitalizar señales de un mundo analógico (y viceversa) ha permitido desarrollar equipos y sistemas con varias ventajas sobre aquellos de procesos puramente analógicos; entre éstas se pueden mencionar las siguientes.

1. Los sistemas digitales pueden realizar cálculos complejos con la misma exactitud a velocidades altas. Particularmente, las computadoras digitales pueden realizar cálculos casi hasta cualquier grado de exactitud deseado, con un incremento de costo relativamente pequeño. En cambio, el costo de las computadoras analógicas aumenta rápidamente con la complejidad de los cálculos, si se desea mantener constante la exactitud.

2. Los equipos digitales son también versátiles; simplemente colocando un nuevo programa se pueden cambiar totalmente las operaciones. Esta característica es particularmente importante si se trata de un sistema de control que ha de recibir información operativa o instrucciones desde algún centro de cómputo, en el que, por ejemplo, se realizan estudios de optimización y análisis económico.

Regresando a los sistemas y sus formas de control, el proceso de señales tanto analógicas como digitales permite una clasificación de éstos en cuatro grandes partes: 1) una planta analógica controlada analógicamente; 2) una planta analógica bajo control digital; 3) una planta digital con control analógico; 4) una planta digital controlada digitalmente. El problema que motivó el desarrollo de este trabajo queda comprendido en el puntos 1 y 2 anteriores; es decir, un sistema acelerador de partículas que procesa señales analógicas pero que sus formas de control eran tanto analógicos como digitales. Al modernizar dicho acelerador con la adición de una microcomputadora personal como medio de operación y control, fue necesario procesar señales analógicas y digitales, afortunadamente en éstas últimas la mayoría de las acciones de control son del tipo “Si-No” o de dos posiciones, lo que se describe a continuación.

4.5) CONTROL DE DOS POSICIONES O “SI-NO”.

Como su nombre lo indica es una forma de control que sólo cuenta con dos posiciones fijas (encendido y apagado) y en la mayoría de los casos sólo conecta y desconecta la energía hacia la planta. Este tipo de control es muy simple y por lo mismo económico, por lo que es ampliamente utilizado tanto en sistemas industriales como domésticos; un ejemplo muy conocido en controles de temperatura es el termostato.

Para este tipo de control, a lazo cerrado, con una salida $m(t)$ y señal de error actuante $e(t)$; la señal $m(t)$ permanece en un valor máximo o mínimo dependiendo del proceso y la señal de error que puede ser positiva o negativa, de tal modo que:

$$\begin{aligned} m(t) &= M_1 && \text{para } e(t) > 0 \\ &= M_2 && \text{para } e(t) \leq 0 \end{aligned}$$

donde M_1 y M_2 son constantes diferentes (generalmente M_1 puede ser un valor mínimo o cero o bien un valor negativo). Este tipo de control se compone generalmente con dispositivos eléctricos y/o electrónicos.

En la figura 4.13 se presenta un diagrama a bloques del control de dos posiciones. En ocasiones se debe desplazar la señal de error actuante antes de que se produzca la conmutación, lo que se llama brecha diferencial y hace que la salida del control $m(t)$ mantenga su valor hasta que la señal de error actuante haya pasado levemente del valor preestablecido para prevenir las acciones excesivamente frecuentes del mecanismo de si-no.

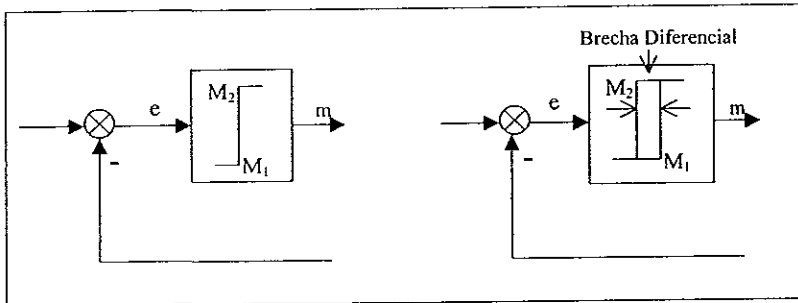


Figura 4.13.- Diagrama por bloques de controladores si-no.

Dependiendo de la brecha se puede reducir la amplitud de las oscilaciones en la salida, pero en consecuencia aumenta la cantidad de conmutaciones por segundo y se reduce la vida útil del controlador (por ejemplo un relevador).

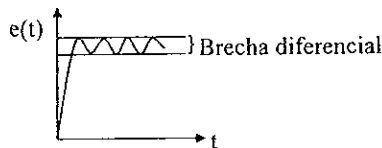


Figura 4.14.- Gráfica de error vs tiempo.

Similarmente al concepto de las señales analógicas y digitales; los sistemas dinámicos pueden clasificarse como de tiempo continuo o discreto, por la forma en que capturan las variables que procesan; lo que se refiere al intercambio de información ya sea para operación y control o bien para el manejo de datos de interés. En los de tiempo continuo cada parámetro es registrado y procesado en todo momento ya sea por técnicas analógicas o su emulación mediante métodos digitales. Los sistemas de tiempo discreto o de datos muestreados, son aquellos en los cuales una o más variables pueden cambiar solamente en ciertos instantes, que se han de indicar por kT ($k = 0, 1, 2, 3, \dots$); pueden especificar el momento en el cual se realiza alguna medición

física o el tiempo en el cual se lee la memoria de una computadora. Los sistemas de tiempo discreto difieren de los de tiempo continuo, en que las señales para el primero aparecen en forma de datos muestreados

En la práctica, se presentan los sistemas de tiempo discreto, cuando se obtienen las mediciones necesarias para el control en forma intermitente o cuando se comparte un control de gran envergadura o computadora entre diversas plantas, de manera que se envía una señal de control a cada planta sólo periódicamente o siempre que se utiliza una computadora para realizar los cálculos necesarios para el control. Muchos sistemas de control industrial modernos, son sistemas de tiempo discreto porque invariablemente influyen algunos elementos cuyas entradas y/o salidas son discretas en el tiempo. Sin embargo, a veces la discretización con operación de muestreo, puede ser enteramente ficticia e introducida únicamente para simplificar el análisis de un sistema de control que en realidad sólo contiene elementos continuos.

Como hay diversos tipos de operaciones de muestreo de importancia práctica, se les indica a continuación:

1. Muestreo periódico (convencional): en este caso los instantes de muestreo están equiespaciados, o sea $t_k = kT$ ($k = n, 1, 2, \dots$).
2. Muestreo de orden múltiple: el esquema de las t_k se repite periódicamente, o sea $t_{k+r} - t_k =$ constante para todo k .
3. Muestreo de ritmo múltiple: en este caso concuerdan simultáneamente dos operaciones de muestreo en $t_k = pT_1 + qT_2$, donde T_1 y T_2 son constantes y p y q son enteros.
4. Muestro al azar. en este caso, los instantes de muestreo son casuales, o sea que t_k es una variable aleatoria.

Lo anterior da lugar a un tercer tipo de señal que es la de tipo discreto, la que aparece solo en determinados momentos y entre otros interesa su magnitud. En comparación con otros tipos y/o procesos de señales, se puede observar en la figura 4.15 una función $x(t)$ discreta en el tiempo, una función cuantificada y una señal discreta y cuantificada. El funcionamiento de sistemas de control digital, involucra la cuantificación, tanto en amplitud como en tiempo.

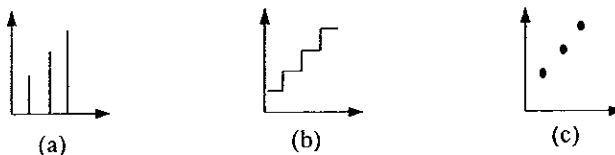


Figura 4.15.- a) Señal discreta, b) cuantificada y c) discreta y cuantificada.

REFERENCIAS DEL CAPÍTULO 4

- 1) Camiña Catalán, Carlos. Sistemas Continuos de Control. Servicio de Publicaciones UPU. Valencia 1989.
- 2) Canales Ruiz, Roberto, Becerra Rivera, Renato. Análisis de Sistemas Dinámicos y Control Automático. Limusa. México 1980.
- 3) D'azzo, John J., Hoopis H., Constantino. Sistemas Realimentados de Control. Paraninfo. Madrid 1989.
- 4) Dorf, Richard C. Sistemas Modemos de Control. Addison-Wesley Iberoamericana. México 1989.
- 5) Franklin F., Gene, Powell, J. David, Workman L., Michael. Digital Control of Dynamic Systems. Addison Wesley. USA 1990.
- 6) Jones D., Larry, Chin Foster, A. Electronic Instruments and Measurements. Prentice Hall. New Jersey 1991.
- 7) Kuo C., Benjamin. Sistemas de Control Automático. Prentice-Hall Hispanoamericana S.A.. México 1996
- 8) Kuo C., Benjamin. Sistemas de Control Digital. CECSA. México 1997.
- 9) Ubierto P., Arthur, Ibañez P., Carabantes. Diseño Básico de Automatismos Electrónicos. Paraninfo. España 1996.

CAPÍTULO 5

COMPUTADORAS E INTERFACES

5.1) ANTECEDENTES

Al construir un circuito electrónico puede surgir la dificultad de no conocer el medio en el cual funcionará, esto es particularmente común si parte del medio es una computadora, debido a su arquitectura y su funcionamiento lógico; lo que cada día se complica por los avances de la tecnología. De hecho, las primeras computadoras ocupaban grandes salas y en la actualidad pueden ser del tamaño de un libro.

La evolución en las computadoras, especialmente las de tipo PC, ha sido rápida; si se recuerda que cuando aparecieron por primera vez en el mercado las computadoras personales IBM con un microprocesador Intel 8086, éstas tenían una capacidad de 64 Kb de RAM (*Random Access Memory*: memoria de acceso aleatorio) y con unidades de disco flexible de 160 Kb. Con tales características el equipo se consideraba un verdadero avance en su tiempo; lo que en la actualidad es insuficiente y obsoleto ya que hoy en día es muy común encontrar computadoras desde 16 Mb de RAM, discos duros de 4.1 Gb y procesadores 80586 a más de 100 MHz.

Cualquier PC cuenta con puertos de entrada y salida, para el *mouse*, el *joystick*, la salida de impresora, salidas seriales y por supuesto el teclado y el monitor; esto es para lo que concierne a los puertos externos. Además la PC cuenta con canales o puertos internos conocidos como *bus* de expansión.

El *hardware* en computación se refiere a la computadora tanto en su maquinaria como al equipamiento interno y externo, es decir la parte física (“si se puede tocar es hardware”); el *software* es el conjunto de instrucciones que le dice que hacer a la computadora (programas, datos, etc.); tanto el *hardware* como el *software* son inútiles si no trabajan conjuntamente.

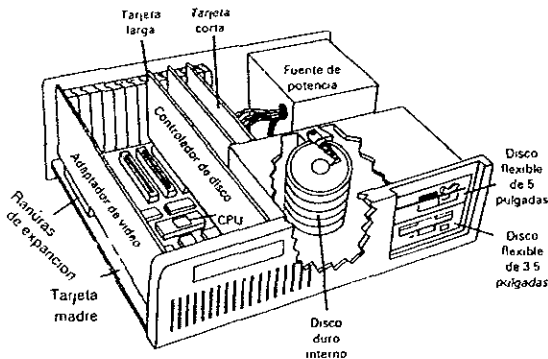


Figura 5.1.- Esquema interno de una computadora

Los modelos comerciales de computadoras tienen diferentes tipos de *bus* como son: ISA, EISA, MicroCanal y PCI. El más popular es el bus ISA con una expansión de 16 bits, conocido también como *bus* AT. En el diseño original de la PC se estructuró un *bus* que admitiera expansiones futuras y que éstas no se limitaran a tarjetas ya desarrolladas únicamente en el momento de su fabricación. Para poder trabajar con el bus interno se requiere distinguir cada una de sus funciones por medio de direcciones; esto es como utilizar el código postal, ya que facilita a la máquina el trabajo de estar localizando el puerto deseado. La capacidad de direccionamiento es función directa del procesador, pero, en general está limitado a 512 puertos (*ports*) muchos de los cuales ya están usados por periféricos, tales como las unidades de disco flexible y duros, o que ya están definidos para un fin específico como el reloj y la memoria, pero aun quedan bastantes puertos libres.

El propósito de entablar una comunicación entre computadoras es el establecer el envío y recepción de información de una computadora a otra computadora o a un equipo externo ya sea para su control y/o procesamiento.

5.2) EL BYTE

Las computadoras son equipos binarios ya que solo operan en dos estados: *apagado* y *encendido*. La distinción entre estas dos condiciones es la más pequeña cantidad de información y se le conoce como dato binario o *bit*; un *bit* puede tomar valores de 1 para encendido y 0 para *apagado* o *viceversa* dependiendo de la lógica utilizada.

La información de un solo *bit* no es lo suficientemente útil, así que para manipular o manejar una cantidad aceptable de información las computadoras requieren el juntar varios bits para formar una unidad mayor de información. El *byte* está conformado por el agrupamiento de 8 bits con lo que es posible tener 256 combinaciones de "0" y "1"; así, con un *byte* se puede expresar un número de 0 a 255 (partiendo del mismo principio de que un *bit* tiene dos condiciones, al tener ocho datos y cada dato independiente para estar en uno de los dos estados da 2^8 combinaciones; generalmente el 0 se considera como una combinación la cual no se utiliza propiamente para realizar alguna función, sino como un estado de espera o *apagado* pero aun así se debe de tomar en cuenta y por ello se tienen $2^8 - 1$ posibilidades).

La memoria de una computadora está especificada en bytes y múltiplos del mismo: kilobytes y megabytes. En la notación del sistema métrico decimal, el kilobyte representaría 1000 bytes. Dado que las computadoras usan una base binaria (base 2 en lugar de la base 10 o decimal), un kilobyte está definido como 1024 bytes ya que el 1024 (2^{10} , 10 bits) es el valor más cercano a 1000. De manera semejante se aplica para un megabyte que equivale a 1024 kilobytes. Por este método de cálculo se explica el porqué 640Kb de memoria tienen 655360 bytes y no sólo 64000 bytes. En la tabla 5.1 se muestran algunas equivalencias entre los sistemas binario y decimal, así como el uso de prefijos para múltiplos.

Binario	2 Elevado a la potencia	Equivalente en decimal	Multiplos
0000000000000001	0	1	
0000000000000010	1	2	
00000000000010000	5	32	
0000000100000000	9	512	
0000001000000000	10	1024	1 k
0001000000000000	13	8192	8 k
1000000000000000	16	65536	64 k

Tabla 5.1.- Tabla binaria

Una computadora puede expresar un número directamente en uno o más bytes, pero el usuario frecuentemente requiere almacenar texto como información; para que la computadora realice esta tarea se le asigna a cada caracter (ya sea letra, número o símbolo) un número de 0 a 255. Cada caracter tiene definido un valor, es decir un código, que lo representa. En la tabla 5.2 se muestran algunos ejemplos en dos tipos de código.

Caracter	ASCII	EBCDIC
punto	01000000	00100000
signo <	01001100	00111100
signo +	01001110	00101011
C	11000011	01000011

Tabla 5.2.- Comparación entre código ASCII y EBCDIC.

Las computadoras antiguas han utilizado distintos conjuntos de código. A partir de las computadoras tipo PC se ha acordado un estándar mundial, por lo menos para lo que se refiere al alfabeto y muchos de los caracteres de puntuación. Este conjunto de código estándar se conoce como el conjunto de caracteres ASCII (*American Standard Code for Information Interchange*). El otro conjunto de código de caracteres que puede encontrarse es el EBCDIC (*Extended Binary Code Decimal Interchange Code*), que es ampliamente usado por la IBM en su ambiente de macrocomputadoras. El ASCII estandariza solamente el código usado para 128 caracteres, dejando 128 números extras del total de un byte. Como un resultado de esto, se ha dejado a la inventiva de los programadores de computadoras el tener dos opciones, pueden ahorrar un bit y almacenar caracteres solamente en 7 bits, o bien usar el bit extra para otros propósitos. Un uso común para los 8 bits (datos) es la creación de conjuntos de caracter ASCII extendido, el único problema con este ASCII extendido es que no se aplica a todas las marcas de computadora. En la figura 5.2 se presenta el código ASCII extendido, de la firma IBM.

000	^Q NUL	037 %	074 J	111 o	148 ö	185 ¶	222 █
001	^A SOH	038 &	075 K	112 p	149 ò	186 ¶¶	223 █
002	^B STX	039 '	076 L	113 q	150 ù	187 ¶¶¶	224 α
003	^C ETX	040 (077 M	114 r	151 ù	188 ¶¶¶¶	225 β
004	^D EOT	041)	078 N	115 s	152 ÿ	189 ¶¶¶¶¶	226 Γ
005	^E ENQ	042 *	079 O	116 t	153 Ö	190 ¶¶¶¶¶¶	227 π
006	^F ACK	043 +	080 P	117 u	154 Ü	191 ¶¶¶¶¶¶¶	228 Σ
007	^G BEL	044 ,	081 Q	118 v	155 Ç	192 ¶¶¶¶¶¶¶¶	229 σ
008	^H BS	045 -	082 R	119 w	156 £	193 ¶¶¶¶¶¶¶¶¶	230 μ
009	^I NT	046 .	083 S	120 x	157 ¥	194 ¶¶¶¶¶¶¶¶¶¶	231 τ
010	^J LF	047 /	084 T	121 y	158 ₣	195 ¶¶¶¶¶¶¶¶¶¶¶	232 ϕ
011	^K VT	048 0	085 U	122 z	159 ₣	196 ¶¶¶¶¶¶¶¶¶¶¶¶	233 θ
012	^L FF	049 1	086 V	123 {	160 á	197 ¶¶¶¶¶¶¶¶¶¶¶¶¶	234 Ω
013	^M CR	050 2	087 W	124	161 í	198 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶	235 δ
014	^N SO	051 3	088 X	125 }	162 ó	199 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	236 ∞
015	^O SI	052 4	089 Y	126 ~	163 ú	200 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	237 φ
016	^P DLE	053 5	090 Z	127	164 ñ	201 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	238 ε
017	^Q DC1	054 6	091 [128 Ç	165 Ñ	202 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	239 Π
018	^R DC2	055 7	092 \	129 ü	166 ñ	203 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	240 ≡
019	^S DC3	056 8	093]	130 é	167 ð	204 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	241 ±
020	^T DC4	057 9	094 ^	131 â	168 ÷	205 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	242 ≥
021	^U NAK	058 :	095 _	132 ä	169 ˆ	206 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	243 ≤
022	^V SYN	059 ;	096 `	133 à	170 ˆ	207 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	244 [
023	^W ETB	060 <	097 a	134 å	171 ½	208 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	245]
024	^X CAN	061 =	098 b	135 ç	172 ¼	209 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	246 ÷
025	^Y EM	062 >	099 c	136 ê	173 ;	210 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	247 ≈
026	^Z SUB	063 ?	100 d	137 ë	174 «	211 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	248 °
027	^[ESC	064 @	101 e	138 è	175 »	212 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	249 •
028	^\ FS	065 A	102 f	139 ï	176 ⋮	213 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	250 ·
029	^] GS	066 B	103 g	140 î	177 ⋮	214 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	251 √
030	^^ RS	067 C	104 h	141 ï	178 ⋮	215 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	252 n
031	^_ US	068 D	105 i	142 Ä	179	216 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	253 z
032	^SPC	069 E	106 j	143 Å	180	217 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	254 ■
033	!	070 F	107 k	144 É	181	218 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	255
034	"	071 G	108 l	145 æ	182 ¶	219 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	
035	#	072 H	109 m	146 Æ	183 ¶¶	220 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	
036	\$	073 I	110 n	147 ô	184 ¶¶¶	221 ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶	

Figura 5.2.- Código ASCII extendido

En lo referente a comunicación entre una computadora y sus periféricos, entre dos computadoras, o entre una computadora y otros equipos; son indispensables las *interfaces*. Una interfaz es una etapa intermedia necesaria para enlazar a dos sistemas o equipos que requieren un intercambio de información; para tal interconexión se deben establecer especificaciones adecuadas al tipo de comunicación deseada. En el entorno de computadoras existen dos tipos principales de interfaces: la paralela y la serie, que permiten a una computadora tener comunicación con equipos externos. A continuación se describe brevemente cada una de ellas

5.3) INTERFAZ PARALELA

La interfaz paralela, como su nombre lo indica, tiene varios canales de datos en paralelo con lo cual le es posible la transmisión de un código de varios bits de forma simultánea; en el puerto paralelo, los datos aparecen en la interfaz como varios canales de un bit; pero también se requiere de una línea adicional, llamada línea de reloj o de sincronización. Los datos son enviados de acuerdo con esta señal y el dispositivo receptor utiliza este reloj para sincronizarse con los datos que se están enviando.

La mayoría de los equipos PC están equipados con un puerto paralelo unidireccional, esto es que solo puede realizar el envío de datos. Como los puertos paralelos están desarrollados para ser utilizados con impresoras, la restricción de que solo envíen y no reciban no supone problema alguno. Otros equipos como el PS/2 de IBM y otras marcas de computadoras vienen con puertos bidireccionales, con estos tipos de interfaces es posible tener una comunicación que permite enviar y recibir datos. En la figura 5.3 se presentan los accesos en este tipo de conector.

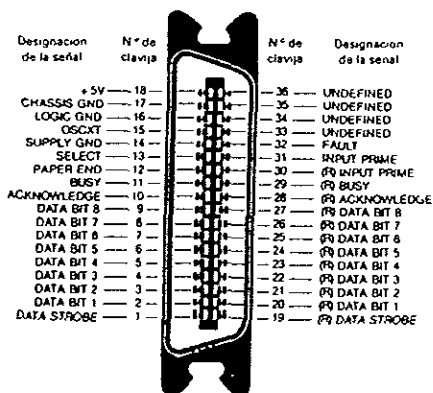


Figura 5.3.- Interfaz paralela centronics. (R) Indica retorno de la señal por tierra.

Una de las características principales de la interfaz paralela es su capacidad de transmitir información a velocidades mas elevadas que la tipo serie; en sí, la interfaz paralela puede transmitir datos a la velocidad máxima del dispositivo con el que esté conectado, como se ilustra en la figura 5.4.

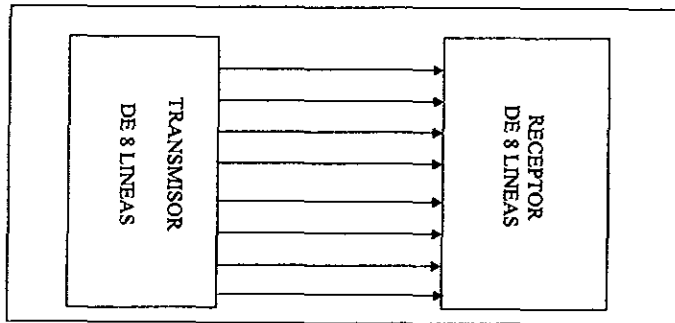


Figura 5.4.- Muestra esquemática de una comunicación paralela.

5.4) INTERFAZ SERIE

Una interfaz serie parece ser menos compleja que la tipo paralelo, ya que los datos se transmiten en serie, es decir que generalmente los bits de datos pasan por un solo canal de uno a uno, en un orden determinado. En esta clase de interfaz se puede tanto enviar como recibir datos, con la restricción de que se tienen velocidades menores para la comunicación, en comparación con la transmisión en paralelo.

En una transferencia en paralelo, como se indicó anteriormente, cada bit tiene su propio canal. Para transferir datos con una interfaz paralela se requiere de tener como mínimo 11 conductores o líneas (8 para datos, 1 de paridad, 1 de tierra y 1 de sincronía); sin incluir las señales de control. En la interfaz serie se requieren de un mínimo de 3 líneas (envío, recepción y tierra), también sin incluir las señales de control. Esta es una de las razones por las cuales se ve limitada la interfaz paralela, ya que se requiere de una mayor cantidad de cables que en la interfaz serie y no puede recibir datos. La desventaja principal de la comunicación serie es que se tienen velocidades fijas de transmisión por lo que los equipos externos se deben ajustar a una de esas velocidades, lo que no ocurre con el caso en paralelo donde es posible ajustar el tiempo de transmisión de la computadora a las necesidades del equipo.

La comunicación entre dos equipos que utilizan el modo serie, se realiza utilizando líneas o canales de transmisión en las modalidades: simplex, semi-duplex y duplex. A continuación se explica brevemente cada una de éstas.

5.5) COMUNICACIÓN SIMPLE O SIMPLEX

Este modo, también conocido como transmisión SX, es capaz de enviar información sólo en un sentido, sin posibilidad de verificar la correcta transmisión mediante alguna señal eléctrica de retorno; por ejemplo, el envío de audio a una bocina. Obviamente en este tipo de comunicación no hay reversibilidad; es decir, no se puede hacer una transmisión de regreso por este sistema. En la figura 5.5 se ilustra la transmisión de información por modulación-demodulación como ejemplo.

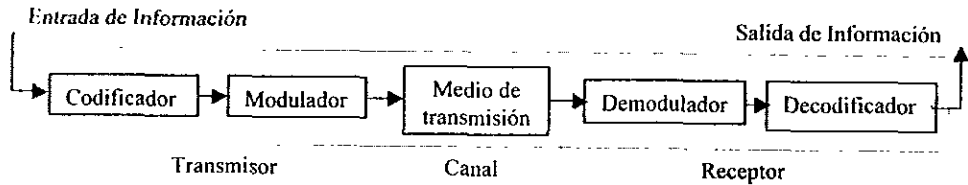


Figura 5.5.-Modulación-demodulación

5.6) COMUNICACIÓN SEMI-DUPLEX

En muchos casos es deseable mantener una comunicación en dos sentidos, o al menos, poder devolver un mensaje a su origen para mantener una posible verificación o control. Una de las maneras de obtener esto es utilizando el mismo canal de manera alterna para transmitir en ambas direcciones, este es el caso del método semi-dúplex (HDX, *half-duplex*). Como se puede ver en la figura 5.6 es posible enviar y recibir información de ambos sentidos pero no al mismo tiempo; es decir, que en un momento dado el flujo de información se realiza en un solo sentido mientras que del lado contrario se mantiene en espera a que termine dicho flujo de información, para después alternar el sentido y estado de espera. Un ejemplo típico de esta clase de comunicación es la transmisión de audio mediante radio de banda civil.

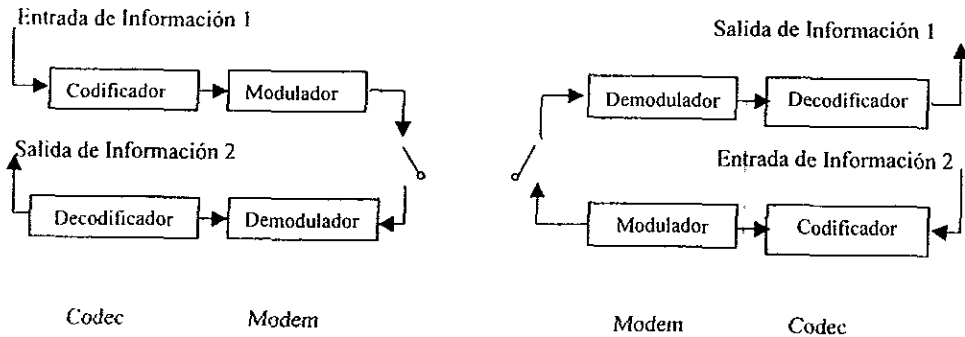


Figura 5.6.-Semi-duplex

5.7) COMUNICACIÓN DÚPLEX

En esta se obtiene comunicación simultánea en ambos sentidos; algo así como cuando dos personas hablan por teléfono. A este método se le conoce como dúplex completo (FDX, *full-duplex*). Con este modo es posible tener una comunicación sin necesidad de que una de las terminales se encuentre en espera. Como se puede notar en las figuras 5.6 y 5.7, tanto en la transmisión HDX como en la FDX, los moduladores y demoduladores operan en parejas. Esta combinación de modulación y demodulación se le llama *modem* (*modulador-demodulador*) en los sistemas de transmisión de datos. También los codificadores y decodificadores trabajan en pares, dando así origen al término *codec* (*codificador-decodificador*).

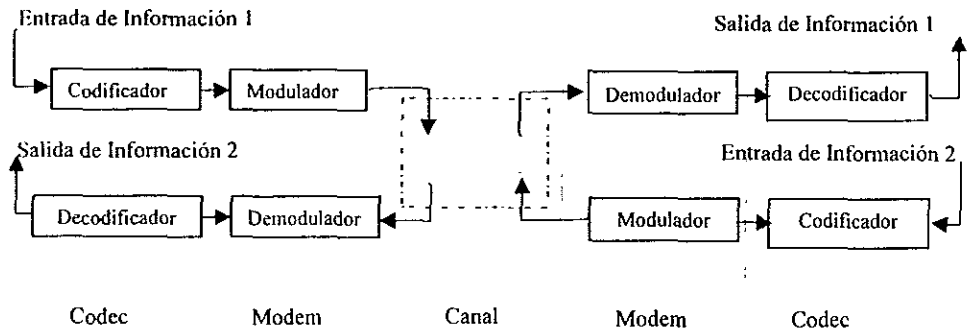


Figura 5.7.-Duplex

A su vez, para la comunicación serie existen dos tipos: la asíncrona y la síncrona; cualquiera de éstos puede ser adoptado por los tres modos de comunicación descritos (simplex, semi-duplex y dúplex). A continuación se describe el tipo asíncrono que es el usado en el desarrollo de este trabajo. Posteriormente se menciona brevemente la comunicación asíncrona.

5.8) COMUNICACIÓN ASÍNCRONA

Aunque la interfaz se presenta en formatos distintos, en el caso de las computadoras tipo PC se utiliza mucho la *transmisión asíncrona*. Recordando que los bits de datos se transmiten por un solo canal de forma consecutiva, la transmisión asíncrona necesita un entorno estable; si se produce cualquier irregularidad en la línea y le sucede algo a un bit de datos, los bits siguientes pueden ser leídos erróneamente. En la figura 5.8 se muestra este tipo de transmisión.

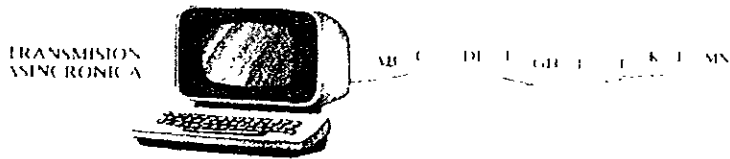


Figura 5.8.- Transmisión de caracteres tipo serie asincrónica.

La transmisión serie asincrónica requiere que la señal tenga las características siguientes: bit de inicio, bit de parada, bits de datos y bit de paridad (figura 5.9). De los anteriores solo el bit de paridad es opcional, todos los demás son indispensables.

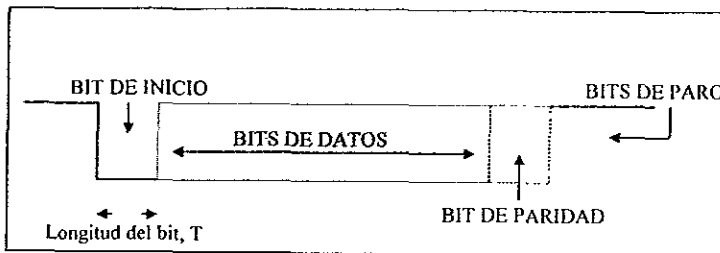


Figura 5.9.- Señal asincrónica

BIT DE INICIO.

Cuando un canal de datos serie está inactivo, permanece en estado de 1 o encendido. La transmisión de un carácter se inicia desactivando el canal de datos, -cambia a estado de 0 o apagado-, esto lo retira de su estado inactivo y le envía lo que se denomina un bit de inicio, que es el primer bit que precede a cada palabra o byte de datos. Cada carácter transmitido necesita un bit de inicio.

Inmediatamente después del bit de inicio, aparece la serie de los bits de datos. No todos los sistemas utilizan los ocho bits de datos; algunos sólo utilizan cinco y otros pueden usar hasta nueve.

BIT DE PARO

Cuando las comunicaciones tipo serie se utilizaban sólo en los teletipos y otros equipos antiguos, estos dispositivos mecánicos necesitaban tiempos de espera entre cada carácter. Por esta

razón, se introdujo un bit de parada o de paro al final de cada caracter. Dependiendo de la espera necesaria, se utilizaban uno, uno y medio o dos bits de paro. Actualmente la única función es la de marcar el final del caracter.

BIT DE PARIDAD

Para asegurarse de que un caracter es recibido correctamente, se añade un bit inmediatamente después de los bits de datos; a éste se le conoce como bit de paridad, el cual no siempre se utiliza. Cuando se usa, la paridad puede ser par o impar y se verifica contando exclusivamente el número de bits de datos que están en el estado "1"; no se toman en cuenta los bits de inicio y parada. Dada la forma binaria de operar con los datos, hay dos formas de manejar la paridad: cuando se utiliza una paridad par, la cantidad de dígitos "1" que se encuentren en los bits de datos deberá ser un número par; en caso de que se tenga un número impar, el bit de paridad será 1 para que se mantenga el número par de "unos"; si dicha cantidad de "unos" en los bits de datos es par, entonces el bit de paridad será 0. Para cuando se maneja una paridad impar es algo similar, cuando la cantidad de unos en los bits de datos sea impar el bit de paridad será 0 y si es par la cantidad de "unos" entonces el bit de paridad será 1 para mantener el número de "unos" en impar. En las figuras 5.10 y 5.11 se muestran tales casos.

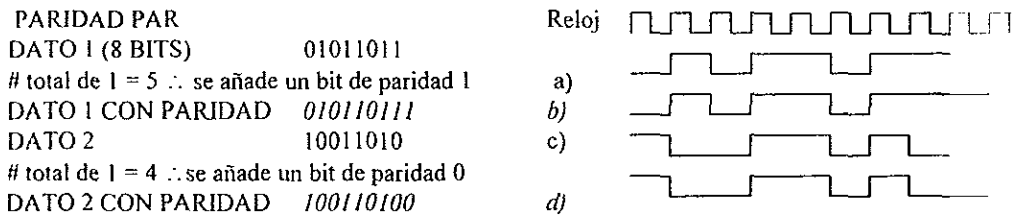


Figura 5.10.- a, c) Datos sin paridad; b, d) datos con paridad par.

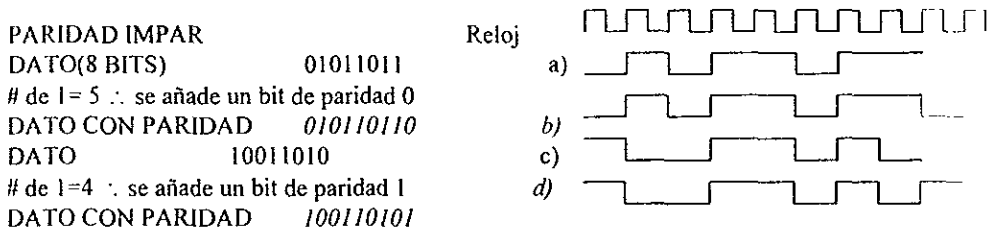


Figura 5.11.- a, c) Datos sin paridad; b, d) datos con paridad impar.

Al dato con paridad se le añaden los bits de inicio y paro. Cuando se recibe un caracter, se cuenta el número de unos y se compara con el bit de paridad, con lo que se verifica si es que hubo algún bit de datos que hubiera cambiado durante la transmisión (debido a ruido o interferencia) Si la transmisión no coincide, se indicará el error, y entonces se solicitará que se realice una retransmisión de los datos; tal verificación depende de las características del equipo que reciba los datos

LONGITUD O DURACIÓN DEL BIT

El bit de inicio indica únicamente el principio del flujo de datos y tiene la misma longitud de un bit de datos. La longitud de un bit depende de la velocidad de transmisión, o sea de cuántos bits por segundo se quieran transmitir. Si el equipo realiza una comunicación a 300 bps, entonces se divide 1 entre 300 para calcular la longitud del un bit

Longitud de un bit = $1/300$ bps = 3.333 mseg.

Si la velocidad es de 1200 bps, se tiene: $1/1200 = 0.833$ mseg.

Cabe aclarar que bits/seg es el régimen de velocidad en bits o tasa de bits (bps); es decir la velocidad de transmisión de un dato codificado en binario. Por otra parte se conoce como *baud* a la unidad para la velocidad de modulación de datos. En los sistemas de transmisión binaria la cantidad de bauds es igual al número de bits por segundo; por ejemplo, si se transmite una señal binaria a 110 bps la rapidez de modulación es igual a 110 bauds. En sistemas multiniveles y en sistemas donde se utiliza la compresión de datos, la cantidad de bauds indica la cantidad de transiciones por segundo, lo cual es sinónimo de velocidad de símbolos

En la mayoría de los programas para comunicación, el texto o información que se escribe se transmite de caracter a caracter. Cada vez que se pulsa una tecla se envía un bit de inicio, los bits de datos, el bit de paridad y el bit de paro. Al agrupar caracteres se crea un bloque o grupo de datos o paquete de datos. El tamaño actual de los paquetes puede ir desde 8 hasta millones de bits. Al enviar datos en paquetes se prescinde del bit de paridad y se utilizan algunos datos extra, como es la suma de verificación o carácter de confirmación que se utiliza para determinar si los datos han llegado sin errores; esto es frecuente en comunicaciones por medio de modem, lo que se ilustra en la figura 5.12.

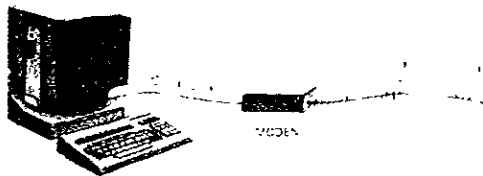


Figura 5.12.- Comunicación por modem.

Cabe recordar que un *modem* es un dispositivo que adapta una terminal o computadora a una línea tal como la telefónica. Convierte los pulsos digitales a frecuencias dentro del intervalo de audio del teléfono y los vuelve a convertir en pulsos en el lado receptor.

Conociendo la longitud del bit y el número de bits que se van a transmitir se puede determinar el tiempo que se requiere para capturar una señal o información, por ejemplo si se envían 8 datos de información, con su bit de inicio, de paro y paridad, esto suman 11 bits, los cuales por la longitud del bit proporcionan el tiempo total de la transmisión de un carácter. Tomando los resultados obtenidos anteriormente si se transmite a 300 bps entonces el tiempo requerido para un carácter es de 0.03666 segundos y a 1200 bps es de 0.00916 segundos, de igual forma ocurre al multiplicar este resultado por el número de caracteres a enviar para tener el tiempo total de transmisión de la palabra o palabras.

5.9) COMUNICACIÓN SÍNCRONA

La comunicación síncrona es una transmisión en la que los bits y caracteres de datos se envían a una velocidad fija, estando sincronizados la computadora emisora y el equipo o computadora receptora; esto elimina la necesidad de los elementos de inicio y final, y ofrece una mayor eficiencia, pero mayor complejidad al elaborar los circuitos de la interface, ya que es necesario sincronizar tiempos.

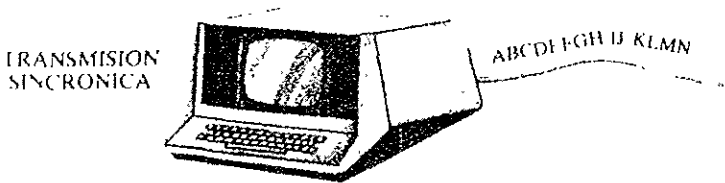


Figura 5.13.- Transmisión de caracteres tipo síncrono.

5.10) EL PUERTO SERIE

A la interfaz serie se le ha generalizado con el nombre de *puerto serie*, pero estrictamente hablando la interfaz es la etapa intermedia en donde se realizan las conversiones de entrada/salida y el puerto es la conexión por donde se realiza dicha entrada o salida de información. La interfaz debe comunicarse tanto con la computadora así como con el equipo conectado en el puerto; ésta debe aceptar ambas transmisiones de información, tanto de datos como de control, así como poder determinar el estado de ambos sistemas. Para establecer una comunicación, la interfaz usa un espacio especial de direcciones dentro de la computadora reservados para la entrada y salida (I/O). de esta manera es como la computadora controla un puerto en específico; es decir puede verificar su estado, leer y escribir datos o cambiar las características del mismo. En la figura 5.14 se muestra la ubicación física de un puerto e interfaz serie.

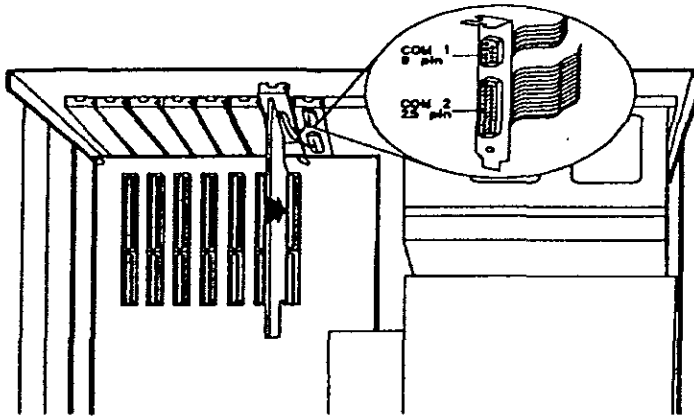


Figura 5.14.- Puertos seriales.

Por añadidura a la transmisión de datos y el estado del puerto serie, se tiene una manera de indicarle a la computadora la atención necesaria en algún momento; hay una señal que interrumpe a la computadora en cualquier estado en el que se pudiera encontrar, cuando los puertos serie tienen información que la computadora debe buscar.

Si se define un puerto de comunicaciones (*COM*) como la identidad de un puerto serie, ésta se obtiene de la combinación de una dirección y un *IRQ* (número de petición de interrupción). Para cada puerto existe una combinación que lo identifica, la cual es única; esto es que no puede haber mas de un puerto con la misma combinación de dirección e *IRQ*.

La dirección le indica al programa donde está el puerto, es decir, cuando el programa necesita entablar comunicación con el puerto *COM1* (puerto serie 1), éste conoce la dirección del puerto y envía la información a dicha dirección.

En una computadora existen varios dispositivos que utilizan señales denominadas *interrupciones (IRQ)*; cuando un puerto genera una interrupción es porque ha recibido un caracter. Esto le indica al programa que necesita dirigirse a ese puerto para capturar el caracter y procesarlo.

Las primeras computadoras tipo PC únicamente definían a dos puertos serie, *COM1* y *COM2*. Con el tiempo éstos no fueron suficientes y los fabricantes de programas y equipos solucionaron esta demanda de puertos con la creación de dos definiciones adicionales: *COM3* y *COM4*. Pero como no hay una definición estándar oficial para estos puertos, no están soportados por todos los productos de programas y de equipos. En la tabla 5.1 se muestran algunas combinaciones de dirección e *IRQ* para puertos.

Puerto para PC	Dirección/IRQ
COM1	03F8h/4
COM2	02F8h/3
COM3	03E8h/4*
COM4	02E8h/3*

Tabla 5.1.- Combinaciones de dirección e IRQ para puertos.

Cabe aclarar que las interrupciones se utilizan para hacer frente a algo importante en algún proceso, por lo que la computadora dejará lo que estaba haciendo; las interrupciones reciben distintas prioridades, de tal forma que las más importantes son manejadas en primer lugar y pueden interrumpir a las de prioridad inferior. Las líneas IRQ utilizadas habitualmente en la comunicación por el puerto serie son el IRQ4 (para COM1 y COM3) e IRQ3 (para COM2 y COM4). La prioridad de una línea IRQ es proporcional a su número, de tal modo el COM2 tiene una prioridad superior que COM1.

5.11) EL UART

Para pasar del formato paralelo del bus de datos de la computadora a la forma serie de la interfaz, -como se mencionó-, se puede realizar el cambio por medio de una matriz de circuitos lógicos o mediante un dispositivo construido específicamente para ello; éste último se conoce como *UART* (*Universal Asynchronous Receiver Transmitter*: receptor-transmisor asíncrono universal).

El UART es un subsistema completo integrado en un microcircuito que convierte datos paralelo a serie y viceversa; es decir, también de serie a paralelo. Se especializa en la comunicación asíncrona tanto para enviar como para recibir datos; también determina, inserta y verifica los bits de paridad, crea el bit de inicio, selecciona e inserta los bits de paro, controla el número de bits por carácter y suprime los bits de formato tras la recepción; e incluso lo almacena todo para tener tiempo de realizar otras tareas.

Los antiguos equipos de PC utilizan el UART 8250, los equipos tipo AT y superiores utilizan los UART 16450 ó 16550. El 8250 presentaba problemas a "alta velocidad" (9200 bauds) los nuevos dispositivos ofrecen un mejor rendimiento con mucho menos problemas. La interacción entre el software de comunicación y los UART recientes presentan un alto nivel de transferencia de información (57600 bauds) con respecto a los anteriores. El software le indica al UART que envíe o reciba caracteres, y que inicie o detenga el flujo de datos. Algunos equipos detectan el tipo de UART instalado en el equipo y realizan de forma autónoma los ajustes adecuados.

El UART también se puede encontrar principalmente en los modems internos; en los externos se utiliza el que viene con la misma computadora ya que se conectan a su puerto serie.

Existe otro tipo de interfaz de comunicaciones como es la *UASRT*, la cual es programable y puede trabajar en modo síncrono y asíncrono.

5.12) INTERFAZ RS-232

A continuación se explican algunas características de la interfaz RS-232 la cual es la de mayor uso para computadoras tipo PC en la comunicación por el puerto serie.

En los inicios de la computación cada fabricante implementaba sus señales de comunicación para las exigencias o características de sus dispositivos, así empresas diferentes usaban voltajes y corrientes distintas al realizar sus conexiones, originando con esto que la comunicación por una interfaz de una computadora a otro equipo de otra marca fuera imposible.

Para conseguir un orden en este caos, el estándar RS-232 fue desarrollado y adoptado en 1969 por la *Electronic Industries Association (EIA)*; en este tiempo la tecnología se encontraba en una revolución.

El estándar RS-232 se internacionalizó, dando como resultado que fuera ocupado para puertos serie. En 1972 adoptó la norma ISO 2110 (International Organization for Standardization), la cual especifica el conector DB-25, hoy familiar por su amplio uso. Aunque el conector DB-25 por sí solo es ampliamente usado no parte exclusivamente del estándar RS-232. *Una interfaz RS-232 puede usar cualquier conector y aun ser estándar.*

La principal victoria del estándar RS-232 ha sido definir los voltajes y el uso de los circuitos de interfaz serie. El estándar define a un 1 binario (o condición de marca) como el voltaje que se encuentre entre -3 y -25 Volts. Un 0 binario (o condición de espacio) se define como cualquier voltaje que se encuentre entre +3 y +25 Volts. Los voltajes comprendidos entre -3 y +3 Volts son considerados como indeterminados o indefinidos. La magnitud de la corriente de salida en corto circuito no debe exceder 0.5 amperes, esto es principalmente para protección de la interfaz. En el caso de que ocurriese un corto, el circuito debe soportarlo sin daño alguno. La impedancia de entrada del receptor debe estar entre 3 k Ω y 7 k Ω . La velocidad de transición (*slew rate*) a la salida del circuito excitador no debe exceder de 30 Volts por microsegundo, pero el tiempo requerido para pasar la transición de -3 a +3 Volts no debe ser mayor de 1 milisegundo ni del 1 % de la longitud de un bit. La capacitancia en paralelo de la terminal no debe exceder de 2500 picofarads, incluyendo la capacitancia del cable. En la figura 5.15 se muestran los niveles típicos de dicha interfaz.

También están definidas la máxima y mínima velocidad de cambio (*switching*) para la resistencia al ruido. Estas especificaciones establecen como límite superior, un cable de 50 pies de longitud con una velocidad de 20000 bps. Para velocidades más lentas se pueden tener cables de mayor longitud. El límite de velocidad superior práctico de una interfaz RS-232 está cerca de los 150000 bps para un cable de 5 pies o menos.

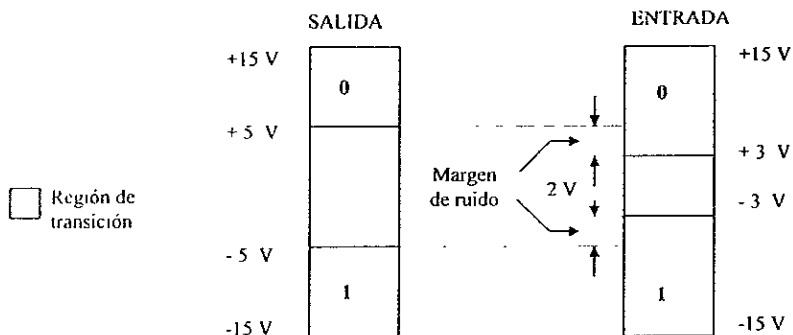


Figura 5.15.- Niveles típicos característicos del RS-232C

El estándar RS-232 fue adoptado rápidamente y es la razón por la que hoy es posible conectar de forma serial a computadoras sin que se requiera de un fabricante o marca en especial. El estándar RS-232 tuvo revisiones mayores antes de llegar a lo que es hoy, el RS-232B y finalmente el RS-232C. En la figura 5.16 se muestra la asignación de terminales para el conector DB-25, posteriormente en la figura 5.17 se ilustra lo correspondiente para el conector DB-9.

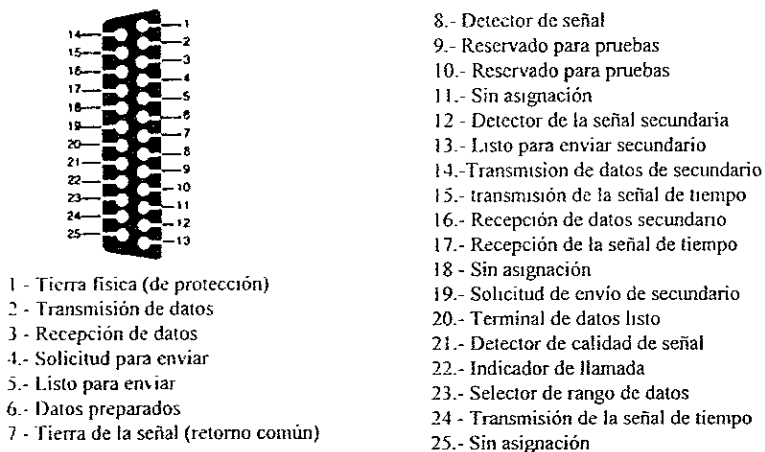
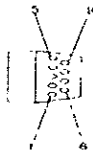


Figura 5.16.- Conector DB-25 y asignación de terminales (pins).



1.- Detector de señal

- 2 - Receptor de datos
- 3.- Transmisor de datos
- 4.- Terminal de datos preparados
- 5.- Tierra de señal
- 6.- Equipos de datos preparado
- 7.- Solicitud para enviar
- 8.- Permiso para enviar
- 9.- Indicador de llamada

Figura 5.17.- Conector DB-9 y asignación de terminales (pins).

El uso de uno u otro tipo de conector (DB-25 o DB-9) es porque raramente son utilizadas las 25 señales, de hecho solamente 9 canales de la interfaz serie son implementados en la PC. Basicamente, el número mínimo de conexiones necesarias para el flujo de datos en ambas direcciones son 3 alambres: pin 2 (transmisión de datos), pin 3 (recepción de datos) y pin 7 (tierra de señal). Tales condiciones son para ambos conectores, únicamente cambiando la asignación de terminales. En la tabla 5.2 se muestra la equivalencia de terminales para dichos conectores.

DB-25	←→	DB-9
2	←→	3
3	←→	2
4	←→	7
5	←→	8
6	←→	6
7	←→	5
8	←→	1
20	←→	4
22	←→	9

Tabla 5.2.- Relación de pins entre DB 25 y DB 9

La transmisión/recepción de datos, básicamente se puede realizar con las conexiones que se muestran en la figura 5.18. Mediante un osciloscopio conectado en la terminal correspondiente del puerto y el software adecuado (apéndice A), es posible observar la salida de datos de la PC como los cambios de niveles antes descritos.

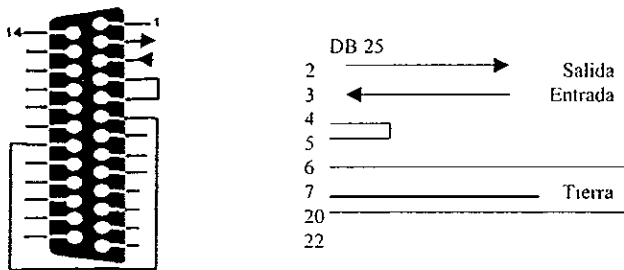


Figura 5.18.- Conexión básica para transmisión y recepción.

REFERENCIAS DEL CAPITULO 5

- 1) Albrecht, Bob. Introducción a los PC. McGraw-Hill, México 1992.
- 2) Bearpark, Keith. Computer Communications. McGraw-Hill, Londres 1993.
- 3) Carballar Falcón, José Antonio. Configuración, Actualización y Mantenimiento: Software y Hardware de su PC. Ra-ma, España 1994.
- 4) Cole, Gerald D. Computer Networking For Systems Programmers. J. Wiley, EUA 1989.
- 5) Davis Bitter, Gordon. Introduction to Computers. McGraw-Hill, México 1977.
- 6) Freedman, Alan Diccionario de Computación McGraw-Hill, México 1994.
- 7) Gookin, Dan. Advanced MS-Dos Batch File Programming Windcrest, EUA 1989.
- 8) Halsall, Fred. Comunicación de Datos, Redes de Computadoras y Sistemas Abiertos. Addison-Wesley, México 1989.
- 9) Kochhar, A K. Sistemas de Producción basados en Computadoras. Continental, México 1981.
- 10) Nania, Georges A. Diccionario de Informática Paraninfo, España 1985
- 11) Perry, William E. Computer Control and Security. J. Wiley, EUA 1981.

CAPÍTULO 6

COMPUTACIÓN DEDICADA A LA INSTRUMENTACIÓN

En el capítulo anterior se describe brevemente la arquitectura de las computadoras personales así como sus principios de funcionamiento; desde la descripción de bit, byte y códigos de comunicación, llegando hasta la necesidad del uso de interfaces para la comunicación de la computadora con otros equipos; de estas últimas se mencionan los tipos paralelo y serie, habiendo abundado más en la descripción sobre esta última, por ser la que se utiliza en el desarrollo de este trabajo. Para terminar de asentar las bases sobre la "conexión" de una computadora a otros equipos y particularmente al acelerador descrito en el capítulo 3; es decir, tener una computadora personal dedicada a dicho sistema; falta todavía describir con más detalle lo referente a la transmisión y recepción de datos, así como el lenguaje de computación usado.

Tanto la transmisión como la recepción de datos por el puerto serie, en protocolo RS-232C, según lo mencionado anteriormente, requiere de varios dispositivos para el manejo de la información. A continuación se describen tales necesidades considerando a la computadora como el equipo base

6.1) TRANSMISIÓN

En los equipos receptores, que comúnmente utilizan circuitos TTL, la señal RS-232C proveniente del puerto de la computadora requiere de modificaciones en sus niveles de voltaje por la forma en que ésta maneja la lógica binaria en comparación con la lógica TTL, donde el estado alto (usualmente el 1) está entre los 3 y 5 Volts, mientras que el otro (usualmente el 0) se ubica entre 0 y 2 Volts. la región entre 2 y 3 Volts es de indeterminación. Por esta razón se requiere de un convertidor de niveles de voltaje que tenga entrada compatible con RS-232C y salida TTL (figura 6.1) con la condición de que éste sólo modifique los niveles de voltaje, mas no la información de la señal.

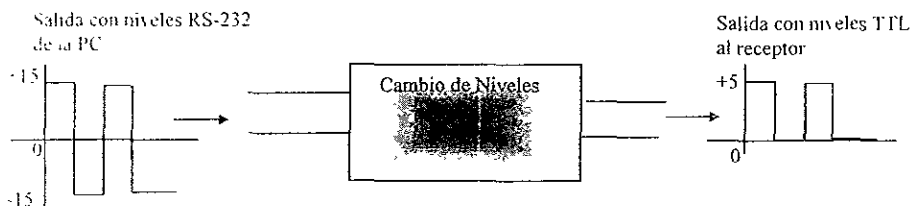


Figura 6.1.- Conversión de RS-232 a TTL

Ya en niveles TTL (entre 0 y 5 Volts), se requiere de sincronizar los datos para su correcto manejo por lo que es necesario un dispositivo que opere en el momento de detectar una señal de

información (como se describió en el capítulo anterior, la señal de arranque puede ser el bit de inicio), además de mantener la sincronización de los bits que son recibidos (figura 6.2).

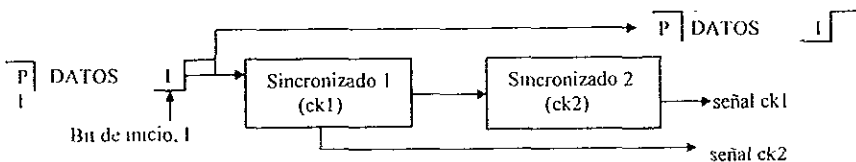


Figura 6.2.- Señales para sincronizar la recepción.

Para utilizar la información enviada en cada byte, se requiere de capturar y almacenar temporalmente cada uno de los bits; y así en lugar de tener una sola línea, a la salida de la captura debe haber tantas líneas como bits sean recibidos; es decir que los bits se convertirán de la forma serie en que son transmitidos, a modo paralelo para su uso en determinado equipo (figura 6.3). En este proceso también se seleccionan los bits útiles (n salidas), desechando los de formato (bits de inicio y paro).

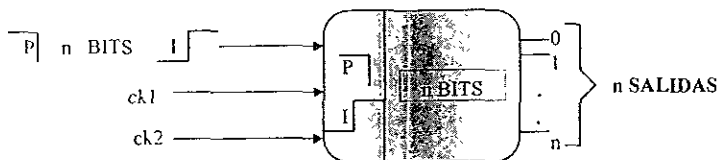


Figura 6.3.- Captura y almacenamiento de n bits de salida.

El uso de los bits (ya en paralelo) en determinado equipo, implica la selección previa del mismo entre varios de su tipo que pueden estar también conectados a las mismas salidas del convertidor serie-paralelo; esto se conoce como direccionamiento del equipo deseado e implica que parte de los bits en la información enviada sean reservados para tal efecto, siendo los bits restantes el dato que en sí procesará el equipo seleccionado. Así, de las n salidas útiles mostradas en la figura 6.3 anterior, siempre hay que tener presente que parte de ellas (m bits) están destinadas a ubicar el equipo que recibe los bits restantes ($n-m$) que constituyen al dato; esto hace necesario el optimizar las salidas disponibles bajo algún método que permita hacer una codificación de los bits (por lo menos los de direccionamiento) antes de ser enviados; esto es útil ya que permite reducir el número de bits de direccionamiento dejando mayor cantidad de los mismos para el dato, o bien no reducirlos y poder contar con mayor cantidad de dispositivos conectados a la salida del convertidor serie-paralelo. Este procedimiento es similar a las líneas de direccionamiento que la misma computadora utiliza para activar o desactivar dispositivos internos tales como puertos, unidades de disco, etc. o efectuar otras acciones.

En los procesos de transmisión (y recepción) no se requiere de un gran número de líneas, como en los direccionamientos internos de la computadora, pero aún así se puede utilizar un método de codificación, donde el número de salidas de dirección está relacionado con el número de bits codificados. Para esto se requiere que en el decodificador se maneje el mismo procedimiento utilizado en la codificación. Hay otros métodos más avanzados de codificación tales como el encriptamiento, que es utilizado cuando se requiere una mayor seguridad en la transmisión de información.

Para aclarar lo anterior, considérese como ejemplo que en la salida del convertidor de la figura 6.3 se tienen 16 bits útiles, de los cuales 4 son de direccionamiento y/o control; y los 12 bits restantes para el dato en sí. Si sólo se utiliza una "interpretación binaria"; es decir una conexión directa con esos 4 bits de direccionamiento, esto a lo más permite el control de 4 equipos en todas sus combinaciones posibles, según se muestra en la tabla 6.1.

				DISPOSITIVOS A CONTROLAR			
A	B	C	D	Equipo 1	Equipo 2	Equipo 3	Equipo 4
0	0	0	0	APAGADO	APAGADO	APAGADO	APAGADO
0	0	0	1	APAGADO	APAGADO	APAGADO	ENCENDIDO
0	0	1	0	APAGADO	APAGADO	ENCENDIDO	APAGADO
0	0	1	1	APAGADO	APAGADO	ENCENDIDO	ENCENDIDO
0	1	0	0	APAGADO	ENCENDIDO	APAGADO	APAGADO
0	1	0	1	APAGADO	ENCENDIDO	APAGADO	ENCENDIDO
0	1	1	0	APAGADO	ENCENDIDO	ENCENDIDO	APAGADO
0	1	1	1	APAGADO	ENCENDIDO	ENCENDIDO	ENCENDIDO
1	0	0	0	ENCENDIDO	APAGADO	APAGADO	APAGADO
1	0	0	1	ENCENDIDO	APAGADO	APAGADO	ENCENDIDO
1	0	1	0	ENCENDIDO	APAGADO	ENCENDIDO	APAGADO
1	0	1	1	ENCENDIDO	APAGADO	ENCENDIDO	ENCENDIDO
1	1	0	0	ENCENDIDO	ENCENDIDO	APAGADO	APAGADO
1	1	0	1	ENCENDIDO	ENCENDIDO	APAGADO	ENCENDIDO
1	1	1	0	ENCENDIDO	ENCENDIDO	ENCENDIDO	APAGADO
1	1	1	1	ENCENDIDO	ENCENDIDO	ENCENDIDO	ENCENDIDO

Tabla 6.1.- Conexión directa con cuatro bits.

Usando un método de direccionamiento que utilice los mismos 4 bits pero que en este caso opere no solo decodificando la dirección sino también efectuando un proceso de almacenamiento adicional, es decir, cuando se seleccione un equipo éste no cambie su estado (encendido-apagado) a menos que se le seleccione y se indique un cambio; entonces, 3 de esos 4 bits se pueden utilizar codificados para la ubicación y el restante para la acción (selección o desección).

En la tabla 6.2 se ilustra esta forma de direccionamiento donde las líneas A, B y C permiten la ubicación codificada de hasta 7 equipos, que pueden estar ya sea encendidos ó apagados, reservando la línea D como bit de activación, según se mencionó anteriormente, donde

si este último está en 1 entonces se cambia el estado del equipo mientras que con el bit D en 0 el equipo ubicado continúa sin cambio

A	B	C	ACCION
0	0	0	Espera, no hay equipo
0	0	1	Equipo 1, encendido/apagado depende de D
0	1	0	Equipo 2, encendido/apagado depende de D
0	1	1	Equipo 3, encendido/apagado depende de D
1	0	0	Equipo 4, encendido/apagado depende de D
1	0	1	Equipo 5, encendido/apagado depende de D
1	1	0	Equipo 6, encendido/apagado depende de D
1	1	1	Equipo 7, encendido/apagado depende de D

Tabla 6.2. Conexión utilizando direccionamiento con tres bits.

El resto de los bits almacenados, como ya se dijo, se utilizan precisamente como datos para el equipo receptor seleccionado, para su conversión digital-analógica (D/A) posterior.

Al respecto, existen distintos tipos de convertidores D/A con características diferentes, en su control, método de conversión y resolución. Dejando a un lado la parte de control y conversión, se hace un enfoque a una de las características más importantes que es la resolución. Esta tiene que ver con el incremento más pequeño correspondiente de una señal analógica a partir del bit menos significativo (LSB) convertido. En los convertidores de este tipo generalmente se expresa la resolución en bits, donde el número de niveles analógicos es igual a 2^n y n es el número de bits o resolución; por lo tanto para conocer el valor de los incrementos del convertidor, se divide el valor máximo de voltaje, denominado voltaje a escala completa ("full scale voltage"), entre 2^n . Mientras mayor sea el número de bits de resolución el número de niveles será mayor, mientras que el incremento o diferencia entre cada nivel será menor.

Por ejemplo, al convertir una entrada digital de 4 bits (resolución de 4 bits) teniendo un voltaje de escala completa de 5 Volts, los incrementos de cada uno de los 16 (2^4) niveles analógicos serán iguales a $5/16 = 0.3125$ Volts; manejando una resolución de 8 bits ($2^8 = 256$) los incrementos serían $5/256 = 0.0195$ Volts, es decir que se pueden definir más valores analógicos y mucho más exactos. Si se observa en las operaciones, se están considerando todas las posibilidades de combinaciones, incluso el 0; pero si el 0 es una combinación entonces no se tendrían 0 Volts debido a que el 0 sería igual al valor de un incremento, como esto no sucede nunca se llega a tener representada la escala completa, ya que cuando la entrada binaria se encuentre en 0 a la salida se tienen 0 Volts, por lo cual falta un incremento o nivel analógico para tener a la salida efectivamente la escala completa. Entre mayor sea la resolución la diferencia entre el máximo valor que se puede alcanzar y el valor máximo real va disminuyendo, como se muestra en la tabla 6.3, considerando $V_{REF} = 5$ Volts (voltaje máximo). En las gráficas 6.1 y 6.2 se ilustra una salida analógica entre 0 y 5 Volts para entradas digitales con 4 y 8 bits respectivamente.

	4 bits				Valor analógico	8 bits								Valor analógico	
	A	B	C	D		A	B	C	D	E	F	G	H		
Escala completa	1	1	1	1	4.6875 V	1	1	1	1	1	1	1	1	1	4.98046875 V
Media escala	1	0	0	0	2.5 V	1	0	0	0	0	0	0	0	0	2.5 V
Cero + LSB	0	0	0	1	0.3125 V	0	0	0	0	0	0	0	0	1	0.01953125 V
Escala cero	0	0	0	0	0 V	0	0	0	0	0	0	0	0	0	0 V

Tabla 6.3. Comparación de salidas entre un DAC de 4 bits y otro de 8 bits.

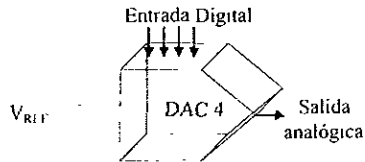


Figura 6. 4. Esquema de un convertidor de 4 bits.

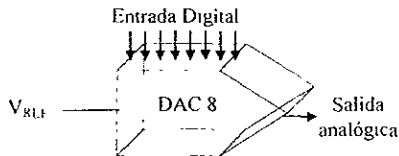
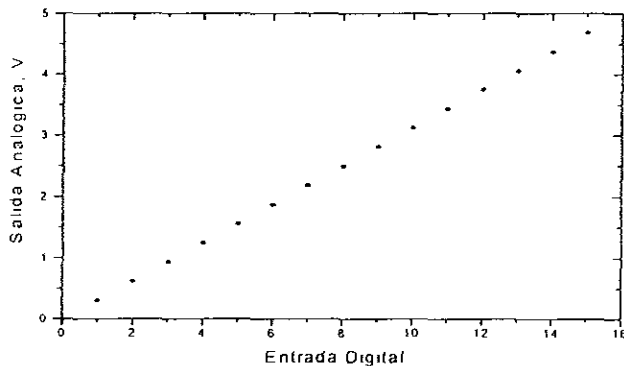
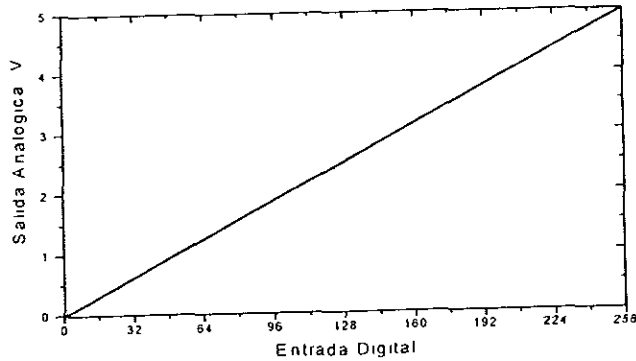


Figura 6. 5. Esquema de un convertidor de 8 bits.



Gráfica 6.1.- Salida ideal de un convertidor D/A de 4 bits, 15 niveles.



Gráfica 6.2.- Salida ideal de un convertidor D/A de 8 bits, 255 niveles.

6.2) RECEPCIÓN

Hay sistemas periféricos que además de requerir información, también la regresan a la microcomputadora como señales analógicas, que requieren de conversión a forma digital para ser canalizadas hacia la máquina. Esto hace necesario un convertidor analógico-digital (A/D). De forma similar al convertidor D/A, existen diferentes tipos de convertidores en los que la principal característica es nuevamente la resolución. Tanto mayor sea la resolución, menores son los cambios entre cada bit, con lo que se tiene una aproximación más exacta al valor real; si la resolución es pobre el valor analógico leído puede o no ser el valor que se obtuvo en la conversión. Dependiendo del tipo de convertidor, varían en el intervalo del voltaje de entrada, en el tiempo de conversión y en la presentación de los datos; además, algunos dispositivos pueden realizar el ajuste o corrección de errores. En la figura 6.6 se muestra un esquema básico del convertidor analógico-digital.

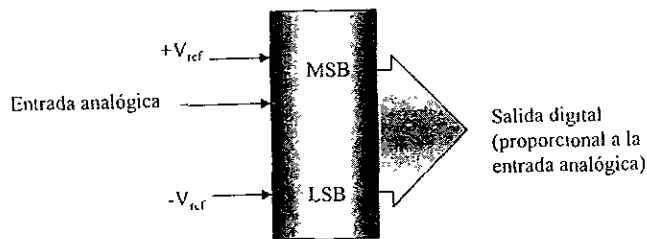


Figura 6. 6.- Esquema de un convertidor A/D.

Tanto en la conversión D/A como en la A/D, una característica secundaria pero también importante es la linealidad. La mayoría de los circuitos integrados dedicados a conversión tienen una linealidad excelente por lo cual ésta no se tomó como parámetro fundamental.

Hay señales que requieren de una "conversión digital" con la resolución de sólo un bit, ya que sólo cambian de estado (encendido o apagado) para indicar únicamente si están activos o listos (figura 6.7). Además estas señales varían en su orden de importancia ya que algunas son fundamentales para que el sistema funcione.

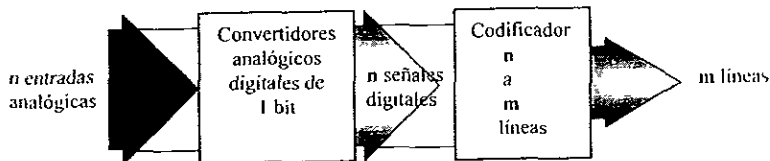


Figura 6. 7.- Conversión y codificación de señales analógicas

Las líneas de entrada analógica pueden ser señales de verificación que indican si se encuentra activo o inactivo cierto equipo o sensor. El número de líneas de entradas analógicas es el mismo número de líneas digitales, esto es porque solo se ajusta cada línea a niveles TTL. En el codificador, n es el número de líneas de entrada y m es el número de líneas de salida.

Nuevamente tomando en consideración lo que se mencionó al inicio de este capítulo, al transmitir o recibir datos se debe de tener una sincronización en ambos equipos, tanto para la inicialización de los datos como para su transferencia (figura 6.8).

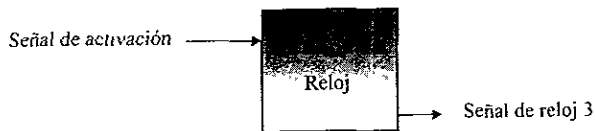


Figura 6. 8.- Reloj para recepción

Con las señales ya digitalizadas, antes de enviarlas a la computadora se requiere dar el formato necesario para su captura; es decir, ordenar las señales digitales del codificador y del convertidor A/D para añadirles los bits de inicio y paro. Dependiendo del orden en el que sean transmitidas, tendrán una representación o interpretación en el momento de que sean agrupadas y desagrupadas nuevamente dentro de la computadora. Por esta razón es necesaria una etapa similar a la de captura, con la diferencia de que en lugar de tomar los datos en serie y la salida en paralelo, se requiere que la entrada sea en paralelo y la salida en serie, en la captura se quitan los bits de inicio y paro, en ésta se requiere de añadirlos, como se muestra en la figura 6.9.

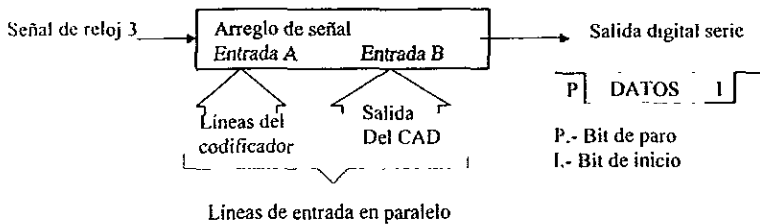


Figura 6. 9.- Conversión paralelo-serie.

Por último, en lo referente a las necesidades de equipo externo, se encuentra la etapa de conversión de nivel; esta vez se cambia la señal TTL a una con niveles RS-232C, cuidando nuevamente la integridad de los bits de información del sistema así como los bits necesarios para la correcta captura de los mismos. Esto se ilustra en la figura 6.10.



Figura 6. 10. Convertidor de TTL a RS-232C.

6.3) LENGUAJES DE COMPUTACIÓN

El lenguaje máquina debería ser el estándar fundamental para la compatibilidad de equipos (hardware), ya que han existido tantas familias de computadoras como fabricantes en el mercado; y cada una de ellas con un lenguaje de máquina propio que las hacía diferentes entre sí y aún dentro de una misma marca; por ejemplo, todas las macrocomputadoras IBM utilizan el mismo lenguaje de máquina, pero la serie tipo PC de IBM maneja un lenguaje de máquina diferente.

Cuando se escribe un programa éste debe ser traducido, ensamblado, compilado o interpretado al lenguaje de máquina propio de la computadora. Para ejecutarse en una máquina distinta, el programa original debe ser reensamblado o recompilado al lenguaje de esa máquina, con los traductores apropiados ó bien reescribirse, -del mismo diagrama de flujo-, en el código que tal máquina utiliza.

La compatibilidad del lenguaje de máquina en la serie tipo PC de la IBM, se logró a partir del uso del microprocesador 8086 de Intel. Posteriormente apareció el 8088 que comprendía todas

las funciones del anterior y con nuevas características, como actualización, que permitía ejecutar cualquier programa del 8086, aunque al revés no siempre era posible. Después del 8088 siguieron los procesadores 80286, 386, 486; hasta llegar al actual 586 mejor conocido como Pentium; manteniéndose en todos ellos una compatibilidad escalonada siempre hacia arriba; es decir, cualquiera de éstos puede ejecutar bien los programas de versiones anteriores, pero al revés no siempre. El desarrollo de los procesadores (hardware) se hizo conjuntamente con la programación (software) correspondiente; esto dio origen a las máquinas con sistemas operativos compatibles y no necesariamente residentes (DOS: "Disk Operating System"), sobre el que se desarrolló toda la programación de interés a usuarios cada vez menos expertos en computación

Lo anterior, de manera natural, dio lugar a la búsqueda de compatibilidad de máquinas inferiores con las superiores, ya que las primeras podían "actualizarse" con relativa poca inversión en algunas características, de tal forma que con un procesador anterior podían comportarse como si tuvieran el procesador reciente. En varios casos, la compatibilidad de lenguajes de máquina se lograba mediante simulación o emulación. Los emuladores son programas que traducen y ejecutan un programa determinado en otro lenguaje de máquina, simulando el funcionamiento de un tipo de máquina dentro de otra; el término simulación se emplea para "representar" el comportamiento de determinado equipo en una computadora, mediante la programación apropiada.

El lenguaje ensamblador es un lenguaje de programación que está a un paso del lenguaje máquina. Cada sentencia del lenguaje ensamblador es traducida a una instrucción de máquina por el ensamblador. Para programar en ensamblador se debe tener una buena documentación y estar familiarizado con la arquitectura de la computadora, los lenguajes que se hacen en ensamblador son difíciles de mantener además que son dependientes del hardware.

Todo programa está escrito en un lenguaje de programación, y hay al menos un lenguaje de programación para cada serie principal de CPU (Unidad Central de Proceso o Procesador). Típicamente, hay un lenguaje ensamblador y una cantidad de lenguajes de alto nivel para cada serie o familia de procesadores. Los lenguajes ensambladores son específicos de cada máquina, y el lenguaje máquina que genera funciona en una sola familia de CPU's. A menos que los lenguajes máquina sean muy similares, es muy difícil traducir un programa en un lenguaje ensamblador de una serie de CPU a otra.

Los términos lenguaje ensamblador y lenguaje máquina son a menudo utilizados como sinónimos; sin embargo, no es lo mismo. Un ejemplo es la siguiente instrucción en ensamblador: COMPARE A, B; que es traducida a una instrucción máquina de la siguiente manera: COMPARE los contenidos de las posiciones de memoria 2340-2350 con el contenido de las posiciones 4567-4577. La instrucción realmente generada en lenguaje máquina a binario es: 10010101001011101010100101001.

Los lenguajes ensambladores son diferentes según las distintas clases de computadoras, en algunas con un mínimo de líneas se puede hacer una tarea que en otra computadora, para realizar lo mismo, sería necesario definir más líneas o viceversa

Los lenguajes de programación de alto nivel fueron creados para eliminar tal dependencia de la máquina. Lenguajes tales como COBOL, FORTAN y BASIC deberían supuestamente poder ejecutarse en diferentes computadoras, sin embargo debido a las variantes de cada lenguaje la compatibilidad no es total. Cada proveedor de compiladores sigue agregando nuevas características a sus lenguajes, haciéndolos *con ello incompatibles con versiones anteriores* o diferentes. Para cuando una nueva característica llega a estandarizarse se han implementado ya una decena de otras características. A veces se desarrollan programas para traducir de un lenguaje a otro; si el programa no puede traducir completamente el archivo, es necesario recurrir a retoques manuales, en ese caso suele ser más sencillo reescribir el programa desde el inicio.

En el pasado, los programas para sistemas (sistemas operativos, administradores de bases de datos, etc.) se escribían en lenguaje ensamblador para optimizar el rendimiento de la máquina. Actualmente suele usarse el lenguaje C, que al igual que el ensamblador, puede manipular los bits a nivel máquina, pero con la diferencia de que un programa en C puede ser compilado a lenguaje de máquina en varios CPU.

C es un lenguaje de programación estructurado, de relativo fácil manejo (casi como un lenguaje de alto nivel), que es capaz de manipular la computadora a bajo nivel. El C puede ser compilado al lenguaje máquina de casi todas las computadoras. Por ejemplo UNIX está escrito en C y se ejecuta en una amplia variedad de micro, mini y macrocomputadoras.

El C se programa como una serie de funciones que se llaman unas a otras para su procesamiento. Las funciones son muy flexibles, permitiendo a los programadores la elección entre el uso de la biblioteca estándar que se provee con el compilador, el uso de funciones de terceros creadas por otros proveedores de C, o el desarrollo de sus propias funciones. Además es un lenguaje en donde no se requiere de una gran cantidad de líneas de programación para activar y definir el o los puertos serie de una computadora.

Por estas razones se seleccionó al lenguaje C para hacer toda la programación que realiza la transferencia de información entre la computadora y el acelerador al cual está dedicada, así como también para la interpretación y el manejo de datos de interés. A continuación se presentan los comandos y su sintaxis para que C utilice los puertos serie.

BIOSCOM

El bioscom es un comando dentro de C, que utiliza la interrupción 0x14 del BIOS para ejecutar varias funciones de comunicación de entrada y salida de los puertos RS-232C. Específicamente, tiene la tarea de habilitar los puertos serie de la computadora. Su sintaxis es la siguiente.

```
int bioscom(int cmd, char abyte, int port);
```

La expresión anterior es la forma en que se declara este comando, en ella:

port : es el valor que le corresponde a la identidad del puerto serie para la entrada y salida. para el COM1 le corresponde un valor de 0, para el COM2 le corresponde el 1 y así consecutivamente.

cmd : es un número entero que corresponde a la función que va a desarrollar. El 0 es para la inicialización del puerto donde toma el valor de *abyte* como los parámetros de comunicación; la función 1 corresponde al envío del caracter asignado en *abyte*; la función 2 es para recibir un caracter y la función 3 regresa el estado actual del puerto (status). Las dos ultimas funciones ignoran el argumento de *abyte*

Cuando *cmd* es 1 el argumento que tenga *abyte* será el valor que se enviará a través del puerto de salida; recordando, como se mencionó anteriormente, que a cada caracter le corresponde un valor asociado entre 0 y 255. Cuando *cmd* es 0 el argumento de *abyte* establece los parámetros de comunicación; éste se puede expresar en hexadecimal, tomando como base la tabla 6.4, se selecciona una combinación de bits de cada grupo:

Datos	Significado	Velocidad de transmisión	
0x02	7 bits de datos	0x00	110 bauds
0x03	8 bits de datos	0x20	150 bauds
Paro o stop		0x40	300 bauds
0x00	1 bit de paro	0x60	600 bauds
0x04	2 bits de paro	0x80	1200 bauds
Paridad		0xA0	2400 bauds
0x00	sin paridad	0xC0	4800 bauds
0x08	paridad par	0xE0	9200 bauds
0x18	paridad impar		

Tabla 6.4.- Parámetros de *abyte* para la iniciación del puerto serie.

Se agrupan en el siguiente orden: velocidad de transmisión, paridad, stop y datos. Por ejemplo al enviar una señal a 2400 bauds, con paridad impar, dos bits de stop y 8 de datos, tomando la lista anterior (0xA0 | 0x18 | 0x04 | 0x03) se tienen el equivalente en hexadecimal al realizar la suma de cada uno de los términos $0xA0+0x18+0x04+0x03 = 0xBF$, por lo tanto *abyte* toma el valor 0xBF. Para tener una mayor claridad en las características del puerto se puede hacer una definición (*define*) de su equivalente (0xA0 | 0x18 | 0x04 | 0x03), lo cual tiene la misma función.

El *bioscom* regresa un valor para todas las funciones (*cmd*), éste es un entero de 16 bits donde los 8 bits más significativos son los bits de estado y los 8 menos significativos varían dependiendo del valor de *cmd*. En la figura 6.11 se muestra la interpretación de cada uno de esos 16 bits

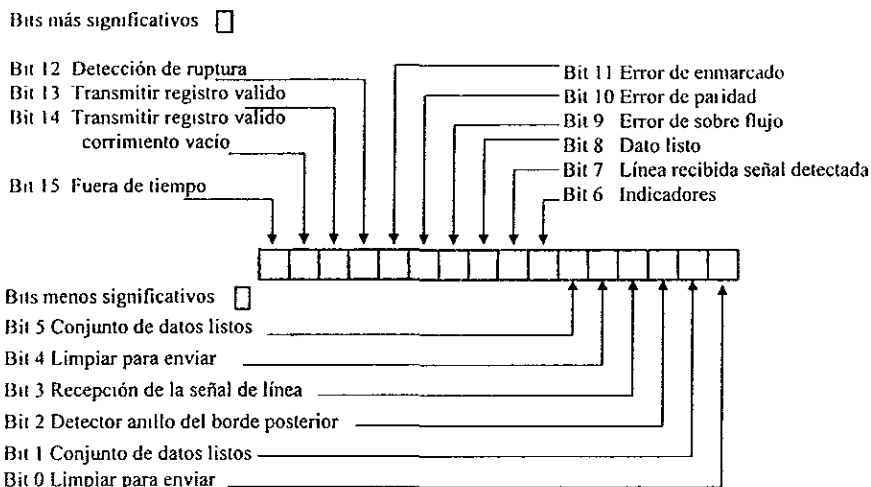


Figura 6. 11.- Información de retorno del puerto serie.

Por ejemplo, si no se pudo enviar el byte, el bit 15 se enciende (1). Cuando el *cmd* es 2 (recepción de datos) el byte leído se encuentra en los 8 bits menos significativos, si no ocurrió algún error, en caso de error por lo menos uno de los bits más significativos estará encendido (1), de otra manera si ninguno está encendido será indicio de que el byte se recibió o capturó sin errores. Cuando el *cmd* es 0 o 3 los bits menos significativos indican el estado del puerto.

INT86

Este comando de C se usa para realizar una interrupción directa, entra al DOS y ejecuta la rutina correspondiente. En su sintaxis, que es la siguiente, usa la declaración "union REGS" para manejar registros específicos de 16 bits, tales como el AX, BX, CX, DX, etc. (ver apéndice A)

```
int int86(int n_int, union REGS *in_regs, union REGS *out_regs),
```

int86 ejecuta la interrupción especificada por *n_int*. Antes de ello, *int86* copia el contenido de *in_reg* en los registros predeterminados. Después de ejecutarse la interrupción, *int86* copia los valores actuales de los registros en *out_regs*. También copia registro de banderas, en el campo *cflag* de *out_regs*.

El valor devuelto por esta función es el valor del registro AX después de la interrupción. Si el valor del campo *cflag* es distinto de 0, quiere decir que ha ocurrido un error y una variable tal como *_doserrno* es puesta al código del error correspondiente.

La interrupción que debe ejecutar este comando para utilizar el puerto serie es la número 14h para *n_m*. De manera similar a la instrucción bioscom, en este caso también se cuenta con las funciones de envío, captura, inicialización y estado del o de los puertos. La función está determinada en el registro AH (ver apéndice A) como sigue:

<i>AH</i>	<i>FUNCIÓN</i>
0	inicializa el puerto de comunicación serie
1	envío de un byte
2	captura o recepción de un byte
3	estado del puerto (status)

Tabla 6.5.- Funciones del registro AH con la interrupción 0x14.

Para AH igual a 0, AL contiene la información del puerto (velocidad de transmisión, bits de paro, de paridad y de datos) en hexadecimal; en DX se especifica el puerto deseado (0 para COM1, 1 para COM2, etc.). Cuando AH es igual a 1, AL contiene el byte que se desea enviar. Sólo cuando AH es 2 o 3, AL no tiene valores de entrada pero si de retorno. Cuando AH es igual a 2 el valor que regresa la función es el byte recibido del puerto, el cual se encuentra en el registro AL de AX. Cuando AH es 3 el valor de retorno se encuentra en AL el cual contiene el estado del puerto

El orden para inicializar el puerto serie esta expresado en 8 bits en el siguiente orden *BBBPPSLL*, donde *BBB* corresponde al valor en binario de la velocidad de transmisión, *PP* es el valor en binario para seleccionar la paridad, *S* el o los bits de stop y *LL* la longitud de los datos, el resultado binario obtenido se convierte a hexadecimal y este valor es el que debe tomar AL cuando AH es 0. Las posibles combinaciones de estos parámetros se muestran en la tabla 6.6.

Como ejemplo, si la velocidad que se requiere es de 4800 bauds (110), con paridad par (01), dos bits de stop (1) y 7 bits de datos (11), queda el número binario 11001111, que cambiado a base hexadecimal es igual a CF. Esto implica generar el código siguiente para el uso de la interrupción

```
union REGS v1,v2;
void rs232_inic()
{
    v1.x.dx=1;           //selecciona el COM 1
    v1.h.al=0xCF;       //selecciona los bauds, paridad, bit stop y
                        //longitud de la palabra (BBBPPSLL)
    v1.h.ah=0;          //función 0 de int 14h a realizar
    int86(0x14,&v1,&v2); //ejecuta la interrupción
}
```

BITS	PARÁMETROS	VALOR DEL BIT		SIGNIFICADO
		EN BINARIO	EN DECIMAL	
7-5	velocidad de transmisión	000	0	110 bauds
		001	1	150 "
		010	2	300 "
		011	3	600 "
		100	4	1200 "
		101	5	2400 "
		110	6	4800 "
		111	7	9600 "
4-3	paridad	00	0	sin paridad
		01	1	paridad par
		10	2	sin asignar
		11	3	paridad impar
2	bits de paro	0	0	1 bit de paro
		1	1	2 bits de paro
1-0	bits de datos	10	0	7 bits de datos
		11	1	8 bits de datos

Tabla 6.6.- Parámetros para inicializar los puertos serie con INT86 ejecutando una interrupción directa al puerto (0x14).

Para completar lo anterior, en cuanto a descripción y manejo de registros, necesarios entre otros para el manejo del comando `int86`, se incluyen algunas explicaciones al respecto en el apéndice A.

REFERENCIAS DEL CAPITULO 6

- 1) Leblanc, G. Turbo C para IBM-PC y compatibles Gustavo Gili, S.A.. España 1988.
- 2) Schildt, Herbert. Guía para usuarios expertos. McGraw-Hill. España 1989.
- 3) Schildt, Herbert. Lenguaje C programación avanzada. McGraw-Hill. México 1989.
- 4) Schildt, Herbert. Programación en lenguaje C. McGraw-Hill. México 1989.
- 5) Schreiber, H Aplicaciones en Computación, Lógica y Regulación. Paraninfo. España 1988.
- 6) Manual: Turbo C, Reference Guide. Borland International. USA 1988.

CAPÍTULO 7

INSTRUMENTACIÓN DEDICADA AL ACELERADOR

En este capítulo se describe la implementación de cada una de las etapas que específicamente constituyeron la instrumentación dedicada al acelerador Van de Graaff de 2 MeV, del Instituto de Física de la UNAM, para su operación y control mediante una microcomputadora tipo PC-AT286 (o superior); en cuanto a la rutina de encendido, el tiempo de exposición y la rutina de apagado del sistema. La planeación de esto se describió en las secciones 3.3 y 3.4 (páginas 18 a 23); a partir de la cual se seleccionaron todas las señales involucradas en la comunicación microcomputadora-acelerador, a través del módulo controlador diseñado y otros circuitos agregados a la consola de controles del acelerador. Con respecto a esto último, según se menciona en la página 21 del capítulo 3, fue necesaria la construcción de un circuito electrónico analógico que al recibir una señal digital (voltaje de c.d., con nivel TTL alto), reprodujese la salida de un variac conectado a línea (120 Vca, 60 Hz), cuando el cursor de éste se gira lenta y paulatinamente desde 0 hasta alcanzar 83 Vrms en su salida (Vpico \approx 117 Volts), con disponibilidad de corriente de 200 mA; a fin de establecer la energía necesaria para la fuente de alimentación de cargas a la banda del acelerador (punto 4 del modo de operación, sección 3.3). El diseño de tal circuito, particularmente denominado *autovariac*, construido e instalado en dicha consola de controles; se describe a continuación para después hacer referencia a todas las señales que envía o recibe la microcomputadora con una breve explicación de sus efectos y/o causas en todo el sistema.

7.1) AUTOVARIAC

El circuito principal de este dispositivo se muestra en la figura 7 1, donde la entrada de la señal digital que lo activa es a través de la terminal indicada como *control* (lado izquierdo del circuito), que se encarga de saturar el transistor 2N2222, activando el relevador Ry. que al cerrar contactos permite que el voltaje + 8Vcd alimente tanto a la fuente de corriente, -formada por el FET 2N5457 y la resistencia de 4.7 k Ω -, como los colectores del arreglo Darlington formado por los transistores BC547 y BC337. Tal fuente de corriente, al cargar el condensador de 1000 μ F, genera un voltaje linealmente creciente (rampa) hasta un máximo de 6.8 V limitados por el diodo zener en paralelo. Así, en un tiempo de aproximadamente 20 segundos, el mismo voltaje linealmente creciente desde 0 pero hasta 5.4 V (valor en el que se estabiliza), aparece en los extremos de la serie formada por la resistencia de 150 Ω y el *led* del encapsulado L-FR (led-fotoresistencia), haciendo que el *led* cambie de su estado de apagado total a emisión de máxima intensidad de luz en forma lenta y paulatina, permaneciendo en el último estado mientras la señal de *control* se mantenga en nivel alto.

Los cambios antes descritos en el *led*, se reflejan en variaciones correspondientes de la fotoresistencia asociada, -desde megaohms hasta cientos de ohms-, logrando con esto cambiar la corriente en el circuito formado por el secundario de un transformador de línea a 12 Vca, un preset de 2.2 k Ω , dicha fotoresistencia y el primario del transformador separador de fases (Tsf) entonado a 60 Hz con un condensador de 3.3 μ F.

Tales cambios de corriente provocan que en el primario del Tsf se genere un voltaje, en forma lenta y paulatina, desde 0 hasta un máximo ajustable por el preset de 2.2 k Ω , en el cual se estabiliza. Nótese que Tsf es la entrada al amplificador de potencia, alimentado por $V_{no\ reg}$, cuya configuración es conocida como "push-pull". Esta etapa incluye los transistores Tip 41, las resistencias de 10 y 500 Ω ; y el transformador de salida (To) que es de línea a 24 Vca con derivación central, 3 A; pero usado al revés (intercambiando primario y secundario).

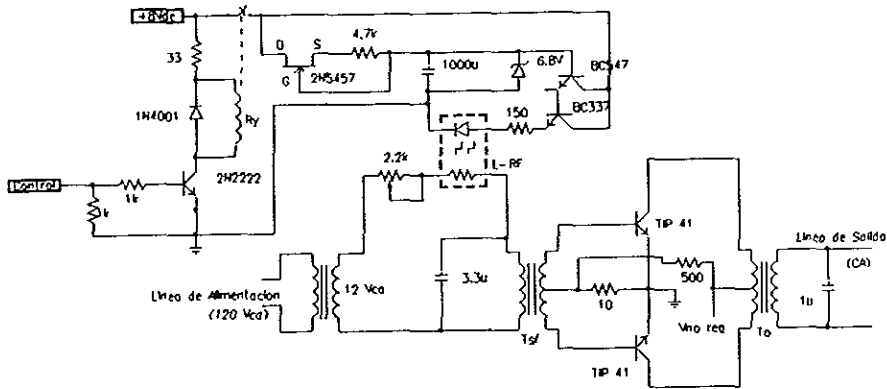


Figura 7.1.- Autovariac.

En resumen, la señal antes descrita se reproduce en la salida del amplificador de potencia, -entonada a 60 Hz por el condensador de 1 μ F-, teniéndose finalmente un voltaje que partiendo de 0 crece lenta y paulatinamente hasta 83 Vrms, con la potencia necesaria requerida por la fuente de alimentación de cargas a la banda del acelerador. De esta forma se reemplazó el uso manual del variac antes mencionado.

En la figura 7.2 se muestra un selector de dos polos tres tiros que permite seleccionar los modos de operación del acelerador, ya sea en forma manual o a través de la computadora, así como el estado de apagado. Nótese que en la posición manual, -en la que se ilustra-, la señal de activación del autovariac es proporcionada por el zener de 5.1 V que a su vez es alimentado por una fuente de 12 Vcd regulados.

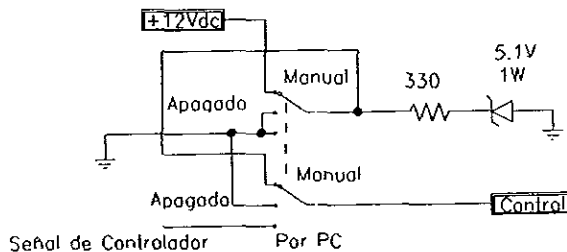


Figura 7.2.- Selector de modo de operación.

Todos los voltajes de alimentación del *autovariac* antes mencionados, son suministrados por la fuente que se ilustra en la figura 7.3, que es una doble configuración típica en la que se usan reguladores 7805. Las disponibilidades de corriente máxima son de 1 A para la salida de 8 Vcd, 2 A para $V_{no\ reg}$ y 2 A para 12 Vcd regulados; esta última salida también alimenta otros relevadores incluidos en la consola, necesarios para la operación automatizada del acelerador, según se describe posteriormente.

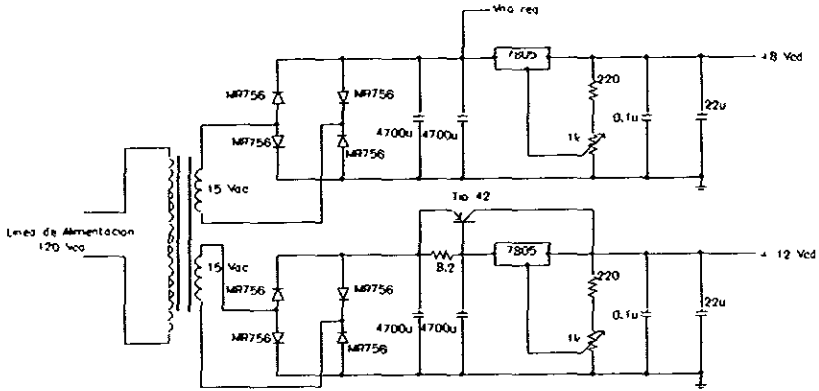


Figura 7.3.- Fuente de alimentación.

7.2) SEÑALES EN EL SISTEMA ACELERADOR

Con respecto a las señales involucradas en el conjunto microcomputadora-acelerador (secciones 3.3 y 3.4), en la tabla 7.1 se resumen las planeadas para, -como mínimo-, cubrir la rutina de encendido, el tiempo de exposición y la rutina de apagado; dejando algunas otras para posibles usos futuros. Cabe recordar que cada una de esas señales se identifican con las combinaciones de las letras S, E, D y A; además del número consecutivo que le corresponde. Tales letras corresponden respectivamente a Salida, Entrada, Digital y Analógica; donde lo de salida y entrada es con referencia a la microcomputadora y el módulo controlador; de esta forma, la señal SD2, por ejemplo, es la señal digital número dos, que envía la microcomputadora (de salida con respecto a esta) a la consola de controles del multicitado acelerador.

De acuerdo a la tabla 7.1, en total se contemplaron 10 señales digitales de salida (SD1 a SD10) y 7 de entrada (ED1 a ED7), todas ellas de 1 solo bit; una señal analógica de salida (SA) bajo conversión D/A de 12 bits; y capacidad para cuatro señales analógicas de entrada (EA1 a EA4) mediante conversión A/D de 12 bits.

NOMBRE	FINALIDAD
SD1	Activa el motor de la banda
ED1	Confirma el aviso de radiación lista
SD2	Inhíbe radiación
ED2	Confirma radiación inhibida
SD3	Activa <i>autovariac</i>
SA	Establece alto voltaje de aceleración
ED3	Confirma radiación lista, ya con alto voltaje
EA1	Monitoreo de alto voltaje de aceleración
SD4	Activa motor a pasos (control de corriente de emisión)
SD5	Avance o retroceso del motor a pasos
SD6	Giro rápido del motor a pasos
SD7	Giro lento del motor a pasos
EA2	Monitoreo de corriente de haz emitido
SD8,SD9	Para selección de la entrada analógica
SD10	Reservada para uso futuro
ED4 a ED7	Monitoreo de parámetros de seguridad
EA3 y EA4	Reservadas para usos futuros

Tabla 7.1.- Resumen de señales involucrada en el acelerador.

Para la implementación de las etapas específicas necesarias para operar con las señales anteriores, mencionadas muy generalmente en el capítulo 6; a continuación se describen los procesos y circuitos involucrados. De manera similar a lo visto en el capítulo 6, se inicia con los dispositivos de transmisión y captura de datos por la PC; y posteriormente se detalla el programa encargado de mantener el control y vigilancia de todo el sistema acelerador.

7.3) TRANSMISIÓN

SEÑAL DE CONTROL

La señal de control tiene las características siguientes: consta de 20 bits dividida en dos porciones o trenes de 10 bits; donde cada tren está constituido por una palabra de 8 bits y adicionalmente para el control de la transmisión: un bit de inicio, sin paridad y un bit de paro. De los dos conjuntos de 10 bits, los 16 bits correspondientes a la suma de dos palabras se reparten en: 4 bits para control o direccionamiento de dispositivos y los 12 bits restantes (dato en sí) se utilizan para su conversión digital-analógica. Se utiliza al puerto serie 2 (COM2), debido a que algunas máquinas ocupan el puerto serie 1 (COM1) para conectar el ratón (*mouse*); se opera a una velocidad de transmisión de 1200 bauds debido a que ésta es soportada por la mayoría de los equipos PC. Cada una de las etapas siguientes utiliza los parámetros de transmisión anteriormente establecidos.

CONVERSIÓN DE RS-232C A TTL

Partiendo de las características del puerto RS-232C, para su conexión con otros equipos, primero se debe de cumplir con el acoplamiento de impedancias; un circuito que es capaz de realizar esta función es el mostrado en la figura 7.4:

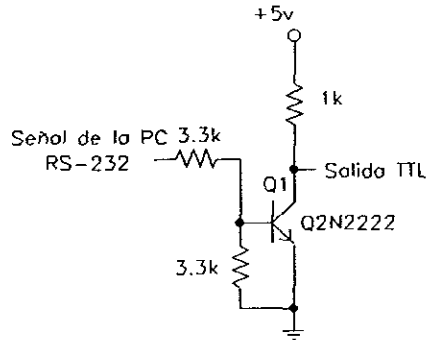


Figura 7.4.- Acoplador de impedancias a transistor bipolar

En el circuito anterior la impedancia de entrada está formada por la resistencia superior de 3.3 k en serie con el paralelo de la unión base emisor del transistor y la otra resistencia de 3.3 k, que equivalen a un valor superior a 3.3 k; esto porque la señal de salida del puerto RS-232C requiere una impedancia de carga entre 3 k Ω y 7 k Ω . Esta configuración de emisor común para el transistor 2N2222 (operado en corte y saturación) permite discriminar los pulsos negativos de la señal RS-232C, que únicamente cortan al transistor, mientras que los positivos lo saturan, teniendo así en la salida de colector una señal negada de 5 Volts de amplitud (TTL) cuya corriente máxima es limitada por la resistencia de 1 k Ω que además protege a la fuente en cada "corto circuito" que ocurre cuando el transistor se encuentra en saturación ($V_{CE} = V_{CESAT}$). La negación de la señal de salida con respecto a la de entrada fue necesaria ya que un 0 en RS-232C corresponde a un nivel de voltaje alto y un 1 a un nivel de voltaje bajo. Con el circuito mencionado que además de la conversión de nivel, realiza dicha inversión de la señal, se obtiene para una entrada de nivel alto (0 lógico RS-232) una salida baja (0 lógico TTL) y viceversa. En la figura 7.5 se ilustra el proceso de conversión de nivel e inversión de señal.

Entrada Lógica (Señal RS-232)		Inversor		Salida Lógica (Señal TTL)
0 (+12 V)	→		→	0 (0 V)
1 (-12 V)	→		→	1 (5 V)

Figura 7.5.- Valores lógicos y niveles de las señales RS-232C y TTL.

LA SINCRONIZACIÓN Y EL RELOJ DE RECEPCIÓN.

Para poder establecer la comunicación se requiere que ambos equipos se sincronicen, lo que se logra mediante una señal de inicio o activación y se debe mantener con un reloj a la frecuencia de transmisión de los bits.

En este caso, como cada tren de bits transmitido tiene un bit de inicio (transmisión asincrónica), este también se utiliza para iniciar la sincronización. El reloj encargado de mantener tal sincronización, requiere también de una señal de control que lo activa únicamente cuando se detecta el bit de inicio de cada uno de los trenes y lo desactiva al terminar el mismo.

Tomando como referencia que la duración de un bit a la velocidad de transmisión de 1200 bauds es: $1/1200 \approx 0.833$ ms y en consecuencia la duración de un tren de 10 bits equivale a 8.33 ms, fue posible diseñar un temporizador y el reloj para estos tiempos, mediante dos microcircuitos LM555, uno operando como monoestable para el primero y el otro como astable para el segundo.

El temporizador requirió de una duración de por lo menos de 9 bits, ya que el bit de paro (stop) no es indispensable para las etapas utilizadas y se puede omitir; el bit de inicio se considera en el momento de que se produce el primer cambio de nivel; con lo que se activa al monoestable con un flanco negativo o de bajada, a partir del que empieza a correr el tiempo de sincronización con duración mínima de $9 \times 0.000833 \approx 0.0075$ segundos. Con este dato y las especificaciones y recomendaciones proporcionadas por el fabricante del LM555, se configuró el temporizador mostrado en la figura 7.6, donde $R_A = 69$ k Ω y $C = 0.1$ μ F.

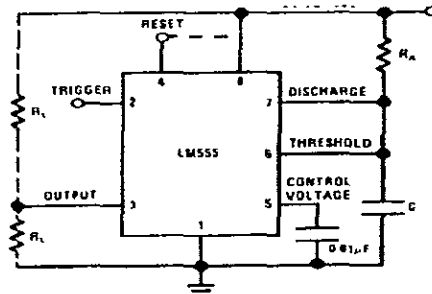


Figura 7.6.- Temporizador mediante un 555 trabajando como monoestable.

El reloj se diseñó con una frecuencia igual al inverso de la duración de un bit, o lo que es lo mismo, con frecuencia igual a la velocidad de transmisión (1200 Hz)

Para determinar los valores de las componentes necesarias al circuito correspondiente, mostrado en la figura 7.7; se seleccionó un ciclo de trabajo (*Duty*): $D = 1/3$; y análogamente al caso anterior, bajo configuración y especificaciones de fábrica del LM555, se usaron $R_A = 5.6$ k Ω , $R_B = 5.6$ k Ω y $C = 0.068$ μ F.

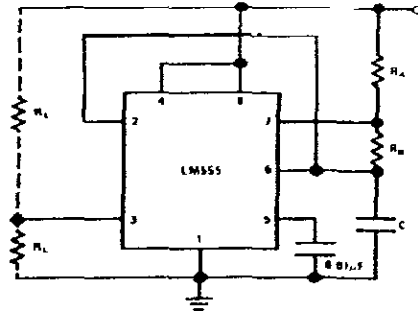


Figura 7.7.- Reloj de captura, con un 555 trabajando como astable.

El reloj depende del temporizador y éste a su vez del bit de inicio de cada tren. Para disminuir el espacio requerido por los dos LM555 se sustituyeron por un doble LM555: el LM556.

En la figura 7.8 se ilustran las señales del temporizador, del reloj y de un dato en dos trenes de 8 bits, en los que en cada uno de ellos se asocia el valor 85 decimal correspondiente a 01010101 en binario. Las señales se interpretan de izquierda a derecha, con la señal de reloj sincronizada con la de datos.

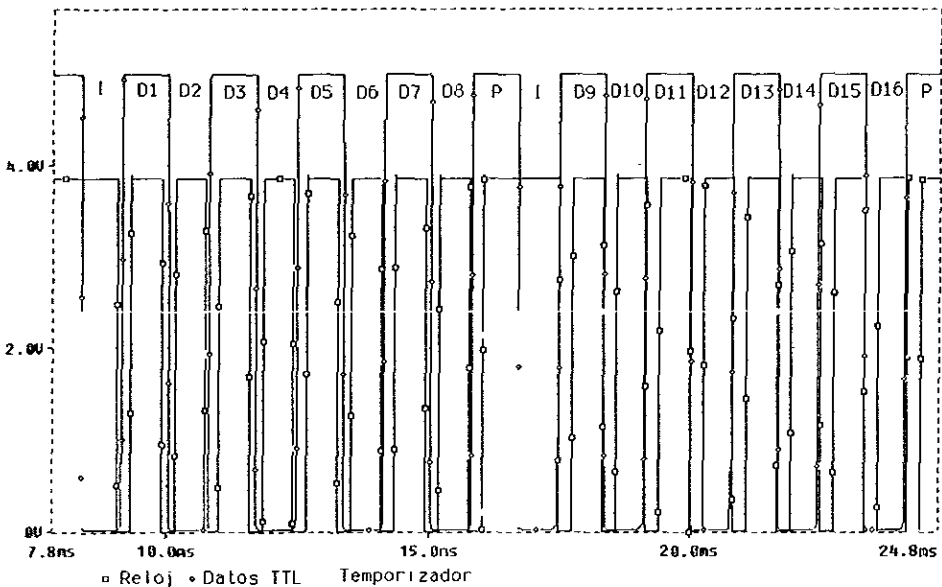


Figura 7.8.- Sincronización de señales.

El paso siguiente fue la captura y conversión de los bits.

CAPTURA Y CONVERSIÓN DE BITS DE SERIE A PARALELO

Como se puede observar en la figura anterior, la señal es enviada bit a bit; es decir que se reciben en el equipo los bits en forma serie. En cada tren de bits transmitido por la computadora, con los respectivos bits de inicio y paro, los que corresponden a la palabra se envían a partir del menos significativo al más significativo. Para realizar la captura se utilizaron registros de corrimiento.

Los registros de corrimiento son dispositivos que van almacenando el o los bits que le llegan, durante un tiempo determinado por una señal de reloj. Un registro sencillo puede ser un flip-flop tipo D, donde el bit de salida depende del bit que se encuentre a la entrada y el cambio ocurre cuando la señal de reloj pasa de 0 a 1. Mientras la señal de reloj permanece en 0 o en 1, el bit de salida no cambia aún si en la entrada se presentan cambios; en este caso las señales de entrada y salida se pueden tomar como serie o paralelo.

Cuando se conectan dos o más flip-flop's en cascada, es decir que la señal de salida de cada uno se conecta a la entrada del siguiente y así sucesivamente, la posición final de cada bit que entra depende de los cambios de la señal de reloj y mientras el reloj funcione, todos los bits estarán cambiando consecutivamente de posición; es decir, que cambiará primero el bit "a" a la salida 1 con lo cual éste se encontrará en la entrada 2. En el proceso de cambio de la entrada 2, la entrada 1 tendrá un nuevo bit "b" el cual también tendrá el mismo comportamiento y desplazará el primer bit enviado a la entrada 3 (salida 2). Lo anterior se aplica para una señal que entra en serie y puede utilizar una salida en serie y/o en paralelo si se conecta en cada salida una línea; pero como la señal de entrada es serie generalmente se utiliza éste para una salida en paralelo, usando salida serie solo en el bit mas significativo, para obtener un mayor número de bits al ser conectados dos o mas registros de corrimiento.

Existen microcircuitos que contienen varios flip-flop's en un solo encapsulado (no necesariamente del tipo D) lo que representa un ahorro de espacio; además no solo manejan entrada serie sino que también hay los que manejan la entrada en paralelo. A continuación se describe el utilizado en este trabajo.

El SN74LS164 contiene dos registros de corrimiento de 8 bits con entrada serie y salida en paralelo. Este circuito integrado requiere de 4 señales de entrada: reloj, dos para datos y un reset maestro; el reloj es el descrito para la sincronización, una entrada de datos y la terminal de reset se conectan a 5 Volts. La salida correspondiente al octavo bit se conecta a la entrada del otro registro para obtener 16 bits.

Como se mencionó en la etapa de sincronización, de cada uno de los dos trenes de 10 bits enviados por la computadora, se capturan los 9 primeros correspondientes a un bit de inicio y ocho de datos; esto ocasiona que en el registro de corrimiento se almacenan solo los 16 bits comprendidos entre el tercer bit del primer tren y el último del segundo, inclusive; con lo que se elimina tanto el primer bit de inicio como el bit menos significativo del primer tren, capturándose totalmente el segundo tren con su bit de inicio incluido. Para no perder dicho bit menos significativo, fue necesario agregar al SN74LS164 un registro de corrimiento de 1 bit (SN74LS74). La salida de datos en paralelo consiste en 15 líneas del primer microcircuito (descartando el bit de inicio del segundo tren) y una del último, en el orden adecuado.

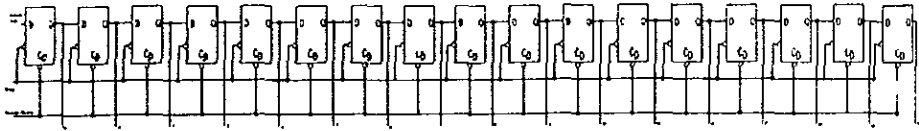


Figura 7.9.- Conversión serie a paralelo

De esas 16 líneas o bits, antes de que puedan ser utilizadas pasan por un proceso de almacenaje, el cual tiene como función la de mantener los bits anteriores durante el proceso de una nueva adquisición de bits; esto es principalmente útil ya que durante la captura (en la salida de los registros) los bits están disponibles en cualquier momento y una lectura "instantánea" no corresponderá a su valor o ubicación real, ya que en el proceso de captura van cambiando los bits hasta que finaliza ésta y cada bit está en el lugar que le corresponde.

ALMACENAMIENTO

Así como los registros de corrimiento utilizan flip-flop's en cascada, también hay arreglos de éstos en paralelo que se utilizan para almacenamiento temporal, a tales dispositivos se les conoce como *latches*. Un *latch* sencillo es un flip-flop simple que en lugar de utilizar una señal de reloj maneja una señal de lectura/muestra; durante la lectura la salida es igual a la señal de entrada y durante la muestra la salida es la representación de la entrada cuando se efectuó la lectura anterior. Esta representación no se modifica aún si en la entrada se encuentra otra señal diferente.

Existen diversos tipos de *latches* donde las características principales son el número de entradas (bits), el tipo de salida (normal o de tres estados); y en menor grado el tipo de señales que requieren para su funcionamiento. Con dos SN74LS373, *latches* de 8 bits, se pueden conectar las 16 líneas de la salida en paralelo de los registros de corrimiento anteriores.

Debido a lo antes mencionado fue necesaria una señal de activación, para accionar a los *latches* en el momento en que se terminaban de enviar los dos trenes de bits. Recordando que en el temporizador se obtienen dos pulsos durante el proceso de captura, si éstos se cuentan y al llegar a dos se manda una señal, ésta última activa durante un tiempo la lectura. La mayoría de los contadores que existen puede adaptarse; el seleccionado fue el SN74LS192, contador arriba/abajo de 4 bits, que inicia su cuenta cuando recibe un cambio en la entrada con flanco positivo (figura 7.10).

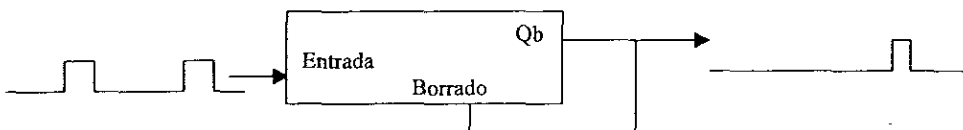


Figura 7.10.- Activador de memoria temporal (latch)..

Al llegar a una cuenta igual a dos, la salida Q_B se pone en alto; así esta señal es la que se ocupa tanto para activar los *latches* como para el *reset* del contador preparándolo para una nueva cuenta. De utilizar directamente tal señal, la lectura se realizaría durante el envío del segundo tren; lo correcto es que la lectura se produzca al terminar la captura de los dos trenes. Esto ocurriría debido a que la señal del temporizador realiza dichos flancos de subida al iniciar cada palabra y no al finalizar; así que para corregir esto fué necesario invertir la señal del temporizador logrando así los flancos de subida al final de cada pulso.

La señal de activación (y *reset*) obtenida del contador, hasta ahora descrita, es un pulso cuya duración es tan corta que los *latches* no alcanzarían a detectarla y no se efectuaría la lectura (el *reset* pone a cero al contador lo que ocasiona que Q_b cambie a 0 que es la causa de la corta duración de la señal). Para solucionar esto se utilizó otro temporizador, nuevamente un LM555 trabajando como monoestable; éste requiere de un cambio de nivel de alto a bajo y del contador se obtiene de bajo a alto, por lo que fué necesaria una inversión para dispararlo. La duración del temporizador fue de aproximadamente 3 ms, de donde se obtuvieron los valores mostrados en el circuito de la figura 7.11, análogamente al caso anterior.

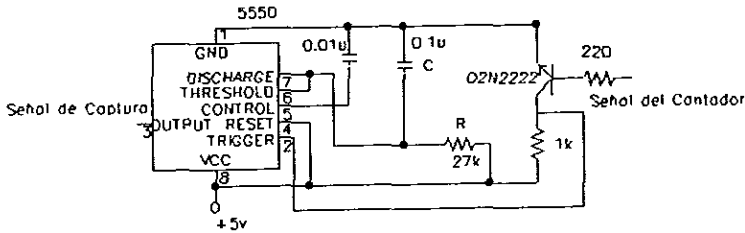


Figura 7.11.- Diagrama de temporizador e inversor.

CONVERSIÓN DIGITAL-ANALÓGICA

Con los 16 bits almacenados en los *latches* el siguiente paso fué distribuirlos en 4 para control y los 12 restantes para el dato en sí, a convertir posteriormente a tipo analógico. Para tal distribución se escogió del primer tren enviado (figura 7.8) a los primeros cuatro bits (después del bit de inicio) como los más significativos del dato, mientras que los otros cuatro bits del mismo tren se reservaron para control. Del segundo tren, todos los ocho bits corresponden, en el orden recibido, a los menos significativos del dato; logrando así en éste un total de 12 bits.

Para la conversión digital-analógica (DAC) en 12 bits se usaron dos convertidores de 8 (circuito integrado DAC08) cuyas correspondientes salidas se sumaron analógicamente mediante una configuración de amplificadores operacionales que además permitió ajustar un factor de ganancia en la respuesta final.

El primer DAC utiliza directamente las salidas del *latch* que contiene los 8 bits menos significativos del dato; en el otro DAC sólo se utilizan las entradas más significativas y los 4 restantes se conectan a tierra para evitar ruido o variaciones que pudieran ser ocasionados al dejar

sin conexión dichas terminales. Las conexiones restantes para el funcionamiento de cada DAC son los propuestos por el fabricante, modificando únicamente el ajuste del nivel de referencia. En la figura 7.12 se muestra dicho convertidor.

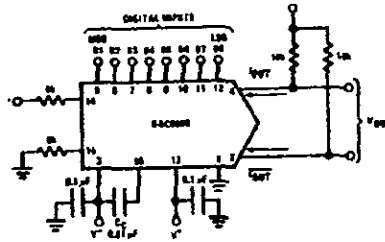


Figura 7.12.- Convertidor Digital-Analógico de 8 bits

Para el primer DAC se toma la salida de la terminal 4; debido a que maneja 8 bits correspondientes a 256 niveles (2^8), para 15 Volts cada nivel tiene un valor equivalente de $15/256=0.05859$ Volts, así la salida máxima es de 14.9414 Volts (255×0.05859 , un nivel equivalente al valor 0 se pierde). El segundo DAC cuenta con la misma resolución y por consiguiente también cada nivel, con la diferencia de que su salida máxima es distinta. Los 4 bits más significativos convertidos en su equivalente decimal considerando su posición, proporcionan un valor igual a 240, que se obtiene de sumar $16+32+64+128$ lo que corresponde al quinto, sexto, séptimo y octavo bit respectivamente, dando como resultado el tener un voltaje de salida máximo dado por $240 \times 0.05859 \approx 14.0625$ Volts.

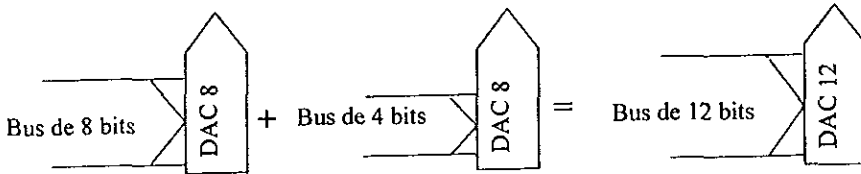


Figura 7.13.- Convertidor Digital Analógico equivalente a 12 bits.

Para obtener de ambos circuitos un DAC equivalente de 12 bits, se requirió sumar ambas señales analógicas, según el esquema de la figura 7.13; para ello se utilizó un amplificador operacional en una configuración de sumador inversor, ajustando la ganancia de cada señal para que los 8 bits del primer DAC sean considerados como los bits menos significativos de los 12; y los 4 restantes del segundo asignándolos como los más significativos. La señal resultante entra a otro amplificador operacional en configuración de inversor con el fin de tener una salida positiva y con ajuste de la ganancia en la misma. El circuito de tal convertidor de 12 bits, se muestra en la figura 7.14, donde $R_1=560\text{ k}\Omega$, $R_2=22\text{ k}\Omega$, $R_3=1.5\text{ k}\Omega$, $R_4=680\text{ }\Omega$, $R_5=10\text{ k}\Omega$, $R_6=1\text{ k}\Omega$, $R_7=470\text{ }\Omega$, $Pot=2.2\text{ k}\Omega$.

ESTA TESTS NO DEBE
SALIR DE LA BIBLIOTECA

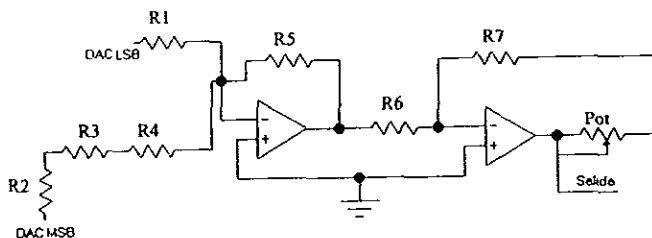


Figura 7.14.- Suma de señales.

DIRECCIONAMIENTO

La interacción con cada uno de los dispositivos del sistema se selecciona con el módulo de direccionamiento, el cual está dedicado tanto para controlar los dispositivos internos como externos. Esta sección es la que recibe las direcciones provenientes de la computadora y activa o desactiva dichos dispositivos, equipos o procesos.

En el capítulo anterior se mencionaron algunas ventajas del direccionamiento que utiliza las cuatro líneas paralelas que restan de la información de los trenes enviados por la computadora (4 de los 16 bits). También se describió brevemente este proceso que ahora se optimiza con algo de lógica adicional para beneficios en el número líneas o dispositivos para control.

Para realizar esto se requirió adicionalmente un bit o señal de activación que proviene del módulo de captura y conversión serie-paralelo; esta señal es la que pone en ceros al contador para indicar que la captura terminó e inicie el almacenamiento; pero también se utiliza en este módulo para indicar la captura de los datos; esta señal pasa por un retardador (delay) utilizando un LM555 funcionando como monoestable. Tal retardo es para asegurar que los datos puedan almacenarse ya que de otra forma el pulso que se produce es de una duración menor a la requerida por el dispositivo de direccionamiento. Este último se compone de tres secciones: la primera consta de un decodificador de 4 a 16 líneas, la segunda es para almacenamiento y la tercera de potencia. Para los dispositivos internos la sección de potencia no es necesaria ya que no se requiere de corrientes altas en las compuertas TTL.

Primero se describen las dos primeras secciones que son generales para los dispositivos externos e internos; la primera parte es simplemente un decodificador 74LS154, la segunda lo componen el 74LS14 que consta de 6 inversores, un 74LS74 que es un par de flip-flop's tipo D y un LM555. El funcionamiento en conjunto consiste en que el decodificador mantiene activada la salida seleccionada con las cuatro líneas almacenadas en él hasta que recibe nuevamente la señal de captura con lo que se desactiva de forma automática la salida anterior y se activa la nueva; por este proceso se tiene solo una línea seleccionada de las 16, cuyo estado depende de que no llegue una nueva dirección (tabla 7.2).

D	C	B	A	Dirección
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	0	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Tabla 7.2.- Direcciones con 4 bits.

Lo anterior permitiría seleccionar solo un receptor a la vez imposibilitando el tipo de direccionamiento ilustrado en la tabla 6.2 del capítulo anterior. Esto hizo necesario el almacenamiento del estado de cada dirección independientemente, por ejemplo, si las direcciones 2, 5 y 8 se encuentran en estado activo pero se desactiva la dirección 5 el resto de las salidas no deben sufrir cambio alguno. Tal proceso se obtiene por medio de un flip-flop para cada dirección; éste requirió de una señal de reloj y otra adicional; la de reloj proviene del decodificador (dirección en sí) que pasa primero por el inversor para obtener una señal con flanco de subida (figura 7.15).

La señal adicional también proviene del decodificador pero a través de un LM555 funcionando como monoestable, que sólo es activado con la dirección 15; en este caso no se efectúa el proceso de almacenar el bit o señal, lo que ocurre es que dicho cambio activa al monoestable y se obtiene una señal con nivel alto durante 10 ms, tiempo suficiente para esperar otro tren con la dirección deseada. Cuando no se usa la dirección 15 la salida del monoestable permanece en nivel bajo.

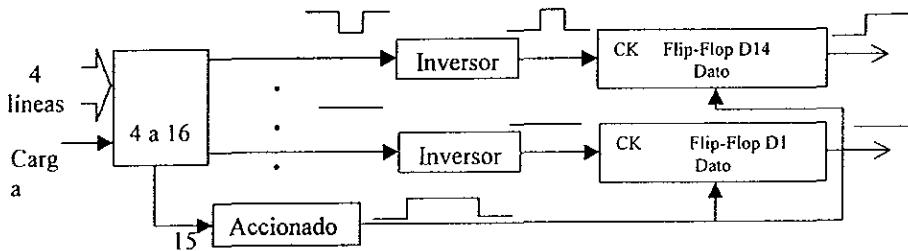


Figura 7.15.- Señales de activación y desactivación.

Cuando se envía el código 15, éste viene seguido de la dirección correspondiente al dispositivo que se activa; para desactivar, únicamente se recibe la dirección del dispositivo deseado. Por ejemplo, según se ilustra en la figura 7.15, si inicialmente las direcciones 14 y 1 tienen un estado bajo, para activar la dirección 14 se manda primero la dirección 15 con lo cual el activador cambia de estado dejando en la entrada de todos los flip-flop un nivel en alto, pero el único que lo captura es el 14 ya que es el que recibe la dirección enviada.

Esto involucra transmitir dos veces la señal, en la primera transmisión (tren de activación) se envían los datos correspondientes a la salida analógica (12 bits) y la dirección de activación (los cuatro bits restantes en 1, dirección 15: 1111); en el segundo tren (de dirección) se envía la misma información para la salida analógica y esta vez la dirección del dispositivo; para desactivar solo se envía el de tren de dirección. En el tren de activación y en el de dirección los bits de datos son los mismos (de A1 a A12) como se muestra en la figura 7.16.

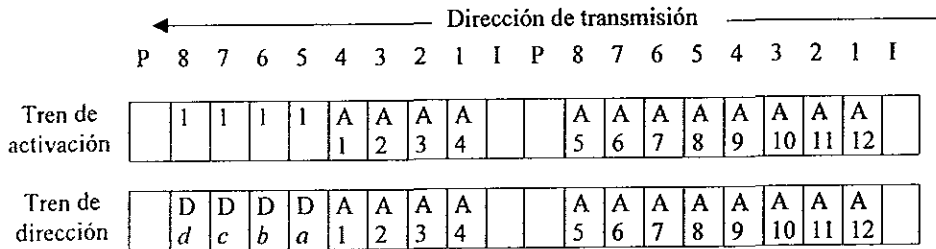


Figura 7.16.- Tren de activación y de dirección *dcb*a.

De las 14 direcciones restantes (la dirección 0 para espera y la 15 para activación) se emplean 3 para controlar el convertidor analógico digital y dos más para seleccionar su entrada analógica, estas últimas son seleccionadas por medio de un multiplexor analógico, en este caso mediante minirelevadores utilizados como se muestra en la tabla 7.3; como se tienen 2 bits existen cuatro opciones, en la figura 7.17 se muestra su forma de conexión.

Q1	Q2	Entrada analógica
0	0	SA4
0	1	SA3
1	0	SA2
1	1	SA1

Tabla 7.3.- Selección de señal de entrada.

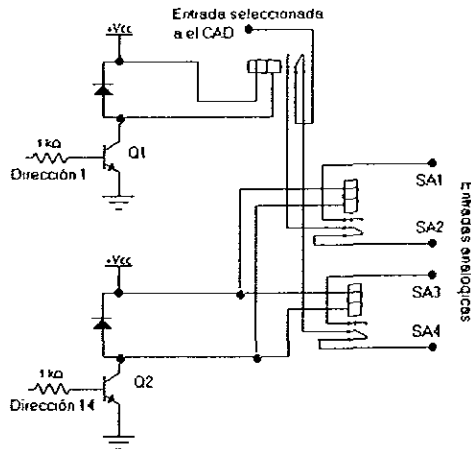


Figura 7.17.-Multiplexor analógico

Para el control de los dispositivos externos se requirió de una corriente mayor, por lo que se utilizaron dos microcircuitos 7406, cada uno de ellos un inversor séxtuplo con salida a colector abierto. A estas compuertas entran las nueve direcciones restantes que se utilizan para accionar los mandos en la consola de control del acelerador, mediante relevadores incluidos en la misma.

7.4) RECEPCIÓN

CONVERSION ANALÓGICA-DIGITAL

Se realizó mediante el circuito integrado ADC12441 que utiliza el método de aproximaciones sucesivas, cuyas características principales son: opera con 13 bits (12 de datos y 1 para el signo), tiene un ciclo de autocalibración para ajustar la linealidad, su error a escala completa es menor a medio LSB, durante cada conversión realiza un ciclo de auto-cero para corregir el error a cero. Debido a que este dispositivo es capaz de convertir señales analógicas positivas y negativas (siempre que la señal a convertir no se encuentre fuera del rango de ± 5 Volts), en el caso de que la señal de entrada sea negativa su representación es una salida complementaria a la positiva y el signo está activado, pero si es positiva la salida se lee directamente y representa cada bit un valor analógico.

Este convertidor requiere de algunas señales adicionales para su control, esto se debe a que, a diferencia del convertidor D/A que no requiere mas que las de referencia, en la conversión A/D es necesario capturar (tomar una muestra de la señal), iniciar la conversión y la salida se obtiene por medio de *latches* por lo que se utiliza una señal para la lectura de los datos al terminar la conversión; en caso contrario presenta una alta impedancia. Las señales de control son pulsos (de nivel alto a bajo); el tiempo entre cada conversión es de aproximadamente 60 ciclos de reloj (ajuste a cero, adquisición y conversión).

Otra diferencia con respecto al DAC es el uso de un reloj para realizar la conversión. El fabricante además de los niveles de voltaje establece una frecuencia mínima y máxima de trabajo, ésta se encuentra desde los 500 kHz a los 4 MHz; utilizando frecuencias menores a los 500 kHz el CAD realiza la conversión pero con menor precisión.

La frecuencia de reloj seleccionada fue de 2 MHz, para esto se utilizó un multivibrador astable como se muestra en la figura 7.18, donde $R_a = 180 \Omega$, $R_b = 18 \text{ K}\Omega$ y $C = 47 \text{ pF}$.

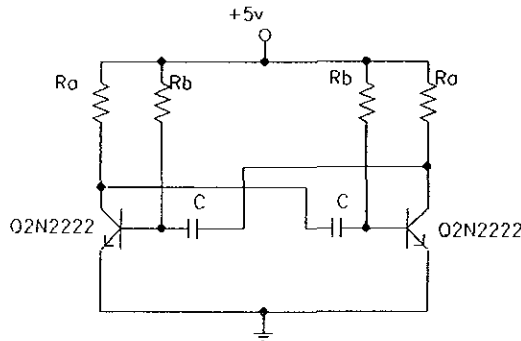


Figura 7.18.- Multivibrador Astable.

Las salidas del CAD pasan por unos *buffers* para conectarlos a circuitos TTL. Los *buffers* utilizados se encuentran en el CD4009, el cual es un circuito integrado tipo CMOS, con 6 *buffers*, muy útil para acoplar con circuitos TTL ya que proporcionan una corriente de salida mayor que los circuitos MOS convencionales. Para cubrir las 13 líneas o bits, se utilizaron dos de estos microcircuitos y un *buffer* discreto de un bit, mediante un transistor configurado con una impedancia de entrada alta. Estas salidas se conectaron a un *latch* (SN74LS245, de 8 bits) para almacenar los datos y prepararlos para su envío a la computadora.

Para la captura y presentación de datos se utilizan dos de las señales del direccionamiento; el CAD requiere de tres señales de las que una es común, por lo que se utiliza una compuerta OR, según se ilustra en la figura 7.19. En la tabla 7.4 se muestran los estados de RD y WR que son salidas del decodificador y de CS en la compuerta OR.

RD	WR	CS	Acción
0	0	0	Alta impedancia de salida
0	1	1	Captura la señal
1	0	1	Presenta los Datos
1	1	1	No se aplica

Tabla 7.4.- Señales para activar el CAD.



Figura 7.19.-Compuerta lógica OR.

ENTRADA DE SEÑALES DIGITALES

Se reservaron 8 entradas digitales para las señales provenientes de la consola del acelerador, que vigilan algunas secciones básicas para el correcto funcionamiento del mismo y otras son de seguridad. Todas son convertidas a niveles TTL para que puedan ingresar a un codificador de prioridad (74LS148), dispositivo que realiza un muestreo de las ocho entradas y a cada una le asigna una prioridad según se ilustra en la tabla 7.5.

0	1	2	3	4	5	6	7	Salida C	Salida B	Salida A	Prioridad
H	H	H	H	H	H	H	H	1	1	1	X
X	X	X	X	X	X	X	L	0	0	0	1ª
X	X	X	X	X	X	L	H	0	1	1	2ª
X	X	X	X	X	L	H	H	0	1	0	3ª
X	X	X	X	L	H	H	H	0	1	1	4ª
X	X	X	L	H	H	H	H	1	0	0	5ª
X	X	L	H	H	H	H	H	1	0	1	6ª
X	L	H	H	H	H	H	H	1	1	0	7ª
L	H	H	H	H	H	H	H	1	1	1	8ª

Tabla 7.5.- Prioridad de señal

La prioridad de la señal, en estado bajo (L), es mayor mientras mas grande sea su número de entrada; es decir que la señal número 7 es la mas importante o tiene la máxima prioridad para ser procesada, ya que no depende del resto de las entradas; esto es por si se presenta en cualquiera de las entradas un cambio a nivel bajo solo se toma en cuenta la entrada mayor y cualquiera que esté por debajo de ésta se ignora (X indica irrelevancia, puede estar en cualquiera de los dos estados); esto es similar al funcionamiento de las interrupciones (IRQ) dentro de la computadora.

ENVIO DE DATOS

Para el envío de datos a la computadora se consideraron 16 bits (13 por parte del CAD y 3 del codificador) que necesitaron de formato previo. Durante cada envío, esos 16 bits se encuentran en paralelo almacenados en los SN74SL245 por lo que fue indispensable convertirlos a tipo serie. En la transmisión se explicó que hay registros de corrimiento que manejan la entrada serie o paralelo y salida en serie, como el SN74LS165 de 8 bits, en el que se utilizaron ambas entradas mediante un pulso de control que realiza la captura en paralelo y el inicio del desplazamiento de los bits. Para su funcionamiento se requirió de una señal de inhibición de reloj,

una de reloj y otra de carga en paralelo. Para cuando la señal de carga en paralelo es activada (nivel bajo o 0) las otras dos señales no tienen efecto en la salida; cuando esta señal está en alto (1) y dependiendo de las otras dos entradas, se efectuará o no el corrimiento. La señal del inhibidor de reloj se pone a 0, cuando el reloj cambia con flanco positivo se realiza el corrimiento y en cualquier otro caso no hay cambio (no se produce corrimiento).

Análogamente a la recepción, la señal a enviar debe de contener los 8 bits de datos, los bits de inicio y paro, sin bit de paridad; de esta forma se manejan dos trenes de 10 bits tales que el primero contiene el bit de inicio, 3 bits del codificador, los 5 mas significativos del dato procedente del CAD y el bit de paro; mientras que el segundo tren lleva su bit de inicio, los 8 menos significativos del dato y el bit de paro. Ese total de 20 bits requirió de 3 registros de corrimiento (SN74LS165) conectados en cascada; esto es, primero realizando la captura en paralelo de cada uno y prosiguiendo con la transmisión en serie. Las 8 entradas de cada chip (un total de 24 bits) se distribuyeron en dos trenes de 12 bits, para mantener una simetría con los circuitos además de incluir bits de espera.

Para iniciar la transmisión debe de ocurrir un cambio, tomando en consideración que la salida de datos esta por la salida negada del último registro. Por lo anterior se inicia el primer bit como un nivel bajo (0), el cual en adelante se considera como nivel de referencia así que el cambio necesario para que se inicie el proceso de envío, debe ser un nivel alto (1) el cual se debe ubicar en el siguiente bit y es a partir de éste cuando los siguientes 8 bits consecutivos corresponden a los primeros 8 bits de datos; el bit de paro es un 0 el cual se ubica a continuación del último bit de los datos, con esto se termina el primer tren bits. El segundo tren tiene una distribución similar al primero, se conecta enseguida del bit de paro del primer tren y así consecutivamente hasta completar los 11 bits restantes. En realidad se están manejando 11 bits por tren ya que se antepone al bit de inicio un bit de referencia (de espera), el cual es muy importante para el primer tren debido a que éste es el bit que aparece en el momento de la carga en paralelo; en el segundo tren no es tan forzoso ya que puede utilizar el bit de paro del primero, pero para mantener una simetría se incluye.

Similarmente a la recepción, para el envío de datos también fue necesario un reloj con una frecuencia igual al inverso de la duración de un bit; es decir, con una frecuencia igual a la velocidad de transmisión ($f = 1200 \text{ Hz}$); mismo que también se implementó con un LM 555 en configuración de multivibrador astable, de igual forma que el construido para la recepción (figura 7.7). El de esta parte se muestra en la figura 7.20.

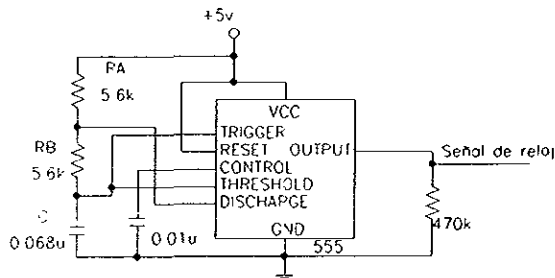


Figura 7.20.- Reloj para envío de datos.

Como se explicó en capítulos anteriores, las señales tanto de salida como para la entrada en niveles RS-232 tienen como estándar el representar un 0 como un nivel alto o voltaje mayor a 3 Volts y a un 1 como nivel bajo o voltaje menor a -3 Volts; lo que hizo necesaria la conversión TTL a RS-232.

CONVERSIÓN TTL A RS-232C

Para finalmente llegar a la computadora solo quedaba convertir la señal de los registros (en niveles TTL) a los niveles RS-232C. Para ello se utilizó el MAX 232, que es un circuito integrado cuyas características principales son las de tener dos entradas tipo RS-232C con salidas respectivas TTL (*receiver*), además de dos entradas TTL con salidas RS-232C (*driver*); opera con una sola fuente de +5V y solo requiere de cuatro capacitores de 1 μ F.

Para concluir el envío de datos se debe tener sincronía con la computadora, por lo que fue necesaria una señal de control; para esto se utilizó una de las señales que conforman la transmisión RS-232 a TTL: *la solicitud para recibir*, la cual es una autorización que da la computadora cuando el puerto esta disponible para recibir información. Esta señal se utilizó para la captura en paralelo de bits en los registros de corrimiento. Cuando la salida TTL del MAX 232 se encuentra en 0 es porque la computadora se encuentra transmitiendo información al equipo y los registros se encuentran en captura; cuando la señal esta en 1 la computadora solicita los datos del equipo y los registros empiezan a enviar los bits. En la figura 7.21 se muestra el circuito del MAX232.

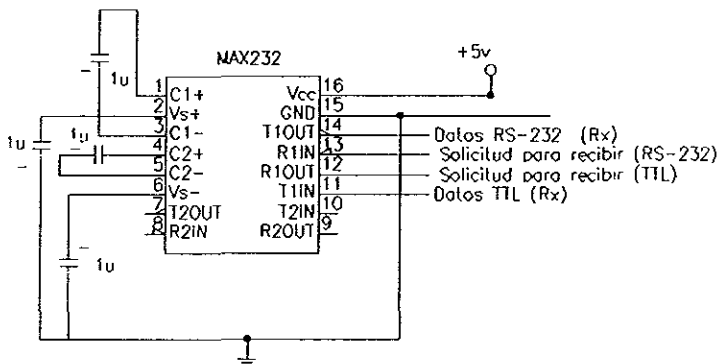


Figura 7.21.- Diagrama del Max 232.

7.5) PROGRAMACIÓN

En los puntos anteriores de este capítulo se describieron los elementos que conforman la parte física del control automático del acelerador (*hardware*), en esta sección se hace referencia a la parte de control pero enfocada a la programación (*software*). La descripción trata únicamente los puntos más relevantes para llevar a cabo la transmisión, adquisición y procesamiento de datos.

Como primer punto se encuentra como base de este proyecto el medio de comunicación. En el capítulo 5 se hizo referencia de algunos de los procesos por los cuales se puede establecer el intercambio de información (de la computadora al equipo de control y viceversa). El utilizar el comando bioscom implica un consumo de tiempo para la conversión a lenguaje maquina, debido a que dicho tiempo esta ligado directamente con la velocidad de reloj del procesador, se optó por utilizar la interrupción al sistema operativo ganando un menor tiempo de procesamiento para la transmisión y recepción de datos. A continuación se muestra y se describe parte del código necesario a dicha interrupción del sistema operativo.

```
{
v1.h.ah=X;
int86(xx,&v1,&v2);
}
```

Los valores válidos para X son 0, 1, 2 y 3 que son las funciones de la interrupción 14 (dirección para el control de los puertos de comunicación serial) utilizadas respectivamente para inicializar, enviar, recibir y verificar el estado del puerto. En el argumento de la interrupción, el valor xx se lee en hexadecimal, por lo que se puede expresar como **0x14 en hexadecimal** ó **20 en decimal** ($20_{10} = 14_{16}$). La variable v1 se utiliza para establecer los parámetros de salida y v2 para los de entrada. Para la inicialización se especifican los valores correspondientes a la función en v1.h.ah = 0, puerto v1.x.dx (0 para el puerto serie 1, 1 para el 2 y así sucesivamente) finalmente para las características de la señal v1.h.al (velocidad, paridad, bit de paro y número de datos), se usa la tabla 6.6 del capítulo anterior; en el envío o transmisión de datos se utiliza v1.h.ah = 1 y en v1.h.al es el caracter a enviar (valor); para la recepción de datos v1.h.ah = 2 y se utiliza v2.h.al para la recepción del caracter o dato; por último el estado del puerto (características) se envía v1.h.ah = 3 y se recibe la información en v2.h.ah y v2.h.al (ver figura 6.11). A continuación se describe un ejemplo más específico.

Ejemplo para transmitir el caracter g por el puerto serial 1, a una velocidad de 9600 bauds, con paridad impar, un bit de paro y 7 de datos la instrucción queda:

<pre>v1.x.dx = 0; v1.h.ah = 0; v1.h.al = 11101010 int86(20,&v1,&v2);</pre>	}	Inicialización del puerto
<pre>v1.h.ah = 1; v1.h.al = g; int86(20,&v1,&v2);</pre>	}	Transmisión de dato

De manera similar se efectúa la adquisición de datos; una vez que se ha definido o inicializado el puerto no es necesario volver a definirlo a menos que se desee cambiar o utilizar otro puerto y/o las características de esta señal.

El caracter enviado se prepara internamente con la interrupción de inicialización, esto es, agregar los bits de inicio, datos y paridad al caracter (palabra o byte), este caracter se convierte en

su equivalente en código ASCII y este a su vez a código binario (unos y ceros) siendo este último el utilizado para la transmisión; en el caso de la recepción se realiza el proceso inverso (eliminar el bit de inicio, paridad y paro) pero en lugar de ser convertido, a su representación conforme al código ASCII, se procesa bit a bit.

El proceso de comunicación está desarrollado para trabajar con 16 bits, de los cuales para la transmisión se dedican 12 bits para la señal analógica (conversión digital-analógica) y 4 para el direccionamiento y activación de dispositivos; en la recepción se manejan 13 para la señal analógica (conversión analógica digital) y 3 para direcciones de retorno. El dato enviado se trabaja de forma binaria, es decir para asignar el voltaje de salida se tiene que realizar una conversión del valor deseado a código binario pero representado de forma decimal.

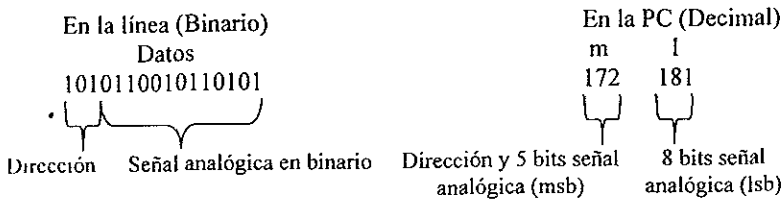
En lugar de escribir varias veces el mismo código para el envío y recepción, se estableció en un procedimiento general para cada función, en donde solo se realiza su llamado asignando los valores de entrada deseados y dirigiendo la salida a unas variables globales; esto además de ahorrar tiempo en programación reduce el tamaño del programa, como se presenta a continuación:

```
void enviar(int m,int l)
{
    v1.h.al=m;
    int86(20,&v1,&v2);
    v1.h.al=l;
    int86(20,&v1,&v2);
}
```

Se asigna el valor m a ser enviado en la primera palabra y se ejecuta la interrupción para el envío
Se asigna la segunda palabra (l) y también se envía.

De la misma forma se crea el proceso de recepción, primero se realiza la comunicación y después se asigna la variable de retorno a una variable distinta.

En la captura se realiza el proceso de conversión a binario, ya que la respuesta está expresada en decimal, por ejemplo si se recibe la información siguiente:



Para la información almacenada en m se deben seleccionar los datos del convertidor analógico-digital y las direcciones de retorno, a este último se le aplica un proceso de decodificación para obtener el dispositivo del cual se recibe la respuesta. El método contrario se utiliza para sumar la información para ser enviada, es decir, se transmiten los datos para el DAC y para el direccionamiento en el mismo conjunto de datos.

En el apéndice C se muestra el listado completo del programa encargado del control.

7.6) EVALUACIONES Y CONCLUSIONES

La mayoría de las etapas desarrolladas, específicamente todas aquellas que tuvieron que incluirse en la consola del acelerador, dentro del contexto de acercar el sistema al medio de operación y control, fueron realmente probadas desde su origen y hasta la fecha; es decir, cada vez que se terminaba de cubrir una etapa de todo este proyecto, ésta era inicialmente evaluada en el laboratorio de electrónica del mismo Instituto e inmediatamente instalada en el acelerador; y se continuaba con la operación del mismo en una forma que puede llamarse semiautomática o semimanual; debido a que dicho conjunto siempre ha tenido y tiene carga de trabajo por parte de los investigadores del Instituto al que pertenece. No hay mejor evaluación al respecto que el reportar que el funcionamiento de dicho acelerador se ha mantenido bajo una operación y control cuya evolución fue notoria durante el desarrollo de este trabajo.

Ya en la parte automatizada, es de mencionarse que el proceso de funcionamiento está dividido en dos módulos de operación: el primero dedicado a únicamente al manejo del acelerador; y el segundo, además del manejo tiene opciones para el mantenimiento y ajuste del sistema. La parte común a ambos modos se basa en el envío de información que se ingresa desde la computadora, esto se realiza proporcionando los datos básicos como son el ajuste del tiempo de exposición y de la cantidad de carga que recibirá la muestra; el resto se realiza secuencialmente por la computadora, desde la activación del motor de la banda y el chequeo de las señales de confirmación, hasta las señales de emergencia que apagan el sistema.

Las opciones de ajuste del sistema se centran básicamente en la recalibración de algunos elementos utilizando valores establecidos o de referencia, otras características son para el chequeo y revisión de la transmisión y recepción.

En ambos casos se presentan al usuario las respuestas obtenidas; por ejemplo se despliega la fecha y hora de inicio de la operación y la hora en el que se terminará; también se presentan guías o mensajes sobre lo que se está haciendo durante el proceso de arranque, por ejemplo en la operación manual se enciende un indicador luminoso "radiación lista", de la misma forma se presenta este mensaje en el monitor de la computadora, esto es para que se pueda tener una idea de lo que se está realizando en ese momento ya que los medidores que se encuentran en la consola solo se verá que se mueven indicando los valores correctos también a criterio de su operador humano, quien en todo momento interactúa y vigila al nuevo conjunto en su totalidad.

Puede pensarse que todo lo desarrollado en el módulo controlador, que permite la interacción del acelerador con la computadora, pudo ser sustituido por una tarjeta de adquisición y manejo de datos, como las que empezaron aparecer en el mercado hace casi una década; entre ellas, las conocidas "*LabVIEW*" y "*LabWindows*", que tuvieron gran auge hace aproximadamente cinco años. De hecho, eso era y sigue siendo posible si solo se hubiera deseado la automatización de dicho acelerador por él mismo; con lo que este trabajo se hubiera reducido sólo a preparar dicho acelerador para su acoplamiento a tales tarjetas, en las que el conocimiento se hubiera limitado a ser usuario de las mismas y en consecuencia continuar dependiendo de tecnologías extranjeras. El enfoque de este trabajo no se dedicó exclusivamente a satisfacer las necesidades de modernización de dicho acelerador, sino que también, en gran parte, se ocupó de conocer a fondo los procesos involucrados en la adquisición, manejo y retorno de datos, aplicados al control

de sistemas; contribuyendo así, aunque sea mínimamente, al conocimiento y desarrollo de la tecnología nacional.

No se ha tratado en ningún momento de competir con tecnologías extranjeras, pero sí de demostrar que en nuestro país y particularmente en nuestra Universidad y Facultad de Ingeniería, existe la cultura y la capacidad de desarrollo para resolver problemas reales.

REFERENCIAS DEL CAPITULO 7

1. Cehmichen, J.P. Como Deben Emplearse los Circuitos Integrados. Madrid 1978.
2. Leblanc, G. Turbo C para IBM-PC y compatibles. Gustavo Gili, S.A.. España 1988.
3. Jones D., Larry, Chin Foster, A. Electronic Instruments and Measurements. Prentice Hall. New Jersey 1991.
4. Morris L., Robert, Miller R., John. Diseño con Circuitos Integrados TTL. CECSA. México 1990.
5. Schildt, Herbert. Lenguaje C programación avanzada. McGraw-Hill. México 1989.
6. Schreiber, H. Aplicaciones en Computación, Lógica y Regulación. Paraninfo. España 1988.
7. Williams B., Arthur. Manual de Circuitos Integrados Volumen I a IV, selección diseño y aplicación. McGraw-Hill. México 1992.
8. Manual: Analog/Interface ICs. Motorola. USA 1996
9. Manual: Comlinear High-Speed Analog and Mixed-Signal Databook. National Semiconductor. USA 1997.
10. Manual: Fast and LS TTL Data. Motorola. USA 1992
11. Manual: National Interface Databook. National Semiconductor. USA 1996.
12. Manual: National Data Acquisition Databook, National Semiconductor. USA 1995
13. Manual: National Operational Amplifiers Databook. National Semiconductor. USA 1995
14. Manual: Optoelectronics. Motorola. USA 1995.

APÉNDICE A

REGISTROS E INTERRUPTACIONES EN EL 8086

En algunos casos y especialmente en los procedimientos de interrupción, es necesario escribir las funciones en algún lenguaje de bajo nivel como el ensamblador; esto ocasiona algunos problemas tales como la denominación y las características de los segmentos de código y de datos. En el lenguaje C lo anterior se simplifica mucho, ya que se puede acceder directamente y en cualquier momento a los registros de la máquina (*hardware*) en el microprocesador, ya sea para lectura o escritura; para ello basta hacer referencia a las *pseudovariantes*. Cuando C las reconoce no les asigna memoria. Las *pseudovariantes* son las que se mencionan a continuación.

REGISTROS DEL 8086

Entre los registros que trabaja el 8086, como son el AX, BX, CX y el DX, llamados registros generales y formados con 16 bits; cada uno de éstos, puede considerarse como compuestos por dos registros de 8 bits cada uno; es decir, que el registro AX se puede descomponer en AH y AL, donde AH contiene los bits más significativos del registro AX y en consecuencia AL los menos significativos. Cualquiera de estos registros, al igual que otros de únicamente 8 bits, se conoce como *pseudovariante*, según se muestra en la tabla A.1.

AX	AH	AL	acumulador	}	
BX	BH	BL	base	}	Registros Generales
CX	CH	CL	contador	}	
DX	DH	DL	dato	}	
				}	
		SP	puntero de la pila	}	Punteros e Índices
		BP	puntero de base	}	
		SI	índice de fuente	}	
		DI	índice de destino	}	
		CS	segmento de código	}	Registros de Segmentos
		DS	segmento de datos	}	
		ES	segmento "extra"	}	
		SS	segmento de la pila	}	

Tabla A.1.- Pseudovariantes utilizadas comúnmente.

El espacio que ocupa un programa puede descomponerse en tres partes: el código, los datos y la pila; que pueden estar físicamente separadas, configuradas u ocupar la misma zona de memoria. A estas partes de la memoria se le llama segmentos y su identificación se guarda en unos registros especiales (los registros de segmento CS y DS). La dirección

del principio de un segmento debe ser un múltiplo de 16, ya que debe coincidir con el principio de un párrafo.

UNIONES

La *union* es una declaración de datos de tipo distinto, que ocupan la misma posición de memoria. Según el momento de la ejecución de un programa, una zona de memoria puede considerarse en realidad como un entero, una cadena de dos caracteres, o un campo de 16 bits. La forma general de una *union* es:

```
union nombre_union {
    elemento 1
    elemento 2
    elemento n
} variable_union
```

donde:

union: es una palabra reservada
nombre_union: nombre dado a la union (describe una información, no una variable)
elemento 1..n: elementos de la memoria
variable_union: nombre dado a la variable

El sistema asigna el espacio de memoria que se requiera para contener a la variable de mayor tamaño. Los miembros de una *union* se accesan de la forma siguiente:

variable_union.nombre_miembro

donde:

nombre_miembro se refiere al elemento seleccionado en la lista declarada; por ejemplo:

```
union prueba{
    Int x;          /* variable entera x
    Char ch[2];    /* cadena de 2 caracteres
    } y;           /* variable de la union
```

Así cuando:

y.x = 263; /* es decir 263 = $\underbrace{00000001}_{0x01} \underbrace{00000111}_{0x07}$

Se puede decir que el valor de cada elemento sería: y.ch[0] = 7
y.ch[1] = 1

En la función *int86*, para los valores de entrada y de salida, se utiliza la instrucción *union REGS*, que sigue las mismas características que la *union* y se compone de la manera siguiente:

```

union REGS {
    struct WORDREGS x;
    struct BYTEREGS h;
};

Struct WORDREGS {
    Unsigned int ax, bx, cx, dx, si, di, cflag;
};

Struct BYTEREGS {
    Unsigned int ax, bx, cx, dx, si, di, cflag;
};

```

union REGS es una unión de dos estructuras. El uso de la estructura *WORDREGS* permite acceder a los registros del procesador (CPU) como registros de 16 bits (palabra completa), mientras que *BYTEREGS* permite un acceso a los registros individuales de 8 bits (un byte); por ejemplo, para acceder la interrupción 16 se debe seguir la secuencia siguiente:

```

union REGS in, out;
in.h.ah = 5
int86(0x16, &in, &out);

```

OTRAS INTERRUPTACIONES CON EL INT86

Otras aplicaciones a las que se puede tener acceso por medio de interrupciones son :

Interrupción	Función
10h	E/S Video
13h	E/S Disco
16h	E/S Teclado
17h	E/S Impresora
21h	llamadas al DOS

Por ejemplo, el manejo de la impresora en forma directa es por medio de la interrupción 17, donde las funciones disponibles (para el registro AH) son:

Función	Descripción
0	Imprime un caracter contenido en AL
1	Inicializa el puerto de la impresora
2	Obtiene el estado de la impresora

La respuesta a cualquiera de esas tres funciones se almacena en la variable AH (un byte), donde cada uno de los bits que la forman se interpreta como sigue:

- 0 Tiempo de espera terminado
- 1 No utilizado
- 2 No utilizado
- 3 0 hay error en la impresora
1 no hay error.
- 4 0 la impresora no está conectada
1 impresora conectada
- 5 0 la impresora tiene papel
1 no tiene papel (out of paper)
- 6 acuse de recibo
- 7 Si está en 0, la impresora está ocupada, hay que esperar antes de enviar datos;
cuando está en 1, la impresora está libre para recibir datos.

Por ejemplo, para solicitar el estado de una impresora conectada en el puerto paralelo 1 (LPT1), se requiere el código siguiente:

```
union REGS in, out;  
in.h.ah = 2;          /* número de la función  
in.x.dx = 0;          /* número de la impresora  
int86(0x17, &in, &out)
```





REFERENCIAS DEL APÉNDICE A

- 1) Carballar Falcón, José Antonio. Configuración, Actualización y Mantenimiento: Software y Hardware de su PC. Ra-ma, España 1994.
- 2) Gookin, Dan. Advanced MS-Dos Batch File Programming. Windcrest, EUA 1989.
- 3) Leblanc, G. Turbo C para IBM-PC y compatibles. Gustavo Gili, S.A.. España 1988.
- 4) Schildt, Herbert. Lenguaje C programación avanzada. McGraw-Hill. México 1989.
- 5) Manual: Turbo C, Reference Guide. Borland International. USA 1988.

APÉNDICE B

DISTRIBUCIÓN Y DIAGRAMAS DE LOS CIRCUITOS

Debido al número de circuitos involucrados y para evitar un circuito impreso demasiado grande, toda la instrumentación se encuentra dividida en 5 tabletas distribuidas por las funciones que desarrollan. Una sexta tableta contiene la fuente de alimentación general a las 5 anteriores; y proporciona los voltajes: ± 5 y ± 15 Volts con respecto a una tierra común. A continuación se presentan los diagramas finales, todos ellos en un tamaño estándar de 15 x 15 cm, acoplados mediante conectores "card-edge" de 15 o 22 terminales. En todas las tabletas, la alimentación de +5 Volts está estandarizada en las dos primeras terminales (A y B). El resto de las terminales de alimentación fueron distribuidas según la necesidad de cada circuito (N/C No tiene conexión)

-  Terminal de entrada X del conector "card-edge".
-  Terminal de salida x del conector "card-edge".
-  Terminal P## y dirección "R-###" (de la tira de postes).
-  Terminal ## del conector DB25.

Tableta 1: Se encarga de la recepción, conversión serie-paralelo, almacenamiento y conversión digital analógica (diagrama B.1a y B.1b).

Conversión de RS-232 a TTL (figura 7.1)

Señal de sincronización y reloj de recepción (figura 7.3 y 7.4)

Conversión Serie-Paralelo (figura 7.6)

Almacenamiento (figura 7.7 y 7.8)

TERMINAL	TERMINAL PARA	DESCRIPCIÓN
A	Alimentación	+ 5 Volts
B	Alimentación	Tierra (0 Volts)
C	-	N/C
D	Entrada	Transmisión de datos, Tx
E	-	N/C
F	Alimentación	Alimentación, + 5 Volts
H	-	N/C
J	Salida	Señal Analógica
K	Salida	Direccionamiento Bit 3
L	Salida	Direccionamiento Bit 2
M	Salida	Direccionamiento Bit 1
N	Salida	Direccionamiento Bit 0
P	Alimentación	+ 15 Volts
R	Alimentación	- 15 Volts
S	Salida	Direccionamiento carga

Tabla B.1.-Descripción de terminales para la tableta 1.

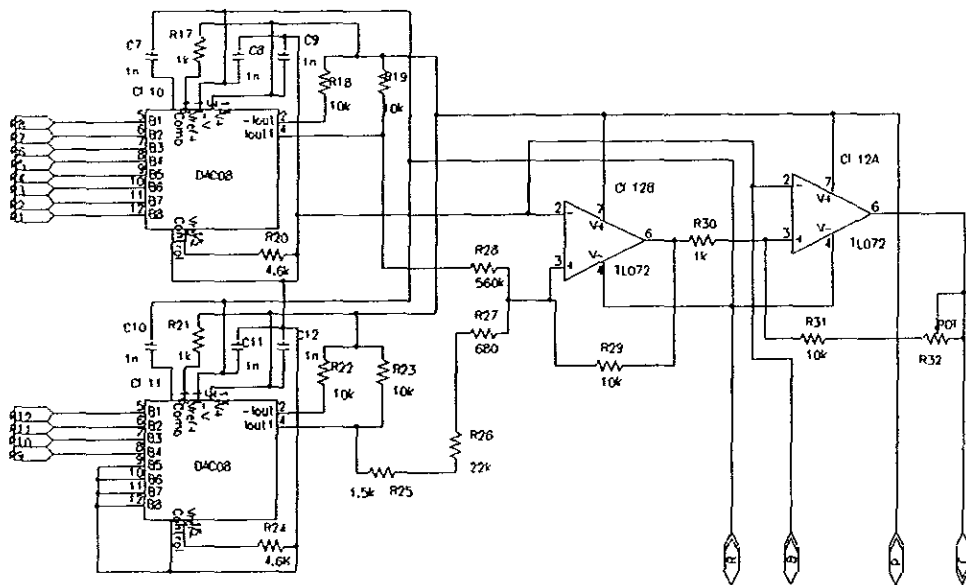


Diagrama B.1b

Tabla 2, etapa de direccionamiento la cual prepara la información para el control de dispositivos internos y externos, diagrama B.2.

TERMINAL	TERMINAL PARA	DESCRIPCIÓN
A	Alimentación	+ 5 Volts
B	Alimentación	Tierra (0 Volts)
C	Entrada	Direccionamiento Bit 3
D	Entrada	Direccionamiento Bit 2
E	Entrada	Direccionamiento Bit 1
F	Entrada	Direccionamiento Bit 0
H	Entrada	Direccionamiento carga
J	Salida	Dirección 13, WR
K	Salida	CS
L	Salida	Dirección 12, RD
M	-	N/C
N	-	N/C
P	-	N/C
R	-	N/C
S	Salida	Dirección 12, CAL

Tabla B. 2.-Descripción de terminales para la tabla 2.

Tableta 3, contiene los dispositivos encargados del control para dispositivos externos (diagrama B.3A), así como también para la recepción de señales analógicas (selección de entrada para el CAD) y la prioridad de señales de chequeo y respuesta (diagrama B.3B).

TERMINAL	TERMINAL PARA	DESCRIPCIÓN
A	Alimentación	Alimentación, + 5 Volts
B	Alimentación	Alimentación, Tierra (0 Volts)
C	-	N/C
D	-	N/C
E	-	N/C
F	-	N/C
H*	Salida	Señal analógica del DAC
J*	Entrada	Señal analógica para el CAD
K	-	N/C
L	-	N/C
M	-	N/C
N	-	N/C
P	-	N/C
R	-	N/C
S	-	N/C
T	Salida	Señal de retorno 2
U	Salida	Señal de retorno 1
V	Salida	Señal de retorno 0
W	-	N/C
X	-	N/C
Y	-	N/C
Z	Alimentación	Alimentación, Tierra (0 Volts)

Tabla B. 3.-Descripción de terminales para la tableta 3.

Esta tableta se encarga de acondicionar las señales de salida digitales (para los relevadores de la consola) y las señales de entrada digital y analógica; la señal de retorno proviene de la codificación de 3 bits (señal de prioridad de las señales digitales de entrada); el control de las salidas externas proviene de la tableta 2 (direccionamiento) y son transferidas a través de un cable listón.

Las salidas y entradas externas se encuentran disponibles por medio de un conector DB-25, en el que se distribuyen las terminales como sigue (ver tabla B.4):

- 1 y 14 son de alimentación externa (+ 5 Volts y tierra)
- 2 y 3 No se conectan
- De la 4 a la 12 son salidas digitales
- De la 15 a la 19 entrada analógica
- 20 salida analógica
- De la 21 a la 25 son entradas digitales

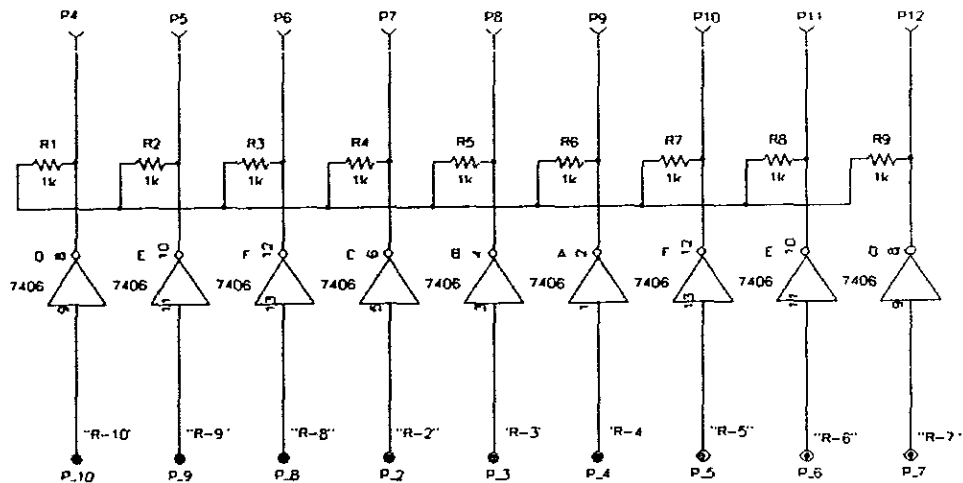


Diagrama B.3a

Terminal	Tipo de señal	Dirección	Termina	Tipo de señal	Dirección
1	Alimentación	+ 5 Volts	14	Alimentación	Tierra (0 Volts)
2	N/C	-	15	Entrada Analógica	SA4 (1 y 14)
3	N/C	-	16	Entrada Analógica	SA3 (14)
4	Salida Digital 10	R10	17	Entrada Analógica	SA2 (1)
5	Salida Digital 9	R9	18	Entrada Analógica	SA1 (default)
6	Salida Digital 8	R8	19	Salida Analógica	-
7	Salida Digital 2	R2	20	Entrada Digital 0	0
8	Salida Digital 3	R3	21	Entrada Digital 1	1
9	Salida Digital 4	R4	22	Entrada Digital 2	2
10	Salida Digital 7	R7	23	Entrada Digital 3	3
11	Salida Digital 6	R6	24	Entrada Digital 4	4
12	Salida Digital 5	R5	25	Entrada Digital 5	5
13	Entrada Digital 6	6			

Tabla B.4.-Terminales para el DB-25 utilizado en la tableta 3.

Tableta 4, realiza la conversión analógica-digital (diagrama B.4).

TERMINAL	TERMINAL PARA	DESCRIPCIÓN
A	Alimentación	+ 5 Volts
B	Alimentación	Tierra (0 Volts)
C	Entrada	Dirección 12, RD
D	Entrada	CS
E	Entrada	Dirección 12, CAL
F	Entrada	Dirección 13, WR
H	Alimentación	- 5 Volts
J	Entrada	Señal Analógica a digitalizar
K	Alimentación	Alimentación, Tierra (0 Volts)
L	Salida	Convertidor A/D Bit 0
M	Salida	Convertidor A/D Bit 1
N	Salida	Convertidor A/D Bit 2
P	Salida	Convertidor A/D Bit 3
R	Salida	Convertidor A/D Bit 4
S	Salida	Convertidor A/D Bit 5
T	Salida	Convertidor A/D Bit 6
U	Salida	Convertidor A/D Bit 7
V	Salida	Convertidor A/D Bit 8
W	Salida	Convertidor A/D Bit 9
X	Salida	Convertidor A/D Bit 10
Y	Salida	Convertidor A/D Bit 11
Z	Salida	Convertidor A/D Bit 12 (signo)

Tabla B.5.-Descripción de terminales para la tableta 4.

Tableta 5, esta tarjeta se encarga de adquirir las señales digitales del CAD y de la tarjeta 3 para la señal de prioridad, así como su preparación para el envío (diagrama B.5).

Para terminar el proceso de comunicación es indispensable sincronizar a los dos equipos, esto se hace por medio de las señales que ingresan por las terminales X y Y; dichas terminales funcionan de la siguiente manera: 1) X (permiso para transmitir) se encuentra en nivel alto (RS-232) durante la transmisión de datos (Tx) y en nivel bajo durante la recepción (Rx), esta señal es enviada del sistema a la computadora; 2) Y (solicitud para recibir) se encuentra en nivel alto en la transmisión y en bajo para la recepción, esta señal proviene de la computadora hacia el sistema. Para ambas señales se está trabajando con niveles RS-232.

La interacción con estas dos señales hace que se efectúe el ciclo de comunicación como sigue: en la transmisión se requiere de la señal X en alto para que la computadora tenga el

permiso para transmitir, es decir que el sistema está listo para recibir información, pero debido a que el sistema siempre está listo esta señal no cambia, Y se encuentra en nivel alto; en la recepción el sistema se activa con la señal Y en bajo, esto se interpreta como que la computadora esta lista o preparada para la recepción de datos; en caso de que ésta cambie a nivel alto se detendrá el envío de información del sistema.

Almacenamiento-envío

Conversión paralelo-serie

Control para envío y conversión de señales (figura 7.17 y 7.18)

TERMINAL	Señal de	DESCRIPCIÓN
A	Alimentación	Alimentación, + 5 Volts
B	Alimentación	Alimentación, Tierra (0 Volts)
C	Entrada	Convertidor A/D Bit 0
D	Entrada	Convertidor A/D Bit 1
E	Entrada	Convertidor A/D Bit 2
F	Entrada	Convertidor A/D Bit 3
H	Entrada	Convertidor A/D Bit 4
J	Entrada	Convertidor A/D Bit 5
K	Entrada	Convertidor A/D Bit 6
L	Entrada	Convertidor A/D Bit 7
M	Entrada	Convertidor A/D Bit 8
N	Entrada	Convertidor A/D Bit 9
P	Entrada	Convertidor A/D Bit 10
R	Entrada	Convertidor A/D Bit 11
S	Entrada	Convertidor A/D Bit 12 (signo)
T	Entrada	Señal de retorno bit 0
U	Entrada	Señal de retorno Bit 1
V	Entrada	Señal de retorno Bit 2
W	-	N/C
X	Salida	Permiso para Transmitir
Y	Entrada	Solicitud para Recibir
Z	Salida	Recepción de Datos, Rx

Tabla B.6.-Descripción de terminales para la tableta 5.

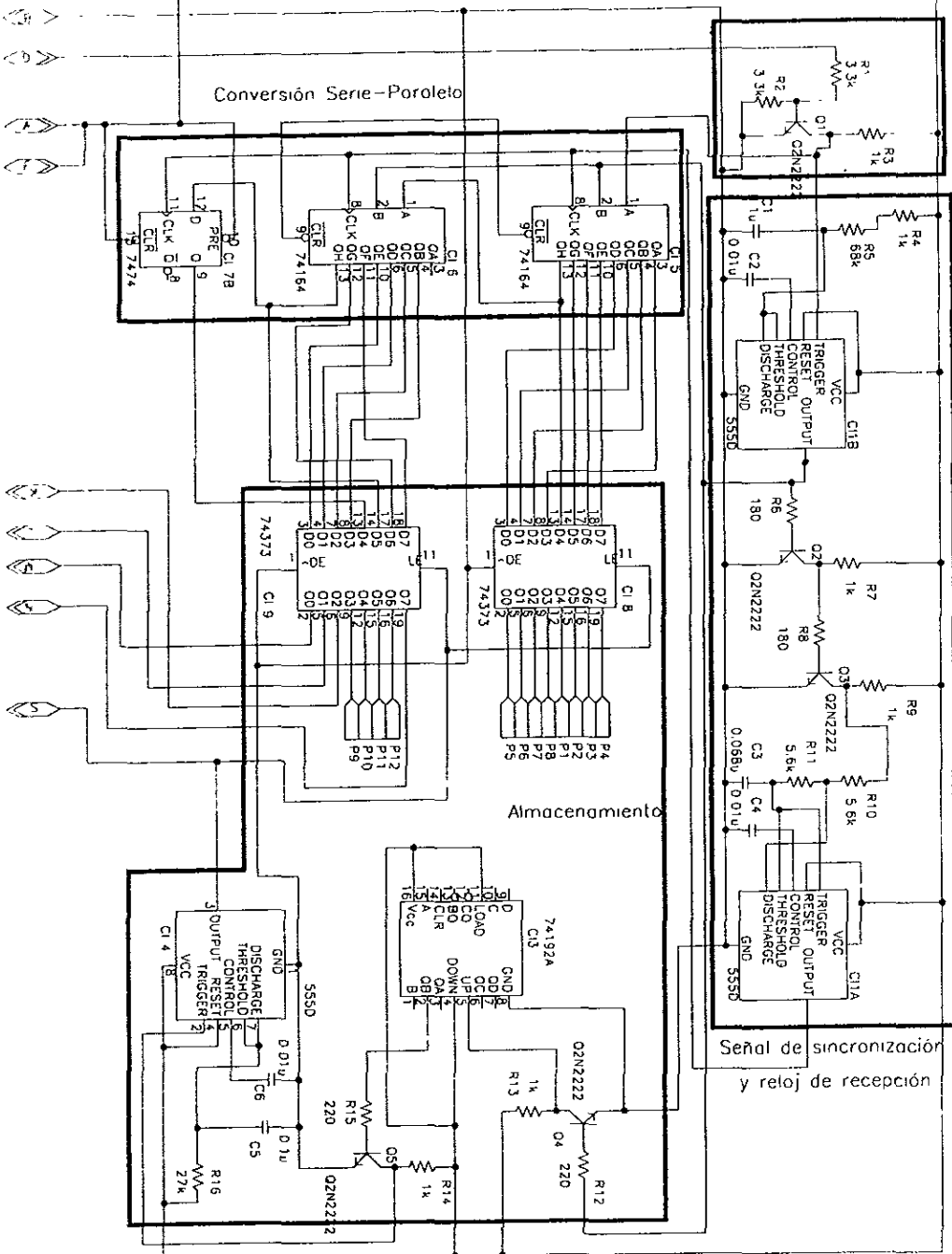


Diagrama B1.a - Captura y conversión de datos Serie-Paralelo

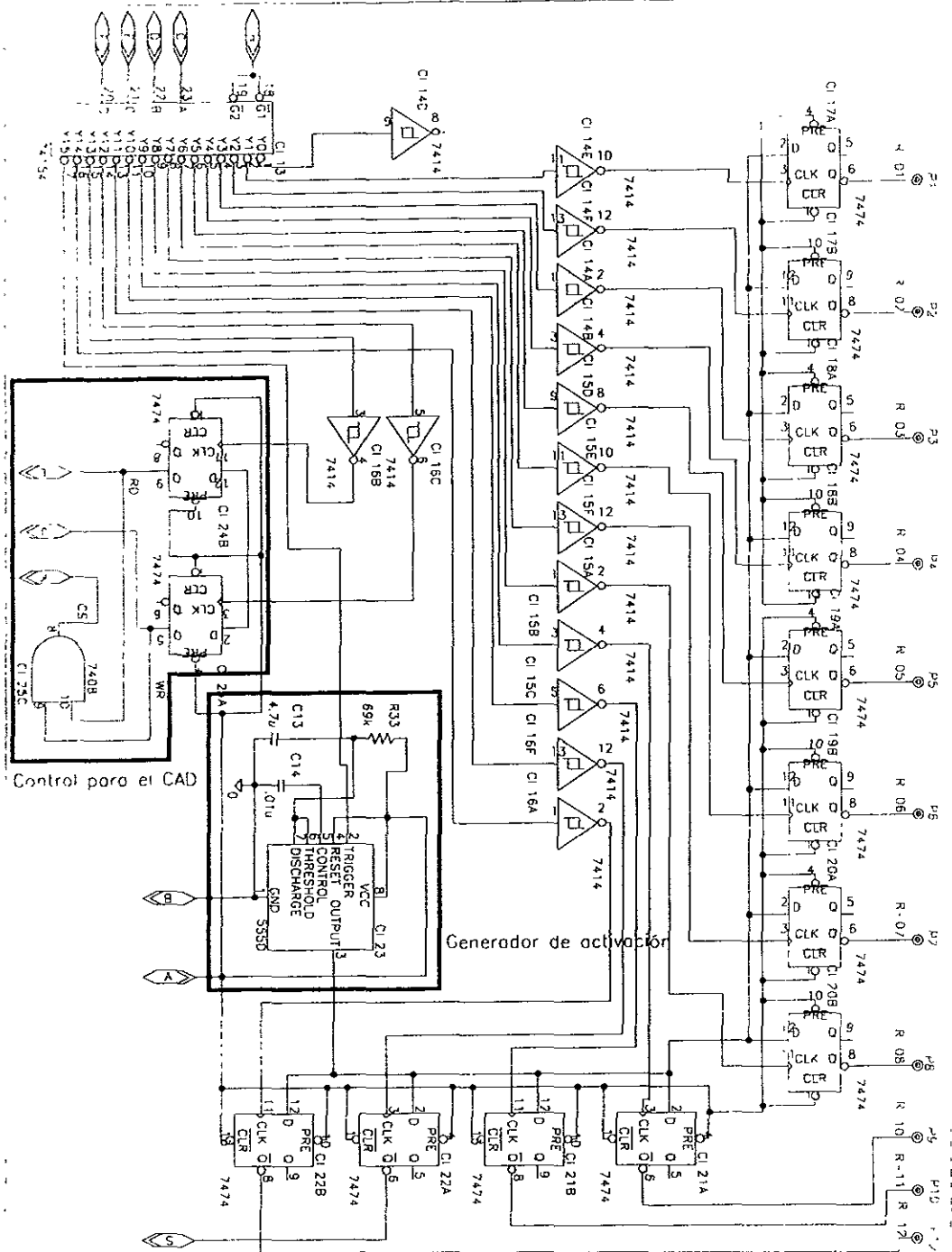
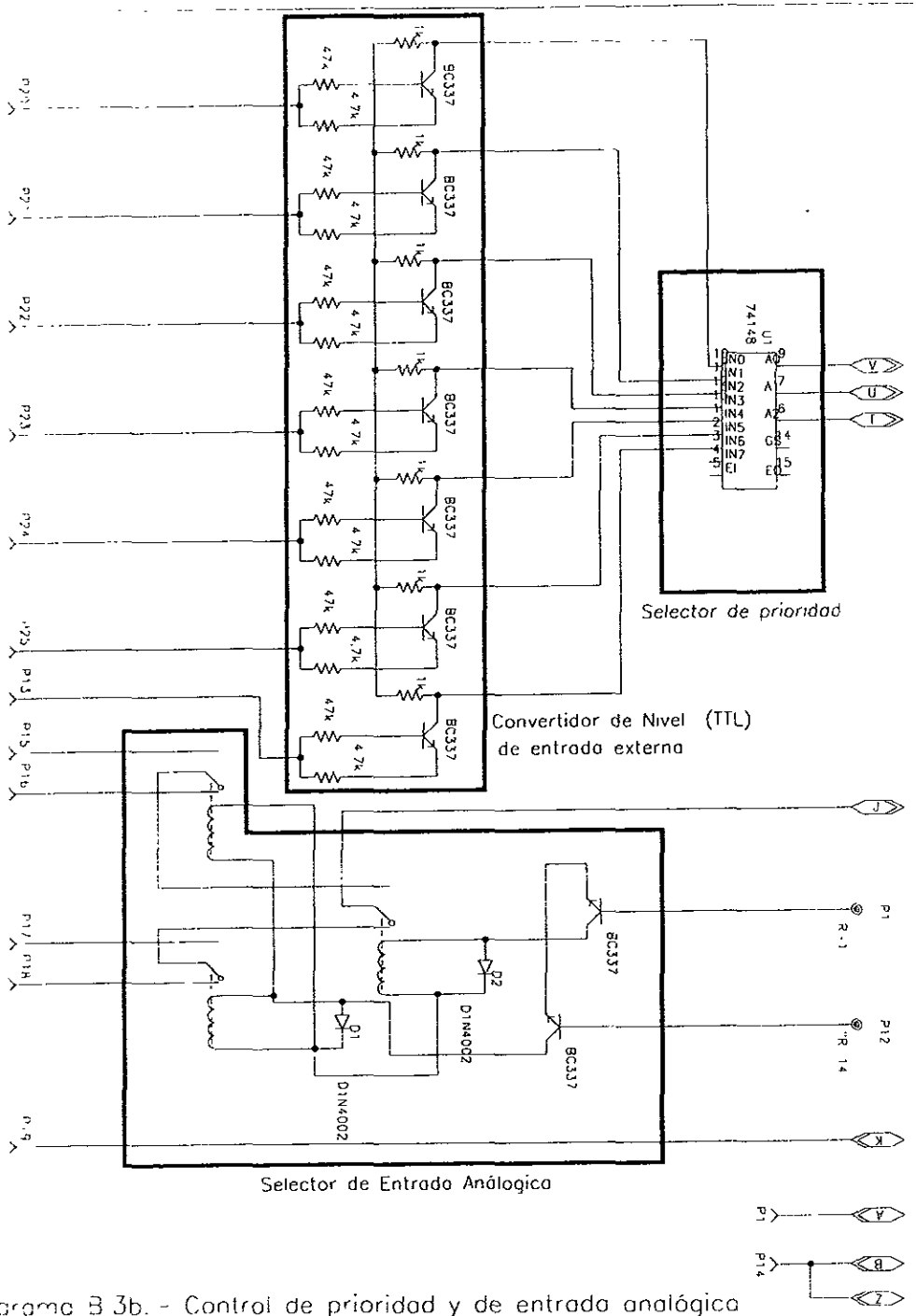


Diagrama B.2- Control de direccionamiento



Programa B 3b. - Control de prioridad y de entrada analógica

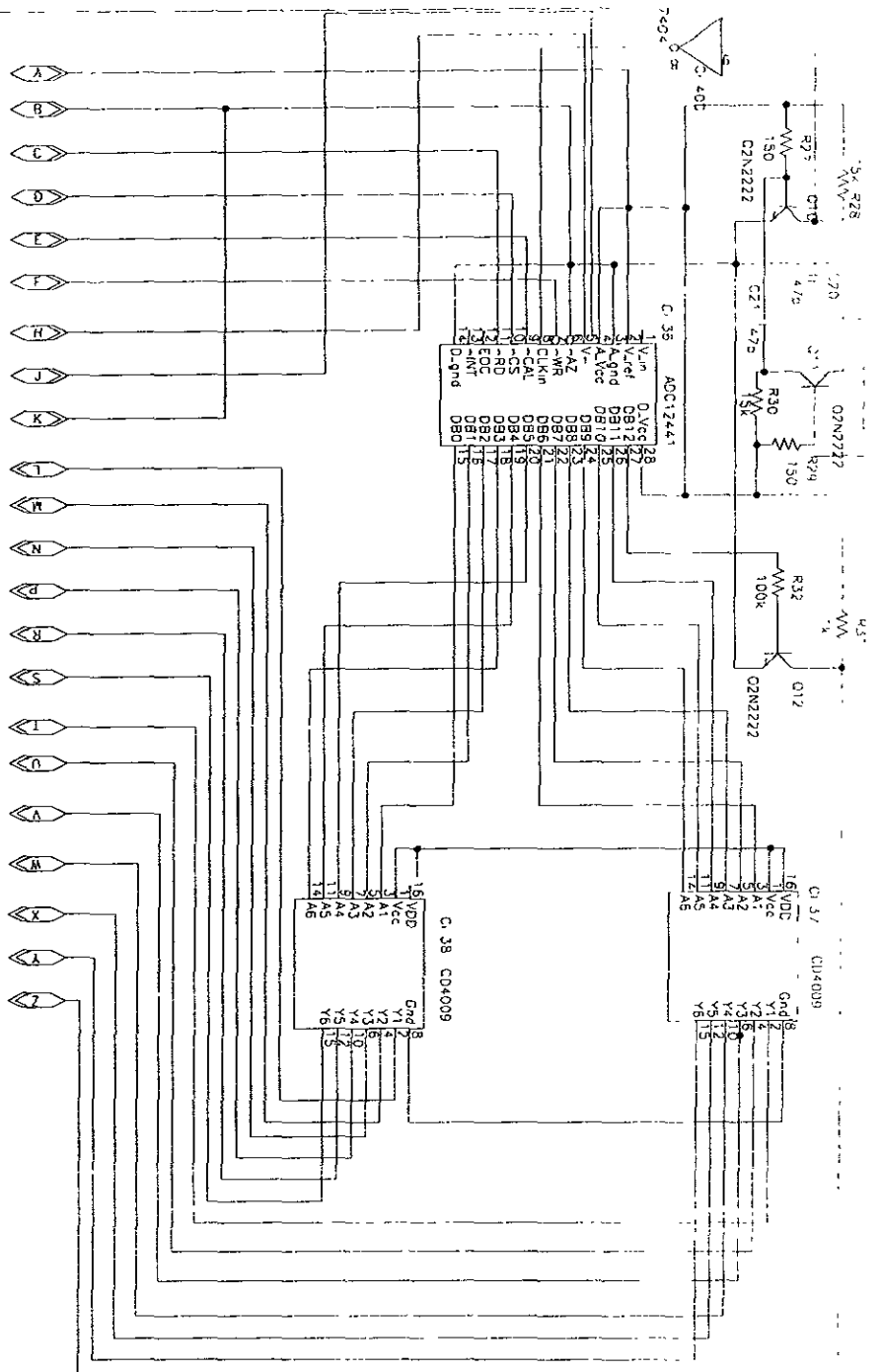
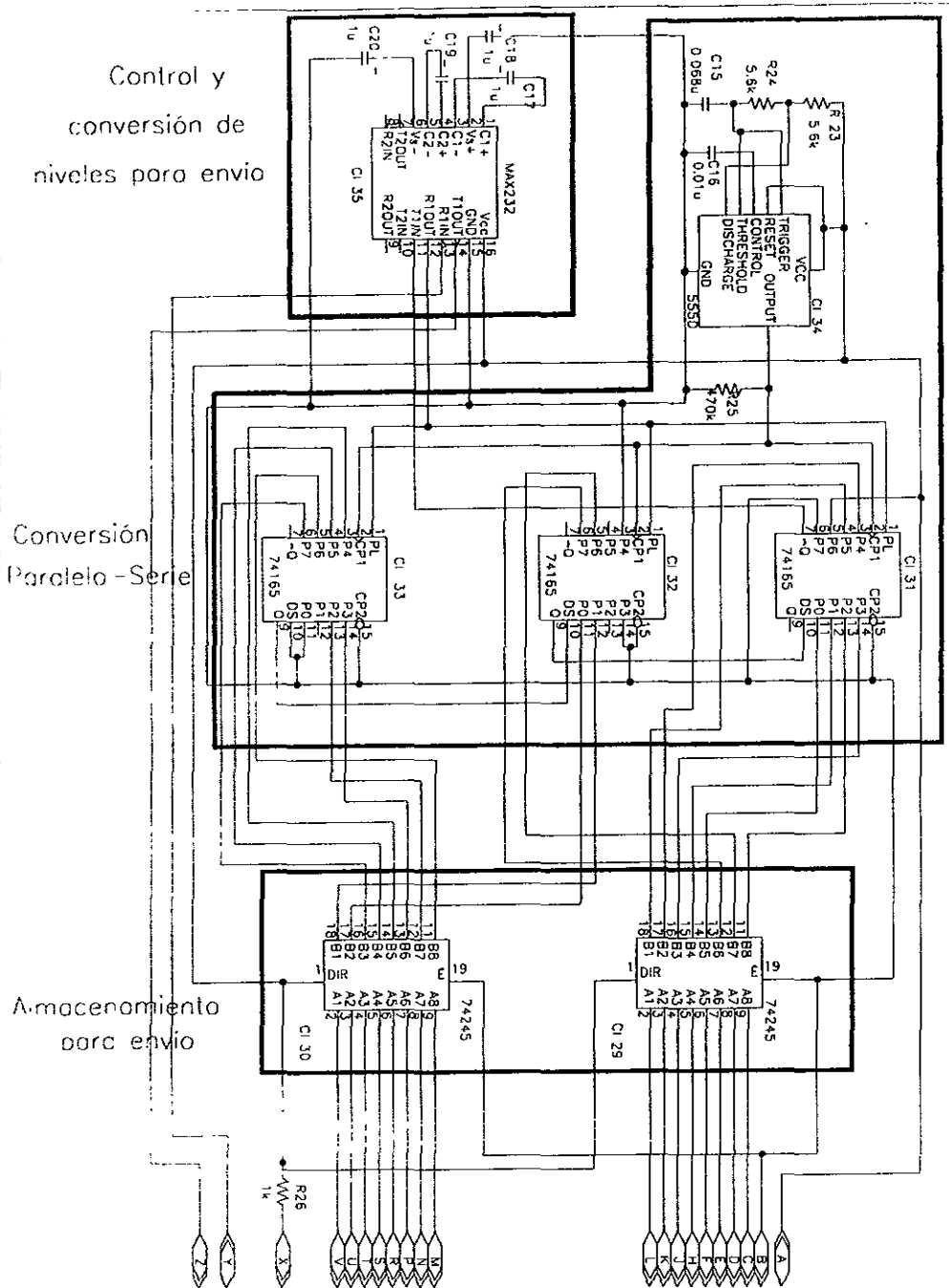


Diagrama B.4. - Conversion Analógico Digital de 12 bits



Control y
conversión de
niveles para envío

Conversión
Paralelo -Serie

Almacenamiento
paralelo envío

Diagrama B 5.- Captura y conversión de datos Paralelo-Serie

APÉNDICE C

LISTADO DEL PROGRAMA

A continuación se presenta el programa de control, el cual esta dividido en secciones dependiendo de su uso.

LIBRERÍA AUXILIAR 1

```
#ifndef _rs232_h_
#define _rs232_h_
extern msb,lsb,sa;
extern float vy;
void rs232_lect();
void rs232_envi();
void rs232_stat();
void rs232_inic();
void enviar(int,int);
inicial();
timer();
reset();
menu1(float,int);
dato();
salida(int,int);
voltaje();
lectura();
#endif
```

LIBRERÍA AUXILIAR 2

```
#if !defined _UTILES_H_
#define _UTILES_H_
#include <conio.h>
#include <dos.h>
enum TeclasFuncion
{F1=187,F2,F3,F4,F5,F6,F7,F8,F9,F10};
enum TeclasDesplaz
{FLEARRIBA=200, FLEIZQA=203, FLEDCHA=205, FLEABAJO=208};
enum AliasTeclas
{ESC=27};
enum DireccionesReles
{R1=16, R2=32, R3=48, R4=64, R5=80, R6=96, R7=112, R8=128, R9=144, R10=160,
R11=176, R12=192, R13=208, R14=224, R15=240};
inline int getkey()
{ int c;
  if ((c=getch())==0) c=getch()+128;
  return c;
}
```

APAGADO

```
#endif#include "rs232.h"
#include "utiles.h"
#include <dos.h>
#include <iostream.h>
#include <conio.h>
#include <dos.h>
// ***** SECUENCIA DE APAGADO NORMAL *****
char qq2;
//bajar steping R8
apagado() {
    sound(190);
    delay(1000);
    nosound();
    clrscr();
    cout<<"bajando Stepping\n";
    enviar((msb+R8),lsb);
    delay(200);
//STEPING RAPIDO R9
    enviar((msb+R15),lsb);
    delay(200);
    enviar((msb+R9),lsb);
    delay(200);
//parar stepping R9
    enviar((msb+144),lsb);
    delay(200);
    cout<<"steping fuera\n Apagando radiacion...";
//bajar elipot R7 R6
    enviar((msb+R7),lsb);
    delay(200);
    enviar((msb+R6),lsb);
    delay(200);
//chechar radiacion encendida
a1:
    lecturacontrol();
    if (cont3==3) cout<<"Radiacion Apagada \n Bajando variac...";
    else goto a1;
//bajar variac R5
    enviar((msb+R4),lsb);//64
    delay(200);
    enviar((msb+R5),lsb);//80
    delay(200);
//chechar radiacion lista
```

```

a2:
lecturacontrol();
if (cont3==2) cout<<"Radiacion lista \n Presionando 1H01";
else goto a2;
//apagar radiacion lista (1H01) R11
enviar((msb+R15),lsb);
delay(200);
enviar((msb+R11),lsb);
delay(200);
//chechar radiacion lista
a3:
lecturacontrol();
if (cont3==1) cout<<"... Radiacion Fuera\n Apagando motor \n";
else goto a3;
//apagar motor R3
enviar((msb+R3),lsb);
delay(200);
cout<<"Cerrar llave para terminar\n";
//chechar llave
a4:
lecturacontrol();
if (cont3==0) cout<<"Contacto abierto \n\n Fin de ciclo de trabajo inicio de espera 5 min";
else goto a4;
sleep(300);
sound(460);
delay(1000);
nosound();
return 0; }

```

ENCENDIDO

```

#include "rs232.h"
#include "utiles.h"
#include <dos.h>
#include <iostream.h>
#include <conio.h>

// ***** SECUENCIA DE ENCENDIDO *****

int cont3;
char qq;
encendido()
{
clrscr();
//leer puerto
cout<<"Llave cerrar\n";
el.
lecturacontrol();

```

```

    if (cont3==1) cout<<"Contacto cerrado \n";
    else goto e1;
//arrancar motor R3
    enviar((msb+R15),lsb);//activa mono
    delay(200);
    enviar((msb+R3),lsb);
    delay(200);
    cout<<"Motor encendido \n Esperando Radiacion Lista..";

//Esperar radiacion lista
e2:
    lecturacontrol();
    if (cont3==2) cout<<" Radiacion Lista\n";
    else goto e2;

//Apagar radiacion lista 1H02 R4
    enviar((msb+R15),lsb);
    delay(100);
    enviar((msb+R4),lsb);
    delay(100);
    cout<<"\n\n Boton 1H02 presionado\n";
//Subir variac R5
    enviar((msb+R15),lsb);
    delay(200);
    enviar((msb+R5),lsb);
    delay(200);
    cout<<"variacion encendido \n";
//Subir alto voltaje a 2.77 (helipot) R6
    enviar((msb+R15),lsb);
    delay(200);
    enviar((msb+R6),lsb);
    delay(200);
    cout<<"Iniciando alto voltaje. Esperando Radiacion... ";
//chechar radiacion encendida
e7:
    lecturacontrol();
    if (cont3==3) cout<<" Radiacion Encendida \n Radiacion A XX\n";
    else goto e7;
//Subir elipt a 3.23 R7
    enviar((msb+R15),lsb);
    delay(200);
    enviar((msb+R7),lsb);
    delay(200);

    cout<<"Subiendo stepping...";
//stepping subir R8
    enviar((msb+R15),lsb);

```

```

        delay(200);
        enviar((msb+R8),lsb);
        delay(200);

//Stepping rapido R9
        enviar((msb+R15),lsb);
        delay(200);
        enviar((msb+R9),lsb);
        delay(200);

//stepping lento R10
        enviar((msb+R9),lsb); //R9
        delay(200);
        enviar((msb+R15),lsb);
        delay(200);
        enviar((msb+R10),lsb);
        delay(200);

        cout<<"En posicion";

//parar stepping R10
        enviar((msb+R10),lsb);
        delay(200);

return 0; }

```

CALIBRACIÓN SALIDA DAC

```

#include <iostream.h>
#include "rs232.h"
int msb,lsb;
dato()
{int ad,lsbd,msbd;
    cout<<endl<<"DATO "; cin>>ad;
    msbd= (ad/256);
    lsbd=255-(ad-(msbd*256));
    msbd=15-msbd;

    msb=msbd;
    lsb=lsbd;
    // codmsb(msbd);
    // codlsb(lsbd);
    // return msbd,lsbd;
return 0;
}

codmsb(int bitm)

```



```

{
msb=bitm;
return 0;
}
codlsb(int bitl)
{

lsb=bitl;
return 0;
}

```

INICIALIZACIÓN DE SISTEMA

```

#include <conio.h>
#include <iostream.h>
#include "rs232.h"
#include <dos.h>
//***** Checado de RESET Correcto *****

int th3,tm3,ts3,tm4,th4,ts4;

inicial()
{
struct time t;
gettime(&t);
th3= t.ti_hour;tm3= t.ti_min;ts3= t.ti_sec;
cout<<"Hora inicio: "<<th3<<":"<<tm3<<":"<<ts3;
th4=th3;tm4=tm3;ts4=ts3+3;
if (ts4>=60){ts4=ts4-60;tm4=tm4+1;}
if (tm4>=60){tm4=tm4-60;th4=th4+1;}

reset();

gettime(&t);
th3= t.ti_hour;tm3= t.ti_min;ts3= t.ti_sec;
if((th4==th3)&(tm4==tm3)&(ts4<=ts3)) {clrscr(); cout<<"Sistema Inicializado,
Preparando Entorno";sleep(2);};

return 0;}

```

TIEMPO DE EXPOSICIÓN

```

#include <ctype.h>
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>

```

```

#include "rs232.h"
#include "utiles.h"

int th,th2,hh,tm,tm2,ts,ts2,a,ss,ss2,s,run,run2;

timer()
{
th2=0;tm2=0;ts2=0;run=0;run2=0;
clrscr();
cout<<"\n AJUSTAR TIEMPO: (Flecha arriba = + Flecha abajo = - \n flecha der= siguiente
flecha izq= regresar";
cout<<"\n Horas: 0 Minutos: 0 Segundos: 0";gotoxy(10,4);
rh:
{switch(getkey()){
case FLEARRIBA: th2++;break;
case FLEABAJO: th2--;break;
case FLEDCHA: s=1; break;
}
if (th2<=0) th2=0;if (th2>=12) th2=12;gotoxy(10,4);if (th2<10) cout<<" ";
cout<<th2;if(s<1) goto rh;}

s=0;gotoxy(26,4);
rm:
{switch(getkey()){
case FLEARRIBA: tm2++;break;
case FLEABAJO: tm2--;break;
case FLEDCHA: s=1;break;
case FLEIZQA: gotoxy(10,4);goto rh;
}
if (tm2<=0) tm2=0;if (tm2>=59) tm2=59;gotoxy(26,4);if (tm2<10) cout<<" ";
cout<<tm2;if(s<1) goto rm;}

s=0;gotoxy(44,4);
rs:
{switch(getkey()){
case FLEARRIBA: ts2++;break;
case FLEABAJO: ts2--;break;
case FLEDCHA: s=1;break;
case FLEIZQA: gotoxy(26,4);goto rm;
}
if (ts2<=0) ts2=0;if (ts2>=59) ts2=59;gotoxy(44,4);if (ts2<10) cout<<" ";
cout<<ts2;if(s<1) goto rs;}

encendido();
ini:
//***** TEMPORIZADOR DE SISTEMA
struct time t;
clrscr();

```

```

gettime(&t);
    th= t.ti_hour;tm= t.ti_min;ts= t.ti_sec;ss2=t.ti_hund;
cout<<"Hora inicio: "<<th<<":"<<tm<<":"<<ts;

    hh=0;
th2=th2+th;tm2=tm2+tm;ts2=ts+ts2;
if (ts2>=60){ts2=ts2-60;tm2=tm2+1;}
if (tm2>=60){tm2=tm2-60;th2=th2+1;}
    cout<<"\nHora fin: "<<th2<<":"<<tm2<<":"<<ts2<<":"<<ss;
        // ***** LOOP DE CONTROL

while(!kbhit() & (hh<=6)){
run2=run2+1;
if(erroses()==1) hh=10;
gettime(&t);
gotoxy(20,5);
th=t.ti_hour;tm=t.ti_min;ts=t.ti_sec;ss=t.ti_hund;
cout<<th<<":"<<tm<<":"<<ts<<":"<<ss;
if((th2<=th)&(tm2<=tm)&(ts2<=ts)&(ss2<=ss)) hh=10;

};
return 0;
}

```

PROGRAMA DE TRONCO COMÚN (MAIN)

```

#include <iostream.h>
#include <bios.h>
#include <conio.h>
#include "rs232.h"
#include "utiles.h"
//***** PROGRAMA BASE *****

int e;
char m1;

void main()
{
    clrscr();
    msb=15;lsb=255;
    gotoxy(25,2);cout<<"Inicializando sistema\n";

    inicial();

    for(e=0;e<=1;e--)
    { clrscr();
        menu1(vy,sa);
    }
}

```

```

ca:
switch(getkey())
{
case F1:clrscr();dato(); break;
case F2:salida(msb,lsb);break;
case F3:clrscr();voltaje();break;
//case F4:clrscr();timer();enviar((msb+sa*16),lsb);break;
case F5:e=6; break;
case F6:lectura();break;
case F7:daccap();break;
case F8:daclec();break;
case F9:dacres();break;
case F10:timer();apagado();break;
case '\n' : goto ca;
default : continue;
}
cout<<"\n\n";
}
;
clrscr();
cout<<"CERRANDO SISTEMA";reset();
}

```

MENÚ

```

#include <iostream.h>
#include <dos.h>
#include "rs232.h"

int mth,mtm,mts,mdy, mdd,mdm;
menu1(float my,int ma)
{
cout<<endl<<"Estado actual del sistema:";
cout<<" Voltaje: "<<my<<" Salida #: "<<ma<<endl;

struct time t;

gettime(&t);
mth= t.ti_hour;mtm= t.ti_min;mts= t.ti_sec;
cout<<"Hora: "<<mth<<":"<<mtm<<":"<<mts;

struct date d;

getdate(&d);
mdy= d.da_year;
mdd= d.da_day;
mdm= d.da_mon;

```

```
cout<<"
"<<mdy<<endl;
```

Instituto de Fisica a "<<mdd<<" del mes "<<mdm<<" de

```
cout<<"          ° Opciones:          °<<endl;
cout<<"          °1)  METER DATOS °<<endl;
cout<<"          °2)  ENVIAR DATOS °<<endl;
cout<<"          °3)  VOLTAJE          °<<endl;
cout<<"          °4)  tempori          °<<endl;
cout<<"          °5)  SALIR          °<<endl;
cout<<"          °6)  cAPTURA"<<endl;
cout<<"          °7)  Cap DAC          °<<endl;
cout<<"          °8)  Lect DAC         °<<endl;
cout<<"          °9)  Reset DAC        °<<endl;
cout<<"          °10) Sistema"<<endl;
cout<<"          °opcion:          °<<endl;
return 0;
}
```

DESPLIEGE DE ERROR

```
#include <iostream.h>
#include "rs232.h"

errores()
{lecturacontrol();
run=run+1;
switch(cont3){
case 1:cout<<" se trata de un error 1";break;//lave cerrada
case 2:cout<<" se trata de un error 2";break;//boton h10
case 3:cout<<" se trata de un error 3";break;//radioacion lista
case 4:cout<<" se trata de un error 4";break;
case 5:cout<<" se trata de un error 5";break;
case 6:cout<<" se trata de un error 6";break;
case 7:cout<<" se trata de un error 7";return 1;//falta proteccion
break;//reset de sistema
};
return 0;
}
```

CONTROL GENERAL PARA EL PUERTO ENTRADA/SALIDA

```
:#include <iostream.h>
#include <conio.h>
#include "rs232.h"
#include "utiles.h"
#include <dos.h>
#include <stdlib.h>
```

```

#define INIT 0 //funcion 0 de interrupcion 14
#define SEND 1 //funcion 1
#define READ 2 //funcion 2
#define STAT 3 //funcion 3
#include <time.h>

union REGS v1,v2;
int b,c,de;
float b1,b2,b3;
int sa,smsb2,pp;
char ap;

//*****TRANSMISIÓN DE 16 BITS CON 4 BITS DE CONTROL
salida(int smsb,int slsb)
{
  clrscr();

  cout<<"Presione el numero del dispositivo a "<<<endl;
  cout<<" enviar"<<endl<<endl;
  cout<<"1) M1      5) 9)   13)"<<endl;
  cout<<"2) M2      6) 10)  14)"<<endl;
  cout<<"3) P1      7) 11)  "<<<endl;
  cout<<"4) P2      8) 12)  "<<<endl;
  cout<<"   99) Cancelar"<<endl;
  vc:
  cout<<": ";
  cin>>sa;
  pp=sa*16;
  cout<<"Apagar o Prender: \n";
  cin>>ap;
  if (ap=='p') { //SELECCIÓN DE ENCENDER O APAGAR EL BIT
SELECCIONADO
    smsb2=smsb+(15*16);
    enviar(smsb2,slsb);//prende al monoestable
    delay(200);}
    smsb2=smsb+pp;
    enviar(smsb2,slsb);

return 0;
}

daccap() //señal de captura para el dac
{ dacr();
  enviar((msb+R15),lsb);
// delay(100);
  enviar((msb+R13),lsb);
  delay(200);
  delay(800);
}

```

```

    enviar((msb+R13),lsb);
    return 0; }

daclec() // lectura de la conversion
{
    enviar((msb+R15),lsb);
    delay(100);
    enviar((msb+R12),lsb);
    delay(200);
    return 0;}

dacres()
{
    enviar((msb+R12),lsb);
    delay(200);
    enviar((msb+R13),lsb);
    delay(200);
    return 0;
}

//***** PROCESO DE CAPTURA DE 16 BITS POR EL PUERTO SERIAL
lectura()
{
    div_t x;
    int y,q,tot1,tot2;
    int l[10];
    {ini:
        enviar(msb,lsb);
        rs232_lect();c=v2.h.al;
        rs232_lect();b=v2.h.al;
        enviar(msb,lsb);
        cout<<b<<" "<<<<endl;

//PROCESO DE ASIGNACIÓN LOS BITS MAS SIGNIFICATIVOS
for (y=0;y<=8;y++){
    if (b>=256) {cout<<"error";goto ini;}
    x = div(b,2);
    b=x.quot;
    m[y]=x.rem;
    } ;
for (y=0;y<=8;y++){ //asignacion lsb
    if (c>=256) {cout<<"error";goto ini;}
    x = div(c,2);
    c=x.quot;
    l[y]=x.rem;
    } ;
} ;

```

```

for (q=7;q>=0;q--){ //MUESTRA DE LA ENTRADA EN BINARIO DE 8 BITS
    cout<<m[q];
} ;cout<<" ";
for (q=7;q>=0;q--){ //MUESTRA DE LA 2 ENTRADA EN BINARIO DE 8 BITS
    cout<<l[q];
} ;

cout<<"\n";

//CONVIERTE A DECIMAL EN DOS BLOQUES DE 4 y 12 BITS

cout<<(m[7]*8+m[6]*4+m[5]*2+m[4]);
cout<<endl<<(totl=m[3]*2048+m[2]*1024+m[1]*512+m[0]*256)<<endl;
c=l[0]*1+l[1]*2+l[2]*4+l[3]*8+l[4]*16+l[5]*32+l[6]*64+l[7]*128;
b3=(totl+c);
if (m[4]==1) b3=- (4095-b3);
vy=b3*.001221;
cout<<"DIGITAL "<<b3<<" ANALOGICO "<<vy<<endl;
sleep(4);
}return 0;}

```

```

void rs232_inic()
{
v1.x.dx=1; //selecciona el COMM 1
v1.h.a1=0x93; //selecciona los bauds, paridad, bit stop y
//longitud de la palabra (BBBPPSLL)
v1.h.ah=INIT; //funcion 0 de int 14h a realizar
int86(0x14,&v1,&v2); //llamada a la funcion
}

```

```

void rs232_lect()
{
v1.h.ah=READ; //funcion 2 int 14h a realizar
int86(20,&v1,&v2);//llamada de la funcion
}

```

```

void rs232_stat()
{
v1.h.ah=STAT; //funcion 3 int 14h
int86(20,&v1,&v2);
}

```

```

void rs232_envi()
{
v1.h.ah=SEND; //funcion 1 int 14h
int86(20,&v1,&v2);
}

```



```

    }

int re,rmsb;
reset()
{   int tc; rs232_inic();
    tc=7; gotoxy(42,4);cout<<" ";
    for(re=15;re<=0;re--)    //ENVIA EL DATO PARA APAGAR CUALQUIER RELE
        {   rmsb=15+16*re;//QUE SE ENCUENTRE ACTIVADO EN EL MOMENTO
            delay(200);enviar((rmsb),255);
            gotoxy(27+w,4);cout<<"±";
            gotoxy(30,5);cout<<tc<<" %";tc=tc+6;
        }
}
return 0;
}

void enviar(int m,int l)
{
    v1.h.al=m;
    rs232_envi();
    v1.h.al=l;
    rs232_envi();
}

int m[10];
lecturacontrol() //***** VERIFICA LAS 4 ENTRADAS DE DATOS DE CONTROL
{
    div_t x;
    int b2,y,q;

    ini:    //chechar que ocurre con los dac 08 al poner esta config
            enviar(msb,lsb);
            rs232_lect();
            rs232_lect();b2=v2.h.al;
            enviar(msb,lsb);

    //PROCESO DE ASIGNACION LOS BITS MAS SIGNIFICATIVOS
    for (y=0;y<=8;y++){
        if (b2>=256) {cout<<"error";goto ini;}
        x = div(b2,2);
        b2=x.quot;
        m[y]=x.rem;}

    cont3=m[7]*4+m[6]*2+m[5]*1;

    return 0;
}

```

ESTABLECER NIVEL DE VOLTAJE

```

#include <iostream.h>
#include <math.h>
#include <conio.h>
#include "rs232.h"
#include "utiles.h"
int va,va2,vlsb,vmsb;
float vy;
voltaje()
{
int h,m,th,tm,s;
h=0;m=0;s=0;
clrscr();
cout<<"\n AJUSTAR voltaje   (Flecha arriba = + Flecha abajo = - \n flecha der= siguiente
flecha izq= regresar";
cout<<"\n Volts: 0   Milivolts: 0";gotoxy(10,4);
rh:
{switch(getkey()){
    case FLEARRIBA: h++;break;
    case FLEABAJO: h--;break;
    case FLEDCHA: s=1; break;
    }
    if (h<=0) h=0;if (h>=12) h=12;gotoxy(10,4);if (h<10) cout<<" ";
    cout<<h;if(s<1) goto rh;}

s=0,gotoxy(26,4);
rm:
{switch(getkey()){
    case FLEARRIBA: m++;break;
    case FLEABAJO: m--;break;
    case FLEDCHA: s=1;break;
    case FLEIZQA: gotoxy(10,4);goto rh;
    }
    if (m<=0) m=0;if (m>=99) m=99;gotoxy(26,4);if (m<10) cout<<" ";
    cout<<m;if(s<1) goto rm;}

    vy=h+m*.01;
    cout<<endl;
    vy=-vy;
    va=(vy-0.009)/0.001299145;va2=va;
    va=abs(va);
    vmsb= (va/256);    vlsb=255-(va-(vmsb*256));
    vmsb=15-vmsb;
    if (abs(vy)>=4.899) {vmsb=0; vlsb=0;}
    codmsb(vmsb);codlsb(vlsb);

return 0;}

```