



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistema electrónico de alarma contra
robos para los cubículos de la
División de Ciencias Básicas**

TESIS

Que para obtener el título de
Ingeniero Mecánico Electricista

P R E S E N T A

Teresa Rosas Mendoza

DIRECTOR DE TESIS

Yukihiro Minami Koyama



Ciudad Universitaria, Cd. Mx., 2004



CONTENIDO

	Página
1 INTRODUCCIÓN	1
2 FUNDAMENTOS DE LOS PUERTOS PARALELOS	5
2.1 Definición de puerto	6
2.2 Evolución de los puertos paralelos	7
2.3 Descripción del conector DB – 25 del puerto LPT1	9
2.4 Bus y tarjeta ISA	12
2.5 Ciclos de lectura y escritura de un puerto de I/O	18
3 SOLUCIONES PROPUESTAS	23
3.1 Procesamiento de los datos del sistema mediante un GAL	24
3.2 Procesamiento de los datos del sistema mediante un microcontrolador	25
3.2.1 PIC	26
3.2.2 Microcontrolador Motorola MC68HC11	27
3.3 Procesamiento de los datos del sistema mediante una PC	28
3.4 Comparación de las soluciones propuestas	28
3.4.1 Propuesta con GAL	28



3.4.2	Propuesta con un PIC	29
3.4.3	Propuesta con el microcontrolador MC668HC11	30
3.4.4	Propuesta con base en una PC 8086	30
3.5	Otras consideraciones	31
3.6	Toma de decisiones para el diseño del sistema	33
4	DISEÑO DEL SISTEMA	35
4.1	Diseño de un puerto paralelo adicional de 8 bits	35
4.2	Hardware	38
4.2.1	Descripción del sistema	38
4.2.2	Diseño y desarrollo de la primera propuesta	39
4.2.3	Diseño y desarrollo de la segunda propuesta	44
4.3	Software	60
4.4	Costos	63
5	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	67
5.1	Resultados	67
5.2	Conclusiones	69
5.3	Recomendaciones	70
	APÉNDICE	73
A.1	Biométrica	73
A.2	Robos en la Facultad de Ciencias	78
A.3	Diseño del bloque despliegue de leds	79
A.4	Software de programación de la GAL16V8	82



BIBLIOGRAFÍA	85
PÁGINAS DE INTERNET	86
PERIÓDICOS Y REVISTAS	87



CAPÍTULO 1

INTRODUCCIÓN

En los últimos años la seguridad se ha convertido en un factor fundamental de nuestra sociedad, debido a esto, los sistemas de alarma contra robo son indispensables para la protección de casas habitación, comercios y empresas.

En su origen los sistemas de alarma utilizaban elementos mecánicos en su diseño. Su central era controlada por un relé y los sensores que utilizaban eran de contacto directo como pares de chapas, que al ser separadas accionaban un mecanismo para notificar una interrupción en el sistema. Con el aprovechamiento del transistor se implementaron circuitos de temporización, los sistemas se ampliaron al uso de sensores magnéticos, y barras luminosas entre otros.

Asimismo, con el uso de los circuitos integrados en el diseño de alarmas, se redujeron los elementos mecánicos y el tamaño de los equipos, en tanto que a los sensores se les incorporó la técnica infrarroja. Actualmente en el diseño de sistemas de alarma es muy común la utilización de microcontroladores, que son sumamente confiables y permiten la utilización mínima de elementos mecánicos. Además, existen los sistemas que utilizan computadoras para su supervisión como una central de monitoreo¹.

¹ Juan José Folguera, Alarmas, Saber Electrónica, p. 7, 13 de marzo de 2001.



También se han comercializado sistemas de alarma sofisticados, con equipos de detección de intrusos mediante controladores de acceso que operan con base en tarjetas personalizadas, o bien, las llaves son reemplazadas por teclados con codificación de claves o transmisores y receptores inalámbricos; otros sistemas están basados en la biométrica. Como lo señala José Antonio Chávez en un artículo del periódico Reforma que transcribo en el apéndice A.1.

Con base en los párrafos anteriores, se concluye que un sistema de alarma puede ser sencillo o muy sofisticado según las necesidades de los usuarios, pero su finalidad es el de dar confianza y tranquilidad a las personas en relación con los robos que afectan a la sociedad.

La inseguridad es un problema que también aqueja a las instalaciones educativas; en los últimos años se ha reportado una alta incidencia de robos de equipo de cómputo, periféricos, y otros artículos de profesores, en las instalaciones de la División de Ciencias Básicas pertenecientes a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

El pasado 20 de febrero de 2004 Arturo Barba y Antimio Cruz publicaron un artículo en el periódico Reforma, respecto de la problemática que enfrenta la Facultad de Ciencias, la transcripción de este artículo está en el apéndice A.2.

Los robos de equipos de cómputo propiedad de la UNAM así como los equipos personales de los académicos son el **antecedente** que origina este proyecto de tesis, que consiste en el diseño de un sistema de alarma contra robos para solucionar un problema técnico real, que surge de la necesidad de seguridad y confianza dentro de las instalaciones de la comunidad académica.

El objetivo es diseñar y construir un sistema de alarma contra robos que sea capaz de prevenirlos por medio del reconocimiento de una clave de cuatro dígitos para permitir el acceso a un cubículo, y en caso de que el código no coincida encender una sirena para proteger los equipos de cómputo, herramientas, proyectos, materiales e instrumentos de trabajo con los que cuentan los profesores para desarrollar sus actividades docentes, y de esta forma propiciar un ambiente de trabajo de seguridad y confianza para los académicos.



Hay que recordar que la seguridad y eficiencia de este sistema implica la discreción con que se maneje el diseño, instalación, ubicación, modo de funcionamiento y claves implementadas. Probablemente el sistema diseñado no sea complejo o avanzado, pero sí muy seguro y confiable al ser personalizado y único, dado que la mayoría de los sistemas comerciales ya son conocidos por los delincuentes.

El sistema de alarma desarrollado cuenta con los siguientes elementos: un control de acceso integrado por un bloque codificador, que permite el ingreso de una clave numérica de cuatro dígitos mediante un teclado de membrana; un bloque memoria en el que previamente se programa una clave de acceso; un bloque sensor para verificar continuamente el estado de la puerta; un indicador luminoso que muestra la cantidad de dígitos de la clave que se ingresan; un bloque demultiplexor para activar y/o desactivar señales de control del sistema; una computadora personal en la que corre un programa de control en lenguaje C para procesar los datos de entrada; un bloque control de sirena como dispositivo de salida que enciende una alarma sonora en caso de que el sistema detecte que la puerta se abre de manera irregular.

El desarrollo de la tesis está dividido en cinco capítulos que se describen a continuación. Este capítulo, corresponde a la introducción, en la que se comenta brevemente el desarrollo de los sistemas de seguridad, lo novedoso que hay en el mercado, los antecedentes que dieron origen a este proyecto, los objetivos por los que desarrolló el sistema de alarma y una descripción de la organización de la tesis por capítulos.

El capítulo 2 contiene la teoría básica sobre puertos paralelos, su definición, su evolución, los elementos que los integran, su funcionamiento, sus procesos de lectura y escritura así como su aplicación en sistemas de control.

Las opciones que pueden satisfacer el diseño de este sistema de alarma contra robo están descritas en el capítulo 3 mediante un esquema básico, en el cual se hace referencia a los aspectos que se tomaron en cuenta en la elección de la solución para desarrollar el sistema.

La descripción de la implementación de un puerto paralelo de 8 bits para una PC de la familia Intel 80x86, su respectivo diagrama electrónico, el desarrollo del hardware del sistema, el proceso evolutivo que originó el circuito final y los diagramas electrónicos se encuentran en el capítulo 4 de esta tesis; aquí se describe cada uno de los bloques y su funcionamiento, y también se menciona cómo trabaja el software que controla al sistema, cuyo programa está elaborado en



lenguaje C. Debido a la extensión y complejidad del programa, se explica con base en un diagrama de flujo general que permite comprender su funcionamiento, y se incluye una lista de materiales electrónicos utilizados y sus precios a fin de presentar un presupuesto del sistema.

Los resultados de los diseños elaborados para el sistema y pruebas que se realizaron se comentan en el capítulo 5, al igual que las conclusiones, aportaciones del proyecto, y una serie de recomendaciones que se consideran como complementos para este sistema de alarma.



CAPÍTULO 2

FUNDAMENTOS DE LOS PUERTOS PARALELOS

En este capítulo presento la información básica para conocer qué es un puerto paralelo, los elementos que lo integran, su funcionamiento y aplicación a sistemas de control¹.

Para ello, primero hago mención del proceso evolutivo por el que han pasado los puertos como resultado del desarrollo de las computadoras a través del tiempo.

Después, identifico las terminales del puerto y describo la función de las mismas, describo la manera en que verifiqué la transmisión de datos de entrada y/o salida entre la PC (Personal Computer, Computadora Personal) y una interfaz.

Posteriormente, presento a la PC como un conjunto de tres unidades elementales que se comunican entre sí, comento la función de las líneas que conectan a dichas unidades, describo las características y terminales de la tarjeta de expansión que utilicé para implementar un puerto paralelo adicional al puerto LPT1 (Line Printer Terminal, Terminal de impresora de línea1) de la impresora. De manera teórica y gráfica, ilustro la forma en la que se pueden leer y/o escribir datos mediante el puerto paralelo.

¹ <http://www.zator.com/hardware/index.html>, <http://www.angelfire/paz/jcr/técnica/PP/pp.html>.



2.1 DEFINICIÓN DE PUERTO

Puerto es la parte de un sistema cuya función es establecer la comunicación con el mundo exterior y a través de él podemos enviar o recibir datos; los puertos de comunicación de la PC, son de particular interés en la electrónica ya que permiten utilizar una computadora personal para controlar cualquier tipo de circuitos electrónicos, principalmente en actividades de automatización de procesos, adquisición de datos, tareas repetitivas y otras actividades.

Existen dos métodos básicos para la transmisión de datos en las computadoras. En un esquema de transmisión de datos en **serie**, un dispositivo envía datos a otro a razón de un bit a la vez mediante un cable; en un esquema de transmisión de datos en **paralelo** un dispositivo envía datos a otro a razón de n bits a través de n cables al mismo tiempo.

Las comunicaciones serie se utilizan para enviar datos a través de largas distancias, ya que las comunicaciones en paralelo exigen demasiado cableado para ser operativas. Los datos serie recibidos desde un modem o otros dispositivos son convertidos a paralelo gracias a lo cual pueden ser manejados por el bus del PC.

Existen uno o más puertos para cada interacción de la unidad de procesamiento principal con sus dispositivos auxiliares. Así, hay un puerto de entrada del teclado, un puerto de salida para el vídeo, un puerto de entrada para el ratón, etc. La PC puede direccionar hasta 65536 (2^{16}) puertos de entrada/salida, I/O de aquí en adelante, por las palabras en inglés Input/Output. Cada puerto se designa por una dirección (un número desde $0000h$ hasta $FFFFh$). A continuación presento en la *tabla 2.1*² las direcciones en hexadecimal de los puertos más usuales de I/O.

Aunque internamente los puertos tienen un número asignado para su identificación, iniciando con el cero se han popularizado los alias empleados por el MS – DOS (Microsoft Disk Operating system, MicroSoft-Sistema Operativo), LPT1, LPT2 y LPT3 que son equivalentes a los números.

² http://www.angelfire.com/pa2/jcgr/tecnica/pp/aspec_gen.html .



Dirección		Descripción
Desde	Hasta	
000	00F	Controlador de DMA (acceso directo a memoria)
020	02F	Controlador de interrupciones maestro
030	03F	Controlador de interrupciones esclavo
040	043	Temporizador
060	060	Teclado
061	061	Altavoz
170	17F	Primer disco duro
200	20F	Puerto de juegos
220	22F	Usualmente tarjetas de sonido
278	27F	LPT2
2E8	2EF	Puerto serie COM4
2F8	2FF	Puerto serie COM2
370	377	Controlador de disco flexible
378	37F	LPT1
3B0	3BB	Adaptador de vídeo monocromo
3BC	3BF	LPT3
3E0	3EF	Puerto serie COM3
3F8	3FF	Puerto serie COM1

Tabla 2.1 Direcciones de puertos más usuales de I/O.

2.2 EVOLUCIÓN DE LOS PUERTOS PARALELOS

Inicialmente las PCs no tenían adaptadores de puerto paralelo, por lo que era necesario usar tarjetas auxiliares en las que se montaban uno o dos puertos paralelos; en otros casos el puerto estaba incluido junto con el adaptador de vídeo.

Los puertos eran sólo de salida, el puerto original es conocido como SPP (Estándar Pallel Port, Puerto Paralelo Estándar) e inicialmente se diseñó para imprimir, tenía capacidad de transmitir 8 bits de datos en dirección de la PC a la impresora, a una velocidad de 150 kbytes/s.

Aunque originalmente se trataba de un puerto de salida, se realizaron adaptaciones para que pudiera recibir datos en grupos de 4 bits; a la forma de operación de 8 bits de salida y 4 bits de entrada se le denomina modo nibble.



La necesidad de conectar dispositivos rápidos y de recibir información por el puerto paralelo llevó al desarrollo de puertos bidireccionales de características mejoradas respecto al SPP tradicional, así se generaron los puertos:

- PS/2 (Puerto Paralelo Bidireccional), que permite una comunicación de 8 bits en ambas direcciones.
- EPP (Enhanced Parallel Port, Puerto Paralelo Extendido), que opera a velocidades entre 0.5 y 2 Mbytes/s, y además de impresoras puede servir para conectar dispositivos como escáneres y unidades de disco o de red externas.
- ECP (Enhanced Capability Port, Puerto de Capacidad Extendida), que dispone de acceso directo a memoria, utiliza generalmente el canal 3 del DMA, y opera entre 2 y 4 Mbytes/s.

Estos dos últimos son al puerto estándar como el procesador Pentium al 80286, el puerto paralelo se utilizó para la comunicación con impresoras. Actualmente se utiliza también para manejar otros periféricos como CD ROM, cintas para copia de respaldo, discos duros, tarjetas de red, escáneres entre otros. En la *tabla 2.2*³ muestro la información sintetizada de cada uno de estos tipos de puertos.

Puerto	SPP	PS/2	EPP	ECP
Fecha de introducción	1981	1987	1994	1994
Fabricante	IBM	IBM	Intel, Xircom y Zenith Data Systems	Hewlett Packard y Microsoft
Bidireccional	No	Sí	Sí	Sí
DMA	No	No	No	Sí
Velocidad	150 kbytes/s	150 kbytes/s	2 Mbytes/s	2 Mbytes/s

Tabla 2.2 Evolución de los puertos paralelos.



2.3 DESCRIPCIÓN DEL CONECTOR DB-25 Y DEL PUERTO LPT1

Los DB-25 son conectores estándar, DB (Data Bus, Bus de Datos) el número 25 está asociado con el número de terminales. Existen dos tipos de este conector, el conector macho y el conector hembra.

- Conector macho: es el que tiene las terminales visibles hacia fuera y se utiliza con un cable de conexión a la PC
- Conector hembra: se encuentra en la placa base de la PC, se distingue por sus orificios en su superficie y se emplea para conectar un periférico o una interfaz.

Las terminales de este conector aparecen en la *figura 2.1* ⁴.

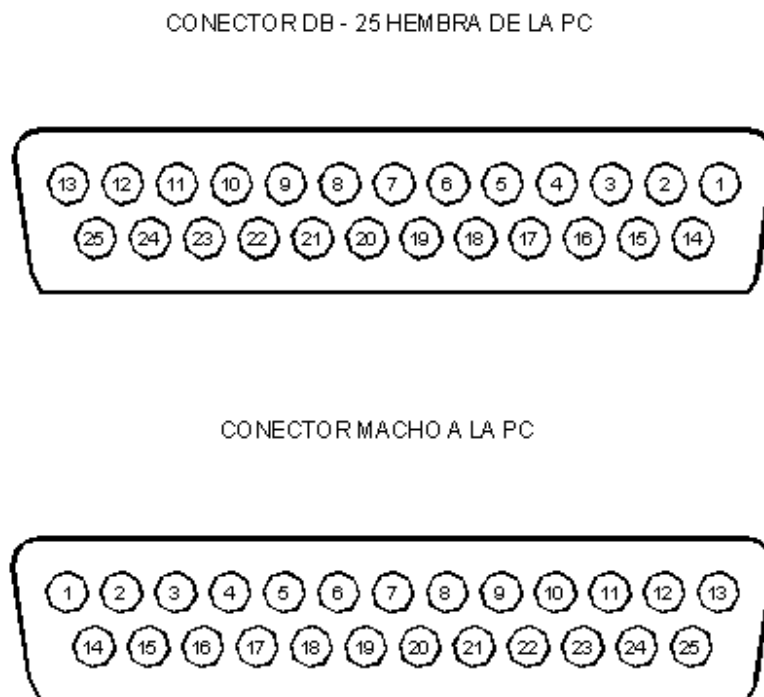


Figura 2.1 Terminales hembra y macho de un conector paralelo DB-25.

³ <http://www.angelfire.com/pa2/jcgr/tecnica/pp/historia.html>.

⁴ <http://www.angelfire.com/pa2/jcgr/tecnica/PP/describ.html>.

En la *figura 2.2*⁵ ilustro el puerto paralelo LTP1 que está formado por 17 líneas de señales y 8 líneas de tierra. Las líneas de señales están formadas por tres grupos:

- 8 líneas de datos
- 4 líneas de control
- 5 líneas de estado.

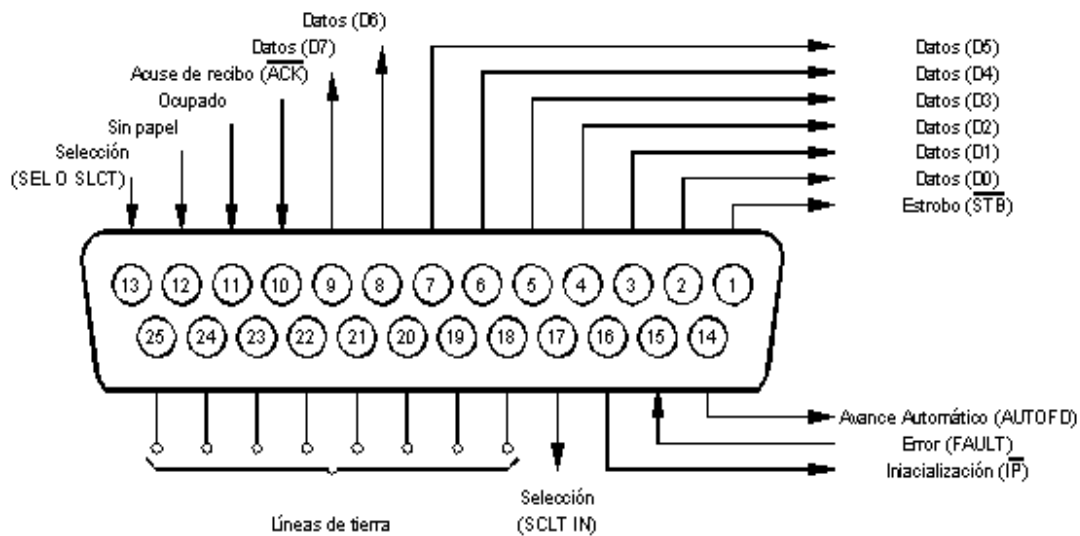


Figura 2.2 Función de cada terminal del conector DB-25.

Las líneas de tierra se indican con circunferencias, y cumplen dos funciones: la primera es que vinculan las líneas de los dos dispositivos que se interconectan en modo que puedan compartir una tierra común como referencia para la señal.

La otra es que la conexión entre los dos dispositivos se realiza a menudo mediante un cable plano y las tierras (conocidas como retornos de tierra en este contexto) actúan como blindajes de las líneas más importantes. En los cables de calidad que no se hacen planos, cada retorno de tierra se retuerce alrededor de una línea de señal formando un par trenzado, para proporcionar un poco de blindaje.

⁵ <http://www.monografias.com/trabajo/paralelos/paralelo.shtml/#arriba>.



Las líneas de datos transfieren información desde la computadora a un periférico en paralelo. Esto se hace con ocho bits (un byte) por vez utilizando las terminales 2 a 9.

D0 se considera el LSB (Low Significant Bit, bit menos significativo) y D7 el MSB (More Significant Bit, bit más significativo), algunos periféricos sólo utilizan datos de 7 bits. En tales circunstancias, el MSB se ignora o a veces se emplea como bit de paridad.

Las líneas de control se utilizan para enviar señales que permitan controlar el flujo de datos, ya que la computadora es mucho más rápida que cualquier periférico con el que se comunica, ésta puede transmitir una mayor tasa de datos que los que un periférico puede manejar.

Por tanto, se utilizan señales especiales para indicarle a la computadora que detenga momentáneamente el envío de datos. Y una vez que el periférico queda libre pide a la computadora que transmita más datos y el proceso continúa.

Las líneas de estado son usadas para intercambio de mensajes, indicadores de estado del periférico a la PC; dicho estado puede afectar el flujo de datos.

Por ejemplo, si una impresora necesita informar a la computadora que se quedó sin papel, puede hacerlo manteniendo alto el bit de sin papel hasta que se aprovisiona nuevamente. Asimismo, un periférico puede informar a la computadora que está energizado y en línea manteniendo alto el bit de "selección" de la terminal 13. Ese es a veces un bit de señal necesario porque algunos periféricos se pueden mantener energizados pero fuera de línea. Un periférico puede incluso solicitar ayuda manteniendo bajo el bit de error para indicar que simplemente está fuera de línea o que se acabó el papel.

En este proyecto utilice dos puertos paralelos, LPT1 y un puerto de 8 bits que adicionalmente implemente, ambos puertos transfieren datos simultáneamente por todas sus líneas, y sus registros tienen direcciones sucesivas que se utilizan para controlar al dispositivo.

Para verificar el funcionamiento de este puerto alámbré una interfaz para los registros, y con el programa de utilidad *debug* del sistema operativo (depurar) transferí datos de entrada del puerto a la PC y datos de salida de la PC al puerto.



Es importante señalar que para la correcta interpretación de datos que se leen o envían a los registros de status o de control, se requiere cambiar los bits invertidos, lo cual puede efectuarse dentro de un programa por medio de la operación lógica XOR, cuyo segundo argumento al que se le suele conocer como máscara, debe tener afirmados los bits que se requieren invertir, y por lo cual a esta operación se le suele denominar enmascaramiento.

Debug es un programa del sistema operativo MS-DOS y es empleado para depurar el software del sistema; tiene diversas funciones, entre otras:

- Comparar dos bloques de la memoria
- Mostrar el contenido de la memoria
- Mostrar y modificar el contenido de la memoria
- Ejecutar un programa con dirección específica
- Cargar datos desde un disquete a la memoria
- Mover el contenido de la memoria
- Enviar datos a los periféricos
- Recibir datos de los periféricos
- Mostrar el contenido del registro de banderas del microprocesador
- Escribir un bloque de memoria a disquete

2.4 BUS Y TARJETA ISA

Una computadora personal, básicamente cuenta con tres elementos:

Procesador, memoria y periféricos.

La forma en la que se comunican entre sí el procesador, la memoria y los periféricos es mediante los *buses de datos, de direcciones y de control*, lo que ilustro en la *figura 2.3*

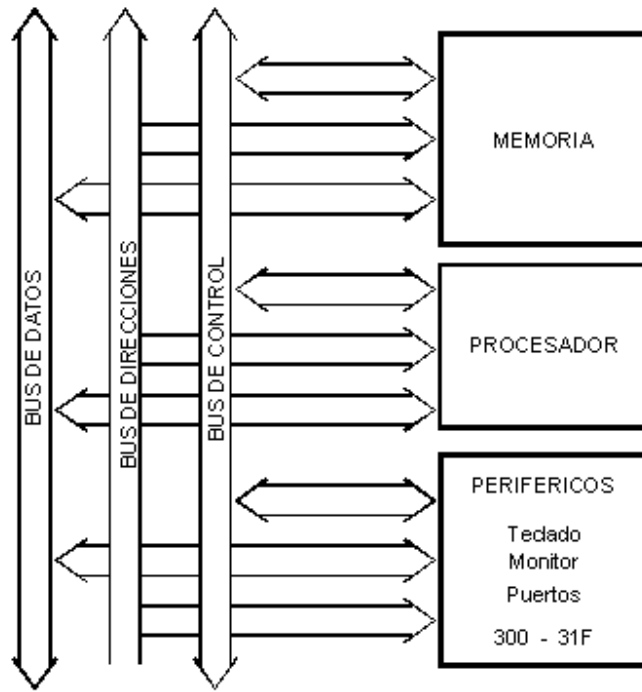


Figura 2.3 Estructura básica de una computadora.

En una PC hay tres buses principales:

Bus de datos: por este bus circulan todos los datos, conecta al CPU (Central Process United, Unidad Central de Procesos), la memoria y otros dispositivos de hardware en una tarjeta principal.

Bus de direcciones: se encarga de indicar la posición concreta de un dato; por lo tanto, para localizar un dato en la memoria principal, la dirección que ocupa éste debe estar presente en el bus de direcciones. Según cuál sea la magnitud del bus de direcciones y la longitud de palabra, así será el tamaño de la memoria que puede disponer la computadora.



Bus de control: por este bus se comandan las instrucciones de los procesos que ejecuta la computadora y se controla el acceso a la memoria y a los periféricos; algunas de las señales que maneja son las siguientes: IOR (Input Output Read, Lectura de un periférico), IOW (Input Output Write, Escritura de un periférico), MEMR (Memory Read, Lectura de memoria), MEMW (Memory Write, Escritura a memoria).

- **Descripción del bus ISA**

El primer bus que se implementó en la arquitectura PC fue el bus ISA⁶ (Industry Standard Architecture, Arquitectura Industrial Estándar); este bus trabaja a una frecuencia máxima de 8.33 MHz. El bus ISA maneja un bus de direcciones de 20 bits y un bus de datos de 8 bits. Permite trabajar con la mayoría de las señales de interrupción de la PC, e incluso utilizar los circuitos de DMA.

En la *figura 2.4* ilustro al bus ISA, con sus dos partes que lo integran. En la primera parte denominada cara A, se encuentran las terminales identificadas desde A1 hasta A31 y corresponden a las de los buses de direcciones y de datos.

La segunda parte denominada cara B, tiene enumeradas sus terminales desde B1 hasta B31 y corresponden a las líneas de alimentación así como las señales relacionadas con las interrupciones y transferencias de datos vía DMA.

Algunos nombres de las terminales ilustradas en la *figura 2.4* tienen el signo negativo, lo que implica que son señales activas bajas, es decir, siempre están en un nivel positivo, y sólo cuando se activan su voltaje baja a cero; en diseño lógico se les llaman *señales lógicas negadas*; mientras que las señales con signo positivo se activan cuando su nivel de voltaje es diferente de cero.

En seguida describo las terminales del bus ISA⁷.

⁶ <http://www.ctv.es/pckids/tutore.html>

⁷ <http://www.ctv.es/pckids/tutore.html>

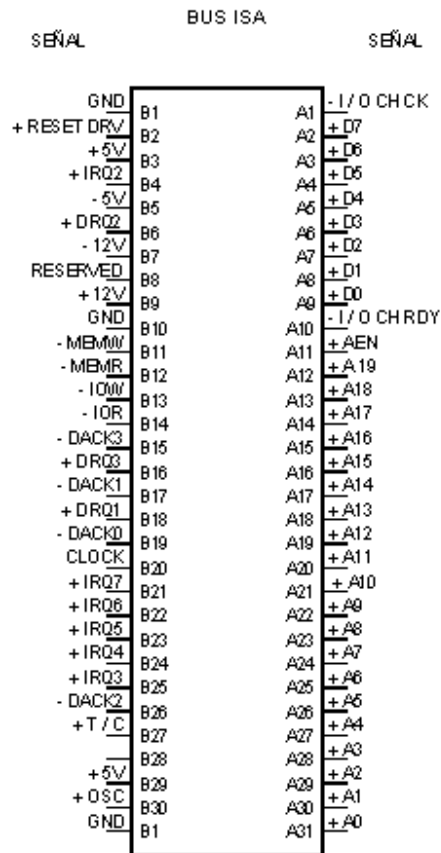


Figura 2.4 Bus ISA.

De la cara A:

A0-A19 (terminales correspondientes A12 hasta A31): el bus de direcciones esta formado por veinte líneas, con las que se pueden direccionar hasta 1MB (2^{20} bytes) de memoria.

-I/O CH CK (I/O Channel Check, Verificación de canal): genera una interrupción no enmascarable.

-I/O CH RDY (I/O Channel Ready, Monitoreo de canal): Se activa por memoria o dispositivos de I/O para retardar el acceso a memoria o los ciclos de I/O.

D0-D7 (terminales correspondientes A2 hasta A9): estas ocho líneas forman el bus de datos.

AEN (Adress Enable, Habilitador de Direcciones) (terminal correspondiente A11): es usada por el controlador de DMA para tomar el control de los buses de datos y de direcciones.



De la cara B:

GND (terminales correspondientes B1, B10, B31): conectadas a la masa del ordenador.

+5V (terminales correspondientes B3, B29): salida de 5 V de continua de la fuente de alimentación.

-5V (terminal correspondiente B5): salida de -5 V

-12V (terminal correspondiente B7): salida -12 V

+12V (terminal correspondiente B9): salida +12 V

MEMW (terminal correspondiente B11): el microprocesador activa esta señal para escribir en la memoria

MEMR (terminal correspondiente B12): el microprocesador activa esta señal para leer de la memoria.

IOW (terminal correspondiente B13): el microprocesador activa esta señal para escribir en un puerto.

IOR (terminal correspondiente B14): el microprocesador activa esta señal para leer de un puerto.

DACK0-DACK3 (acceso directo a memoria) (terminales correspondientes B15, B17, B19 y B26): el controlador de DMA activa estas señales para hacer saber a un dispositivo que el controlador de DMA tiene el control de los buses.

DRQ1-DRQ3 (petición de acceso a memoria) (terminales correspondientes B6, B16 y B18): permite a un periférico reclamar el uso del DMA.

T/C (terminal correspondiente B27): el controlador de DMA activa esta señal para hacer saber a un periférico que el número programado de bytes ha sido enviado.

IRQ2-IRQ7 (Interruption Request, Petición de interrupción) (terminales correspondientes B4, B21, B22, B23, B24 y B25): señales de solicitud de interrupción; los dispositivos periféricos activan estas señales para reclamar la atención del microprocesador.

ALE (Address Latch Enable, Habilidad de Retención de Direcciones), (terminal correspondiente 28): esta señal es usada por el microprocesador para retener los 16 bits de menos peso del bus de datos en un latch durante un ciclo de lectura/ escritura en la memoria o en un puerto.

CLOCK (terminal correspondiente 20): es el reloj del sistema.

OSC (terminal correspondiente 30): es un reloj de alta frecuencia que puede ser usado por las tarjetas de entrada salida



En la *figura 2.5*⁸ se muestran las dimensiones de la tarjeta de expansión ISA y en la *tabla 2.3*⁹, sus características eléctricas.

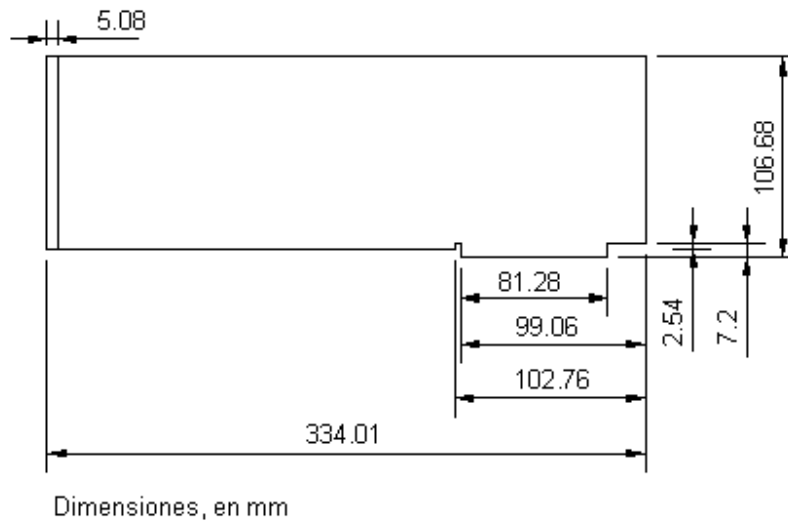


Figura 2.5 Placa ISA

Voltaje de la fuente	Máx (Vdc)	Mín (Vdc)	Corriente	Potencia	Corriente típica / terminal
			(A)	(W)	(A)
+ 5 V dc	5.25	4.8	7	35	0.7
- 5 V dc	-5.5	-4.6	-0.3	1.5	-0.03
+12 V dc	12.6	11.52	2	24	0.1
-12 V dc	-13.2	-10.92	-0.25	3	-0.05

Tabla 2.3 Especificaciones eléctricas de una placa ISA.

⁸ Eggebrecht, *Interfacing to the IBM personal computer*, pag. 79.

⁹ Idem.



2.5 CICLO DE LECTURA Y ESCRITURA DE UN PUERTO DE I/O

- **Ciclo de lectura**

En un ciclo de lectura¹⁰ como el que muestro en la *figura 2.6*, lo primero que hace el microprocesador es subir la señal del ALE a un nivel alto y luego envía la dirección del puerto a través de A0-A19. Después, la señal ALE vuelve a un nivel bajo y en adelante la dirección del puerto a ser leído quedara retenida en un latch (circuitro electrónico para retener provisionalmente un dato binario). Entonces el procesador pone -IOR en nivel bajo, previamente el dispositivo direccionado debió colocar un byte de datos a través de las líneas D0-D7 del bus de datos, el microprocesador leerá el bus de datos y pondrá la señal -IOR en nivel alto de nuevo.

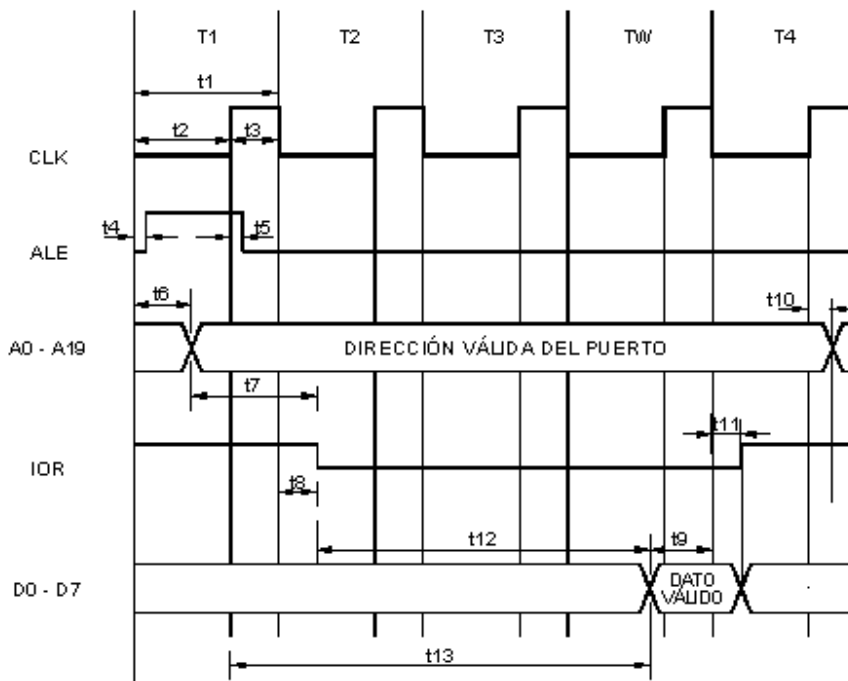


Figura 2.6 Diagrama de tiempos de lectura de un puerto de I/O.

¹⁰ Eggebrecht, Interfacing to the IBM personal computer, p. 64.



La *tabla 2.4* contiene la duración de los intervalos de tiempo que involucra el proceso de lectura de un puerto en un periodo específico.

Símbolo tiempo	Máximo (ns)	Mínimo (ns)
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16
t7	-	91.5
t8	35	10
t9	-	42
t10	-	10
t11	35	10
t12	-	551.5
t13	-	668

Tabla 2.4 Tiempos de lectura de un puerto de I/O.



- **Ciclo de escritura**

Un ciclo de escritura¹¹ en un puerto como el que muestro en la *figura 2.7* funciona de la siguiente manera: el microprocesador sube la señal ALE a "1", luego envía la dirección del puerto a través de A0-A19. ALE es puesta en nivel bajo, el microprocesador envía el byte de datos que será escrito, y posteriormente pone un "0" en -IOW. Después de cierto tiempo durante el cual el dispositivo ha tenido tiempo de leer el byte, el microprocesador pone la señal -IOW en nivel alto de nuevo.

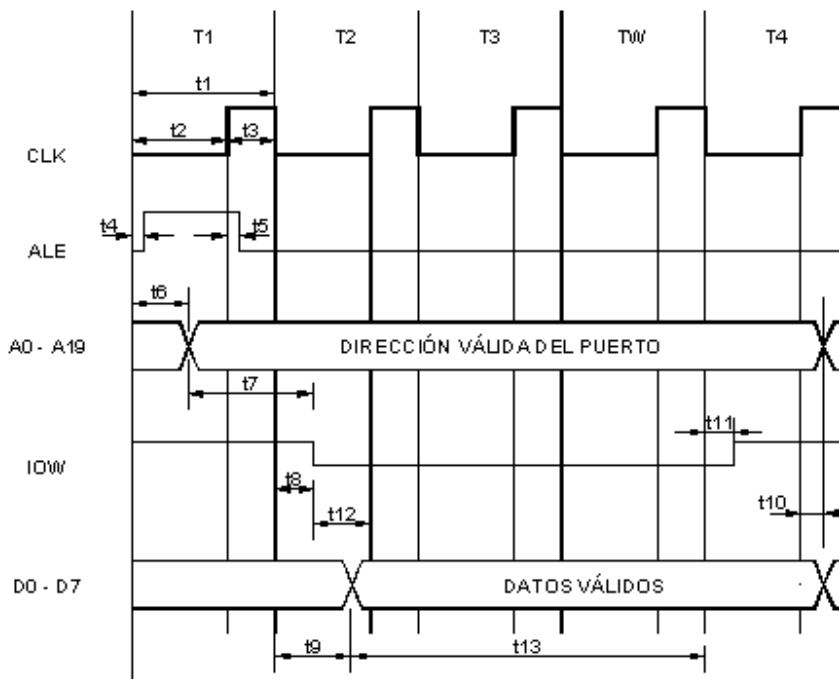


Figura 2.7 Diagrama de tiempos de escritura de un puerto de I/O.

¹¹ Eggebrecht, *Interfacing to the IBM personal computer*, p. 65.



La *tabla 2.5* contiene los intervalos de tiempo que se involucran en el proceso de escritura a un puerto paralelo, en un período específico.

Símbolo tiempo	Máximo (ns)	Mínimo (ns)
t1	-	209.5
t2	-	124.5
t3	-	71.8
t4	15	-
t5	15	-
t6	128	16
t7	-	91.5
t8	35	10
t9	122	14
t10	-	10
t11	35	10
t12	112	-
t13	-	506.5

Tabla 2.5 Tiempos de escritura de un puerto de I/O.

La única diferencia entre un ciclo de I/O a memoria y un ciclo de I/O a un puerto consiste en que en el primer ciclo se utilizaran las señales -MEMR y -MEMW y en el segundo ciclo mencionado se necesitan las señales -IOR e -IOW.



CAPÍTULO 3

SOLUCIONES PROPUESTAS

En este capítulo presento tres posibles propuestas para dar solución al problema planteado en la introducción.

La idea inicial es desarrollar un sistema de seguridad que permita a los profesores la apertura de la puerta de su cubículo; mediante un dispositivo electrónico que reconozca una clave de acceso de 4 dígitos; los requerimientos mínimos que implica el sistema son los siguientes:

- Reconocimiento de la clave de acceso y cambio de clave de forma ilimitada
- Apertura de la puerta después de haber ingresado la clave correctamente y en un lapso menor o igual a 15 s
- Contar con una opción de reinicio, la cual se utilizará en caso de que se interrumpa el proceso de apertura o cambio de clave en el sistema
- Que el sistema continúe funcionando en caso de corte del suministro eléctrico comercial
- Activar una alerta sonora si el sistema detecta que la puerta se abre de forma irregular, presentándose los siguientes casos: Si se abre la puerta sin introducir la clave de acceso, si se introduce una clave incorrecta y la puerta se abre, si se destruye el dispositivo electrónico y si la clave es correcta y se abre después del retardo de tiempo de 15 s.



3.1 PROCESAMIENTO DE LOS DATOS DEL SISTEMA MEDIANTE UN GAL

Una GAL es un Arreglo Lógico Genérico, los dispositivos programables por el usuario como el GAL, contienen una arquitectura general predefinida, gracias a la cual el usuario programa el diseño final empleando un conjunto de herramientas de desarrollo. Las arquitecturas generales de estos dispositivos pueden variar, pero normalmente consisten en una o más matrices de compuertas AND y OR para implementar funciones lógicas.

Otros dispositivos también contienen combinaciones de flip-flops y latches que pueden usarse como elementos de almacenaje para entrada y salida del dispositivo. Los dispositivos más complejos contienen macrocélulas. Las macrocélulas permiten al usuario configurar el tipo de entradas y salidas necesarias en su diseño.

Para este trabajo, la solución al problema mediante su implementación con arreglos lógicos programables es viable, dado que es una manera de "personalizar" los diseños lógicos, pensando en el hardware propio. Para ello, primero defino exactamente el problema a resolver y lo planteo de una manera clara, establezco la descripción funcional del comportamiento del sistema, y dibujo un diagrama de estados.

Del diagrama de estados y la secuencia planteada obtengo la tabla de estados, en la cual defino las características del comportamiento del problema a resolver.

Después traduzco la tabla de asignación de estados a un mapa de Karnaugh, para encontrar las ecuaciones lógicas booleanas que gobiernan el sistema de alarma. Estas ecuaciones las debo editar en un archivo que se grabará con extensión .eqn.

Para programar el GAL16V8 se emplea el compilador OPAL, que es un software con el que se definen las entradas y las salidas de los circuitos de acuerdo a las variables empleadas, y las ecuaciones.

Con base en el archivo que contiene las ecuaciones, el OPAL realiza las operaciones para generar los archivos necesarios que permiten observar los estados de entrada y salida del circuito secuencial en una gráfica de pulsos generada automáticamente por el software.



Si la simulación produce resultados satisfactorios, se graba el programa en el GAL mediante un programador de dispositivos programables y quedará listo para su funcionamiento.

Después, se arma el circuito y por último, se realizan pruebas eléctricas del diseño final.

El GAL que seleccioné para ser empleado en el circuito es el GAL16V8 cuyas características principales son las muestro a continuación.

- 20 terminales
- Fuente de alimentación es $V_{cc} = 5\text{ V} \pm 5\%$
- Consumo de corriente 90 mA
- Frecuencia máxima del reloj para los flip-flop, 41.6 MHz
- Voltaje de entrada en nivel alto 2 V
- Voltaje de salida en nivel alto 2.4 V
- Voltaje de entrada en nivel bajo 0.8 V
- Voltaje de salida en nivel bajo 0.5 V
- Temperatura de operación de 0 a 75 °C
- Admite 16 variables de entrada diferentes y 8 variables de salida diferentes.

3.2 PROCESAMIENTO DE LOS DATOS DEL SISTEMA MEDIANTE UN MICROCONTROLADOR

Otra solución al problema planteado es el diseño del sistema de alarma con base en un microcontrolador.

Un microcontrolador¹ es un microprocesador optimizado que incorpora en su interior las tres unidades fundamentales de un ordenador: CPU, memoria, unidades de I/O, es una microcomputadora que se dedica a resolver tareas específicas en su mayoría control de dispositivos, por lo que suele ir incrustado en dicho dispositivo.

¹ <http://es.wikipedia.org/wiki/Microcontroladores>



En el mercado se encuentra una gran gama de microcontroladores a elegir dependiendo de la aplicación que se le dará; pero sin duda la elección dependerá del proceso y del sistema a plantear, pues los fabricantes tienen diferentes modelos enfocados a tareas específicas. Esta selección deberá ir de la mano con la optimización tanto económica como de la funcionalidad y adecuación al dispositivo considerado.

A continuación presento una opción de solución al problema planteado, mediante la programación de un microcontrolador de la familia Microchip, y una segunda opción con un microcontrolador de la familia Motorola.

3.2.1 PIC

Los microcontroladores PICs (Programmable Interrupt Controller, Controlador programable de instrucciones) funcionan como una computadora que se programa para que cumpla una función específica.

Primero hay que definir los elementos del sistema, y la relación entre ellos, generar un programa para que el PIC ejecute las rutinas necesarias de control y activación de los elementos que integran al sistema.

Un PIC adecuado para esta tarea es el 16C84 entre cuyas características² están:

- Encapsulado de 18 terminales
- Frecuencia de trabajo: 10 MHz máxima
- Temporizadores
- Líneas de I/O digitales: 13 líneas (5 Puerto A y 8 Puerto B)
- Voltaje de alimentación (V_{dd}): De 2 a 6 V DC
- Voltaje de grabación (V_{pp}): De 12 a 14 V DC
- Memoria de programa EEPROM (Electrically Erasable Programmable Read Only, Memoria Eléctricamente Programable y Borrable) de 1k x 14 bits
- Memoria de datos dividida en 2 áreas:
 - Área RAM formada por 22 registros de propósito específico, y 36 de propósito general
 - Área EEPROM formada de 64 bytes
- ALU de 8 bits



- Contador de programa de 13 bit
- 35 instrucciones de programación de las que comúnmente se utilizan 15

3.2.2 Microcontrolador Motorola MC68HC11

La siguiente propuesta emplea la posibilidad de usar un sistema mínimo basado en el microcontrolador MC68HC11 de Motorola para el control del sistema.

Motorola describe al 68HC11³ como un microcontrolador de 8 bits fabricado con tecnología HCMOS (High Density Complementary Metal Oxide Semiconductor, Semiconductor de Metal Oxido Complementario de Alta Densidad), con una frecuencia de bus de 2 MHz y con una amplia lista de recursos internos. Es capaz de ejecutar todas las instrucciones de los microcontroladores M6800, M6801 y 91 más que se le han incorporado.

Los modelos más importantes que componen la familia Motorola tienen como principal diferencia entre ellos, la cantidad de RAM, ROM, EPROM y EEPROM.

Algunos recursos internos disponibles:

- 256 bytes de memoria RAM
- 5 puertos de 8 bits, con pines de entrada, salida y de I/O
- Convertidor analógico-digital de 8 canales y 8 bits de resolución
- Una UART (Universal Asynchronous receiver transceiver, Receptor/Transmisor Asíncrono Universal) para comunicación serie asíncrona
- Un módulo de comunicación serie síncrona
- 5 comparadores
- Temporizador principal de 16 bits
- Interrupciones en tiempo real
- 2 entradas de interrupciones externas.

Para poner a funcionar este microprocesador, todo lo que se necesita es desarrollar un programa de control que permita activar los dispositivos para abrir la puerta o, en su caso activar una sirena.

² Horacio D. Vallejo, Todo sobre PICs, Saber Electrónica, p.5-20

³ http://www.itson.mx/die/eromero/biblioelec/bsistdig3/Man_microb.pdf



3.3 PROCESAMIENTO DE LOS DATOS DEL SISTEMA MEDIANTE UNA PC

La siguiente es una propuesta para resolver el problema de esta tesis e implica el uso y la programación de los puertos paralelos de una PC con procesador 80X86, con el fin de controlar una interfaz electrónica que permita la apertura de una puerta mediante una clave de acceso.

Esta interfaz consiste en el diseño de un circuito electrónico que es controlado por un programa que corre en la PC, he considerado la necesidad de implementar un puerto paralelo de 8 ó 16 bits adicional al puerto LPT1 de la PC, con la finalidad de que el hardware sea robusto.

Las siguientes son algunas características del procesador intel 80X86:

- Se diseñaron originalmente con una arquitectura interna de 16 bits y pueden trabajar con operandos de 8 y 16 bits, con una capacidad de direccionamiento de 20 bits (hasta 1 MB) y comparten el mismo conjunto de instrucciones.
- Las frecuencias internas de reloj típicas son 4.77 MHz en la versión 8086; 8 MHz en la versión 8086-2 y 10 MHz en la 8086-1. Hay que recordar que un MHz son un millón de ciclos de reloj, por lo que una PC *estándar* a 4.77 MHz puede ejecutar de 20.000 a un medio millón de instrucciones por segundo, según la complejidad de las mismas.

3.4 COMPARACIÓN DE LAS SOLUCIONES PROPUESTAS

Las propuestas de solución planteadas tienen ventajas y desventajas, unas con respecto de las otras; a continuación describo las ventajas primero, y posteriormente las desventajas de cada una de las propuestas.

3.4.1 Propuesta con GAL

La tarea del diseñador lógico se simplifica por la existencia de programas con los que actualmente se realizan tareas de diseño y minimización de lógica secuencial, esto es, representación booliana de alto nivel; el software facilita la selección de dispositivos y ayuda a generar vectores de prueba para la simulación del circuito antes de su implementación final.



El GAL16V8 es un dispositivo de la familia de los PLD'S (Programmable Logic Devices, Dispositivos Lógicos Programables), que contiene importantes mejoras sobre sus antecesores y que lo hacen versátil y funcional.

La principal ventaja es su reprogramabilidad, es decir, el GAL16V8 ofrece la opción de borrar las ecuaciones grabadas en él mediante pulsos eléctricos; su celda básica es una EECMOS (Electrical Erase Complementary Metal Oxide Semiconductor, Semiconductor de Metal Oxido Complementario Eléctricamente Borrable), que le proporciona la característica de borrrable, además ofrece un bajo consumo de corriente.

La principal desventaja de los circuitos GAL es su sensibilidad a las variaciones de voltaje de alimentación, que pueden provocar mal funcionamiento e inestabilidad en su desempeño, obligando al diseñador a asegurar que la tarjeta en la que se le coloque no exista una variación considerable de voltaje.

3.4.2 Propuesta con un PIC

La ventaja que se tiene con estos dispositivos es que se pueden reprogramar las veces que sean necesarias.

La razón de ser una opción para el diseño del sistema se debe al tipo de memoria de programa que posee y a que el PIC 16C84 tiene cualidades que comparten los microcontroladores RISC (Reduced Instruction Set Computer, Computadora de Conjunto de Instrucciones Reducido), cuya característica principal es tener un juego reducido de instrucciones, sencillas y su ejecución por lo general requiere de un sólo ciclo de máquina. En este caso, un ciclo de máquina equivale a cuatro ciclos de reloj.

La principal desventaja de estos circuitos integrados es que ofrecen una cantidad limitada a 13 líneas de entrada y/o salida, por lo que serían necesarios dos PIC interconectados para desarrollar el sistema propuesto.



3.4.3 Propuesta con el microcontrolador MC68HC11

Aún cuando existen diversos microcontroladores en el mercado, el 68HC11 destaca por sus recursos, simplicidad y relativa facilidad de manejo, razones por las cuales es otra opción para resolver esta tarea.

Sin embargo, para desarrollar aplicaciones con el microcontrolador 68HC11 se necesita una tarjeta de entrenamiento y desarrollo. Estas tarjetas permiten cargar programas en la RAM interna del microcontrolador desde la PC. Una vez que el programa funciona correctamente, se graba en la EEPROM interna o en una EPROM externa.

Aquí se hace referencia a la tarjeta CT6811 del Grupo J&J. Esta tarjeta contiene lo mínimo que se necesita para poder trabajar con el 68HC11. Lo interesante es que puede servir para el desarrollo de aplicaciones (modo entrenador), y para usarla como producto terminado (modo autónomo)

3.4.4 Propuesta con base en una PC 8086

Algunas ventajas que ofrece este sistema son las siguientes.

- El sistema funciona con base en un software, lo que permite reprogramar las veces que sea necesario; además, puede funcionar en una PC con ranuras ISA o PCI (Peripheral Component Interface, Conexión de Componentes Periféricos) y al menos un procesador del tipo 80X86.
- Tiene posibilidades de incrementar la cantidad y diversificar los dispositivos de detección mediante la modificación del software de control.

En comparación con las propuestas anteriores su principal desventaja es el tamaño, en virtud de que el espacio que ocupa una PC es más grande que el que utiliza un arreglo de GAL o un microcontrolador.



3.5 OTRAS CONSIDERACIONES

La Ingeniería como profesión, disciplina que se enfoca a la resolución de problemas técnicos, debe contemplar el impacto ambiental de la tecnología.⁴

Desde mediados de los años 80, la comercialización excesiva de la electrónica y de la computación llenó los hogares, oficinas, empresas y escuelas de aparatos de consumo. La lista es interminable, pero un buen ejemplo de esto es el de las PC's y sus periféricos, que debido al mejoramiento de la tecnología y a costos cada vez más bajos, los equipos son reemplazados por otros más modernos con mayor frecuencia.

Más allá de todo lo que se pueda decir sobre el consumismo que esta situación ha generado, apenas unos pocos años después de esta comercialización de la electrónica aparece un nuevo problema, aún sin solución: la basura electrónica.

Se denomina de esta manera a los equipos electrónicos (especialmente de consumo) que se encuentran cerca del final de su vida útil, que no funcionan o bien que se dejaron de utilizar por hacerse obsoletos.

Entre los grandes problemas que generan estos desechos figuran los que listo a continuación.

- Muchos de sus componentes son peligrosos y/o altamente contaminantes.
- No se sabe muy bien qué hacer con ellos.
- Los procesos de reciclaje son costosos y nadie quiere hacerse cargo de los mismos.
- El principal "productor" de basura electrónica son los EUA, hecho que debemos considerar debido a la cercanía territorial.
- La falta de regulación en materia legal, origina que algunas empresas y países se desentiendan del problema "exportando" esta basura a otros países.
- Las computadoras y equipos electrónicos en general contienen metales tóxicos pesados como plomo, cadmio, mercurio y cromo, todos ellos cancerígenos y no fácilmente biodegradables.

⁴ <http://facom.udp.cl/CEM/TDC/artic/basura/basura.htm>, <http://www.canietigdl.com.mx/Articulo%202/>



- Estos desechos van a parar a lugares deshabitados o faltos de regulación legal en esta materia, lo que origina un enorme problema de contaminación ambiental.
- En algunos de estos lugares existe el peligro de que los metales pesados penetren en las capas de agua.

La SVTC (Silicon Valley Toxics Coalición, Coalición para evitar los tóxicos de Silicón Valley) estima que para el año 2004, sólo en los EUA habrá más de 315 millones de computadoras en desuso, que contendrán más de 600 millones de kilos de plomo, un millón de kilos de cadmio y 200 mil de mercurio, más mil millones de kilos de plástico.

De acuerdo con la Agencia de Protección Ambiental de los EUA, alrededor de 20 millones de computadoras se volvieron obsoletas en el año 1998, lo cual generó entre 5 y 7 millones de toneladas de desechos. Solo el 14% de este desperdicio fue reciclado.

En México, cada habitante produce aproximadamente 311.4 kg de basura al año. Toda esa basura tiene que ir a algún lado. En algunos lugares la basura fácilmente se entierra en algún tiradero. En otros, la basura se ha convertido en un problema crítico a causa de la falta de espacios para tiraderos, incineradoras y al creciente costo para deshacerse de la basura.

Los problemas por la basura electrónica en nuestro país se incrementarán cuando en el año 2004, como parte del TLCAN (Tratado de Libre Comercio de América del Norte), se pueda importar a México, sin permiso y con arancel cero, equipo electrónico usado procedente de los EUA. Mucho de ese equipo, al ser obsoleto en su país de origen, no estará fabricado de acuerdo a las nuevas normas que están entrando en vigor.

Un ejemplo de este problema ambiental es el que afecta al sur de China, en la provincia de Guizhou, donde la mayoría de sus habitantes se dedican al comercio y reciclaje de equipos descartados de empresas norteamericanas, ya que los campesinos desarmen los componentes de plástico y metal con sus manos desnudas en busca de pedazos que puedan vender, y se exponen a químicos tóxicos.

Al igual que en otros tiraderos, como los que existen en India y en Filipinas, los trabajadores inmigrantes ganan unos cuantos centavos por abrir estuches de plástico con el logo de IBM o de Hewlett-Packard para recuperar los procesadores.



Se sumergen tableros de circuito y chips en ácido para recuperar pequeñas cantidades de oro; durante todo el proceso inhalan tóxicos y los ácidos se vacían en el cercano río Pearl en China. Debido a esto se reportan problemas respiratorios, cutáneos, estomacales y el aumento de abortos espontáneos.

3.6 TOMA DE DECISIONES PARA EL DISEÑO DEL SISTEMA

Con base en el análisis de soluciones propuestas, opté por la solución de implementar el sistema de alarma mediante una PC, porque la disponibilidad de contar ya con más de una PC, reduce el tiempo y costo para desarrollar el sistema.

En las instalaciones de la UNAM se cuenta con equipos de cómputo en sus diferentes áreas de trabajo como laboratorios, bibliotecas y cubículos de profesores. Además dentro de la Facultad de Ingeniería se estima que al menos el 50% del equipo de computo pronto será obsoleto y de no darle otra utilidad será basura electrónica.

Atendiendo a la necesidad de la conservación del medio ambiente, esta propuesta implica el reutilizar equipo de cómputo obsoleto, con procesadores Intel 80286, 80386 ó 80486, cuyas características de velocidad de procesamiento y capacidad de memoria son suficientes para el funcionamiento adecuado del sistema de alarma.



CAPÍTULO 4

DISEÑO DEL SISTEMA

En este capítulo presento una descripción de la implementación de un puerto paralelo de 8 bits para un a PC con procesador Intel 80X86, e incluyo su diagrama electrónico, y en seguida el hardware del sistema de alarma, que está integrado por varios bloques, de los que comento su función y la relación que hay entre ellos.

Después comento el diseño y funcionamiento por bloques del primer sistema implementado. Con base en las pruebas que realicé a dicho sistema, identifiqué la necesidad de realizar cambios de algunos bloques, a fin de hacer un diseño más robusto; estas consideraciones me llevaron al diseño de un segundo sistema de alarma, para ambos casos presento diagramas de bloques generales y fotografías.

También incluyo una descripción del software que controla al sistema, basando mi explicación en un diagrama de flujo.

4.1 DISEÑO DE UN PUERTO PARALELO ADICIONAL DE 8 BITS

Implementé un puerto paralelo de 8 bits para una PC de la familia Intel 80X86 en una tarjeta de expansión ISA; a continuación en la *figura 4.1*, muestro el diagrama de bloques del puerto.

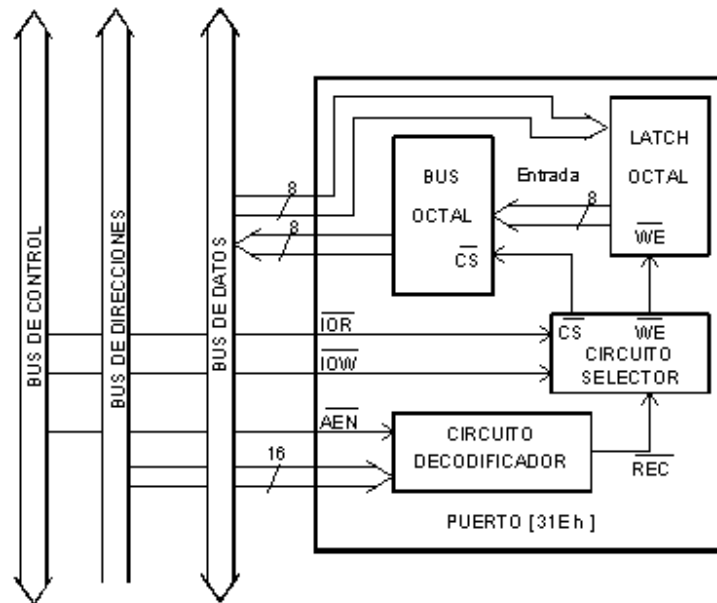


Figura 4.1 Diagrama de bloques del puerto paralelo 31Eh.

Para trabajar con el puerto paralelo es necesario en primer lugar conocer su dirección. Cada periférico tiene una dirección de 16 bits, y debe reconocerse con un decodificador personalizado, es decir, identificar cuándo se desea interactuar con el puerto.

En las PC's compatibles con IBM las direcciones disponibles para el usuario van desde la 300h hasta la 31Fh, las tres más recomendadas por la cantidad de 1 que la componen son: 31Bh, 31Eh, 31Dh, de las cuales elegí la dirección 31Eh.

Para reconocer dicha dirección implemente un circuito decodificador donde conecté los bits A0, A5, A6, A7, A10 y AEN de la tarjeta ISA a circuitos inversores y las terminales A1, A2, A3, A4, A8 y A9 directamente a las entradas de una compuerta NAND, siendo su salida una señal que llamé -REC (reconocimiento) activa baja.

Dado el funcionamiento de esta compuerta, para la condición de tener 0 a la salida se debe satisfacer que todas las entradas sean 1, es decir, siempre que detecte el número 31Eh las entradas se consideran todas 1 y la salida será un cero lógico.

En cualquier otro caso las entradas provocarán un 1 lógico a la salida, lo que significa que está leyendo una dirección diferente a la que se necesita.



Las señales -IOR e -IOW provenientes del bus de control no pueden activarse simultáneamente, para la detección de estas señales, implementé un arreglo lógico con dos compuertas OR, para generar un circuito de selección de señal con implica dos casos diferentes:

1. Que se presenten las señales -IOR, -REC como entradas a una compuerta OR y que su salida sea -CS (Chip Select) donde -CS habilita el ingreso de datos a la PC del exterior.
2. El conjunto de señales -IOW, -REC como entradas a una compuerta OR y que su salida sea -WE (Write enable) donde -WE es habilitador de escritura del latch octal del puerto.

En la *figura 4.2* muestro el diagrama del circuito electrónico para el puerto de 8 bits.

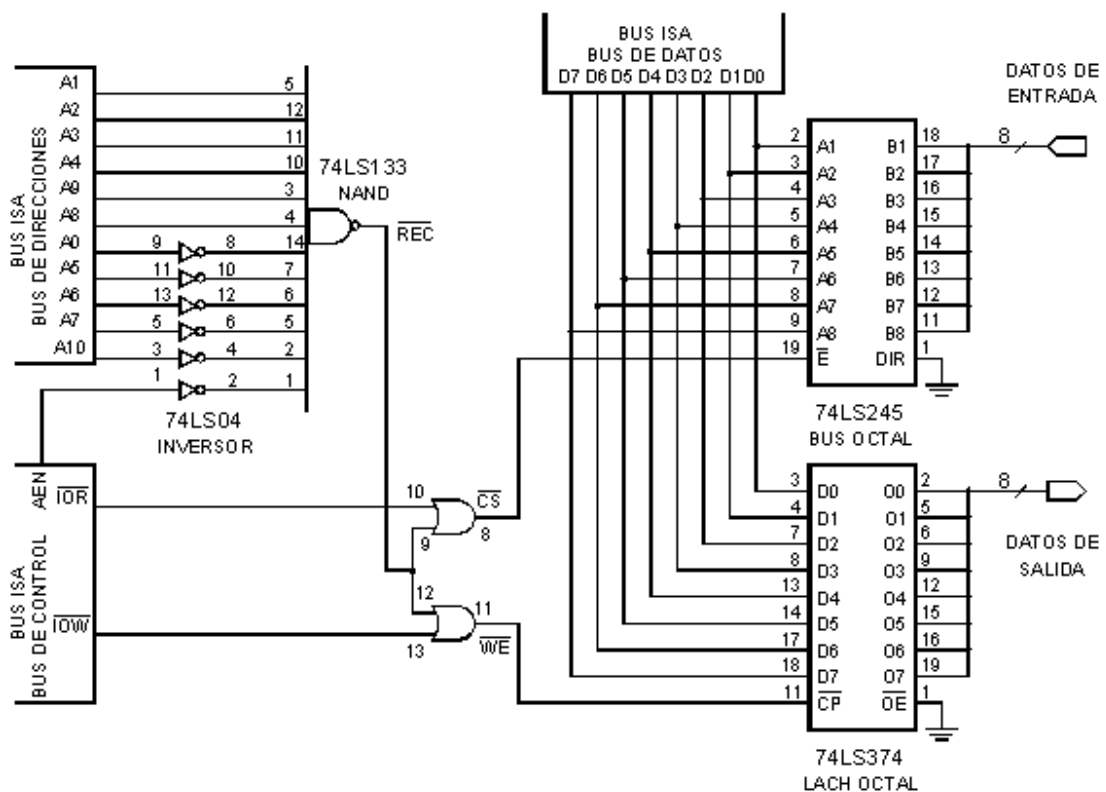


Figura 4.2 Diagrama esquemático del puerto paralelo 31Eh.



4.2 HARDWARE

4.2.1 Descripción del sistema

El sistema de alarma propuesto debe reconocer una clave numérica que se ingresa mediante un teclado, el programa de control del sistema realiza la comparación de la clave ingresada con una clave grabada en una memoria, si las claves son iguales se tiene 15 s para abrir la puerta, con la puerta abierta puede realizarse el cambio de clave; un fototransistor verifica continuamente el estado de la puerta y lo envía a la PC, si el programa detecta que el estado de la puerta es diferente del que espera la sirena se activa.

Para satisfacer los requerimientos del sistema propuesto implemente una interfaz electrónica compuesta por los bloques: teclado, despliegue, buffer y memoria, buffer demultiplexor, sensor y control de sirena.

Como muestro en la *figura 4.3*, estos bloques interactúan entre sí, transmiten y reciben datos de la PC mediante los puertos paralelos.

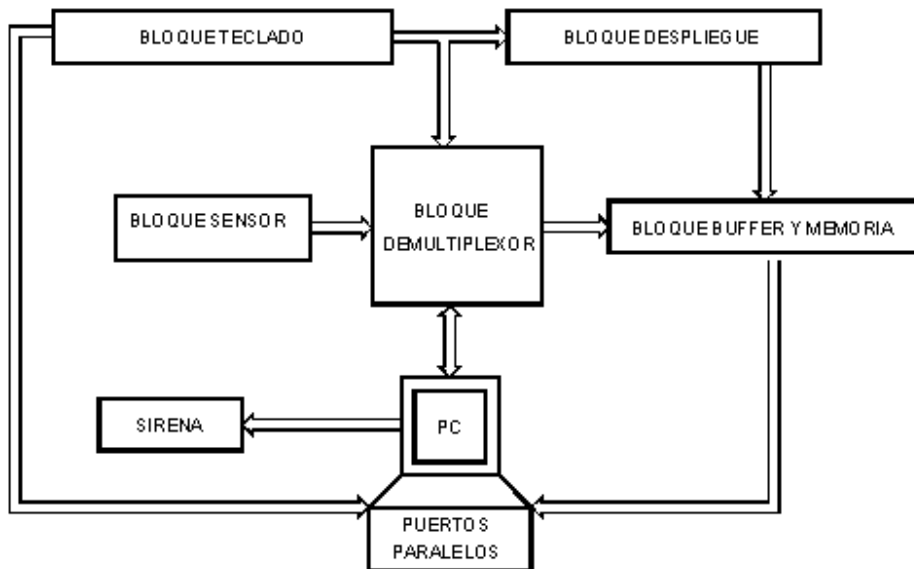


Figura 4.3 Diagrama de bloques del sistema.



Los bloques teclado y despliegue estarán colocados en la puerta, el primero permite el ingreso de la clave numérica y el segundo bloque muestra los dígitos ingresados por lo que sus salidas servirán como datos de entrada al software y guía en caso de corregir o cambiar la clave ingresada; por medio del bloque buffer y memoria se puede leer o escribir en la memoria, mientras que el bloque buffer demultiplexor realiza la transferencia de la clave desde el bloque teclado al puerto LTPI; el bloque sensor tiene la función de monitorear continuamente el estado de la puerta y el bloque control de sirena se encarga de activar la alerta sonora del sistema de seguridad.

4.2.2 Diseño y desarrollo de la primera propuesta

En los siguientes párrafos hago una breve descripción de la función de los bloques que integran a la primera propuesta del sistema.

En la primera propuesta, implemente al bloque teclado con un teclado de membrana; su salida es la entrada a un codificador que genera una salida de la forma BCD, esta salida es enviada por un lado a una compuerta NOR para generar una señal activa baja que llamé -BOT, para indicar al sistema que se presionó una tecla; y por otro lado la salida BCD y dicha señal -BOT son las entradas al bloque despliegue.

Por un lado la señal BOT es la entrada a un circuito reloj, cuya salida se transfiere a un registro de corrimiento, para controlar el almacenamiento de los cuatro dígitos de la clave; por otro lado, la salida BCD generada en el bloque teclado es la entrada a un circuito llamado registro y que integré con cuatro flip-flop para almacenar temporalmente el dígito ingresado, las salidas de este registro las conecte a decodificadores BCD a 7 segmentos, para mostrar la clave ingresada durante las pruebas de funcionamiento del sistema.

La memoria EEPROM requiere ser puesta en modo de escritura o lectura mediante señales de habilitación que se envían desde el puerto paralelo LPT1 y a través del bloque buffer demultiplexor. Para leer el contenido de la memoria deben habilitarse sus terminales correspondientes al modo lectura, y los datos leídos se envían al puerto adicional para que la procese el software del sistema. Es importante señalar que durante este proceso, los buffer que conecte a la memoria deben estar en alta impedancia para evitar conflictos por interconectar entre sí las dos salidas de los buffer. En el caso de escribir una nueva clave, la memoria se



habilita en modo escritura y las salidas BCD del bloque despliegue son transferidas a dos buffer octales van a la memoria.

El bloque buffer demultiplexor permite la transferencia de datos desde el circuito electrónico del sistema a la PC, diseñé este bloque con dos circuitos denominados buffers octales, el programa control envía a este bloque una señal de habilitación a cada buffer de manera que se habilite uno u otro según lo requiera el sistema.

Inicialmente simule al bloque sensor con un interruptor y su salida la conecte a una terminal del bloque demultiplexor. Cuando el bloque envía al software un estado de la puerta diferente al que espera se enciende un led indicador y se activa la sirena que simule con una pequeña bocina.

En la *figura 4.4* ilustro de manera esquemática los bloques que implemente en la propuesta inicial del sistema, donde aparece la dirección de la transmisión de señales y/o datos; las líneas delgadas indican que viaja un solo dato o señal, en tanto que las líneas de bus van acompañadas de un número que indica la cantidad de datos o señales transmitidas.

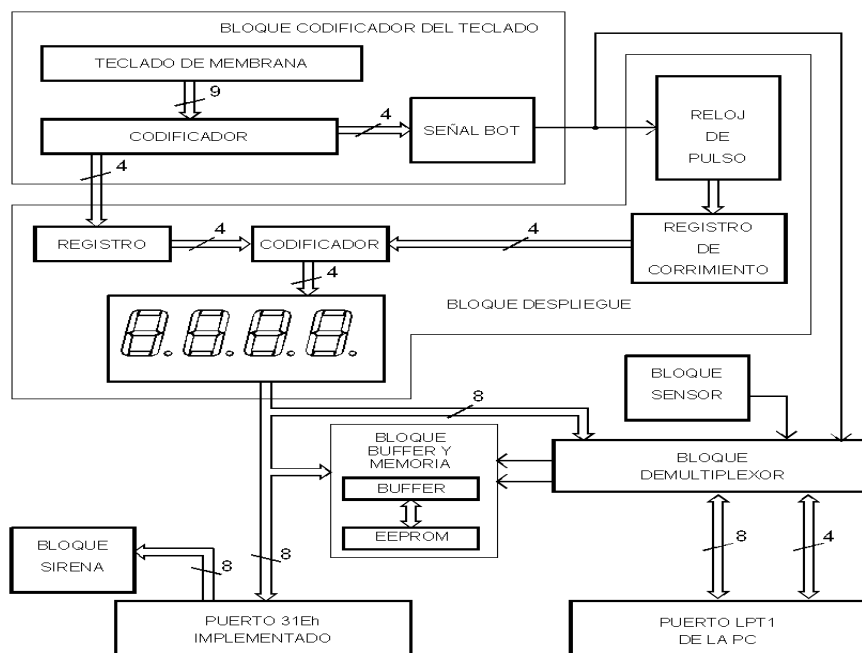


Figura 4.4 Diagrama de bloques detallado de la propuesta inicial.



Resultados de la primera propuesta

Para este primer circuito implementado desarrollé un software para realizar pruebas de funcionamiento e identificar posibilidades de mejora.

Lo primero fue verificar el correcto funcionamiento de cada bloque del circuito para asegurar que la información que enviaba al puerto fuera adecuada para que el programa la procesara. En esta fase se presentaron las situaciones que describo a continuación.

Este sistema incluía la posibilidad de reiniciar pulsando la tecla con el número “4”, pero no siempre era reconocida; esto me impedía que el sistema regresara a su estado inicial donde espera el ingreso de la clave.

Después de ingresar cada dígito de la clave de acceso, era necesario pulsar la tecla con el número “2” para continuar con el proceso, de no ser así la sirena se activaba.

Para borrar la clave durante su ingreso, era necesario pulsar la tecla con el número “1”, el inconveniente es que sólo se borraba dígito por dígito y debía borrarse toda la clave independientemente de la cantidad de números ingresados.

En la medida en que fui evaluando el sistema, el hardware se volvía cada vez más inestable; medí la resistencia en varios puntos de contacto con respecto a la línea de tierra, observé que variaba desde 1 hasta 20 Ω lo cual causaba inestabilidad en el circuito. En las *figuras 4.5 y 4.6* muestro los circuitos de la propuesta inicial.

Con base en lo descrito en párrafos anteriores, decidí rediseñar nuevamente el sistema incluyendo las siguientes modificaciones.

- Reinicio automático del sistema al estado inicial de espera.
- Asignación de la tecla “*” para borrar y la tecla “#” para continuar el proceso;
- Eliminación de la clave pulsando la tecla “*” se debe eliminar toda la clave ingresada, y regresar al estado de inicio del programa.

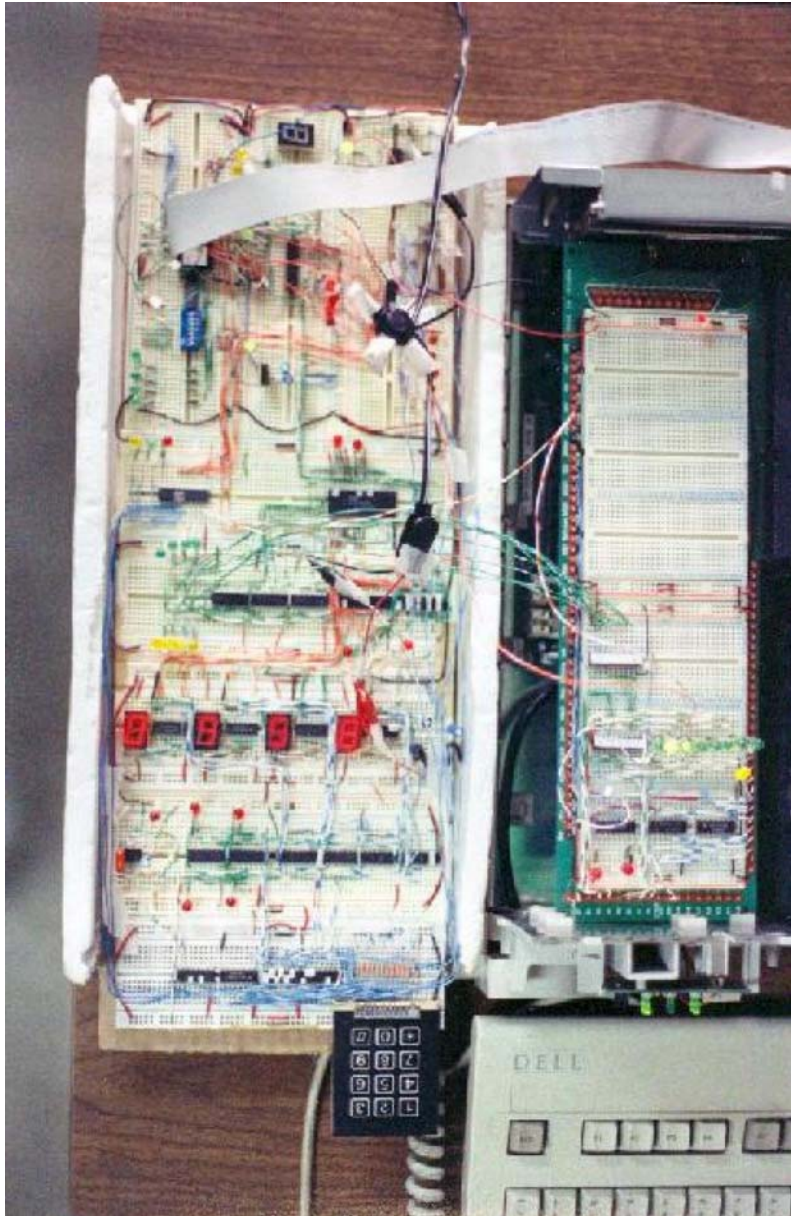


Figura 4.5 Hardware de la propuesta inicial



Figura 4.6 Sistema de alarma completo de la propuesta inicial.



4.2.3 Diseño y desarrollo de la segunda propuesta

Enseguida presento la segunda propuesta del sistema con base en las pruebas realizadas en el primer diseño.

En el bloque teclado sustituí la compuerta NOR por una NAND de trece entradas para alambrear las teclas “ * ” y “ # ”, además coloque condensadores en cada terminal del bloque teclado para evitar los rebotes que se generaban al presionar una tecla.

Por razones de seguridad y para evitar mostrar la clave de acceso, en esta propuesta sustituí los despliegues de 7 segmentos con sus respectivos codificadores y registros por un despliegue de arreglo de 4 leds controlados por una GAL16V8, este circuito sólo muestra el orden y la cantidad de dígitos ingresados. Para enviar datos desde la PC al bloque de leds implemente un nuevo bloque con dos buffers, al que llamo bloque control.

A diferencia del diseño anterior en que el proceso de lectura y/o escritura de la memoria requería de transferir los datos desde el bloque despliegue, mejore el diseño de manera que la transferencia se realice mediante el puerto paralelo 31Eh; con esta modificación el bloque requiere únicamente de la memoria EEPROM y un sólo buffer octal.

En esta propuesta sustituí el interruptor tradicional por un interruptor óptico, también llamado fototransistor¹ utilizo el H21A1 para monitorear el estado de la puerta, un 0 lógico indica que la puerta está cerrada y un 1 lógico que la puerta está abierta, el fototransistor resulta de la combinación de un fotodiodo y un transistor bipolar npn, cuando la luz incide sobre este elemento hace las veces de corriente de base, razón por la cual la terminal de la base no está en el transistor.

Al inicializar el sistema de alarma, el programa que corre en la PC genera un código de bits que envía permanentemente al bloque sirena, en este bloque conecte un codificador para reconocer a dicho código, de manera que si desconecta o destruye el sistema de alarma, la sirena sonara.

¹ http://www.unicrom.com/Tut_fototransistor.asp, http://www.anzwers.org/free/arsabe/Opto/opto_a.htm.



A los elementos de la segunda propuesta del sistema los ilustro en la *figura 4.7*. En este diagrama indicio los nombres de las terminales de entrada y salida de cada bloque, las líneas señalan la dirección del flujo de los datos y la intercomunicación entre los elementos del sistema.

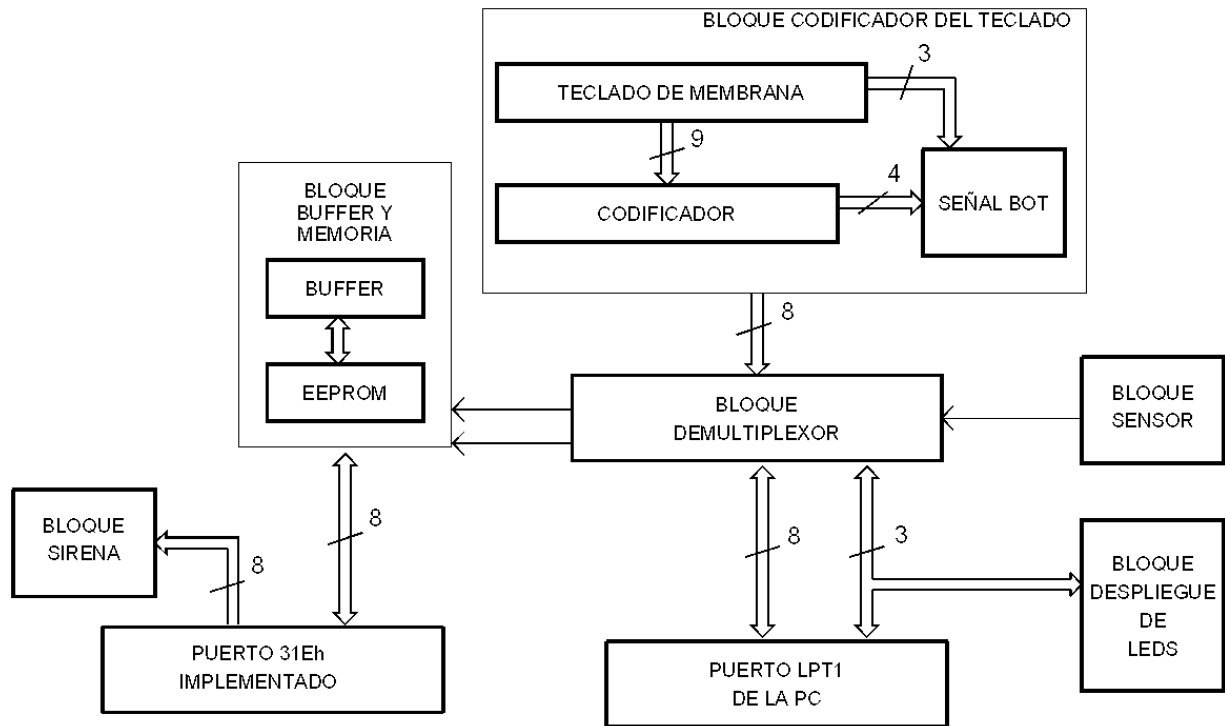


Figura 4.7 Diagrama de bloques detallado de la segunda propuesta.

Resultados de la segunda propuesta

En los siguientes párrafos explico el diagrama electrónico y funcionamiento de los circuitos electrónicos que resultaron a partir de la segunda propuesta.

Así en la *figura 4.8* muestro el diagrama general del hardware del sistema, en el que ilustro a los 7 bloques que integran al sistema con sus respectivas terminales y nombres; la dirección del flujo de datos y la intercomunicación con los puertos paralelos.

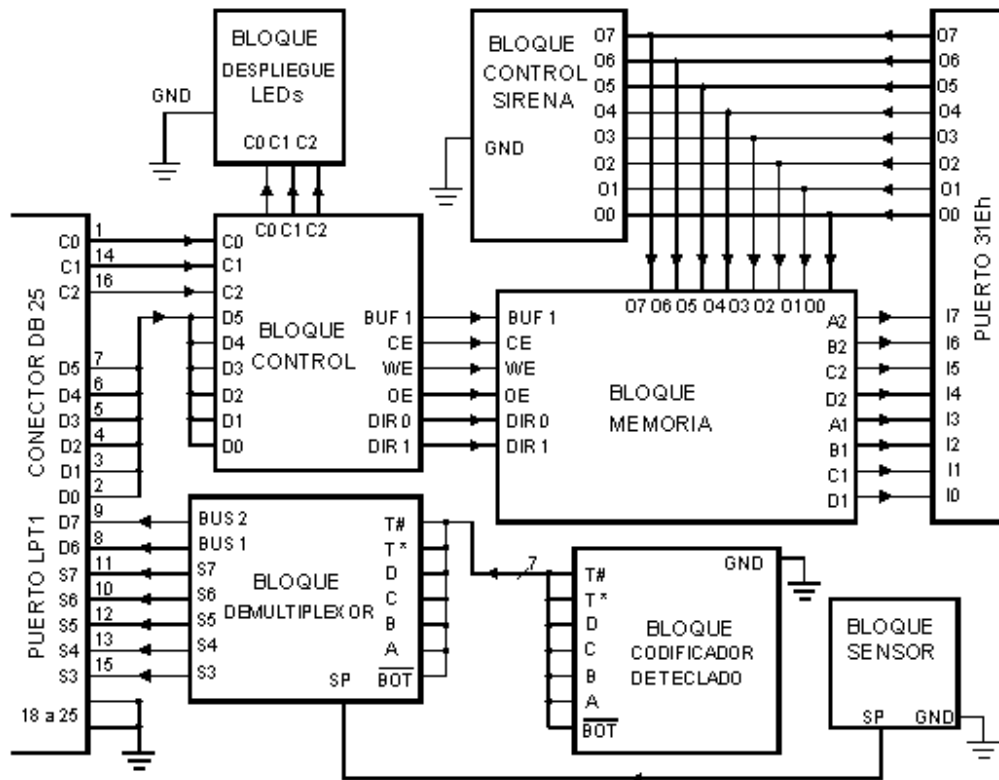


Figura 4.8 Diagrama esquemático del hardware para el sistema.

A continuación hago una descripción de cada bloque.

El bloque codificador de teclado está formado por un teclado de membrana de 4 x 3 líneas, un circuito codificador de 10 a 4 líneas, una compuerta *NAND* de trece entradas y 7 inversores. El teclado está polarizado con una resistencia de 22 kΩ en cada una de sus terminales a *Vcc* y su correspondiente conexión a tierra con un condensador de 2.2μF para reducir los rebotes que se generan en el teclado de membrana.

Los dígitos del 1 al 9 son entradas al codificador, siendo sus cuatro salidas *A*, *B*, *C* y *D*. Cada dígito se codifica como un número binario de cuatro bits, es decir del tipo *BCD*; estas salidas y las terminales del teclado con símbolo “0”, “*” y “#”, son las entradas a la compuerta *NAND*, la cual genera la señal *BOT* que indica cuándo se presionó una tecla.



En el primer diseño fue necesario conectar inversores a las cuatro salidas del codificador así como a las líneas de las teclas “0”, “*”, y “#”, para enlazar directamente dichas salidas a los despliegues de 7 segmentos, debido a la complejidad del alambrado del circuito, decidí conservar los inversores y hacer las adaptaciones convenientes con los bloques relacionados a este, en la *figura 4.9* muestro el diagrama de este bloque.

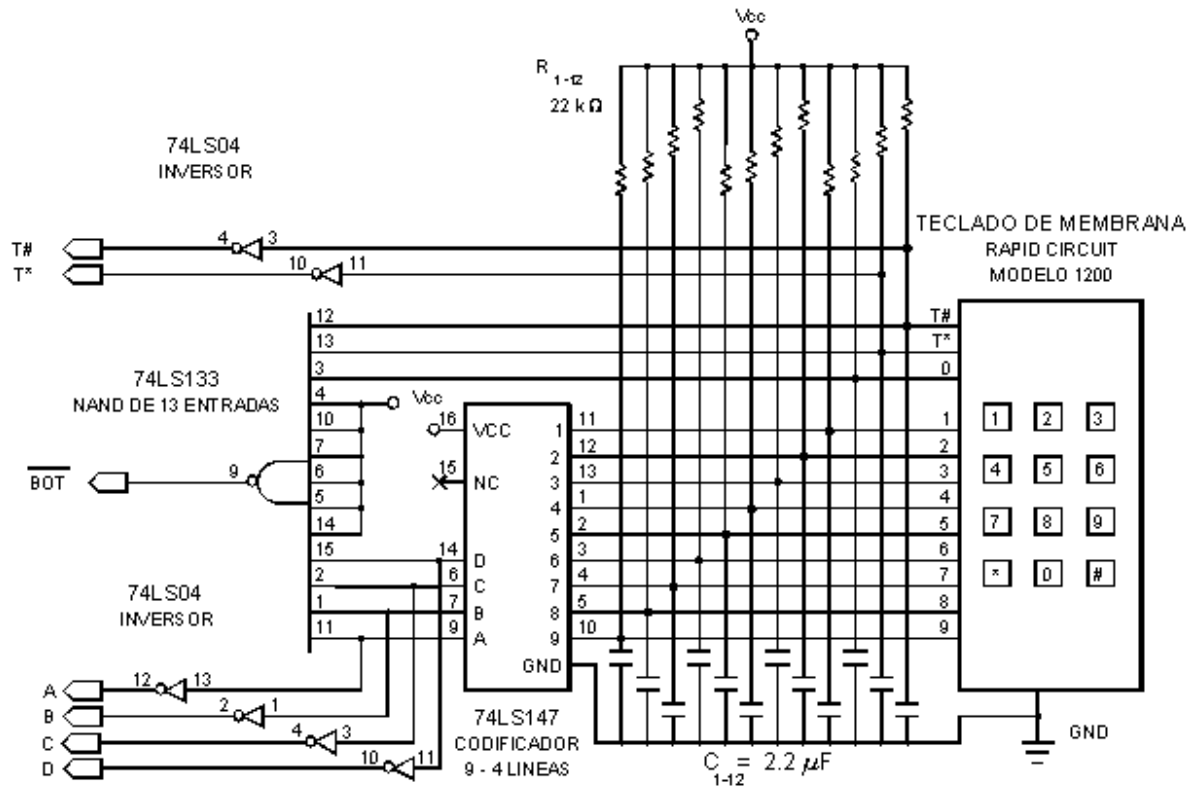


Figura 4.9 Diagrama esquemático del bloque codificador de teclado.

El sistema cuenta con un despliegue de leds formado por cuatro leds que tienen cinco modos de encendido.

En el modo **0** los leds encienden uno a uno de manera secuencial, con un intervalo entre encendidos de $0.5 s$, y que se activa cuando la puerta está cerrada y está aguardando que se ingrese la clave numérica.



En el modo **1** enciende el primer led; en el modo **2** encienden el primero y segundo leds; en el modo **3** encienden el primero, segundo y tercer leds; en el modo **4**, enciende el primero, el segundo, el tercero y el cuarto led; todos los leds encienden de izquierda a derecha y de forma intermitente, con intervalo de encendido/apagado de 0.5 s , y se activan cuando se están introduciendo los dígitos de la clave, ya sea para la apertura de la puerta, o bien, para el proceso de cambio de clave.

El encendido de los leds se controla con una GAL16V8 en la que grabé previamente un programa que generé con base en una máquina de estados. En la *figura 4.10* presento el diagrama del circuito y en el apéndice muestro el diseño de este bloque.

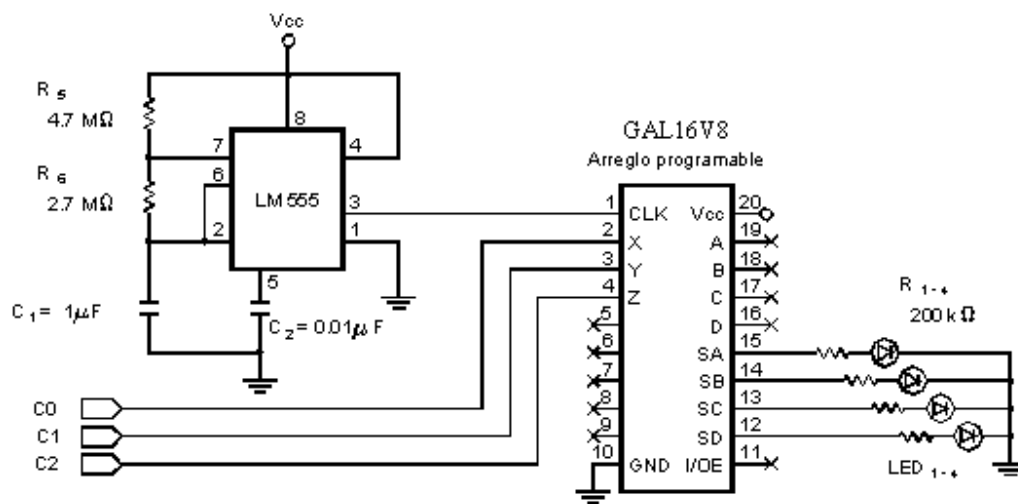


Figura 4.10 Diagrama esquemático del bloque de leds.

Como ya mencioné, la memoria utilizada para guardar la clave es una EEPROM con 8 terminales de entrada/salida, que se conectan al puerto paralelo 31Eh.

El buffer maneja 8 bits y sus salidas son de tipo tres estados, de tal manera que cuando se necesita usar las terminales de la memoria como entradas, por medio del software se hace que el buffer se habilite y lleve sus 8 bits a la memoria.



Cuando las terminales de la memoria se utilizan como salidas, las terminales del buffer se encuentran en alta impedancia. De esta forma se evita que se generen conflictos al interconectar dos salidas a un mismo punto eléctrico.

La memoria tiene 13 terminales de dirección, de las que sólo utilicé dos; considerando que sus combinaciones de cero y uno lógico, pueden generar 4 direcciones diferentes, se pueden guardar los cuatro números de 4 bits que corresponden a cada uno de los dígitos de la clave. Las terminales restantes se conectan a tierra. Las dos terminales para establecer las direcciones así como las terminales *CE* (habilitar memoria), *WE* (habilitar escritura), *OE* (habilitar salidas), *BI* (habilitar buffer 1), son controladas por el programa que corre en la PC.

El circuito correspondiente a este bloque se encuentra en la *figura 4.11* que presento a continuación.

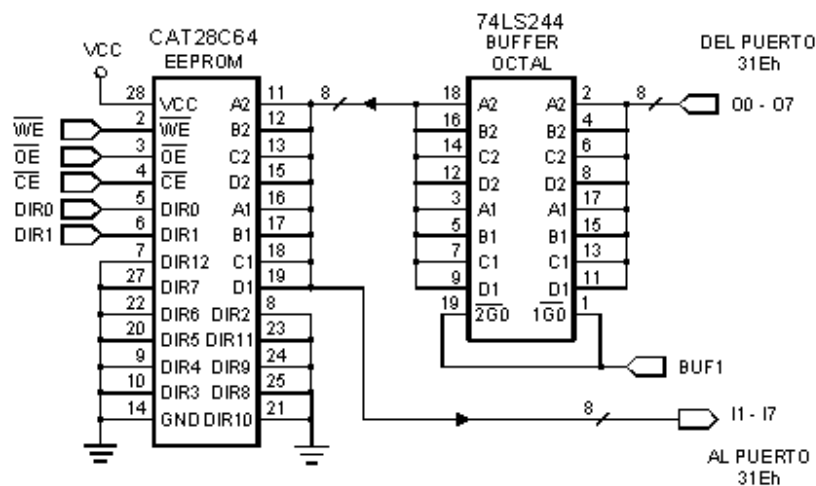


Figura 4.11 Diagrama esquemático del bloque buffer y memoria.

El bloque demultiplexor está integrado por dos buffer octales, sus entradas están conectadas al bloque codificador de teclado, y sus salidas al conector *DB 25*, para llevar la información al registro de status del puerto paralelo *LPT1* de la *PC*. Las terminales de habilitación de estos circuitos las conecte al puerto paralelo 31Eh, en la *figura 4.12* muestro el diagrama de este circuito.

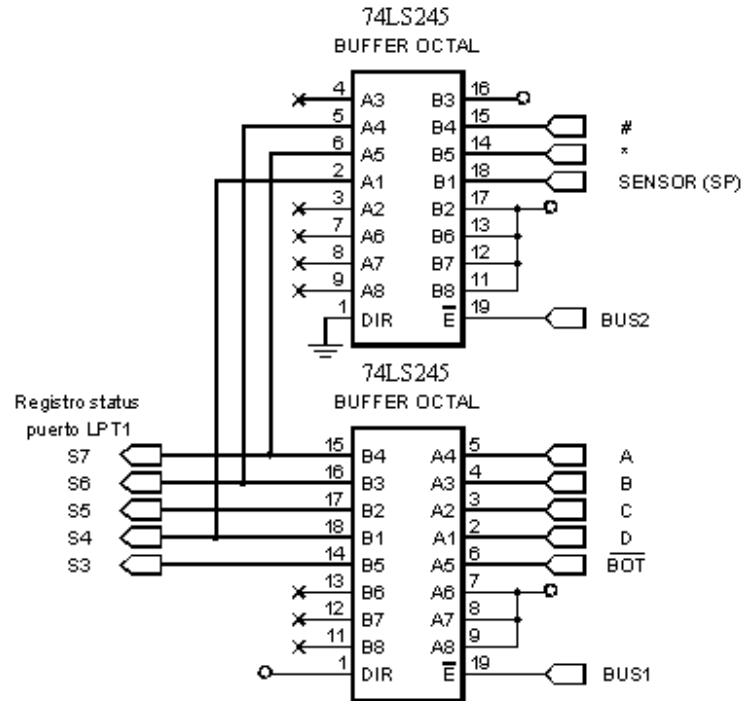


Figura 4.12 Diagrama esquemático del bloque buffer demultiplexor.

El bloque control tiene dos buffer, el primero recibe en sus entradas a los bits que provienen del registro de datos del puerto paralelo *LPT1* y sus salidas son las señales que permiten controlar a la memoria. El segundo buffer tiene cableadas sus entradas al registro de control del puerto paralelo *LPT1* y sus salidas se conectan al bloque de leds. El diagrama esquemático de este circuito esta en la *figura 4.13*.

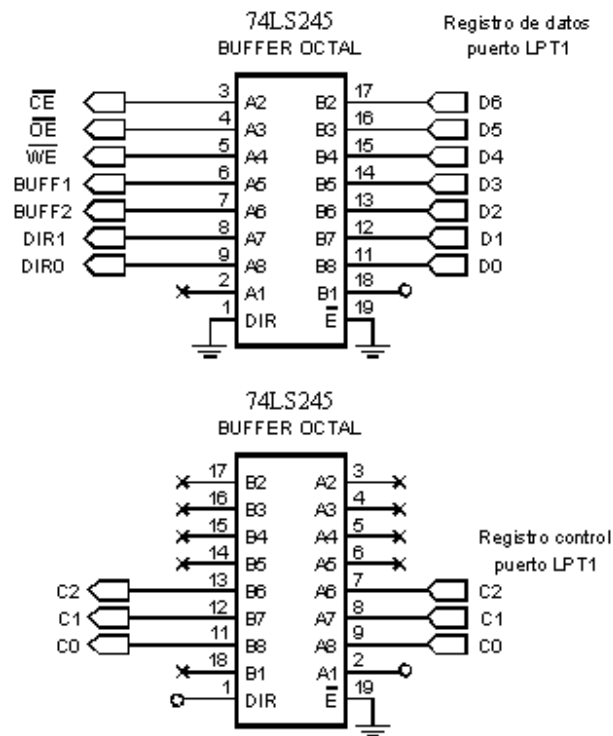


Figura 4.13 Diagrama esquemático del bloque control.

El bloque sensor está integrado por un fototransistor cuya función es verificar el estado de la puerta y devolver el dato leído al bloque demultiplexor y de ahí al puerto *LPT1* para que el software procese dicho dato. Si se requiere aumentar la sensibilidad del transistor a causa de baja iluminación, será necesario incrementar la corriente de la base con ayuda de polarización externa.



En la *figura 4.14* muestro el diagrama esquemático de este circuito.

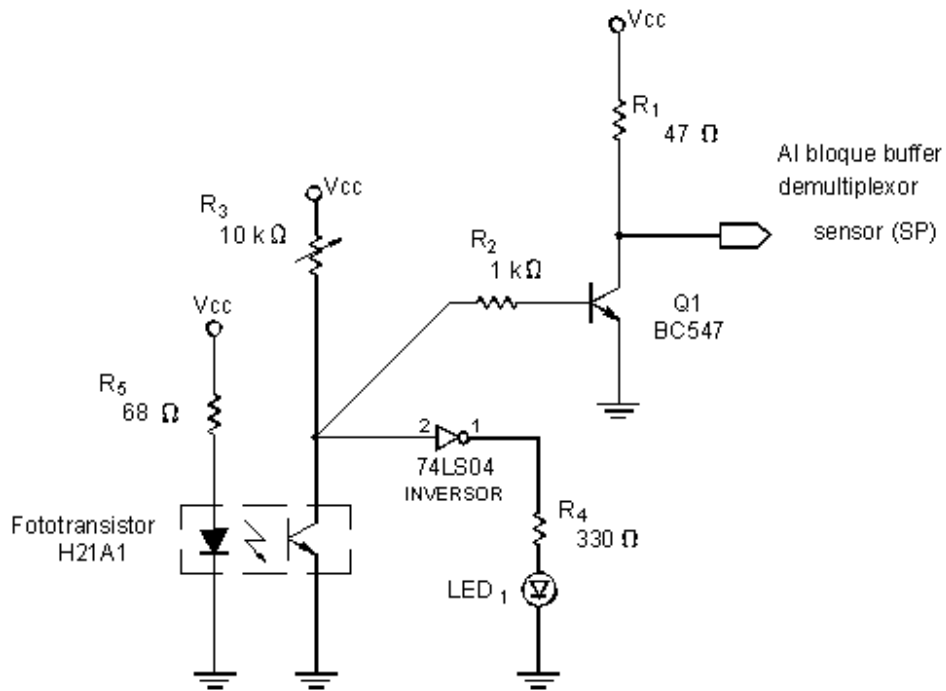


Figura 4.14 Diagrama eléctrico del bloque sensor.

Para el cálculo de las resistencias considere los valores prácticos de diseño de los siguientes elementos:

Diodos

$$I_D = 10 \text{ mA}$$

$$I_{LED} = 10 - 20 \text{ mA (rojos)}$$

$$V_D = 1.5 - 2.2 \text{ V}$$



TBJ (Transistor bipolar de juntura)

Silicio

$$V_{CEsat} = 0.2 \text{ V}$$

$$V_{BE} = 0.7 - 0.8 \text{ V}$$

$$\beta = 100$$

Para las regiones de operación se debe cumplir:

Corte	Amplificación	Saturación
$I_B \leq 0$	$I_B = \frac{I_C}{\beta}$	$I_B \geq \frac{I_C}{\beta}$

TTL (Transistor Transistor Logic, lógica de transistor a transistor)

V_{IL}	V_{IH}	V_{OL}	V_{OH}	Unidades
0.8	2.0	0.4	2.4	V
<small>MÁX</small>	<small>MÍN</small>	<small>MÁX</small>	<small>MÍN</small>	
I_{IL}				
4				mA

El transistor Q_1 debe operar en corte y saturación y a su vez saturar a un transistor Q_2 que esta en la terminal de entrada del TTL al que se envía el estado del sensor, si $R_2 = 1 \text{ K}\Omega$ el valor de I_B se obtiene con la siguiente expresión.

$$I_B = \frac{V_I - V_{BE}}{R_B} = 3.3 \text{ mA}$$

entonces

$$I_B \beta = 330 \text{ mA}$$

Para obtener I_C

$$I_C = I_L + I_{IL}$$

$$\text{Con } R_1 = 330 \Omega$$

$$I_C = \frac{V_{CC}}{R_C} + 4 = 19.5 \text{ mA}$$



Para verificar que el transistor Q_1 y Q_2 en la terminal de entrada del TTL esta en saturación, se debe satisfacer la siguiente expresión.

$$I_C < I_B \beta$$

Para este caso $19.5 \text{ mA} < 300 \text{ mA}$

Para calcular R_3 considere nuevamente el estado de saturación para el fototransistor.

Donde $V_O = V_{CE_{SAT}}$

entonces
$$R = \frac{V}{I} = \frac{V_{CC} - V_{CE_{SAT}}}{I_{IL}} = 1.2 \text{ K}\Omega$$

El valor de I_B en el fototransistor depende de la luminosidad, entre mayor sea el valor de R_3 , el valor de I_B disminuye, entonces R_3 debe permitir desde el mínimo valor de saturación razón por la cual coloque una resistencia variable.

Para asignar el valor de R_4 considere la siguiente expresión.

$$R_4 = \frac{V_{OH} - V_D}{I_{LED}} = 280 \Omega$$

con $V_{OH} = 4.5 \text{ V}$ $I_{LED} = 10 \text{ mA}$

Finalmente para R_5

$$R_5 = \frac{V_{CC} - V_D}{I_D} = 330 \Omega$$

con $V_D = 1.7 \text{ V}$

El bloque control de la sirena se encarga de comunicar a la interfaz del sistema con la sirena, y se activa cuando recibe el código de ocho bits FFh (1111,1111) a través de las salidas O_0 , O_1 , O_2 , O_3 , O_4 , O_5 , O_6 y O_7 del puerto paralelo 31EH, en caso de que detecte que el estado de la puerta es diferente al requerido en ese instante por el programa control.



De esta manera sí se llegara a cortar la conexión con la PC, ya sea por falta de suministro de energía eléctrica o porque se corten los cables, entonces el código FFh llegara al control de la sirena y ésta empezara a sonar.

El diagrama de este circuito lo muestro en la *figura 4.15*.

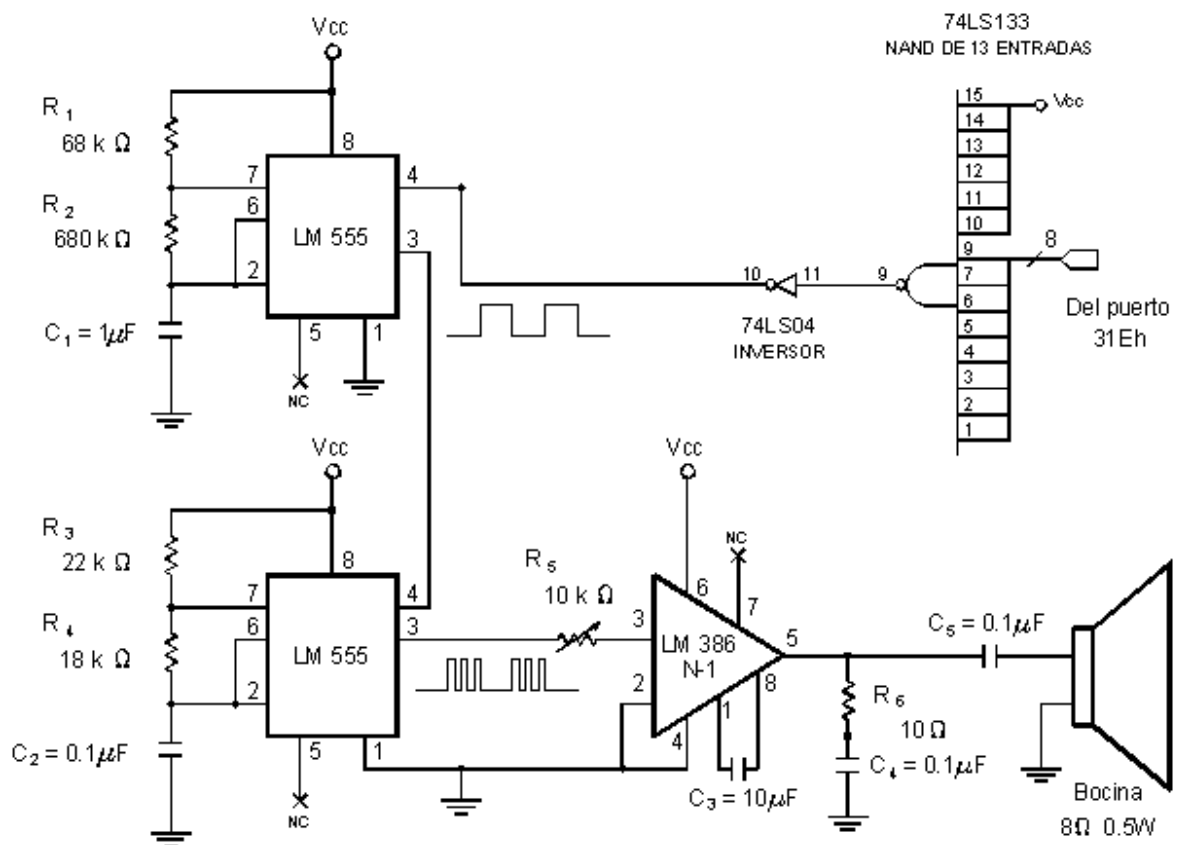


Figura 4.15 Diagrama esquemático del bloque control de sirena.

Este bloque está compuesto por una compuerta NAND para reconocer el código de activación de la sirena; si recibe el código FFh su salida será un “0” lógico. Dicha salida se lleva a un inversor para activar a un par de circuitos LM555 conectados en cascada y modo astable, cuya salida es la



entrada a un amplificador de sonido LM386 con ganancia igual a 200, su señal de salida se lleva a la bocina para generar la alerta sonora.

Para la frecuencia
$$f = \frac{1}{T} = \frac{1.44}{(R_1 + 2R_2)C} \quad (1)$$

Periodo total
$$T = t_1 + t_2 = 0.693(R_A + 2R_B)C \quad (2)$$

Donde
$$t_1 = 0.693(R_1 + R_2)C \quad \text{y} \quad t_2 = 0.693(R_2)C$$

Ciclo de trabajo
$$D = \frac{R_2}{R_1 + R_2} \times 100 \quad (3)$$

Con las ecuaciones 1, 2 y 3, para R_1 y R_2 , obtuve los siguientes valores

$$f = 1.008 \text{ Hz}$$

$$T = 989.604 \text{ ms}$$

con
$$t_1 = 518.364 \text{ ms} \quad \text{y} \quad t_2 = 471.24 \text{ ms}$$

$$D = 73 \%$$

Con las ecuaciones 1, 2 y 3, para R_3 y R_4 , obtuve los siguientes valores

$$f = 248.276 \text{ Hz}$$

$$T = 40.194 \text{ ms}$$

con
$$t_1 = 2.772 \text{ ms} \quad \text{y} \quad t_2 = 1.2474 \text{ ms}$$

$$D = 68 \%$$

Se tiene contemplado el diseño de este controlador de manera que sea capaz de transmitir inalámbricamente el código a la sirena.



A continuación muestro tres fotografías del sistema de alarma implementado.



Figura 4.16 Segunda propuesta para el sistema de alarma.



Figura 4.17 Puerto adicional de 8 bits.

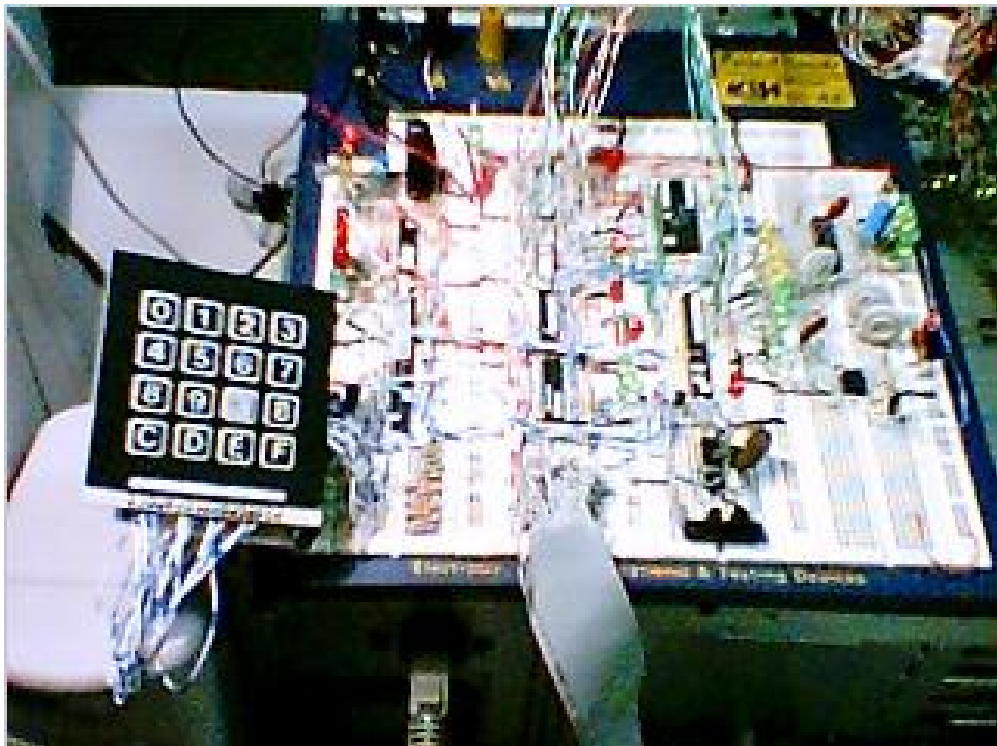


Figura 4.18 Hardware de la segunda propuesta.



4.3 SOFTWARE

El programa que desarrollé en Lenguaje C tiene por objeto controlar la transmisión y recepción de los datos entre la interfaz electrónica y la PC; a continuación describo brevemente las principales partes del programa y su función.

El programa principal está integrado por las siguientes subrutinas.

- Subrutina `determine_time_k`: calcula la constante de tiempo para convertir el valor de un contador de ciclos a tiempo en milisegundos
- Subrutina `delay`: establece retrasos de tiempo para la correcta lectura de los datos en los puertos paralelos
- Subrutina `read_door_sens`: verifica el estado de la puerta abierta/cerrada
- Subrutina `kbd_read`: detecta cuando se oprimió algún carácter del teclado de membrana
- Subrutina `read_code`: lee la clave que se ingresa por medio del bloque teclado
- Subrutina `read_aster_code`: borra la clave y/o continúa el proceso después del ingreso de la clave
- Subrutina `mem_read`: lee la memoria en modo de lectura
- Subrutina `code_compare`: comparar la clave ingresada con la clave grabada en la memoria
- Subrutina `mem_write`: establece el código de salida del registro de datos de manera que se habilite la memoria en modo de escritura
- Subrutina `siren_on`: envía un código de bits al bloque sirena

Para la ejecución del programa se considera que la condición inicial es que la puerta esté cerrada, y sólo en este caso el programa pasará al siguiente paso en el que automáticamente calculará la base de tiempo de la PC en la que está corriendo el programa, a fin de contabilizar los tiempos requeridos correctamente; si la puerta está abierta es necesario cerrarla para que el bloque sensor detecte el estado de la puerta y pueda iniciar el programa.

Una vez realizado lo anterior, el programa de control pasará a un estado de espera, hasta que se introduzca alguna clave numérica mediante el teclado de membrana.

La condición de espera continúa hasta que se detecte el ingreso de un primer dígito, condición necesaria para proporcionar intervalos de menos de cinco segundos entre el ingreso de cada uno de los siguientes tres números, mismos que el usuario le proporcionará.



Si transcurren los 5 segundos y no se ingresa el siguiente número, el programa regresará al estado de espera para detectar nuevamente la clave de acceso.

En total se ingresan cuatro dígitos y la tecla “#”. Si no se cumple este formato, regresa a su estado inicial de espera.

Al cumplirse las condiciones señaladas en el párrafo anterior, el programa procede a leer los cuatro dígitos de la clave pregrabada en la *EEPROM* y comparar si es igual a la introducida por el teclado. En caso negativo, regresa a su estado inicial. Si en cualquier momento intermedio se abre la puerta, esto provocará que el dispositivo habilite al controlador de la sirena, el cual hará que ésta empiece a sonar, indicando que la puerta se abrió de manera inapropiada o violenta.

Si los números introducidos por el usuario y los números grabados en la memoria son iguales, se cuenta con un intervalo de tiempo de quince segundos para abrir la puerta sin que se dispare la alarma. Esta situación obedece a que si el “personal no autorizado a ingresar en las instalaciones de interés” conociese la clave, al no contar con las llaves de las cerraduras de la puerta tardaría más de quince segundos en abrir, lo que causará el encendido de la sirena.

En caso de que se haya abierto la puerta luego de introducir la clave correcta y dentro del lapso de tiempo mencionado, el programa pasa a un estado de espera similar al de inicio, con la salvedad de que en el momento en que se cierre la puerta, el sistema regresa a su estado inicial. Si la puerta permanece abierta, es posible cambiar la clave volviendo a oprimir la clave de acceso seguida de la tecla “#”. Si la clave introducida nuevamente es igual a la de la memoria, se tiene la posibilidad de cambiar la clave rescribiendo el nuevo código en la *EEPROM*, para lo que se requiere introducir correctamente por el teclado de membrana la nueva clave dos veces seguidas, con la finalidad de que el usuario esté bien seguro del código que introdujo al sistema, y se logre disminuir la posibilidad de una equivocación involuntaria. Después de rescribir el nuevo código en la memoria, el sistema pasa al mismo estado de espera hasta que se cierre la puerta, o a que una vez más oprima su clave para un nuevo proceso de cambio de código.



La figura 4.19 corresponde al diagrama de flujo del programa e ilustra lo que describí en los párrafos anteriores.

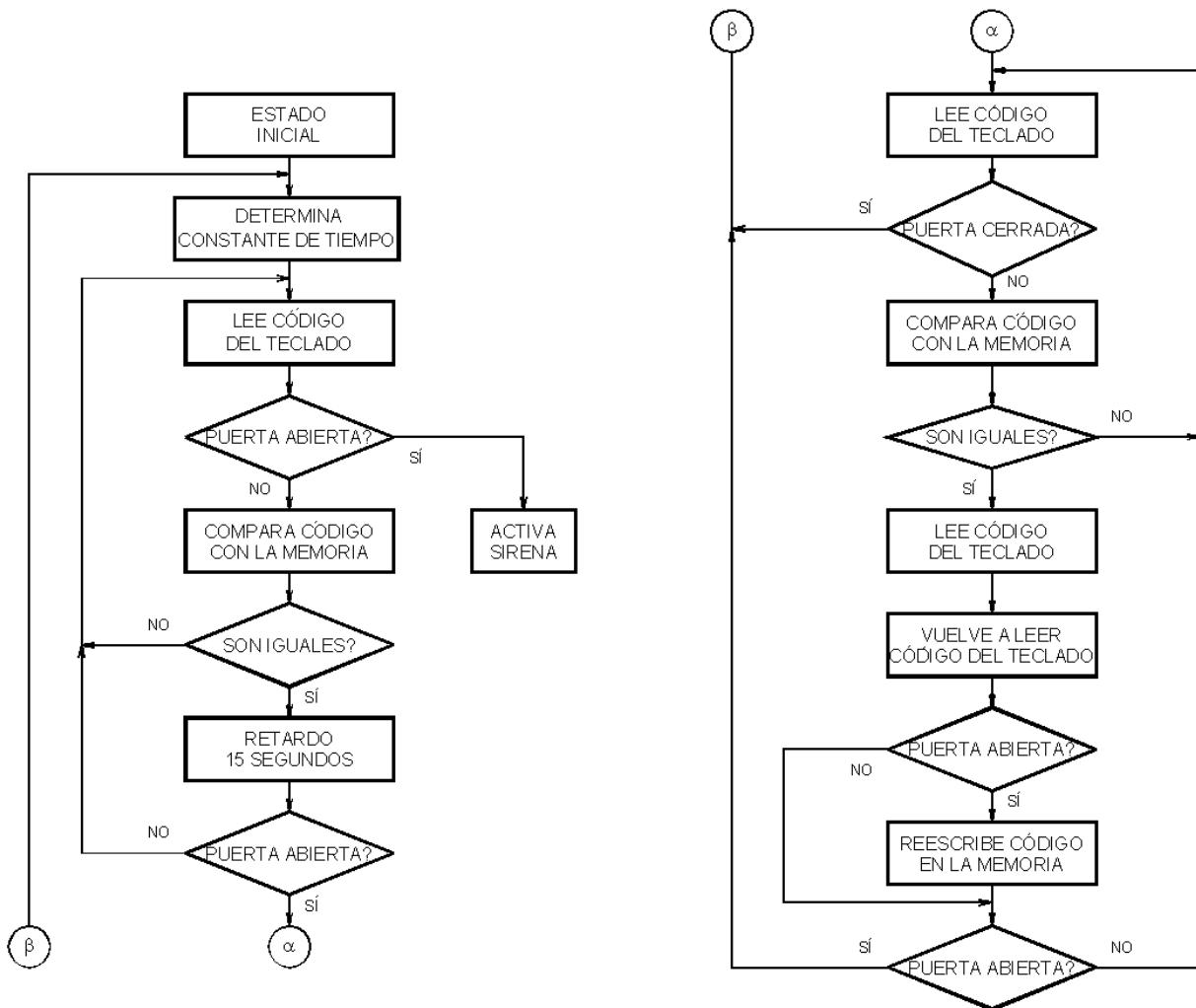


Figura 4.19 Diagrama de flujo del software del sistema.



4.4 COSTOS

Los costos son un aspecto muy importante en este proyecto en términos de su fabricación en serie pensando en su instalación real del sistema de alarma en la División de Ciencias Básicas.

En virtud de esto busqué adquirir los dispositivos indispensables para los circuitos, así como reutilizar las PC's con procesador Intel 80x86, tarjeta de expansión ISA

A continuación presento una lista de dispositivos que utilicé para la implementación de los circuitos finales del sistema de alarma, con su costo comercial en la *tabla 4.1*

Teniendo presente que los precios de los dispositivos varían con el tiempo y lugar de compra, pretendo mostrar un costo representativo del sistema.

Para ser congruente con la descripción de bloques, presento una lista de precios de los elementos empleados, así primero aparece el nombre del bloque, después el concepto, seguido de la cantidad de componentes, su precio por unidad y su precio total de los elementos.



Bloque	Concepto	Cantidad	Precio por unidad (\$)	Total por unidades (\$)
Puerto paralelo LPT1	Conector DB - 25	1	25	25
	Cable plano 26 hilos	1	12 x metro	12
Bloque memoria	EEPROM CAT28C64	1	100	100
	74LS244	2	6	12
Bloque demultiplexor	74LS245	1	7	28
Bloque despliegue de leds	LM555	4	1	4
	GAL16V8	1	16	16
	Resistores	3	0.5	1.5
	Condensadores	2	2	4
	Resistor variable	2	6	12
	Leds	3	4	12
Bloque codificador de teclado	Teclado membrana 4X3	1	30	30
	Resistores	12	0.5	6
	Condensadores	12	2	24
	74LS147	1	8	8
	74LS133	1	8	8
	74LS04	2	4	8
Bloque sensor	Resistores	5	0.5	2.5
	TBJ 547C	1	5	5
	Resistor variable	1	6	12
	Sensor H21A1	1	20	20



Bloque	Concepto	Cantidad	Precio por unidad (\$)	Total por unidades (\$)
Puerto paralelo 31Eh	74LS133	1	8	8
	74LS32	1	4	4
	74LS04	1	4	4
	74LS245	1	7	7
	74LS374	1	7	7
	Resistores	8	0.5	4
	Leds	4	4	16
Bloque sirena	LM555	2	4	8
	LM386	1	5	5
	74LS133	1	8	8
	74LS04	1	4	4
	Condensadores	5	2	10
	Resistores	5	0.5	2.5
	Resistor variable	1	15	15
	Bocina 8Ω a 0.5W	1	25	25

Tabla 4.1 Lista de precios de dispositivos utilizados.

De la tabla anterior se puede verificar que el costo total de los dispositivos listados es de \$477.50 M.N.

El precio de una sirena varía desde \$180.00, en tanto que las fuentes de alimentación para caso de interrupción en el suministro eléctrico se encuentran desde \$900.00, este precio es de una batería para alimentar a dos computadoras como máximo.



CAPÍTULO 5

RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

5.1 RESULTADOS

Para llegar al actual sistema de alarma, fue necesario pasar por los diseños de hardware y software, comentados en el capítulo anterior. Durante el proceso de diseño del sistema tuve diversas dificultades que impedían su buen funcionamiento; los principales problemas identificados con sus respectivas soluciones son los que describo a continuación.

Para el diseño del bloque despliegue de leds elaboré, el diagrama de estados y mapas de Karnaugh considerando la secuencia de encendido que debían seguir los leds, y luego de obtener las ecuaciones booleanas del circuito realicé un programa para grabar la GAL16V8 con el programa *OPAL* con el cual generé archivos tanto de simulación como para la grabación de dicho programa en el dispositivo.

Los problemas que tuve al probar el circuito fueron básicamente de polarización. De las hojas de especificaciones se sabe que el voltaje de polarización es de 5 V; en una primera prueba alimenté el circuito con la misma fuente que se polariza al resto del hardware, esto fue con 4.5 V debido a la caída de voltaje, lo que mantenía a los leds encendidos pero no realizaba los cambios programados. Verifiqué nuevamente la simulación, y el diseño de diagramas de estados así como las ecuaciones, después agregué al circuito un regulador de voltaje y le coloqué su propia fuente de alimentación, al probarlo el circuito encendía a los leds de la manera programada pero en la secuencia se perdía un pulso ocasionando que el apagado simultáneo de los leds 3 y 4 no fuera



correcto; este problema lo corregí al llevar todas las terminales Vcc a un mismo punto y todas las tierras a otro punto común; por lo tanto, para que la GAL funcione correctamente necesita ser alimentada estrictamente con el voltaje especificado por el fabricante.

Los problemas que tuve en el desarrollo del software eran referentes a la estructura de programación; al principio tenía gran cantidad de ciclos e instrucciones repetitivas, lo que lo hizo extenso y tedioso, por lo que modifiqué la estructura del programa mediante subrutinas, y en consecuencia mejoré la calidad de programación y funcionamiento.

El crear subrutinas y una estructura nueva de funcionamiento me permitió disminuir parte de los circuitos involucrados; fue una tarea difícil adecuar el hardware a este nuevo software, dado que para solucionar este problema desarrollé programas individuales que permitieran el correcto funcionamiento del sistema, en especial el control para la memoria EEPROM previo a eliminar los bloques sobrantes.

El programa de control tiene dos partes fundamentales: la primera requiere como condición de entrada que la puerta esté cerrada; siendo de este modo, el bloque sensor envía a la PC un “0” lógico; en la segunda parte del programa se establece como condición de entrada que el estado de la puerta sea un “1” lógico, en caso de desear entrar en el proceso de cambio de clave.

Por error no consideré adecuadamente dichas condiciones en las subrutinas que permiten leer el teclado, lo que provocó que sólo en la primera parte se pudiera borrar la clave en caso de error, mientras que en la segunda parte, sólo podía borrar dígito por dígito. Este problema lo resolví rastreando el valor que tomaban las variables en cada instrucción dentro de la subrutina kb_read y kb1_read, e incluí esta consideración en la subrutina aster_code.

Para lograr que el programa vuelva a su estado inicial de espera en caso de interrumpir el proceso de la introducción de la clave de ingreso o cambio de clave, fue necesario intercalar a lo largo del programa la subrutina que verifica continuamente el estado de la puerta mediante el sensor optoelectrónico.

Cuando ingresaba los dígitos de la clave, se generaban errores al procesar los datos en el programa, esto a causa de los rebotes que se producen en el teclado de membrana; para eliminar este problema incorporé subrutinas de retardo de tiempo, con objeto de que los bits que lleguen a los puertos sean interpretados correctamente.



5.2 CONCLUSIONES

El diseño de un sistema de alarma contra robos para cubículos de académicos universitarios fue una actividad sumamente enriquecedora, en la que pude aplicar los conocimientos adquiridos sobre circuitos lógicos, tanto combinacionales como secuenciales, además de incluir ampliamente el conocimiento de la conformación y el manejo de puertos paralelos de las PC's comerciales.

Además, me proporciono la visión necesaria para ubicar en mi mente la función de la ingeniería como la resolución de problemas de la comunidad, redundando en mejores condiciones de vida para la sociedad.

Conocer el manejo de entrada y salida de datos al puerto paralelo brinda la posibilidad de controlar mecanismos y circuitos electrónicos. Los principales problemas que se presentaron durante la puesta en operación del puerto paralelo LPT1, del puerto propio de 8 bits así como del sistema, fueron por la falta de cuidado cuando se interconectan entre sí dos o más salidas, y que puede causar conflictos en la transmisión de datos.

En este circuito se incluyó intensivamente el uso los puertos paralelos de una *PC* comercial, lo que demuestra que reutilizar equipo de cómputo que está en desuso como una nueva opción para la implementación de sistemas permite obtener soluciones de bajo costo y buena calidad. Por ello, uno de los objetivos del presente trabajo es difundir el diseño de soluciones de control electrónico de dispositivos con la convicción de aprovechar equipo obsoleto en condiciones de operación. De este modo, se contribuye a la preservación del medio ambiente y a la economía de las instituciones educativas o empresas.

El control de acceso que utiliza una clave de cuatro dígitos introducida por medio de un teclado, podría ser modificado por uno que incluya un lector de código de barras, o bien un lector magnético. Incluso podría pensarse en construir un sistema que se basara en reconocimiento de imágenes y/o de voz.

Espero que la propuesta final de este trabajo pueda servir para mostrar a las autoridades de la Facultad de Ingeniería de la UNAM las bondades del sistema diseñado, a fin de que se le tome en cuenta como solución factible para su instalación en varios de los cubículos de los profesores.



Contribuciones del proyecto

El sistema de alarma desarrollado en este trabajo, al ser instalado en los cubículos de los profesores, les podrá ofrecer una mayor tranquilidad y seguridad personal, lo que se reflejará en un mejor desempeño en su actividad académica. La clave de acceso puede compartirse con su personal de confianza, y puede modificarse en el momento en que el usuario así lo considere conveniente, sin necesidad de recurrir a los diseñadores.

Este dispositivo permite eliminar el uso de otro tipo de protecciones, tales como rejas o puertas adicionales, que no sólo desarmonizan con el ambiente de la Universidad, sino que en situaciones de emergencia o pánico, ponen en riesgo la vida de las personas que se encuentran en el interior de las instalaciones.

5.3 RECOMENDACIONES

Como complemento a futuro para hacer más eficiente al sistema de alarma propuesto, se recomienda contar con una cámara de video y un módulo de control, como se ilustra a continuación.

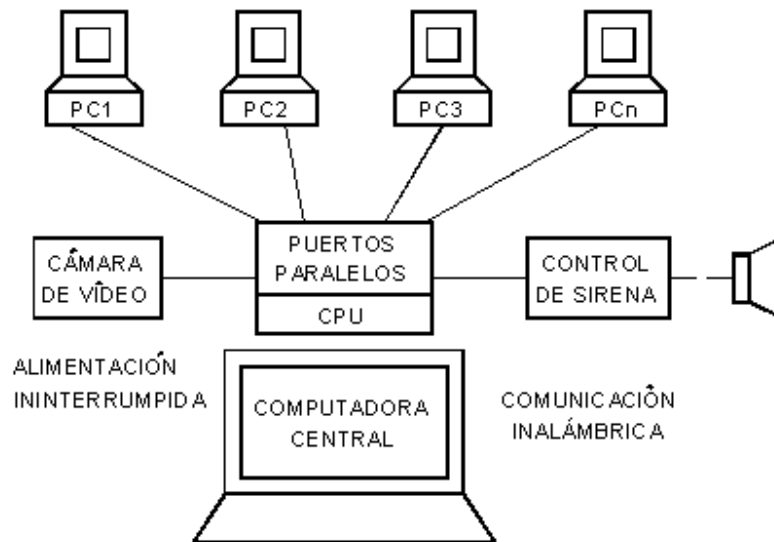


Figura 5.1 Diagrama de bloques en el que muestro los elementos que podrían añadirse para mejorar el funcionamiento del sistema propuesto.



La cámara de vídeo que se elija para complementar el sistema, deberá iniciar la grabación de imágenes en caso de que se detecte la presencia de personas no autorizadas dentro de las instalaciones protegidas en horas no laborables.

La cámara que se podría recomendar para esta parte es una webcam con las siguientes características:

Modelo: PC CAM350

Marca : Creative

Precio de Lista: \$1,150.00 pesos M.N.

Sirve para tomar fotos digitales instantáneas durante el día o la noche, tiene flash con estroboscopio y 8 MB de memoria. Tiene capacidades extensivas para tomar 76 fotos instantáneas con una resolución de 640x480 pixeles o bien, 220 fotos con una resolución de 352x288 pixeles y 498 fotos con una resolución de 176x144 pixeles, 8 MB PhotoSafe de memoria mantienen las imágenes seguras en caso de que la batería se agote, tiene un sensor CMOS 640x480 pixeles, pantalla LCD, puerto USB, flash con modo automático y reducción de ojos rojos y no se requieren fuentes de alimentación por separado.

La PC a la se que conecta la cámara debe contar al menos con un procesador de 233 MHz, sistema operativo Windows 98/00/Me/XP, memoria 64 MB, CD-Rom 4X y un puerto USB

Cabe señalar que la PC CAM puede ser sustituida por una cámara infrarroja, para efectos de mejor confiabilidad en la captación de imágenes y robustez del sistema. A continuación listo algunas de estas cámaras con sus respectivas características.

- Cámara infrarroja

La imagen puede ser desplegada en cualquier equipo con entradas RCA como: televisores, sistemas de vídeo VHS, y PC con captura de vídeo entre otras. 320 líneas de resolución, 0.1 lx, aproximadamente 1 metro de visión bajo oscuridad total, requiere una alimentación de 12 V y entre 100 mA y 300 mA.



- Cámara infrarroja espía

12 V, opera en total oscuridad, 350 líneas de resolución, 0.0 lx, auto iris, 1/60 a 1/15000 velocidad, foco ajustable, lentes de cristal.

- Infrarroja CCD Sony

Con 420 líneas de resolución, 0 lx, lentes de 36 mm, sirve para día y noche, resistencia al agua.

Si se considera que en cada cubículo de profesor se instalará una cerradura electrónica, y que cada una de éstas utilizará una *PC*, será conveniente contar con una computadora central o servidor, que verifique continuamente que el sistema está funcionando correctamente, mediante una red local.

Conviene que la sirena del sistema se comunique de forma inalámbrica con la computadora central para impedir que esta sea desactivada al cortar la comunicación por cable.

Para evitar que el sistema deje de funcionar por fallas en el suministro eléctrico, se recomienda conectar todos los dispositivos activos del sistema a fuentes ininterrumpidas de poder.



APÉNDICE

A.1 BIOMETRICA

“ La biométrica¹ es una ciencia relativamente nueva cuyo objetivo es lograr que las computadoras sean capaces de reconocer con certeza a cada persona con base en sus características únicas, como la huella digital, las facciones o señas de sus ojos; existen varios tipos de mediciones para la biométrica entre los cuales destaca los siguientes.

- Reconocimiento facial: Identifica a personas por medio del análisis de rasgos fáciles que no son fácilmente alterables, como el contorno superior de las cavidades oculares, el área alrededor de los pómulos y los extremos de la boca.
- Huellas digitales: Este es uno de los sistemas más conocidos y utilizados, esta tecnología analiza la imagen de la huella digital y analiza ciertos puntos clave de la huella para compararlos con huellas conocidas.
- Geometría de la mano: Este sistema toma 96 mediciones de la mano, incluyendo su anchura, largo, largo de cada dedo, distancia entre articulaciones y forma de los nudillos. La forma de la mano no cambia durante la vida de un adulto, de modo que éste es un sistema muy confiable.
- Reconocimiento del iris: En esta técnica, se fotografía el área de color que rodea a la pupila del ojo. El iris es una buena fuente de información biométrica, pues contiene una 266

¹ José Antonio Chávez. Pase por el escáner, sección A, Interfase, p. 1, Lunes 9 de febrero de 2004.



características distintivas. Las características del iris no varían durante toda la vida del individuo.

- Reconocimiento de la retina: Este sistema fotografía el pequeño nervio del fondo del ojo que procesa la luz que entra. Cada retina tiene un arreglo único de vasos sanguíneos, e incluso las retinas de gemelos idénticos son distintas. La retina, sin embargo, puede ser alterada por algunas enfermedades. Además, dado que la retina es pequeña e interna, es más difícil de implementar estos sistemas, pues el individuo debe acercarse mucho al lector y permanecer quieto más de un minuto, mientras se fotografía su retina.
- Reconocimiento de firma: En esta técnica se autentifica la identidad midiendo la manera en que la persona realiza su firma. El sistema mide el ritmo de movimiento, la aceleración de la pluma y la presión.
- Reconocimiento de voz: El sistema compara la voz de un sujeto, que habla en un micrófono, con la almacenada en un banco de datos, usando características como tono, cadencia y gravedad. Es especialmente útil cuando la voz es el único dato disponible, como en llamadas telefónicas.

La aplicación preferida de la biométrica es la seguridad; muchas empresas utilizan estos sistemas para restringir el acceso a ciertas áreas, y para verificar la asistencia de sus empleados.

Un estudio realizado por el NIST (Instituto Nacional de Estándares y Tecnología de Estados Unidos de América) a finales del año 2002 afirma que: “Nuestros sistemas indican que un sistema biométrico dual, que incluya dos huellas digitales y una imagen facial puede ser necesario para alcanzar los requerimientos de un sistema útil ”; de acuerdo con el análisis del Instituto, la medición con una sola huella digital es confiable en un 85 por ciento, con un nivel de falsos positivos de uno por ciento, si la base de datos contra la que se compara tiene 100 mil registros. Adicionalmente, el dos por ciento de los usuarios tiene las huellas digitales demasiado dañadas. Por ello, es necesario agregar el reconocimiento facial, que ofrece niveles de confiabilidad, de acuerdo con el estudio, del 77 por ciento.

Esta tecnología tan probada no está exenta de problemas. De acuerdo con los trabajos publicados por Tsutomu Matsumoto, profesor de criptografía y matemáticas en la Universidad Nacional de



Yokohama de Japón, todos los lectores actuales de huellas digitales pueden ser engañados de manera relativamente simple.

La tecnología biométrica ya se encuentra disponible en el mercado mexicano, principalmente en aplicaciones de lectura de huella digital, y de hecho ya se dirige incluso a los hogares, afirmó Delio Montalvo, director general de Secuone de México.

Además de empresas que ya utilizan lectores de huellas digitales, también en las casas pueden colocarse cerraduras electrónicas, que permiten el acceso solamente a quienes tienen huellas digitales registradas en el sistema.

Monclavo aclaró que son productos listos para ser instalados incluso por el mismo usuario, las cerraduras tienen un costo desde los 550 dólares y son a prueba de problemas; prácticamente todas tienen algún sistema de respaldo para el caso de que haya problemas con el lector, que puede ser una llave o un código numérico”.

En las figuras 1.1, 1.2, 1.3 y 1.4 muestro ejemplos de los sistemas de reconocimiento basado en la biométrica.²



Figura 1.1 Lector de huella digital.

² Las figuras mostradas en este capítulo son tomadas del periódico Reforma y tienen la siguiente referencia. José Antonio Chávez. Llega biométrica a México, sección A, Interfase, p. 6, Lunes 9 de febrero de 2004.



Figura 1.2 Cerradura con lector de huella digital y/o clave numérica.



Figura 1.3 Lector digital en combinación con tarjeta inteligente.



Figura 1.4 Cerradura con reconocimiento de huella digital.



A.2 ROBOS EN LA FACULTAD DE CIENCIAS

“ La Facultad de Ciencias de la UNAM³ ha padecido en el último año más de 50 robos de equipo de cómputo y de laboratorio, lo que ha afectado directamente el trabajo de 30 investigadores. Consultados al respecto, los investigadores aseguran que el problema se remonta a los últimos 12 meses, en los que han sido robados 38 computadoras y 12 microscopios, varios especializados, como el espectrográfico, con un valor cercano a los 200 mil pesos.

El problema se ha vuelto tan grave que 20 doctores y ocho maestros en ciencias están dispuestos a realizar por las noches rondines en la facultad. “ Queremos llamar la atención de las autoridades para resolver el problema”, dijo Lazcano. “Estamos dispuestos a convertirnos en una especie de veladores con posgrado”.

José Antonio Vela, secretario de Servicios a la Comunidad Universitaria, señaló que, si bien los robos aislados siempre han existido, esta situación se ha agudizado en el último mes.

“Por ello se aprobó un presupuesto de un millón 900 mil pesos para comprar un sistema de seguridad”, informó.

Dicho sistema –que será adquirido a través de la administración central de la UNAM- se encuentra actualmente en licitación, y quizá tarde varias semanas o meses en empezar a operar.

Según Francisco Javier García, secretario administrativo de la facultad, se buscará frenar los robos con la adquisición de un sistema de vigilancia por circuito cerrado y la instalación de nuevas cerraduras. Además, los vigilantes de la facultad ya han sido alertados y serán capacitados para manejar el nuevo equipo.

Hasta ahora, las autoridades universitarias ignoran el número de denuncias levantadas en los últimos meses y el curso de las investigaciones.

Víctor Velasco, del Laboratorio de Genética consideró que los robos no tienen un solo modus operandi; algunos son “de ocasión”, pero todos van dirigidos ya que se han llevado equipo que “no cualquiera puede vender”, como los microscopios.

³ Artículo tomado del periódico Reforma, Arturo Barba y Antimio Cruz. Pegan robos en UNAM a proyectos científicos, sección C, Cultura, p. 6, Viernes 20 de febrero de 2004.



Las sustracciones han tenido lugar a toda hora: los fines de semana, en vacaciones, en días hábiles, por la noche, e incluso a plena luz del día.

Raúl Gómez, jefe del Laboratorio de Física Atómica y molecular, fue una de las víctimas. Hace tres semanas le robaron un procesador que contenía información de varios proyectos de investigación que realizaba desde hace meses.

En este mismo laboratorio se robaron una impresora láser y una cámara de vídeo, afectando trabajos en el área de espectroscopia, superconductores y análisis de meteoritos.

También ha habido hurtos en los laboratorios de análisis numérico, fisiología animal, ficología, palobiología, el museo de zoología y aulas de enseñanza de la biología.

Según Lazcano, esto ha provocado un ambiente de paranoia, ya que los investigadores temen por sus proyectos y su propia persona”.

A.3 DISEÑO DEL BLOQUE DESPLIEGUE DE LEDS

Para la implementación del despliegue leds diseñe un circuito lógico que enciende un grupo de leds en los modos que a continuación explico.

- Modo 0: los leds encienden uno a uno de manera secuencial, con un intervalo de 0.5 s; la secuencia se activa cuando la puerta está cerrada y aguarda a que se ingrese la clave numérica.
- Modo 1: enciende el primero led
- Modo 2: enciende el primero y segundo leds
- Modo 3: enciende el primero, segundo y tercer leds
- Modo 4: enciende el primero, segundo, tercero y cuarto leds, con un intervalo de encendido/apagado de 0.5 s; se activan cuando se van introduciendo los dígitos de la clave.

El diagrama de estados que utilicé para la resolución del problema planteado, lo muestro en la *figura A.1*

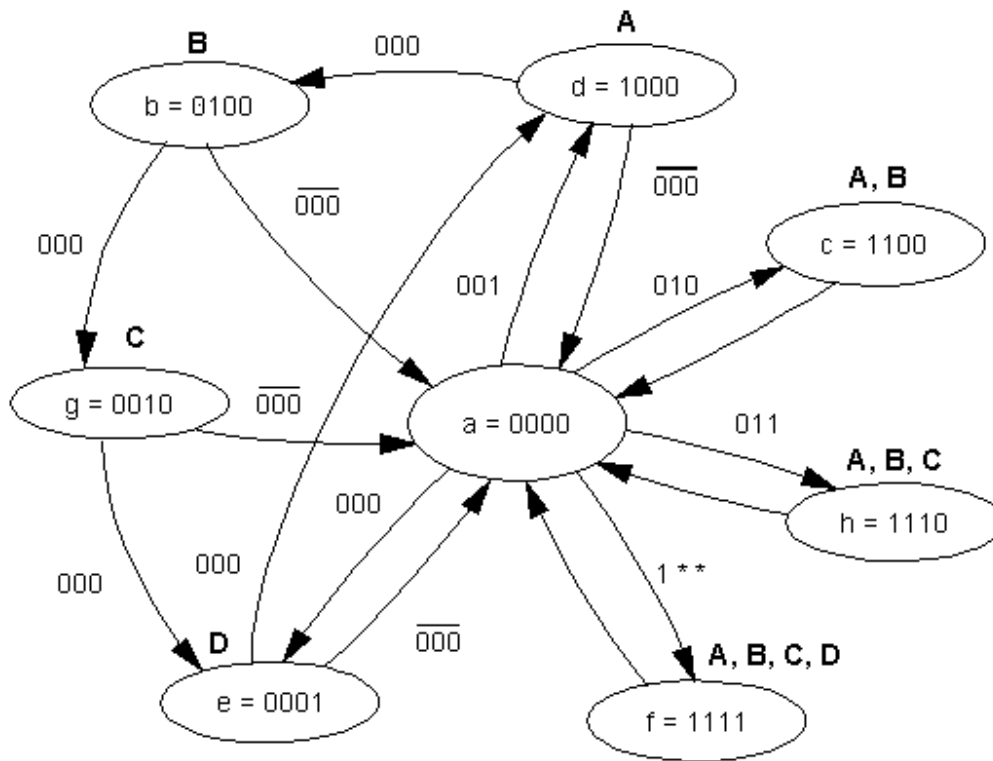


Figura A.1 Diagrama de estados para control del bloque de leds.

Con base en el análisis de este diagrama determiné los siguientes mapas de Karnaugh y obtuve sus respectivas ecuaciones de estado.

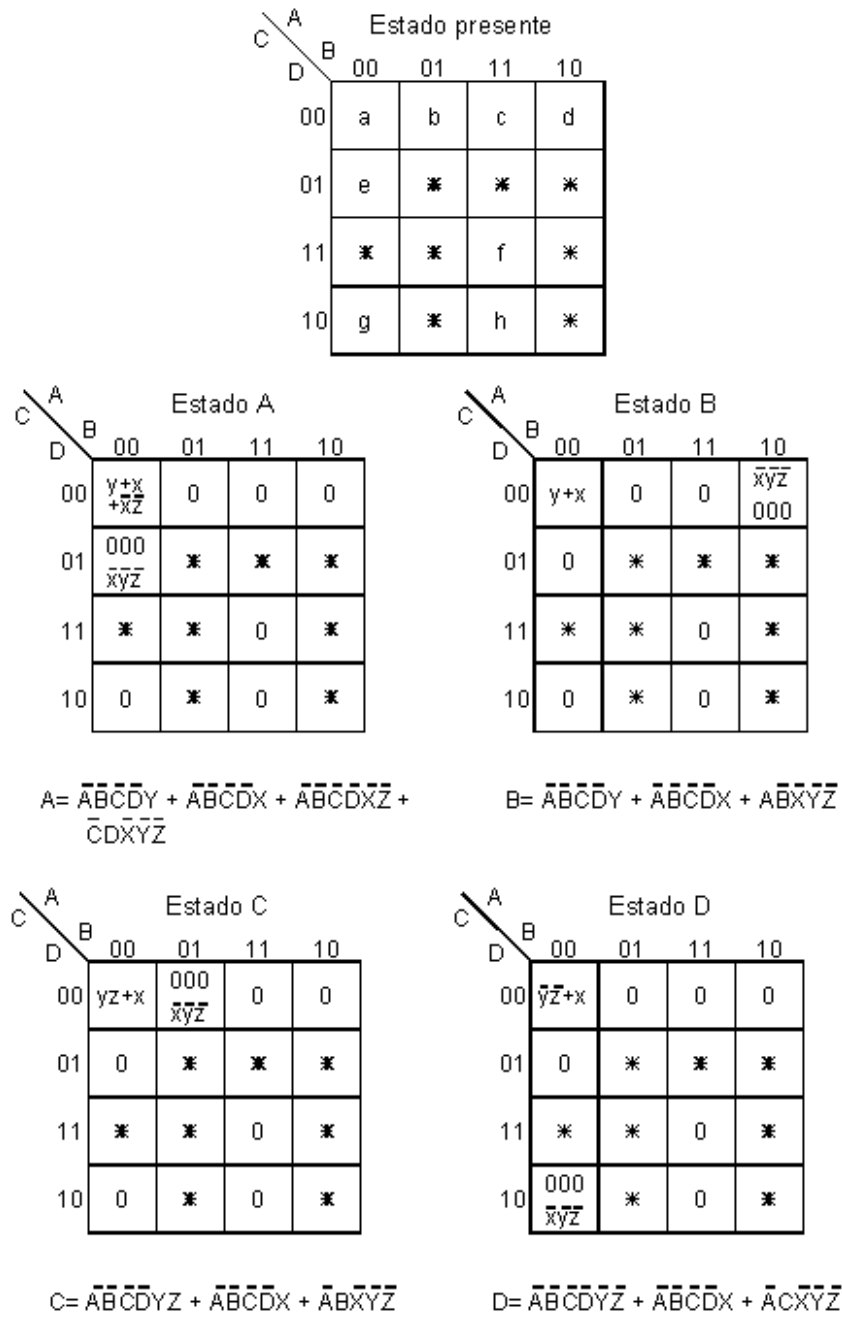


Figura A.2 Mapas y ecuaciones de estado para el control de bloque de leds.



A.4 SOFTWARE DE PROGRAMACIÓN DE LA GAL16V8

BEGIN HEADER

Programa para implementar un control de *leds*, con cinco modos de encendido diseñado en GAL 16V8.

Circuito: Bloque de *leds* para un control de acceso

Autor: Teresa Rosas

Fecha: -diciembre de 2002

END HEADER

BEGIN DEFINITION

DEVICE GAL16V8;

INPUTS CLK = 1, X =2 , Y =3, Z =4, OE =11;

OUTPUTS SA = 15, SB =14, SC =13, SD =12;

BEGIN EQUATIONS

$$A = /A * /B * /C * /D * Y + /A * /B * /C * /D * X + /A * /B * /C * /D * /X * /Z + /C * D * /X * /Y * /Z;$$
$$B = /A * /B * /C * /D * Y + /A * /B * /C * /D * X + A * /B * /X * /Y * /Z;$$
$$C = /A * /B * /C * /D * Y * Z + /A * /B * /C * /D * X + A * B * /X * /Y * /Z;$$
$$D = /A * /B * /C * /D * /Y * /Z + /A * /B * /C * /D * X + A * C * /X * /Y * /Z;$$

SA = A;

SB = B;

SC = C;

SD = D;

END EQUATIONS



BEGIN VECTOR

CLK, X, Y, Z, OE

```
0 0 0 0 0
1 0 0 0 0
0 0 0 0 0
1 0 0 0 0
0 0 0 0 0
1 0 0 1 0
0 0 0 1 0
1 0 0 1 0
0 0 0 1 0
1 0 0 1 0
0 0 0 1 0
1 0 1 0 0
0 0 1 0 0
1 0 1 0 0
0 0 1 0 0
1 0 1 0 0
0 0 1 1 0
1 0 1 1 0
0 0 1 1 0
1 0 1 1 0
0 0 1 1 0
1 1 0 0 0
0 1 0 0 0
1 1 0 0 0
0 1 0 0 0
1 1 0 0 0
0 1 0 1 0
1 1 0 1 0
0 1 0 1 0
1 1 0 1 0
0 1 0 1 0
```



1 1 1 0 0
0 1 1 0 0
1 1 1 0 0
0 1 1 0 0
1 1 1 0 0
0 1 1 1 0
1 1 1 1 0
0 1 1 1 0
1 1 1 1 0
0 1 1 1 0

END VECTOR



BIBLIOGRAFÍA

- 1 Axelson, J, *How to use a PC's parallel port for monitoring and control purposes*, Microcomputer Journal, (1994).
- 2 Brey, B *The Intel Microprocessors 8086/8088, 80286, 80386 and 80486, Architecture, Programming and Interfacing*, Merrill, USA, 452 pp. (1991).
- 3 Eggebrecht Lewis C. *Interfacing to the IBM personal computer*. Howard W, Sams & Company, USA, 246 pp. (1988).
- 4 Gottfried, B, S, *Programación en C*, Mc Graw-Hill, México, p. 590, (1996).
- 5 Kernighan, B, W, & Ritchie, D, M, *Lenguaje de Programación C*, Prentice Hall Hispanoamericana, México, 294pp. (1991).
- 6 Mano, M, M., *Diseño Digital*, Prentice Hall Hispanoamericana, México, 491 pp. (1993).
- 7 Tocci, R, J, *Sistemas digitales, principios y aplicaciones*, Prentice Hall Hispanoamericana, México, 479pp. (1986).
- 8 *Fast and LS TTL Data*, Motorola, USA, “LS TTL family”, (1994).



PÁGINAS DE INTERNET

- 1 Zator Systems. *Tecnología del PC, puertos E/S*.
http://www.zator.com/Hardware/H2_5_1.htm - TOP
- 2 JCGR Web Site. *El Puerto paralelo*.
<http://www.angelfire.com/pa2/jcgr/tecnica/tecnica.htm>
- 3 Gianluca Gentili. *Todo acerca de los puertos paralelos*.
<http://www.monografias.com/trabajos/paralelos/paralelos.shtml> - arriba
- 4 PC Kits. *El bus ISA*, <http://www.ctv.es/pckits/tISAe.html>
- 5 *Microcontrolador*, <http://es.wikipedia.org/wiki/Microcontroladores>
- 6 Doblado Alcázar Cristina, Gonzáles Gómez Juan, et. al.
Microcontrolador MCHC11 fundamentos, recursos y programación.
http://www.itson.mx/die/eromero/biblioelec/bsistdig3/Man_microb.pdf
- 7 Martín Flacker. *Desperdicios de computación envenenan aldeas Chinas*
<http://facom.udp.cl/CEM/TDC/artic/basura/basura.htm>
- 8 Mural/Negocios. *México teme la invasión de “chatarra tecnológica”*.
<http://www.canietigdl.com.mx/Articulo%202/>
- 9 Fototransistor. http://www.unicrom.com/Tut_fototransistor.asp
- 10 Fototransistor. http://www.anzwers.org/free/arsabe/Opto/opto_a.htm.



PERIÓDICOS Y REVISTAS

- 1 José Antonio Chávez. *Pase por el escáner*. Periódico Reforma, sección A, Interface, p. 1, Lunes 9 de febrero de 2004.
- 2 José Antonio Chávez. *Llega biométrica a México*. Periódico Reforma, sección A, Interface, p. 6, Lunes 9 de febrero de 2004.
- 3 Arturo Barba y Antimio Cruz. *Pegan robos en UNAM a proyectos científicos*. Periódico Reforma, sección C, Cultura, p. 6, Viernes 20 de febrero de 2004.
- 4 Juan José Fulguera, *Alarmas*, Revista Saber Electrónica, Quark, Buenos Aires Argentina, 2001.
- 5 Horacio D. Vallejo, *Todo sobre PIC's*, Revista Saber Electrónica, Quark, Buenos Aires Argentina, febrero de 2001.