UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UN SISTEMA DE SEGURIDAD Y CONTROL DE ACCESO

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

JOSE ALONSO FLORES

Asesor: Ing. Felipe Rauda García

MEXICO, D. F, 1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

RESUMEN

El objetivo de esta tesis es el diseño y construcción de un sistema de seguridad y controlador de acceso basado en un microcontrolador, dicho sistema opera en forma independiente, sin embargo, tiene la capacidad de interactuar con una computadora personal (PC).

El sistema que se describe en el presente trabajo es en general económico y con las características principales que requiere un sistema de seguridad; tiene la capacidad de supervisar 15 mallas de sensores y controlar el acceso de una área restringida. Además, la opción de ser operado vía PC, incrementando así sus funciones.

La computadora permite llevar un registro estadístico de los eventos ocurridos durante el día. Además, la programación de parámetros de operación se vuelve mucho más fácil, dado que se muestran diálogos interactivos fácilmente entendibles por el usuario.

Cuando se concede el acceso a la zona restringida, la computadora registra la hora y la fecha en que este evento ocurre, así como los datos de la persona que se introduce. Si se activa un sensor, entonces se registra la hora y la fecha, el número de la malla en que se activó y una breve descripción de zona cubierta por dicha malla. Esta información se almacena en disco para su análisis posterior, ya que se pueden generar reportes en pantalla o impresos.

Se cuenta con 254 claves programables que se pueden distribuir a personas exclusivas o a departamentos completos; pueden estar formadas de hasta 8 dígitos.

Además, se contempla el uso de sensores dedicados a la detección de intrusos como son: contactos magnéticos, sensores infrarrojos, etc., y con sensores preventivos e informativos como son los sensores de humo y de temperatura.

Los parámetros de operación son totalmente programables mediante el propio sistema o a través de la PC. Al respecto se tiene una clave de seguridad ("Password"), la cual permite que la programación sea efectuada sólo por aquella persona que cuente con esta clave, evitando así posibles sabotajes.

INTRODUCCION

Desde su aparición, el ser humano ha tenido que protegerse tanto de la naturaleza como de sus semejantes. Esta razón lo ha llevado a pensar siempre en nuevos y diferentes dispositivos de protección, los cuales han tenido una constante evolución a medida que la ciencia se ha desarrollado.

Los sistemas de seguridad actualmente se constituyen en su gran mayoría por dispositivos electrónicos, ya que los costos de los componentes se han reducido considerablemente, gracias al desarrollo tecnológico y a los altos volúmenes de producción.

Anteriormente, sólo se disponían en lugares en los que se deseaba evitar el robo, el atraco o el incendio. En su mayoría estos lugares eran grandes almacenes, bancos, etc. En la actualidad se aplican en pequeños negocios, fábricas y hogares, además de las entidades bancarias y de ahorro; estos sistemas van desde el más sencillo hasta el controlado por una computadora de propósito especial. Existen diferentes fabricantes y arquitecturas, pero todos tienen un costo elevado, dado que son dispositivos que utilizan tecnología de punta, para proteger valores de una empresa o del hogar.

El presente documento está dividido en dos partes fundamentales, la primera trata en forma general a los sistemas de seguridad, a los sistemas de control de acceso y a los diferentes sensores que se utilizan; de estos últimos se describen ventajas y limitaciones, así como el sitio adecuado para su instalación. En la segunda parte, se describe el diseño e implementación del sistema desarrollado, además de su funcionamiento e instalación.

El sistema desarrollado está formado por una unidad de control, sensores y detectores de intrusos e incendios, indicadores de alarma, dispositivos para el control de acceso, respaldo de baterías y como dispositivo opcional una computadora personal (PC).

La unidad de control se encarga de procesar y controlar la información de los diferentes dispositivos que forman el sistema. Dicha unidad, supervisa a los diferentes sensores, activa y desactiva señales de alarma dependiendo de las condiciones indicadas en ellos, es responsable del control de acceso y de la comunicación con la computadora personal.

El bloque de sensores y detectores está formado por un grupo de contactos magnéticos colocados estratégicamente para detectar cuando alguna puerta o ventana es forzada por algún intruso; por sensores infrarrojos que detectan la presencia de éste; detectores de ruptura de cristal que se activan cuando un cristal se golpea o rompe y por sensores de humo.

Los indicadores de alarma son dispositivos audibles y visuales que se activan por la unidad de control cuando se detecta algún intruso o un incendio. Esta parte se encuentra implementada con sirenas y dispositivos emisores de luz.

El control de acceso se lleva a cabo por medio de un teclado y claves en memoria con una capacidad de hasta 254 claves programables de hasta ocho dígitos.

El respaldo de baterías está diseñado para entrar en operación cuando la energía de la línea cae por debajo de un cierto valor o desaparece y soporta al sistema durante aproximadamente 12 horas.

La comunicación con la computadora se realiza de forma asíncrona a una velocidad de 9600 bauds a través de su puerto serie (com 1).

1.0 SISTEMAS DE SEGURIDAD

Hoy en día, los sistemas de seguridad se aplican en pequeños negocios, fábricas, hogares, en las entidades bancarias y ahora también en algunos procesos industriales, en las centrales nucleares, en los centros de investigación, etc.

Dichos sistemas están formados por dos partes principales. Una lo constituyen los dispositivos de seguridad y la otra un sistema de vigilancia electrónico.

La seguridad propiamente dicha es proporcionada por los dispositivos de seguridad, tales como cerraduras, cajas fuertes, bóvedas, y estructuras como paredes, cercas y puertas. Los sistemas de vigilancia electrónico proveen de detectores de intrusos, por ejemplo, controles de acceso, cámaras y monitores de televisión, señales de alarma, etc.

Físicamente, un sistema de seguridad deberá cumplir con cuatro funciones [1] indispensables e igualmente importantes, las cuales son: retardo, detección, alerta y respuesta. El retardo está dado principalmente por los dispositivos de seguridad como puertas, paredes y cerraduras entre otros, los cuales se encargan de retardar lo más posible la acción del intruso. La segunda función, detectar, es proporcionada por los sensores que detectan la presencia de personas no autorizadas o señales de alarma. La tercera función de un sistema de seguridad, alertar, está dada por los sistemas de alarma y supervisión que especifican la zona que se ha violado. La última función es ejecutar la acción correspondiente a la señal de alarma. Si alguna de estas funciones no es implementada, dicho sistema no puede brindar protección en el grado máximo absoluto.

La detección de intrusos es una de las partes más importantes de un sistema de seguridad, ya que si se sabotea prácticamente todo el sistema es anulado. El área a proteger debe ser dividida en cinco zonas de detección [1]. La primera corresponde a la parte exterior del área a la que se brinda protección; es decir, el intruso se puede detectar cuando intenta cruzar el perímetro de

la zona protegida, pudiendo ser éste una barda de concreto o una cerca. La segunda zona de detección está comprendida entre la cerca y la propia construcción a proteger. En ella se utilizan detectores de microondas, barreras detectoras por infrarrojo y algunos sensores sísmicos enterrados. La tercera zona es el perímetro de la construcción; la detección del intruso se lleva cabo, cuando éste penetra al edificio forzando alguna puerta o ventana, utilizando detectores de vibración y contactos magnéticos principalmente. La cuarta zona de detección está comprendida entre el perímetro del edificio y el bien que se protege (caja fuerte, bóveda, cuadros, etc.); los detectores de movimiento, son los dispositivos más comúnmente usados para detectar al intruso en esta zona y que pueden ser por infrarrojo, microondas o ultrasonido. La última zona corresponde al bien que se protege, la detección puede llevarse a cabo utilizando sensores de proximidad, interruptores de piso, etc.

Un sistema de seguridad puede estar formado por sistemas normalmente abiertos, normalmente cerrados o una combinación de ambos [2].

Aquellos en circuito cerrado, se activa una alarma cuando algún sensor abre el circuito. Este tipo de sistemas tienen la ventaja de que si un intruso corta los hilos que cierran el circuito, la alarma se activa inmediatamente. Pero también tienen inconvenientes, uno de ellos es que necesita consumir energía todo el tiempo, aunque en los sistemas actuales sólo se sensa el estado lógico, además, no es recomendable para líneas largas debido a la resistencia que presentan los conductores.

Los sistemas en circuito abierto son aquellos en los que no hay flujo de corriente hasta que se activa algún sensor; se utilizan normalmente cuando se requiere un grado de protección menor al máximo absoluto.

Un sistema de seguridad debe ser ante todo confiable, una falla en el momento adecuado, anularía el fin para el que fue instalado. Por otro lado, aquel que es propenso a dar falsas

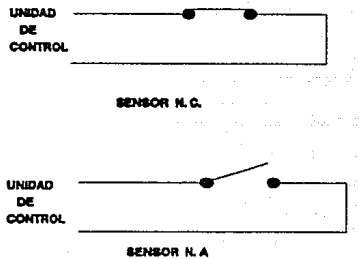


Fig. 1.1 Tipos de sensores.

alarmas, es tan malo como el que puede ser saboteado y anulado completamente en sus funciones, ya que generalmente es ignorado y desconectado.

Puede ser tan robusto como grado de protección se requiera y de lo que se esté dipuesto a pagar, ya que el nivel de robustés está en función del costo. Con el desarrollo de la tecnología, los sistemas actuales tienen un costo mínimo de operación, pues su consumo de energía es reducido o al menos que el sistema sea bastante grande.

1.1 INSTALACION DE LOS SISTEMAS DE SEGURIDAD

La protección que ofrece un sistema de seguridad depende de la forma en que se instala y de la confiabilidad de sus componentes; si éstos se conectan con alambres fácilmente accesibles para los intrusos y la interconexiones se realizan con empalmes de alambres descubiertos, el nivel de

seguridad será bajo. Un elemento que se ha diseñado para funcionar dentro de un local no es seguro si se usa en el exterior. Al instalar un sistema, se puede caer en el error de adquirir componentes baratos, lo que puede llevar a tener problemas de falsas alarmas y costos elevados de mantenimiento.

El instalador deberá ponerse en el lugar del ladrón y saber los máximos riesgos de incendio de cada zona. Se deben proteger todos los accesos externos y los internos para detectar con seguridad la presencia del intruso. La unidad de control deberá situarse en un lugar oculto, de tal forma, que no sea visible al intruso y también debe disimularse el cableado, sobre todo del indicador de alarma, ya que si se sabotea, la totalidad del sistema estará fuera de servicio. No existen normas generales en este tipo de instalación, ya que cada caso es independiente y necesita soluciones distintas.

1.2 NIVELES DE SEGURIDAD

Las áreas que requieren protección deben ser clasificadas en alguno de los siguientes cuatro niveles [1] de seguridad: nivel A, nivel B, nivel C y nivel D. El nivel A es para brindar un nivel mínimo de protección contra un adversario altamente especializado, mientras que el nivel D es propuesto para brindar un nivel mínimo contra un adversario inexperto. El nivel de seguridad A se recomienda cuando se tiene que proteger a un alto valor monetario o información estratégica. El nivel C se propone para material robable, dinero y artículos. El nivel D es propuesto para control administrativo.

NIVEL DE SEGURIDAD EN BASE AL TIPO DE AMENAZA

	INTERNO	EXTERNO
A	ALTAMENTE ESPECIALIZADO	ALTAMENTE ESPECIALIZADO
B	ALTAMENTE ESPECIALIZADO	ESPECIALIZADO
C	ESPECIALIZADO	INEXPERTO
D	INEXPERTO	INEXPERTO

Tab. 1.2.1

1.3 SENSORES Y DETECTORES DE INTRUSOS.

En un sistema de seguridad, los sensores de intrusos se pueden clasificar para su descripción en dos grandes gruposy es debido principalmente por su ubicación física dentro del sistema. Los grupos son los siguientes: detectores externos y detectores internos de intrusos.

1.3.1 DETECTORES EXTERNOS DE INTRUSOS

Los detectores externos son aquellos dispositivos que sensan la presencia de un intruso antes de que éste logre introducirse hasta el edificio donde se encuentran los valores protegidos, entre ellos se tienen los siguientes:

1) Transductores Electromecánicos.

Los transductores electromecánicos son interruptores normalmente cerrados, aunque también pueden ser normalmente abiertos. Dichos interruptores se utilizan para detectar al intruso cuando trata de introducirse a la zona protegida a través de la cerca. Cuando el intruso trepa o corta los hilos, en la cerca se producen vibraciones de alta frecuencia que hacen que los interruptores abran y cierren un circuito, generando así, una serie de pulsos los cuales son enviados a un procesador para ser analizados.

El ancho de los pulsos depende directamente de la frecuencia de la vibración. El número de pulsos y la duración de éstos, depende de la forma de escalar y el tiempo requerido por el intruso para penetrar por la cerca.

Son dos los tipos los básicos utilizados, los interruptores de inercia mecánica y los de mercurio. Los de inercia mecánica están formados básicamente por una masa de metal sísmico soportado por dos o tres contactos formando un interruptor normalmente cerrado. Los interruptores de mercurio son normalmente abiertos y hacen un contacto instantáneo en el momento del impacto, estos consisten en una botella con una cantidad pequeña de mercurio y dos contactos eléctricos cercanos al mercurio. Cuando ocurre el impacto en la cerca, el mercurio se desplaza de su posición en reposo y toca los contactos eléctricos generando momentáneamente un circuito cerrado. Este tipo de sensor, también se utiliza para detectar vibraciones en paredes y estructuras que pudiera romper el intruso para penetrar al edificio.

2) Transductores piezoeléctricos.

Los transductores piezoeléctricos convierten las vibraciones mecánicas en una señal eléctrica. La señal generada varia proporcionalmente en amplitud y frecuencia de las vibraciones

mecánicas producidas en la malla por un intruso o a la presión inducida en el sensor cuando el intruso cruza por una línea de sensores piezoeléctricos. Este transductor está formado por un cristal de cuarzo que al aplicarle una presión externa genera un voltaje proporcional a la presión aplicada. Las señales obtenidas son amplificadas y filtradas para eliminar las bajas frecuencias causadas por disturbios naturales.

3) Transductor "geofónico".

El transductor gefónico consiste en un imán permanente de forma cilíndrica en cuyo interior se tiene una masa sísmica de forma tubular, sobre la cual se enrrolla una bobina de alambre fino. Cuando el transductor experimenta vibraciones mecánicas, la masa sísmica se mueve junto con la bobina sobre el campo magnético del imán cortando las líneas de campo, esto hace que en la bobina se induzca una tensión eléctrica proporcional a las vibraciones mecánicas que experimenta el transductor.

Puede ser colocado en una cerca para detectar al intruso cuando intente escalarla, sensando las vibraciones mecánicas provocadas por el ataque. También, es usado para detectar el paso de un intruso en la zona protegida, sensando las vibraciones que sus pisadas causan sobre la tierra, esto se logra enterrando transductores a lo largo de la zona protegida.

Los actos de penetración en forma general, provocan vibraciones y disturbios en el medio, que son de frecuencia y amplitud mayor que las provocadas por el viento y la lluvia. Los sensores utilizados, deben tener la capacidad de discriminar las señales débiles y de baja frecuencia para reducir el número de falsas alarmas.

4) Sensores infrarrojos.

Los detectores infrarrojos generan un patrón de haces múltiples de radiación infrarroja y cuando uno o alguno de los haces se interrumpe por el intruso se inicia la alarma. En la figura 1.3.1 se muestra un detector de barrera del tipo descrito.

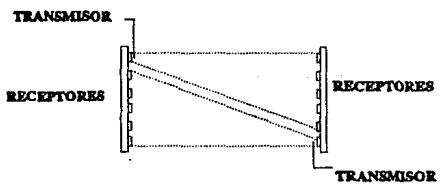


Fig. 2.3.1 Detector de barrera infrarrojo.

La radiación infrarroja es una emisión electromagnética que se extiende por debajo de la parte de luz visible del espectro, pero mucho mas alta que la banda de radiofrecuencia. Es generada también por fuentes que producen calor. El cuerpo humano o el de cualquier animal, una lámpara, una estufa, etc., son generadores de infrarrojos. Cualquiera de éstas fuentes podría interferir al receptor y un intruso podría engañarlo encendiendo una lámpara dirigida hacia el. Para evitar esta situación se modula en amplitud, de forma que el receptor sólo reconozca la señal modulada, reduciendo así, la posibilidad de que la fuente puede ser sustituida por el intruso. La fuente de radiación infrarroja más comúnmente usada, es el diodo emisor de luz

(LED) de arseniuro de galio (AsGa). El haz de luz generado por el transductor es colimado y dirigido hacia el receptor, el cual está formado por lentes colectores que enfocan la energía captada hacia una celda fotoeléctrica. Dicha celda está formada por un dispositivo semiconductor que convierte la radiación infrarroja en una señal eléctrica proporcional a la energía recibida.

El receptor supervisa la señal eléctrica y si ésta cae por debajo de un cierto nivel, entonces se inicia la alarma. El nivel deberá caer por lo menos al 90% de su valor durante 75 milisegundos para que se inicie la alarma, ésto reduce la posibilidad de falsas alarmas que pudieran ser provocadas por animales pequeños y polvo.

Este tipo de sensores se configuran en forma de barrera de detección. El margen de longitud en estas barreras va desde 10 m hasta 30 m en la unidades más grandes. En algunos casos, es necesario usar espejos para formar la barrera, entonces se debe considerar la perdida de energía por la reflexión.

5) Detectores por microondas.

Los detectores de microondas, generan un angosto haz de energía electromagnética en la banda de microondas que es recibida por su correspondiente receptor. Cuando alguien atraviesa el haz de radiación electromagnética, el receptor sensa una variación en la cantidad de energía recibida o la distorsión causada por el intruso y entonces inicia la alarma. En figura 1.3.2 se muestra el funcionamiento de este dispositivo.

La distancia entre el transmisor y receptor depende de el tipo de antena, de la configuración de ésta y de la frecuencia de operación del sistema. Los sensores de microondas, también se configuran en forma de barrera invisible para detectar al intruso en la parte externa del edificio.

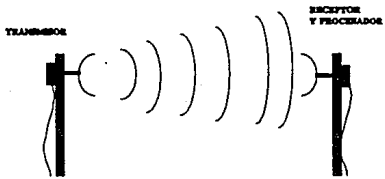


Fig. 1.3.2 Detector por microondas.

La frecuencia de operación de estos dispositivos está situada generalmente entre 800 Mhz y 15 Ghz. Las microondas pueden atravesar la madera, el cristal, el yeso e incluso con una extensión limitada los ladrillos. Esto significa tener cierta ventajas sobre los detectores infrarrojos.

En las figuras 1.3.3 y 1.3.4 se muestran algunos arreglos para proteger una zona utilizando detectores de microondas.

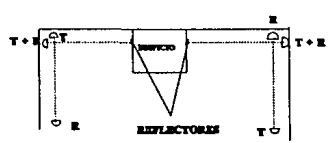


Fig. 1.3.3 Configuración utilizando reflectores.

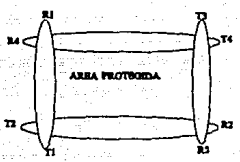


Fig. 1.3.4 Arreglo de detectores de microondas.

6) Sensor de campo eléctrico.

Los sensores de campo eléctrico generan un campo electrostático entre un arreglo de conductores y tierra. Cuando alguien se aproxima o toca la malla de conductores se detecta una alteración en el campo eléctrico generando una alarma.

7) Sensor de electrocapacidad.

Los sensores de electrocapacidad, detectan la presencia de un intruso, mediante una medida en el cambio de la capacitancia eléctrica entre los cables sensores y la tierra eléctrica.

Cuando un intruso toca los cables sensores una alarma es generada. Los cables pueden ser montados a lo largo de una malla, en lo alto de una pared o cualquier lugar donde alguien intente escalar.

Los cables sensores de capacitancia se dividen en dos secciones de alarma, derecha e izquierda de igual longitud, para la detección en modo diferencial. La operación en modo diferencial, minimiza las falsas alarmas, ya que los cambios comunes producidos por el viento, la lluvia, la niebla y el alumbrado afectan a ambas mitades igualmente y cancelan la señal del procesador. El cambio causado por el intruso no es común y puede ser detectado en la parte derecha o izquierdo cuando el intruso toca los cables sensores, dando además, una referencia de la zona de ataque.

La sensibilidad del sensor es ajustable, permitiendo detectar cambios en la capacitancia tan pequeños como 10 picofaradios hasta tan altos como 150 pifaradios, permitiendo al sensor conocer los requerimientos específicos de el lugar de la instalación.

8) Sensor de presión balanceado.

El sensor de presión balanceado detecta a personas y vehículos, sensando los ondas de presión generados por el movimiento sobre la superficie de la tierra. El sensor consiste en dos secciones tubulares de igual longitud y presión, la longitud tubular de cada sección puede ser hasta de 100 metros con una separación de un metro. El sensor produce una señal analógica proporcional a la diferencia de presión en las dos secciones tubulares, esta señal es enviada a un procesador para discriminar las señales que pudieran generar falsas alarmas.

1.3.2 DETECTORES DE INTRUSOS INTERNOS.

1) Detectores de movimiento por ultrasonido.

Los detectores de movimiento por ultrasonido constan de un transmisor, un receptor y una unidad de control. El principio se basa en que un oscilador electrónico genera una frecuencia

ultrasónica que alimenta a uno o más transductores. Los sonidos de alta frecuencia se producen en un espacio protegido y se reciben después por un sensor alojado en la misma unidad que el transductor.

El transmisor genera un patrón de señal acústica dirigido hacia la zona de detección. La energía reflejada por las paredes, el techo o por objetos es recibida y enviada al procesador.

El receptor capta el sonido del emisor y también el reflejado procedentes de varias superficies del recinto. Si hay ningún movimiento dentro de la zona de detección, el sonido reflejado procedente del objeto en movimiento experimenta un cambio de frecuencia debido al efecto *Doppler-Fizeau*, el cual es un corrimiento en frecuencia de la señal reflejada.

Si se mezclan dos frecuencias cercanas se producirá una tercera que es la diferencia entre las anteriores. También, se genera la suma de ambas, pero se desecha por ser muy alta para estas aplicaciones. El valor de la frecuencia *Doppler* depende de la velocidad del móvil relativa a detector, pero siempre será mucho más baja que la frecuencia ultrasónica.

La desventaja que presentan estos detectores es que son propensos dar falsas alarmas. Una frecuencia *Doppler* filtrada puede ser producida por el aire a través del cual viaja el sonido, moviéndose de la misma forma que lo hace la superficie reflectora. Debido a esto, los sensores ultrasónicos no se utilizan en el exterior, pues la más ligera brisa podría dar la alarma. En el interior se deben evitar los corrientes de aire, el movimiento de cortinas y animales.

Operan a una frecuencia específica comprendida en el rango de 19 a 40 Khz. La energía acústica generada a 19 Khz es considerada inaudible en promedio para el ser humano y es definida como frecuencia de ultrasonido.

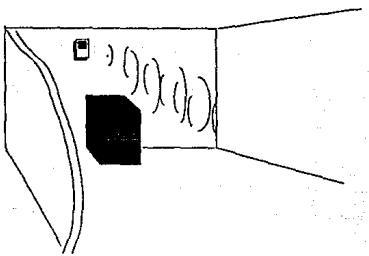


Fig. 1.3.5 Detector de movimiento por ultrasonido.

2) Detector de movimiento por infrarrojo (PIR).

El detector de movimiento por infrarrojo, sensa un cambio en el patrón de energía térmica, resultado del movimiento del intruso dentro del espacio de inspección. Además, detecta el calor del cuerpo humano e inicia la alarma cuando el cambio en la energía satisface el criterio del detector. El campo de inspección para el detector infrarrojo es el área frente al elemento sensor.

Todos los objetos con temperaturas sobre el cero absoluto (0 K), radian energía térmica. La magnitud y la frecuencia de la energía radiada depende de la temperatura absoluta y de la superficie de los objetos. Las características del patrón de energía de fondo, depende de si el cuerpo es opaco o brillante asumiendo que todos están a la misma temperatura. Lo objetos opacos son radiadores de calor más eficientes que los brillantes, por eso aquellos con diferente

terminado causan variaciones de la energía térmica radiada en el fondo de la zona de inspección. En la figura 1.3.6 se muestra el patrón de radiación de un sensor infrarrojo pasivo (PIR).

El sensor responde a la energía térmica de longitud de onda entre 1 y 1,000 micras , sin embargo, el sensor más popular es para el rango de infrarrojo entre 8 y 14 micras.

El termistor y la termopila son los sensores más utilizados para detectar la energía térmica en el rango de frecuencia infrarrojo. El termistor es un dispositivo semiconductor el cual varia su resistencia cuando detecta una variación de la energía térmica recibida.

La termopila es una multi-unión de termopares. Un termopar consiste en un par de uniones termoeléctricas de metales diferentes. Una juntura es protegida de la radiación incidente y la segunda se opaca para mejorar la absorción de calor. La diferencia de temperaturas en la unión entre dos metales diferentes, produce una *fem*, la cual es amplificada y procesada.

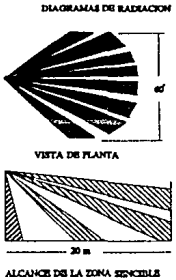


Fig. 1.3.6 Patrón de radiación del detector de movimiento por infrarrojo.

Los detectores de infrarrojo del tipo pasivo se encuentran disponibles con un solo haz o con múltiples haces de inspección. Los de simple haz se utilizan para proteger pasillos y corredores, y los de haz múltiple se usan para proteger grandes espacios.

3) Detector pasivo de audio.

Un detector pasivo de audio, detecta el ruido generado por el intruso al forzar una entrada o el producido por el rompimiento de objetos dentro del área protegida. El dispositivo consta de un número de micrófonos estratégicamente instalados para detectar sonidos dentro de la banda de frecuencias audibles.

4) Detector de vibraciones en estructuras.

Las vibraciones mecánicas en las estructuras, generados por la acción de un intruso en su afán de penetrar al edificio, son sensadas principalmente por transductores piezoeléctricos e interruptores electromecánicos. Los sensores son diseñados para responder a las frecuencias bajas, ya que desde el punto donde se produce el ataque hasta donde se encuentra el sensor la onda de vibración es atenuada en las frecuencias altas.

La señal eléctrica generada por el transductor es recogida y enviada al procesador a través de un cable coaxial.

5) Detector de ruptura de cristal.

El detector de ruptura de cristal es similar al de vibraciones de estructuras, sólo que el transductor es diseñado para responder a las vibraciones de alta frecuencia, generados por el

rompimiento de un cristal. El dispositivo inicia la alarma cuando detecta el primer pulso de alta frecuencia.

6) Contactos Magnéticos.

Idealmente, cualquier sensor no debería tener partes móviles y debería estar completamente protegido en su contorno. Los contactos magnéticos reúnen estas características, ya que tienen en su interior un relevador accionado por un imán externo por proximidad. Al separar el imán del relevador, este abre o cierra el circuito dependiendo de la configuración.

Este dispositivo es utilizado para detectar la apertura de una puerta o ventana sin autorización. En la figura 2.3.7 se muestra una aplicación típica.

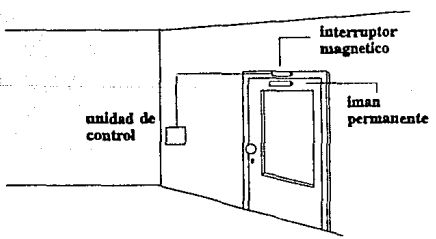


Fig. 1.3.7 Aplicación para contactos Magnéticos.

7) Sensor capacitivo de proximidad.

El sensor opera de igual manera que un capacitor eléctrico y consta de dos placas de material conductor separadas por un medio dieléctrico. Para este caso, una de las placas lo constituyen los objetos metálicos existentes en el área protegida y la otra el plano de tierra bajo y alrededor de los objetos; el medio dieléctrico es el aire. Cuando ocurre un cambio en la carga eléctrica o en el dieléctrico, ocurre un cambio en el valor de la capacitancia, que indicaría la presencia de un intruso.

La sensibilidad del detector puede ser afectado por el cambio en la humedad relativa del medio y la relocalización de los objetos metálicos. La humedad relativa causa variaciones en el medio dieléctrico, al incrementarse la humedad relativa, causa que la conductividad del medio aislante se incremente y reduzca la capacitancia.

8) Sensores de incendio.

Cuando se inicia un incendio, normalmente se desprenden gases y humo producto de la combustión que no son visibles al ojo humano, pero que si alteran las condiciones ambientales del sitio donde ocurre la combustión. El detector iónico es el dispositivo más utilizado para la detección de incendios. El sensor se compone básicamente de dos cámaras separadas entre sí, ambas están ionizadas por una fuente de material radiactivo situado en al cámara interior. Esta atmósfera ionizada, al aplicarle una tensión entre las dos cámaras, crea una débil corriente de iones que en condiciones normales es idéntica en ambas cámaras, manteniendo una diferencia de potencial constante. Cuando se produce un desprendimiento de gases o humos producto de la combustión, éstos llegan al detector penetrando en la cámara exterior, chocan con las corrientes de iones, haciendo que las partículas ionizadas sean más pesadas, con lo que se

produce un desequilibrio entre las dos cámaras, aumentando la diferencia de potencial que activa una señal de alarma. La radiación de la fuente es en forma de rayos alfa y gama. El campo de los rayos *alfa* es muy corto, aproximadamente de unos 4 cm, por lo que no pueden pasar la cámara exterior. Los rayos *gama* sólo son emitidos por el sensor, cuya radiación se controla para que no sobrepase el valor de 1.5 microcuries por hora a una distancia de 5 cm desde la superficie del sensor, este valor está por debajo de la radiación normal del ambiente considerada en 5 microcuries por hora.

Otro tipo sensores utilizados son los detectores ópticos de humo, se forman por una cámara en la que se encuentra montado un diodo emisor de luz (LED) y una celda fotoeléctrica, de forma tal, que la luz emitida por el diodo no llegue a la fotocelda si no es por refracción sobre las partículas de humo que penetran al detector. Cuando la luz reflejada es detectada por la fotocelda (fototransistor) se activa el circuito de alarma.

1.4 INDICADORES DE ALARMAS.

Un indicador de alarma puede ser una lámpara, un "zumbador", una sirena, un timbre, etc. Estos deben tener un sonido fuerte de tal forma que pueda ser oído en gran extensión. Estos dispositivos tienen un gran riesgo, ya que si son saboteados todo el sistema de seguridad queda anulado. Por ello conviene instalar más de una sirena o timbre con este propósito.

El oído humano no responde linealmente al nivel acústico generado por un dispositivo sonoro. Lo hace de una forma logarítmica. Esta característica, permite contener con un amplio margen, sonidos muy altos y escuchar perfectamente sonido muy débiles. Cualquier ruido por encima de 80 dB se considera alto. El umbral de dolor se alcanza a 130 dB. La presión sonora decrece con la distancia. El dispositivo de alarma más común es el timbre, pero cuando se requiere que el sonido sea de alta intensidad, entonces se deberán utilizar sirenas.

1. BARNART, ROBERT L.
Intrusion Detection System. 2a. Ed.
Buterworkths, 1988. pp. 1-5, pp.20-60, pp60-90.

2. VAUGHN, MARTIN [Y] DEAN DEVIS
Proyectos de Seguridad.
CEAC,1989. pp.15-90.

2.0 SISTEMAS CONTROLADORES DE ACCESO

La función primordial de los sistemas controladores de acceso es controlar la entrada y algunas veces la salida de áreas críticas. Esta función se lleva a cabo por medio de la identificación del individuo y permitiendo la entrada sólo al personal autorizado. Además, del control de acceso común, los sistemas automáticos pueden llevar un registro estadístico de la asistencia, supervisar el tiempo de estancia del personal, efectuar turnos de guardia, etc.

El control de acceso puede llevarse a cabo tanto por guardias de seguridad o por sistemas automáticos, aunque en la mayoría de los casos se usa una combinación de ambos. En términos generales es indispensable la presencia de guardias para tener un buen control. Los guardias se sitúan en los puntos de entrada directa para la revisión de equipaje de personas no autorizadas o visitantes. Además de las funciones de control de acceso, los guardias deben responder a las alarmas. Una alarma puede ser activada cuando alguna persona no autorizada intenta entrar, cuando una puerta permanezca demasiado tiempo abierta, o se detecta una señal de humo.

Los sistemas automáticos reducen el número de guardias, ya que algunas de sus funciones son tomadas por dichos sistemas. El tipo elegido depende del nivel de seguridad requerido, del número de personas autorizadas, de el sitio donde se desea instalar, etc. No hay sistema que pueda satisfacer todos los requerimientos. Además, todos tienen puntos vulnerables especialmente si no tienen una aplicación adecuada o están mal instalados. De aquí que cada parte que lo forma debe ser probada y contar con la mejor calidad; también se requiere de un mantenimiento adecuado y la perfecta instrucción del personal de seguridad.

2.1 TECNICAS DE CONTROL DE ACCESO

Los sistemas automáticos controlan el acceso sin la ayuda de algún guardia; permiten el acceso o la negación basándose en el reconocimiento de un código en memoria que se da por medio de un teclado, de una tarjeta codificada o por algún otro medio.

Un número importante de factores son considerados al seleccionar un sistema de control de acceso. El más importante es la resistencia a la falsificación. El grado de resistencia requerido es función de la importancia y de que tan crítica es el área que se esta controlando. La resistencia a la falsificación es una medida de la dificultad para duplicar el código de acceso. Los sistemas de reconocimiento del habla, la retina y la firma son considerados los más resistentes debido a la interacción dinámica necesaria para la identificación personal. Los sistemas de reconocimiento de geometría de la mano y huellas digitales se consideran como de medios a alto, mientras que cuando el acceso es dado por tarjetas codificadas, códigos en memoria y teclado ofrecen baja resistencia a la falsificación.

Otro factor importante a considerar en el control de acceso es el tiempo que tarda una persona normal en entrar, ya que mientras la puerta permanezca abierta otra podría introducirse. Algunos sistemas dan una señal de alarma cuando la puerta permanece demasiado tiempo abierta. Además, es necesario considerar el tipo de cerradura.

2.2 PRINCIPALES SISTEMAS CONTROLADORES DE ACCESO.

Existen diferentes dispositivos para controlar el acceso de una área, la selección de este dispositivo está en función del nivel de seguridad deseado y del costo. A continuación se describen algunos de los sistemas más comunes:

2.2.1 Teclado y código en memoria.

En ellos se debe dar un código que se encuentra en memoria con la secuencia adecuada usando un teclado. Cuando el código es correcto se otorga el acceso activando inmediatamente la cerradura que abre la puerta.

Un punto vulnerable que presenta este sistema es que una vez que se ha abierto la puerta más de una persona puede entrar, pudiendo ser alguna no autorizada. Algunos, ofrecen seguridad adicional, activando una alarma cuando la puerta permanece demasiado tiempo abierta.

Los sistemas con teclado y código en memoria proveen relativamente un nivel bajo de seguridad por lo que es importante darle una aplicación adecuada dependiendo de la necesidad que se requiera cubrir.

2.2.2 Tarjetas codificadas.

La mayoría de los sistemas de control de acceso usan tarjetas codificadas con sus respectivas lectoras. Para lograr el acceso la persona introduce o presenta su tarjeta a la lectora. Dicha tarjeta en tamaño y apariencia se parece a una tarjeta de crédito. Aunque las técnicas de grabación del código varían en cada fabricante, se pueden almacenar millones de combinaciones. La codificación puede ser magnética o electrónicamente con los datos necesarios para la completa identificación de la persona. Algunas, proporcionan una fotografía y las características propias del portador para una posible revisión complementaria.

Su aplicación requiere tener un procesador central con los datos de cada usuario, conectando a éste, las lectoras de tarjetas remotas. El procesador puede controlar el acceso y la salida de cientos de empleados usando lectoras en varios lugares. Cuando una tarjeta se presenta a la lectora, ésta sensa la información codificada y la transmite al procesador, el cual recibe la información y la compara con los datos en memoria, y en unos milisegundos decide si negar o acceder la entrada. Cuando el acceso es concedido el controlador manda una señal que abre inmediatamente la puerta.

La unidad central de control permite al operador realizar muchas funciones, una de las cuales es cancelar tarjetas perdidas o robadas. Es necesario que la cancelación de tarjetas sea tan rápido y fácil como sea posible.

Una de la funciones adicionales en algunos sistemas es que no se permite el uso de la tarjeta para entrar hasta que ésta haya sido usada para salir del área de control. Con lo que se evita que la tarjeta pase de una persona que se encuentra adentro a otra que quiera entrar.

Las lectoras de tarjetas identifican a la tarjeta no al portador. La vulnerabilidad más común es la pérdida y que su propietario no se percate. Una combinación teclado y código en memoria robustece al sistema.

Las tarjetas pueden ser codificadas para dar información adicional, por ejemplo, el puesto del usuario, cuando y a que hora está permitido su acceso.

A continuación se describen algunas formas más populares de control de acceso por tarjetas codificadas:

1) Tarjeta de identificación por foto.

Este tipo de tarjeta puede ser la credencial de empleado con la fotografía del propietario, la cual que puede ser inspeccionada por un guardia. Es difícil cuantificar la efectividad de este tipo de control de acceso, debido a que entra en juego el criterio del guardia cuando examina la credencial. Otro factor que interviene es el número de personas que estén entrando al mismo tiempo, además, de que la credencial es fácil de falsificar.

2) Tarjetas con código magnético.

Las tarjetas con código magnético tienen una hoja flexible de material magnético, entre dos hojas de material plástico, en la que se graba un arreglo de marcas magnetizadas permanentemente. El código es determinado por la polaridad de las marcas.

La desventaja que presentan estas tarjetas es que el código se puede borrar si es expuesta a un fuerte campo magnético. Es posible falsificarlas, pero en general no presentan problemas.

3) Tarjetas con tira magnética.

Una tarjeta de este tipo presenta una tira magnética a lo largo de uno de sus lados, la cual se codifica con los datos de identificación del portador. Algunos sistemas utilizan codificación alfanumérica permitiendo identificar el nombre y datos adicionales.

Los sistemas que usan tarjetas con tira magnética tienen actualmente un amplio uso; pueden ser falsificadas y también existe el riesgo de un borrado accidental.

4) Tarjetas de código óptico.

Estas tarjetas son codificadas por un arreglo geométrico (códigos de barras) grabado en cintas, los espacios representan datos codificados. La ventaja de estos códigos es que no requiere un lector sofisticado. Puede ser leído pasando un detector óptico sobre la tarjeta.

La desventaja es que el código es visible y puede ser fácilmente duplicado, aunque en las versiones recientes el código sólo puede ser leído usando luz ultravioleta o infrarroja.

5) Tarjetas de proximidad.

Las tarjetas de proximidad se codifican eléctricamente, un campo electromagnético es transmitido por una unidad estacionaria de interrogación ubicado junto a la entrada de acceso. Cuando la tarjeta está expuesta al campo electromagnético se induce un voltaje en la tarjeta que activa un circuito eléctrico pasivo, entonces la unidad interrogadora sensa la información y la envía a la unidad de control, si el dato es válido se abre la puerta. La ventaja de estos sistemas es que no es necesario insertar la tarjeta en la lectora.

2.2.3 Comparación por video.

En la comparación por video, se utiliza un circuito cerrado de televisión y en combinación con el personal de seguridad se realiza el control de acceso. Dicho control, es manual ello implica que sea más lento y además depende de la dedicación y concentración del operador para el buen desempeño del sistema.

Las áreas que requieren de alta seguridad, el sistema controlador de acceso, verifica la identidad del solicitante a través del reconocimiento de huellas digitales, geometría de la mano, patrón de voz y algunas otras características que hacen única a una persona.

Para ésto, se requiere digitalizar previamente los datos que identifican al usuario. Es muy común, que la persona que desea el acceso se identifique con una tarjeta codificada, posteriormente, para confrontar sus datos se procede a la verificación.

La información de entrada es digitalizada y comparada a alta velocidad con los datos de referencia para que en pocos segundos el acceso pueda ser concedido dependiendo de el resultado de la comparación.

Desafortunadamente, las características físicas de la gente cambian en tiempos relativamente cortos. Por ejemplo, las huellas digitales pueden verse afectados por una herida, por el desgaste extremo y por estar en contacto con superficies abrasivas dependiendo de la actividad que el usuario realice. El patrón de voz y la firma pueden ser afectados por el estress y la fatiga. Por estas razones el sistema debe tolerar un porcentaje de errores.

A continuación se describen algunos de los sistemas mas comunes de verificación.

2.2.4 Sistemas de reconocimiento de huella digital.

En estos sistemas, la huella que se desea reconocer se encuentra entintada en una superficie determinada. El área de la huella digital es explorada por métodos ópticos, digitalizada y transmitida a la unidad de control, la cual guarda esta información junto con los datos de la persona que posteriormente solicitará el acceso.

La identificación se lleva a cabo por comparación, es decir, la unidad de control compara la huella leída con la patrón guardada en memoria. Se comparan los datos de las pequeñas interrupciones, la terminación de arrugas y ramificaciones de un número de aproximadamente cien marcas impresas en una huella.

La terminal de este sistema cuenta con un "display", un teclado o lector de tarjeta, un dispositivo sobre el cual se pone el dedo del solicitante y un explorador óptico ("scaner") para obtener la información de la imagen de la huella digital. El teclado o la lectora de tarjeta, son utilizados para identificar a la persona, ya sea que introduzca su tarjeta o que teclee un número asignado previamente. En el display se indican los pasos a seguir para lograr el acceso.

Es muy común, que el sistema guarde información de más de un dedo, debido a que pueden producirse heridas o daños que causarían un error en la comparación de huellas, si esto ocurre se tiene la opción de cambio de dedo.

2.2.5 Reconocimiento de la firma.

El sistema de reconocimiento de la firma, se basa en la comparación de las características dinámicas del firmante. Estas características son la presión ejercida al ejecutar la firma y la velocidad con que se realiza. Dos son las técnicas utilizadas para identificar la firma. Una usa un sensor de presión especial puesto sobre el escritorio, el cual sensa la fuerza aplicada al escritorio por el firmante. Con ésta técnica no se requiere de una pluma especial. La segunda técnica, utiliza una pluma especial que sensa el movimiento de la punta y además la presión aplicada por el firmante.

El patrón de presión y movimiento de la pluma son diferentes en cada firmante lo que da un alto grado de certidumbre sobre la autenticidad de la firma. La falsificación de la firma original es muy difícil, debido a que la velocidad de escritura y la presión no están directamente relacionados con la apariencia.

2.2.6 Reconocimiento de la geometría de la mano.

El sistema para el reconocimiento de la geometría de la mano puede ser programado para operar con diferentes configuraciones, pero básicamente miden la longitud. La medición se realiza utilizando un método fotoeléctrico; cuando la mano es puesta sobre el sensor, la luz que ilumina el fotosensor es parcialmente obscurecida por los dedos. La información obtenida es enviada a la central para efectuar la comparación correspondiente.

2.2.7 Reconocimiento del patrón de voz.

La persona que desea el acceso al área controlada, entra primero a una cabina para prueba de voz, en donde se identifica a través de un teclado o tarjeta codificada. Debe previamente recordar el mensaje individual que debe repetir frente a un micrófono, dicho mensaje generalmente está formado por cuatro palabras de dieciséis monosílabos aproximadamente. Estas frases tienen una duración de alrededor de dos segundos. La repetición de la frase en el micrófono es procesada y comparada con los datos en memoria. El sistema compara la amplitud de la onda de voz, además de la frecuencia y el tiempo.

2.2.8 Reconocimiento de retina.

Para un control individual de acceso, en estos sistemas, se analiza el patrón arterial de la retina del ojo . El ojo es expuesto a una cámara que explora el área circular de la retina con un haz de luz infrarrojo de extremadamente baja intensidad. La luz reflejada por el fondo del ojo, es enfocada a un fotosensor que mide la magnitud de la luz en varios puntos distintos a lo largo de 420°. El resultado describe una forma de onda formada por los datos de los puntos.

3.0 DESCRIPCION GENERAL DEL SISTEMA

El sistema de seguridad y control de acceso que se describe en el presente trabajo es en general económico, pero que satisface las necesidades básicas de seguridad y control de acceso; está diseñado para utilizar componentes de bajo costo y accesibles en el mercado nacional; tiene capacidad para supervisar varios sensores conectados en forma de mallas y controlar el acceso de una área restringida.

El diseño está basado en un *microcontrolador*, el cual permite que las dimensiones del "hardware" sean mínimas, ya que en una sola pastilla se incluyen varios dispositivos como son: memorias, líneas de entrada/salida, etc., este dispositivo también facilita el diseño de equipos amigables y de dimensiones pequeñas.

El sistema desarrollado, además de ser económico es funcional y de fácil manejo, ya que cuenta con un "display" alfanumérico mediante el cual se visualizan las operaciones realizadas, así como los parámetros programados. El tamaño del sistema (Unidad de Control) es de 20 X 30 cm que resulta muy práctico, ya que no ocupa mucho espacio. Además, tiene un consumo de corriente mínima, del orden de 250 mA en operación normal.

El acceso al sistema se hace a través de una clave de seguridad ("password"), con el fin de evitar que cualquier persona pueda alterar los parámetros de operación y sólo pueda hacerlo la persona indicada. Esta clave a su vez puede ser cambiada cuando se activa la llave que se encuentra en el módulo de control.

Sólo la persona que tenga acceso al sistema puede observar los parámetros de operación o modificarlos según convenga. Los parámetros que se pueden cambiar son: definición de los sensores de una malla, claves de acceso y estado de las señales de alarma.

Se tiene capacidad para interactuar con una computadora personal (PC), mediante la cual se puede llevar un registro estadístico de los eventos sensados por la unidad de control. Cabe destacar que el sistema puede operar de manera autónoma realizando la mayoría de las tareas encomendadas, el uso de la PC es opcional.

Los eventos que se registran en la computadora son: la hora y la fecha en que se detecto algún sensor activado, así como la ubicación de éste dentro del área de cobertura; la hora y la fecha en que un usuario logró el acceso a la zona restringida, además del nombre y departamento correspondiente. Se genera un registro cada vez que se efectúan cambios en los parámetros de operación, así como también, cuando se realiza una prueba al sistema, ya que se recomienda que sea cada mes para asegurar su correcto funcionamiento.

El "software" de la computadora permite también programar los parámetros de operación, ésta es la forma más fácil y rápida, aunque puede hacerse a través de la Unidad de Control.

El sistema descrito está formado de las siguientes partes principales:

1.- Unidad de control;
2.- Sensores y detectores;
3.- Indicadores de alarma;
4.- Control de acceso;
5.- Fuente y respaldo de baterías, e
6.- Interfase Pc.

La unidad de control es la parte fundamental, ya que en ella se encuentra toda la circuitería de control y la información de todos los parámetros de configuración; del buen funcionamiento de ésta depende el éxito del sistema.

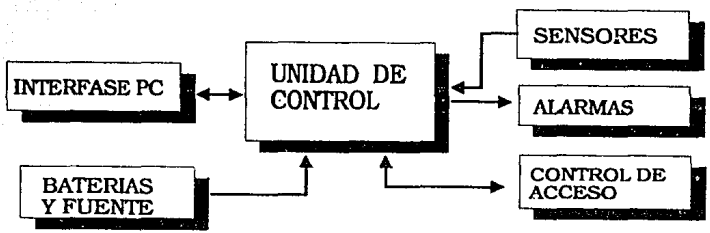


FIG. 3.1 DIAGRAMA DE BLOQUES DE SISTEMA.

Se tiene capacidad para supervisar hasta 15 mallas de sensores, las cuales están formadas por sensores de intrusos e incendios, que al activarse abren o cierran un par de contactos o entregan un nivel lógico. La información proporcionada por dichos sensores es recogida por la unidad de control que ejecuta la acción correspondiente.

Los indicadores de alarma son dispositivos que emiten señales audibles o visuales, por medio de las cuales, el sistema indica cuando es detectado algún intruso o incendio dentro de la zona protegida.

Todo sistema que consume energía eléctrica generalmente depende de la compañía de luz, pero dado que ésta puede interrumpir su servicio por fallas en sus líneas de distribución o transmisión, nuestro sistema garantiza su operación continua durante un período de 12 horas sin activación de señales de alarma. Esto se logra a través de un respaldo de baterías.

El control de acceso es una de las partes importantes del sistema, por esta razón, se analizan varios parámetros, tales como, tiempo máximo requerido para que una persona normal logre el acceso, el tiempo máximo para teclear su clave, etc. El acceso a la zona protegida se logra a través de teclear un código que identifica al solicitante.

La comunicación con la PC se lleva a cabo por medio del puerto serie (com 1) de la computadora a una velocidad de 9600 bauds. El propósito de la comunicación con la PC es básicamente con fines estadísticos.

3.1 UNIDAD DE CONTROL

La unidad de control está formada básicamente por un *microcontrolador (mC)* de la familia mcs-51 de Intel. Pero además, se tienen dispositivos adicionales como son: memorias para el manejo de información y configuración, un "display" alfanumérico para visualizar las condiciones de operación y para auxiliar en la programación, ya que todos los parámetros que se manejan son programables. También, se tiene un circuito de auto prueba, el cual se encarga de vigilar la correcta operación del sistema.

La unidad de control está integrada, también, por un convertidor analógico digital (A/D) con ocho canales multiplexados de 8 bits para sensar variables analógicas. Además, se tiene una interfase paralelo (PPI 8255) para incrementar el número de líneas de entrada/salida (E/S), así como, circuitos para la comunicación RS-232C con la PC.

En los últimos años el uso de los microcontroladores se ha visto incrementado dada su flexibilidad y al poco espacio que ocupan dentro de una tarjeta electrónica, pues en una sola pastilla se tiene una unidad central de procesos (CPU), memoria de varios tipos, contadores y

temporizadores, puertos de entrada salida y algunos otros circuitos, por ejemplo, puerto serial, etc., dependiendo de la versión.

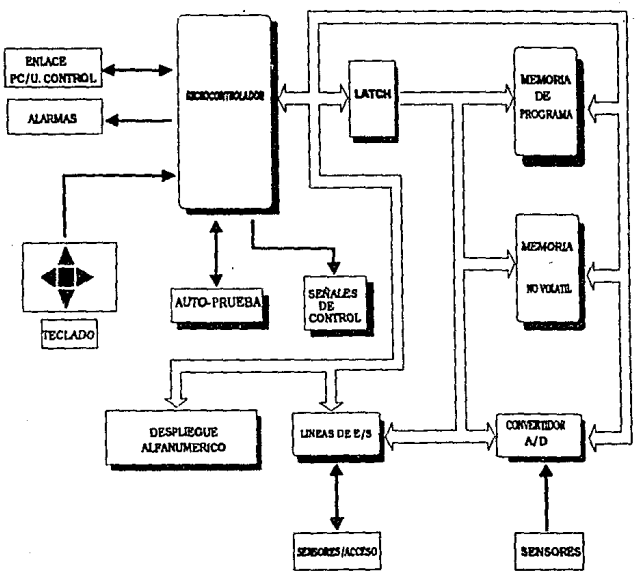


FIG. 3.1.1 Diagrama de bloques de la Unidad de Control

En el sistema desarrollado se utilizó el microcontrolador 8031, el cual pertenece a la familia MCS - 51 de Intel que en general tienen las siguientes características principales: CPU de 8 bits, 4 puertos de entrada/salida, 128 bytes de memoria RAM dentro de la misma pastilla, además, cuentan con puerto serie de comunicación full duplex y capacidad para direccionar hasta 64k de memoria. En la figura 3.1.2 se muestra un diagrama de bloques de esta familia de microcontroladores.

En el diseño, como ya se dijo, se busca reducir el costo es por eso que se utiliza el 8031, ya que es un *microcontrolador* que no cuenta con memoria de programa interna ("EPROM") y esto hace que su costo sea bajo.

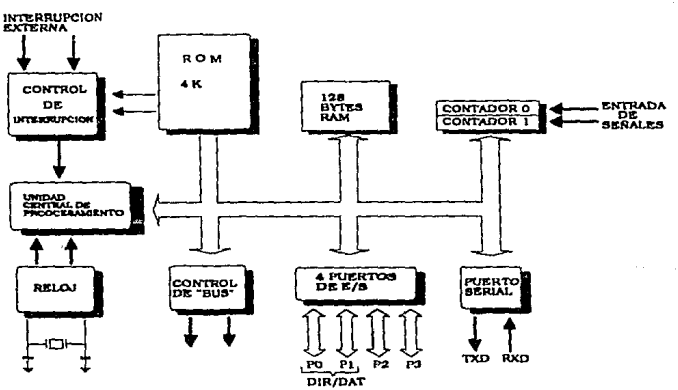


Fig.3.1.2 Diagrama de bloques del 8051

3.1.1 MEMORIAS

Los procesamientos digitales requieren generalmente de guardar información, y con éste propósito se utilizan memorias. Las memorias son dispositivos que pueden clasificarse en dos categorías: Volátil y No volátil. Las memorias de tipo no volátil retienen la información aun si la energía que las alimenta desaparece. Estas memorias son utilizadas para guardar información que posteriormente será requerida. Dentro de las memorias no volátiles se incluyen los discos y cintas magnéticas, pero en el desarrollo de este trabajo se hace referencia a las memorias del tipo semiconductor.

1) MEMORIA VOLATIL (R/W)

Las memorias volátiles permiten la lectura y escritura, el termino de memoria volátil esta asociado a las memorias RAM ("Randon Acces Memory"). Existen actualmente dos tipos de memoria RAM, la RAM estática y la memoria RAM dinámica.

La memoria RAM dinámica es de alta capacidad y bajo consumo, además de tener un costo menor. La carga en este tipo de memorias, es fijada generalmente por un capacitor y dependiendo de la carga será el nivel lógico que se interprete, pero como el capacitor no es ideal, tiende a perder su carga, ésto hace necesario tener un ciclo de refresco.

La memoria RAM estática está construida por lógica de flip-flops y por lo tanto no requiere de un ciclo de refresco. En ambos tipos de memoria la información almacenada se pierde si se elimina la fuente de alimentación.

2) MEMORIA NO VOLATIL

Las memorias ROM ("Read Only Memory") se utilizan para fijar información que no esta sujeta a cambios, en este tipo de memorias si desaparece la energía que los alimenta estas conservan la información almacenada. Generalmente a esta memoria se le conoce como memoria de programa y a la RAM como memoria de datos.

Las primeras ROM'S contenían arreglos de celdas secuenciales de 1's y 0's por conexión metálica durante su fabricación, posteriormente aparecen las PROM que ya pueden ser programadas y que estaban hechas a base de fusibles; actualmente existen las EPROM que pueden ser programadas y borradas por el usuario, el borrado se realiza exponiéndolas a rayos ultravioleta. Ahora también, se tienen memorias del tipo PROM, pero que pueden ser borradas electricamente EEPROM.

El sistema desarrollado cuenta con memoria de programa, de datos y memoria no volátil. La memoria de programa lo constituye una EPROM de 8K X 8 (2764), la de datos está formado por una RAM que incluye el microcontrolador. La parte de memoria no volátil está formada por dos memorias, una RAM no volátil (X2444) y una EEPROM (2816).

a) MEMORIA DE PROGRAMA

La memoria de programa de la Unidad de Control está formada por una EPROM 2764, cuyas características se enumeran a continuación:

Tiempo de acceso ------ 150 ns.
Polarización ---------- 5 V.
Capacidad ------------ 65,536 bits.

En ella se encuentra grabada toda la secuencia de instrucciones ejecutadas por el 8031. En la figura 3.1.3 se muestra un diagrama de bloques de esta memoria.

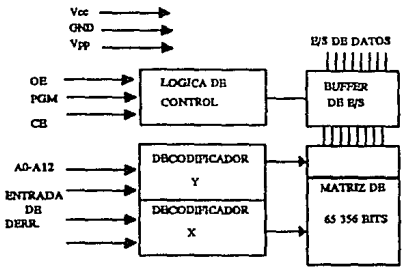


Fig. 3.1.3 Diagrama de bloques de la EPROM 2764.

b) DESCRIPCION DE LA MEMORIA X2444

La memoria X2444 es una RAM no volátil de Xicor y consta de 256 bits. Dicha memoria es de tipo serial configurada en un arreglo de 16 X 16 bits sobrepuestos a una EEPROM; está fabricada con tecnología MOS de canal N. La transferencia de datos entre las dos memorias (RAM,EEPROM) se puede llevar a cabo por medio de "software" o "hardware".

La X2444 opera de la siguiente manera, los datos que requieren ser guardados en la EEPROM primero se cargan en la RAM y posteriormente por "software" o "hardware" se bajan a la

EEPROM, y cuando los datos de la EEPROM requieren ser utilizados, entonces se copian a la RAM. Esta memoria está diseñada para un número ilimitado de operaciones (escritura) en RAM, así como el de recargar los datos a la RAM desde la EEPROM. La fijación de datos está garantizada para 100,000 operaciones de escritura en EEPROM y la información almacenada se garantiza por 100 años. En la figura 3.1.4 se muestra un diagrama funcional de bloques.

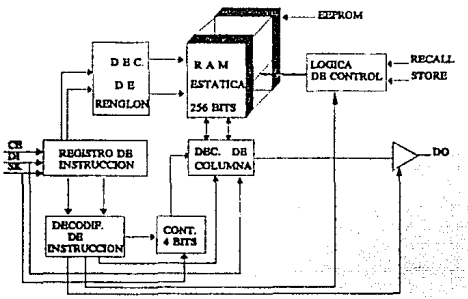


Fig 3.1.4 Diagrama funcional de la X2444.

CARACTERISTICAS ELECTRICAS:

Polarización ---------- 5 V
Corriente activa ---------- 15 mA
Corriente de espera ---------- 6 mA
Corriente en modo de reposo -- 5 mA

En esta memoria, el pasar los datos de la RAM a la EEPROM y viceversa se puede realizar por "hardware" o "software" como se muestra en la tabla 3.1.1.

OPERACION	STORE	RECALL	INST.
"hardware" *Recall*	1	0	_____
"software" *Recall*	1	1	RCL
"hardware" *Store*	0	1	_____
"software" *Store*	1	1	STO

Tab. 3.1.1

Store: Operación de cargar el contenido de la RAM en la EEPROM.
Recall: Operación de cargar los datos contenidos en la EEPROM a la RAM.

Las instrucciones se transmiten en forma serial como se muestra en la figura 3.1.5. La instrucción de *Sleep* remueve la polarización de la RAM, poniendo a la memoria en estado de bajo consumo (reposo) y los datos de la RAM se pierden. Para salir de este modo es necesario recargar a la RAM con los datos de la EEPROM mediante la instrucción de recarga (*Recall*).

Dadas las características del puerto serie del 8031 y de la memoria, se encontró que éste podía accesar directamente a la memoria a través de su puerto serie trabajando éste en modo cero. En la figura 3.1.6 se muestra la interconexión de la memoria con el microcontrolador.

INSTRUCCION	FORMATO I2 I1 I0	OPERACION
WRDS	1 X X X X 0 0 0	Habilita la escritura
STO	1 X X X X 0 0 1	Pasa el contenido de RAM en EEPROM
SLEEP	1 X X X X 0 1 0	Modo de reposo
WRITE	1 A A A A 0 1 1	Escribe en Ram en la dir. AAAA
WREN	1 X X X X 1 0 0	Habilita la escritura
RCL	1 X X X X 1 0 1	Pasa el contenido de EEPROM a RAM
READ	1 A A A A 1 1 X	Lee la dir. AAAA de RAM

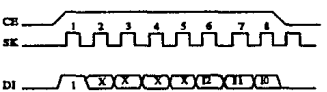


Fig. 3.1.5 Instrucciones de la memoria X2444.

Las operaciones de la memoria se realizan por "software", con el fin de no incrementar más el "hardware" del sistema. El conectar la memoria por el puerto serie del 8031 hace necesario modificar las instrucciones. Esto se debe a que cuando el puerto serie está inactivo, la línea de recepción de datos (RXD) se encuentra en nivel alto ("1"), entonces cuando se habilita la memoria, este uno lógico es tomado como primer dato. Además, el puerto serie transmite primero el bit menos significativo. El formato de instrucciones quedó de la forma que muestra la tabla 3.1.2.

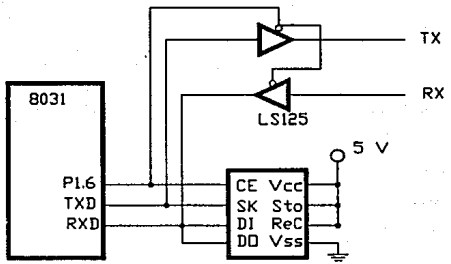


Fig. 3.1.6 Interconexión entre la X2444 y el 8031.

Instrucción	7	6	5	4	3	2	1	0 (bits)
sleep	x	0	1	0	x	x	x	x
store	x	1	0	0	x	x	x	x
write	x	1	1	0	A	A	A	A
wren	x	0	0	1	x	x	x	x
recall	x	1	0	1	x	x	x	x
read	1	x	1	1	A	A	A	A

Tab 3.1.2.

El puerto serie del *microcontrolador* en el modo cero, genera 8 pulsos que sirven de reloj a la memoria, estos pulsos se transmiten por la línea TXD cuando un dato es cargado en el SBUF para ser enviado a la memoria y también cuando la bandera de interrupción (TI) del puerto serie es puesta a cero, para recibir datos.

c) DESCRIPCION DE LA MEMORIA EEPROM

La EEPROM utilizada es una memoria de Samsung (2816) que tiene una capacidad de 16,384 bits. Esta memoria es completamente borrable y programabre utilizando únicamente una fuente de 5V aunque también tiene la opción de alto voltaje.

En la figura 3.1.7. se muestra el diagrama de bloques de esta memoria.

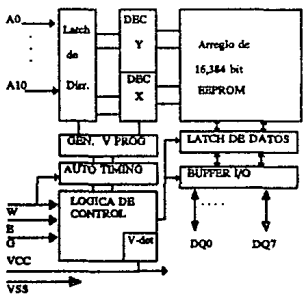


Fig.3.1.7. Diagrama de bloques de la memoria 2816.

CARACTERISTICAS

- 5 V Borrar/escribir /leer.
- Protección contra arranque y paro de la fuente de alimentación.

- 300 ns tiempo máximo de acceso.
- 110 mA corriente activa.
- 50 mA corriente de spera.
- Opción de alto voltaje.
- Genera los tiempos necesarios para su operación internamente.
- Sólo tiene 2 líneas de control.

3.1.2 CONVERTIDOR ANALOGICO/DIGITAL (A/D)

Una señal digital es una forma de onda que tiene transiciones abruptas entre dos valores. La señal que puede asumir varios valores en el rango continuo se denomina señal analógica. Cuando una señal analógica requiere ser procesada es muy común se digitalize para facilitar su procesamiento.

El procedimiento de convertir una señal analógica a su equivalente digital, involucra una secuencia individual de procesos llamados **muestreo, retenimiento, cuantización y codificación.** Estos procesos no siempre se ejecutan de manera separada, generalmente el muestreo y retenimiento los realiza un sólo circuito, la cuantificación y codificación otro.

TEOREMA DEL MUESTREO

Para que el procesamiento de una señal analógica en su equivalente digital sea válido, depende fundamentalmente del teorema del muestreo.
El teorema del muestreo establece lo siguiente:
Una señal analógica de banda limitada a B Hz, queda limitada por sus dos valores a intervalos uniformes con separación menor de 1/(2B) segundos. El teorema establece que una señal de

banda limitada **B**, puede reconstruirse a través de sus muestras tomadas uniformemente a una razón no menor de 2 **B** muestras por segundo.
Si Ts es el perfodo de muestreo, entonces TS <= 1/(2 B) o la razón de muestreo sea mayor que 2B muestras por segundo. El máximo permisible, Ts = 1/(2 B), se conoce como intervalo de **Nyquist**.

La señal analógica puede recuperarse a través de su equivalente muestreada haciéndola pasar por un filtro paso-bajas de ganancia Ts y ancho de banda **B**.

EL CONVERTIDOR A/D ADC0809

El ADC0809 es un convertidor analógico digital de 8 bits de tecnología CMOS con 8 canales multiplexados. El multiplexaje es controlado por tres bits de direccionamiento, este dispositivo realiza la conversión analógica digital (A/D) por aproximaciones sucesivas. En la figura 3.1.8 se muestra un diagrama funcional de bloques del ADC0809.

El multiplexor analógico selecciona uno de los ocho canales a través del decodificador de direcciones, con una transición de bajo a alto de la señal de inicio de conversión (start). El registro de aproximaciones sucesivas (SAR) se inicializa, también, con el flanco positivo de ésta señal.

Con el flanco de subida de la señal de inicio se toma la muestra a convertir. La conversión puede ser interrumpida por el nuevo pulso de la señal de inicio antes de 64 ciclos de reloj.

El detector de umbral en el sistema de conversión por aproximaciones sucesivas, determina cada bit al examinar el voltaje de una serie de resistencias de ponderación binaria (red de 256 R). En primer lugar, la entrada analógica es muestreada y al mismo tiempo se inicia el proceso de examinar el voltaje en las resistencias de ponderación. En el siguiente paso del proceso de conversión, el detector de umbral inicia la identificación del valor de voltaje relativo a la referencia de voltaje. En la figura 3.1.9 se muestra el diagrama de tiempos de este convertidor.

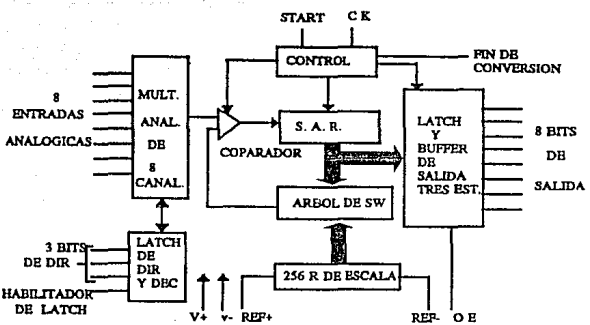


Fig 3.1.8 Diagrama de bloques del ADC0809.

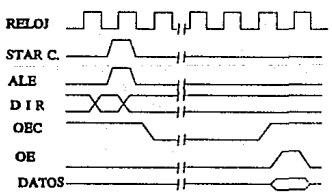


Fig. 3.1.9 Diagrama de tiempos del ADC0809.

El reloj (CK), opera en el rango de 20 Khz a 1.2 MHz, esta señal se pensó obtener de varias fuentes, como poner un oscilador exclusivo o utilizar las señales de ALE y PSEN que genera el 8031. Después de analizar estas últimas, se encontró que estaban presentes todo el tiempo y dentro del rango permitido por lo que no sería necesario incluir un divisor de frecuencia. Finalmente, el convertidor quedó conectado a la línea de PSEN del *microcontrolador*, ya que presenta un ciclo de trabajo del 50% aproximadamente. La señal PSEN se activa siempre que se hace un acceso a la memoria de programa (2764) y esto ocurre dos veces en cada ciclo de máquina. El ciclo de máquina esta definido por el cristal del sistema que es de 7.3728 MHz, pero cada ciclo requiere de 12 períodos de reloj, por lo que la señal de PSEN tiene una frecuencia de 1.2 MHz. En la figura 3.1.10 se muestra la interconexión entre el microcontrolador y convertidor.

La referencia positiva se conecta a 5 V, como puede verse en la figura, mientras que la referencia negativa del convertidor es conectada a una referencia de voltaje de 2.5 V (LM 336); con lo que se logra una resolución de 9.76 mV, la razón de conectar en ésta forma la referencia de voltaje, es por que se necesita tomar constantemente lecturas de un sensor de temperatura que entrega mediciones en grados absolutos (grados Kelvin), por lo que a 0°C se tiene un valor de 2.73 Volts con una resolución de 10 mV/ °C. Esto hizo necesario incluir la referencia (LM 336) para poder sensar incrementos de temperarura de un grado.

3.1.3 LINEAS DE ENTRADA/SALIDA (E/S)

En un sistema electrónico, las líneas de entrada/salida son el medio por el cual la Unidad Central de Procesos se comunica con el medio exterior. En el sistema, estas líneas son suministradas por una interfase paralelo PPI 8255 que proporciona 24 líneas y por el puerto 1 del microcontrolador (P1.0-P1.7) que suma 8 líneas, dando un total de 32 líneas de E/S.

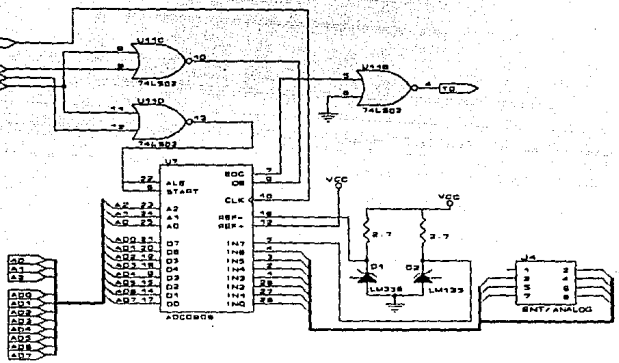


Fig. 3.1.10 Diagrama que ilustra la conexión del convertidor A/D con el *microcontrolador*.

El PPI 8255 se utiliza para supervisar a los diferentes sensores, al teclado de acceso y el usado para programar el sistema. Los sensores se colocan en las líneas de los puertos A y B que se definen como entradas. Cada línea de ambos puertos tiene un nivel lógico fijo dado por una resistencia de 15K hacia el positivo de la fuente, esto hace que el nivel de entrada sea siempre "1".

Generalmente, los sensores tienen contactos normalmente cerrados, cuando alguno se activa abre el circuito. En el puerto cuando el sensor no se encuentra activo el *microcontrolador* lee

siempre un valor lógico de "0", pero en el momento que el sensor se activa el circuito se abre y entonces leerá un valor de "1" .

El puerto C del 8255 es dividido en dos partes, la parte baja (PC.0 - PC.3) es definida como salidas y la parte alta (PC.4-PC.7) como entradas. Este puerto se utiliza para supervisar tanto al teclado de acceso como a cuatro teclas en la unidad de control que sirven para programar al sistema.

3.1.4 MODULO DE CRISTAL LIQUIDO AND 771.

El "display" alfanumérico se utiliza para obsevar las condiciones de operación del sistema, dispone de dos líneas de 24 caracteres cada una.

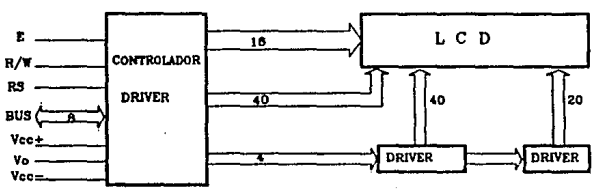


Fig. 3.1.11 Diagrama de bloques del módulo AND 771.

El módulo AND 771 incluye una matriz de cristal líquido, un *microntrolador* y memoria RAM. Con este módulo se pueden manejar caracteres gráficos y alfanuméricos según el código

ASCII (*American Standard Code for Information Interchange*) con un patrón de 5 X 8 puntos. En la figura 3.1.11 se muestra un diagrama de bloques de este dispositivo.

Tiene dos registros de 8 bits, el registro de instrucción y el registro de datos. El registro de instrucción es sólo de lectura y en el se encuentran los códigos de comandos, tales como limpiar, corrimiento del cursor, etc. El registro de datos almacena temporalmente los datos que serán escritos en la RAM de datos o en la RAM generadora de caracteres. Los datos escritos en el registro de datos son enviados inmediatamente, también en este registro se almacenan los datos que serán leídos por el dispositivo externo, la dirección del dato a leer es almacenado en el registro de instrucción y el dato en el registro de datos.

La señal *RS* es la encargada de seleccionar el registro dependiendo de la operación que se desee realizar. En la tabla 3.1.3 se muestra la operación básica del módulo de cristal líquido.

Señal	Función
RS	"1" : Entrada de datos "0" :entrada de comandos
E	"1": Habilita al módulo.
Vo	Voltaje de contraste del display.
R/W	"1": Lectura de datos (módulo → CPU "0":escritura de datos (CPU → módulo)

Tab. 3.1.3

Para sincronizar el 8031 con el módulo de cristal líquido se tuvieron algunos problemas, ya que las señales no son compatibles. La señal de lectura y escritura del display se debe activar primero que la de habilitación (E), como se puede ver en la figura 3.1.12., en el *microcontrolador* las señales R/W se presentan después que la señal de habilitación. También, al mismo tiempo que la señal R/W se debe presentar al módulo la señal de RS que le indica el

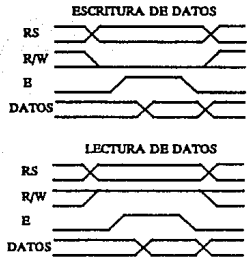


Fig. 3.1.12 Esquema de tiempos del módulo AND 771.

tipo de dato siguiente, que puede ser un dato o un comando. Por estas razones, fue necesario utilizar algunas compuertas lógicas para proporcionar la secuencia correcta de las señales. La interconexión del módulo con el *microcontrolador* del sistema es mostrado en la figura 3.1.13. En esta figura puede verse como las señales RS y R/W son proporcionados por las direcciones A0 y A1, la señal de habilitación (E) por una combinación de *RD* y *WR*, y por *Y2* del decodificador de periféricos 74LS138 después de que las compuertas *AND* y *NOR* ejecutan la operación lógica de la ecuación 3.1.2 para proporcionar la secuencia correcta de esta señal.

$$E = (RD'WR' + Y2')' \qquad \text{.....(3.1.2)}$$

3.1.5 AUTO PRUEBA (AUTO-RESET)

La operación de un sistema de seguridad debe ser confiable con el fin de reducir el número de falsas alarmas y evitar que sea saboteado fácilmente. El garantizar el correcto funcionamiento

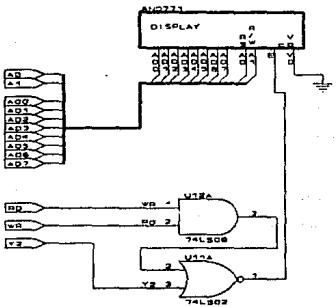


Fig. 3.1.13. Diagrama que ilustra la conexión del "display" con el 8031.

implica detectar cuando su operación es anormal y forzarlo a volver a su secuencia correcta de programa. El sistema desarrollado cuenta con un dispositivo de auto-prueba, el cual permite que éste siempre funcione correctamente.

El circuito de auto-prueba supervisa el funcionamiento del *microcontrolador* y de sus dispositivos periféricos. El *microcontrolador* se puede salir de su secuencia de programa si existe una fuente de ruido muy grande, que en un momento determinado pudiera ser interpretado como un nivel lógico, ejecutando códigos de operación incorrectos que llevarían el sistema a la falla. Otro de los factores que pueden afectar es el ruido en la fuente de alimentación, generado por las transiciones de encendido y apagado que pueden darse accidentalmente o por la acción de un intruso experimentado. El circuito de auto-prueba detecta cuando la operación del es incorrecta y genera la señal de inicio (*Reset*).

El dispositivo implementado pregunta continuamente al *microcontrolador* por sus condiciones de operación y la de sus periféricos, éste debe responder con una señal que el dispositivo de auto-prueba evalúa; si la señal no concuerda con la esperada se genera el pulso de inicio (*Reset*), ya que implica que se ha salido de su secuencia de programa.

Existen dos formas de inicializar al sistema, una de ellas es dada por "software" y la otra "hardware". El inicio por "software" se da cuando el *microcontrolador* sigue en secuencia correcta, pero algunos de sus periféricos ya ha perdido sus parámetros de programación, entonces el *microcontrolador* salta a su dirección de inicio (00h). La otra forma es mediante "hardware", es decir cuando el circuito de auto prueba pregunta al *microcontrolador* sobre sus condiciones y este definitivamente no responde o lo hace de manera incorrecta.

El inicio por "hardware" esta formado por un circuito de lógica combinacional que opera de la forma siguiente: Un oscilador independiente al reloj del sistema genera una señal que interrumpe al *microcontrolador* 5 veces por segundo, la señal de este reloj se aplica también por P1.5. El *microcontrolador* contesta a las interrupciones con una serie de pulsos defasados 180 grados respecto al oscilador que provoca las interrupciones, a través de P1.7; si el defasamiento no es correcto se genera la señal de inicio.

El dispositivo está formado por un oscilador de aproximadamente 2.5 hertz y un grupo de compuertas lógicas. Para generar las interrupciones, los pulsos de 2.5 Hertz se dafasan 180 grados a través de la compuerta XOR(U10A) como se muestra en la figura 3.1.14. Los pulsos invertidos se retardan por medio del capacitor C3, posteriormente esta señal invertida y retardada se aplica a la compuerta XOR(U10B) que genera los pulsos que interrumpen al *microcontrolador* por *INT 0*. A pesar de que el reloj es de 2.5 Hertz utilizando la lógica descrita el 8031 es interrumpido en aproximadamente 5 veces por segundo, el doble de la frecuencia del

oscilador. Cuando el *microcontrolador* es interrumpido inmediatamente debe responder con un pulso negativo a la señal de reloj aplicada en P1.5. La señal que entrega es comparada con el reloj por medio de la compuerta XOR (U10C), si ambas señales están defasadas 180 grados, una respcto a la otra, se produce una salida de "1" que no genera la señal de inicio, pero si están en fase o defasadas cualquier otro ángulo, entonces se tiene un nivel de "0" que es equivalente al pulso de inicio (*Reset*); este nivel es invertido por la compuerta XOR (U10D) configurada como inversor dado que el *microcontrolador* interpreta como señal de inicio a un nivel alto.

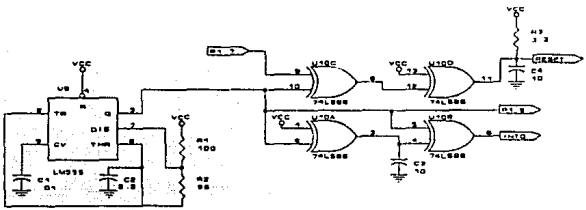


Fig 3.1.14. Circuito de auto-prueba.

El oscilador esta formado por el multi-vibrador 555 en configuración astable. La forma astable de este circuito genera una serie de pulsos cuya frecuencia es determinada por un par de resistencias y un capacitor, según la ecuación 3.1.2.

$$f=\frac{1.44}{(R_1+2R_2)\,C_1} \quad \ldots\ldots\ldots\ldots (3.1.2)$$

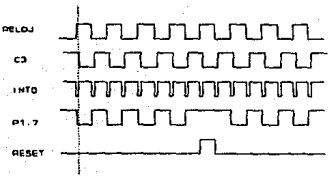


Fig. 3.1.15 Diagrama de tiempos del circuito de auto-prueba.

El inicio por "software" se debe a que algunos de los dispositivos periféricos como el PPI 8255 y el módulo de cristal líquido (AND 771) requieren ser programados previamente para poder operar correctamente. Si por alguna razón pierden sus parámetros de configuración, el *microcontrolador* (8031) no va ha accesarlos correctamente. Para evitar esta situación, cada vez que el *microcontrolador* se interrumpe por el circuito de auto prueba, carga nuevamente la palabra de programa en el 8255 y posteriormente lee un carácter en el display, el cual se ha escrito en la interrupción anterior. Si el carácter leído corresponde al escrito anteriormente, entonces sólo actualiza y continua, de lo contrario salta a la dirección 00h e inicia nuevamente el programa.

3.1.6 PROGRAMACION AUTONOMA

Como ya se dijo, el sistema es completamente programable de forma autónoma o a través de la computadora (PC). Si no se dispone de una PC, entonces todos los parámetros son programados utilizando cinco teclas localizadas en la parte frontal de la Unidad de Control

Dicho tablero está formado por el "display" alfanumérico, señalizadores de condición de fuente de alimentación y cinco teclas con aspecto de punta de flecha, como puede observarse en la figura 3.1.16.

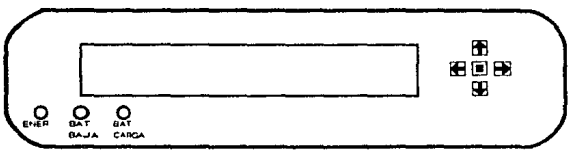


Fig. 3.1.16 Tablero de la unidad de control.

Las teclas realizan varias funciones durante la programación que depende de la opción seleccionada. Aquellas equivalentes a las flechas son manejadas por el *microcontrolador* a través del puerto C del 8255 y la tecla central se conecta directamente al 8031 por la interrupción externa 1 (INT 1) como se muestra en la figura. 3.1.17.

1) IDENTIFICACION DE TECLA OPRIMIDA

El teclado es uno de los medios por el cual el usuario se comunica con la máquina, por lo que es muy importante saber en que momento y que tecla se oprime. Las teclas son contactos mecánicos normalmente abiertos (N.A.) que debido a su fabricación en el momento en que son presionados presentan una serie de rebotes, los cuales deben ser eliminados para no provocar errores en la identificación de la tecla, la eliminación se puede hacer de dos formas, por "software" o "hardware".

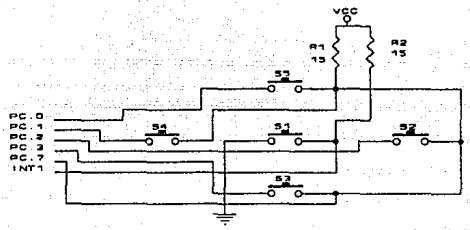


Fig. 3.1.17. Teclado del tablero de control.

Eliminarlos por "hardware" implica incrementar el número de componentes del sistema y por consiguiente un aumento en su costo. En el sistema se eliminan por "software", mediante el cual, si se detecta que alguna tecla se encuentra oprimida, entonces se ejecuta una rutina de retardo en la que se calcula el tiempo en pasen todos los transitorios y posteriormente, si aún se encuentra oprimida, se procede a su identificación, de esta forma también se garantiza que la señal detectada no sea ruido.

La identificación de las teclas se realiza utilizando el método de poleo ("polling"), en el cual se rota un cero (lógico) por un extremo del contacto de cada una de las teclas que se desee identificar.

En el instante **t1** se pone un "0" en la línea PC.0 y se lee PC.7, si la tecla está oprimida, en PC.7 se leerá también un "0" de lo contrario se tendrá un "1". En **t2** se pone ahora PC.1 a cero y se pregunta si PC.7 está también a cero, de ser así, se identifica la tecla de lo contrario se continua con la búsqueda. Esta forma de búsqueda se ejecuta siempre que el sistema se encuentre en operación. La figura 3.1.18 muestra lo descrito anteriormente.

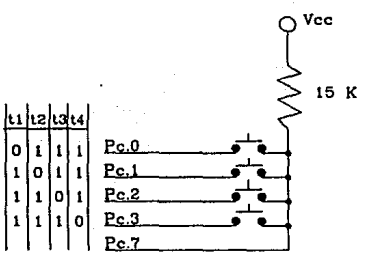


Fig. 3.1.18 Identificación de tecla.

El *microcontrolador* interactua con sus diferentes periféricos mediante direcciones, ya que no cuenta con una señal que identifique la comunicación entre memoria y puertos. En la tabla 3.1.4 se muestra el mapa de direcciones del sistema.

DIRECCION	DISPOSITIVO
0000H 1FFFH	MEMORIA DE PROGRAMA
2000H 2FFFH	MEMORIA EEPROM
4000H 4003H	DISPLAY ALFANUMERICO
6000H 6007H	CONVERTIDOR ANALOGICO/DIGITAL
8000H 8003H	PPI 8255

Tab. 3.1.4 Mapa de dir. del sistema

3.1.7 CLAVE DE SEGURIDAD

El sistema cuenta con una clave de SEGURIDAD que permite sólo a la persona autorizada modificar los parámetros de operación. La clave es una forma de proteger al sistema, pues es necesario introducirla correctamente cada vez que se desee cambiar algún parámetro. Dicha clave puede ser cambiada cuando así se requiera a través de activar una cerradura, para la cual se requiere de una llave específica. Esta cerradura se localiza en la parte lateral de la unidad de control.

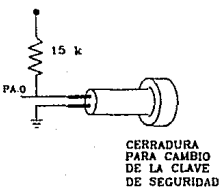


Fig. 3.1.19

Cuando la llave abre la cerradura, pone la línea PA.0 del 8255 a "0" como puede verse en la figura 3.1.19. En el instante en que este bit es leído, el mC reconoce una petición de cambio de clave de seguridad, enviando un mensaje al display para indicar al usuario que el sistema ya esta listo para recibir la nueva clave.

Terminada la operación, se regresa la llave a su pocición original quedando la nueva clave programada en el sistema. La información de la clave de seguridad se almacena en la RAM no volátil X2444, junto otros parámetros de configuración.

3.2 SENSORES Y DETECTORES

Los sensores utilizados actualmente en sistemas de seguridad, tienen la característica que cuando se activan sólo abren o cierran un par de contactos o en su defecto entregan un nivel lógico. Entre los éstos se tienen a los sensores de humo, de movimiento, contactos magnéticos, etc. Todos los ajustes necesarios se realizan dentro del módulo sensor, estas características permiten formar arreglos de sensores para incrementar el área de cobertura y sólo supervisar una señal. Los arreglos que pueden formarse son serie ó paralelo en el caso de que la salida sea un relevador.

Cuando se tienen sensores cuyos contactos son normalmente abiertos (N.A.) se pueden formar mallas con sensores conectados en paralelo como se muestra en la figura 3.2.1. Si los sensores son del tipo normalmente cerrados (N.C), entonces pueden configurarse mallas con sensores conectados en serie como muestra la figura 3.2.2.

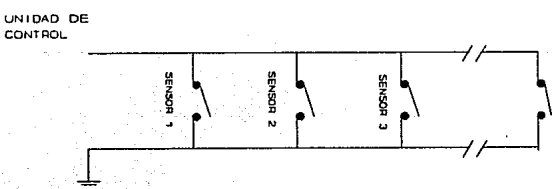


Fig. 3.2.1. Sensores conectados en paralelo.

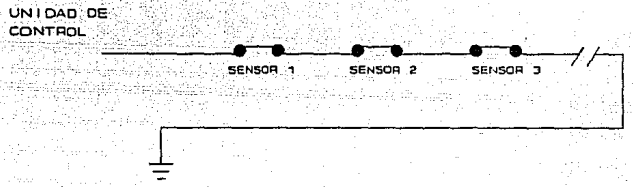


Fig 3.2.2. Sensores conectados en serie.

En las mallas no se conoce exactamente cual es el sensor que se encuentra activado, pero se sabe que en el área cubierta por dicha malla se ha detectado un intruso o un incendio, según sea el caso. Cuando se requiera saber específicamente el sensor que se activa es necesario supervisar individualmente a cada uno.

El sistema que se describe en el presente trabajo, cuenta con 15 mallas programables para sensores normalmente abiertos o cerrados, las cuales son supervisados por el *microcontrolador* a través del los puertos A y B del 8255 configurados como entradas. Cada línea dedicada a supervisar sensores tiene una resistencia a Vcc que fija en nivel de entrada a 5 V. Cuando los sensores que forman una malla son N.C. en operación normal el *microcontrolador* leerá en esa línea un nivel de "0" lógico; cuando en la malla se activa algún sensor, entonces se abre el circuito y en la línea de entrada del 8255 aparecerá el nivel de Vcc. Este tipo de sensores son utilizados cuando se requiere detectar intrusos, ya que si se cortan los alambres conductores de todos modos abre el circuito generando el mismo efecto que si se activara.

En el caso que la malla este formada por sensores N.A. en operación normal se tendrá en la línea de entrada del 8255 el valor de Vcc y cuando un intruso o incendio active algún sensor se

tendrá un "0" lógico. Estos sensores son comunes cuando el sistema requiere de detectar incendios.

A continuación se describen algunos de los sensores comerciales utilizados en las pruebas del sistema.

3.2.1 SENSOR PASIVO INFRARROJO

El sensor pasivo infrarrojo de movimiento (PIR) XJ-660 de Intellisense, tiene un denso patrón de haces con una cobertura de 18m x 18m, con 37 zonas ópticas, incluyendo seis zonas de sabotaje. Además, cuenta con compensación automática de temperatura, inmunidad a la luz blanca e interferencia por radio frecuencia (RFI) y un consumo menor que 20 mA a 12 V. En la figura 3.2.3 se muestra el aspecto físico de este sensor.

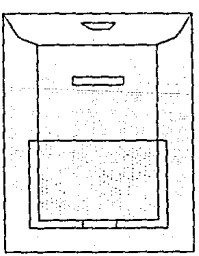


Fig. 3.2.3 Sensor pasivo infrarrojo XJ-660.

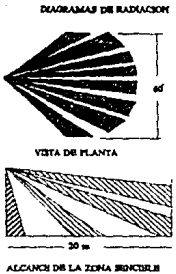


Fig. 3.2.4 Patrón de radiación (PIR) del XJ-660.

El sensor XJ-660 está diseñado para usarse en interiores, la zona de detección deberá estar libre de objetos, ya que la radiación infrarroja no puede penetrar objetos sólidos. De ser posible el sensor debe estar apuntando hacia los interiores lejos de las fuentes de calor.

Este dipositivo cuenta con ajuste del área de inspección mediante el deslizamiento de la tarjeta de circuito impreso, es decir el ajuste se hace cambiando la posición del elemento sensor. El rango de alcance está determinado por la posición del elemento sensor dentro del módulo y la altura a que se encuentre instalado dicho módulo. La tarjeta de circuito impreso tiene una serie de marcas que determinan la posición del elemento sensor dentro del módulo, mediante una tabla que relaciona la posición del elemento sensor y la altura del módulo se encuentra el rango de alcance. En la figura 3.2.5 se muestra las marcas y la tabla equivalente al alcance del sensor.

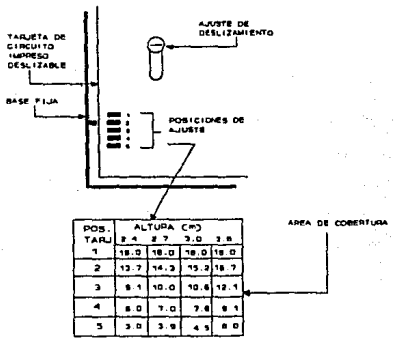


Fig. 3.2.5 Cobertura del sensor.

Características:

- Rango
  18m X 18m

- Relevador de alarma
  Normalmente cerrado
  100 mA, 30 V dc

- Alimentación
  6 - 14 V dc

- Inmunidad a RFI
  100 watts a 1.5 m
  100 a 1000 Mhz

- Campos de visión del PIR
  22 zonas lejanas
  6 Intermedias
  3 zonas cercanas
  6 zonas de sabotaje

- Dimensiones
  13 cm x 7 cm x 6 cm

- Peso
  226.4 g

2.2.2 SENSOR DE HUMO:

El sensor de humo utilizado, es el modelo SA91CH de Firt Alert, éste se alimenta con una batería de 9 V en su forma original, pero al incorporarlo al sistema se hicieron algunas modificaciones. El módulo sensor tiene una cámara de ionización para detectar el humo producto de una combustión. Además cuenta con una serie de circuitos de control y "zumbador" piezoeléctrico como alarma sonora.

El SA91CH, no detecta la presencia de gas, calor o flama, el detector supervisa el aire y cuando la cantidad de humo sensado rebasa una cierto límite inmediatamente se inicia la alarma.

La alimentación de 9 V que requiere el sensor se toma de la fuente del sistema. En este caso el sensor no cuenta con relevador para indicar cuando se activa, pero entrega un valor de voltaje proporcional a la cantidad de humo detectado, se adicionó un comparador de voltaje para obtener un valor lógico para indicar cuando rebasó ya la cantidad de humo permitido y activar la alarma general. En realidad la alarma que incluye el sensor se utiliza como alarma local para indicar el sitio preciso donde se detectó el humo. En la figura 3.2.6 se muestra la conexión de este sensor con la unidad de control.

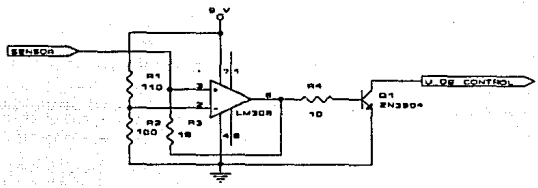


Fig. 3.2.6 Acondicionamiento de la seña del sensor de humo.

2.2.3 CONTACTOS MAGNETICOS

Es muy importante detectar cuando alguna puerta o ventana es abierta sin autorización, pues podría tratarse de un intruso que ha logrado introducirse hasta el edificio burlado los dispositivos de seguridad anteriores. Una forma de hacerlo es utilizando contactos magnéticos.

Los contactos magnéticos empleados son de la marca *Sentrol*, los cuales están formados por un par de pequeñas laminillas metálicas separadas una cierta distancia y encapsulados en una bombilla de vidrio, que ante la presencia de un imán permanente hacen contacto formando un circuito cerrado. El imán se coloca sobre la puerta y el interruptor mágnetico en el marco, de tal forma que cuando la puerta o ventana está cerrada, el par de laminillas y el imán se encuentran alineados, por lo tanto el circuito se encuentra cerrado, pero cuando la puerta se abre, entonces se separan y el contacto se abre.

Pueden conectarse a unidad de control formando mallas o de forma independiente, haciendo que en operación normal la unidad de control cense un "0" lógico en las líneas destinadas con

este propósito. Cuando en alguna de estas líneas se lee un "1" se inicia la alarma. En la figura 3.2.7 se muestra el aspecto físico de este sensor.

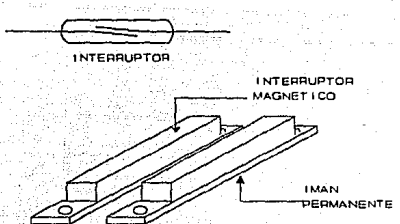


Fig. 3.2.7 Contactos magnéticos.

3.3 ALARMAS

Una señal de alarma es indispensable en un sistema de seguridad, pues es la forma en que el sistema comunica al usuario la detección de un intruso o incendio dentro de la zona protegida. También una señal de alarma puede inclusive evitar el robo, ya que avisa al intruso que se presencia ha sido detectada y puede el residente tomar las medidas necesarias para evitar ser robado, por ejemplo, avisar a la policía, etc. En el caso de un incendio, una alarma puede dar tiempo suficiente al usuario para abandonar el área y evitar una tragedia.

El sistema descrito, tiene tres salidas para alarmas dos de ella son con relevador de 1 Amp. y la otra con transistor "darlington" de 5 Amp. Las salidas con relevador están dispuestas para manejar directamente señales audibles o visuales de baja potencia o relevadores de mayor potencia, la otra salida puede manejar directamente una sirena de 12 V a 3 A o activar otro relevador.

Las alarmas son alimentadas todas por la fuente del sistema, con el fin de evitar que su operación dependa directamente de la línea de 120 V a.c., y puedan ser accionadas, en caso de que exista una falla en la línea de a.c., por el respaldo de baterías.

Para las sirenas es recomendable que cuenten con un relevador de protección para detectar cuando los hilos conductores sean cortados por la acción de algún intruso, además, los conductores deben estar entubados y enterrados para hacer más difícil el sabotaje.

En la figura 3.3.1 se muestra un diagrama de las salidas destinadas a activar las alarmas del sistema. Los relevadores utilizados son de encapsulado *dip* relativamente pequeños, con una repuesta de 4 ms.

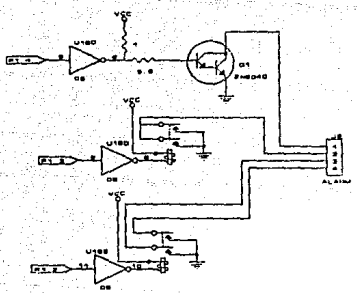


Fig 3.3.1. Salidas para las señales de alarma.

3.4 CONTROL DE ACCESO

El acceso al área restringida es controlado por el sistema a través de teclado y código en memoria que identifica al usuario. La persona que desea el acceso, introduce su clave y si ésta es correcta la unidad de control activa la cerradura (contra eléctrica) que abre la puerta.

El control de acceso está formado por el teclado de 12 teclas, un "zumbador" piezoléctrico para indicar con un sonido cuando se presiona una tecla y el dispositivo utilizado para abrir la puerta.

El teclado es manejado por el mC a través del puerto C del 8255, el "zumbador" y la contra que activa la cerradura de la puerta se controlan por el puerto 1 del 8031 a través de P1.0 y P1.1 respectivamente.

El sistema tiene inicialmente capacidad para manejar 9 claves de acceso diferentes. Las claves se encuentran programadas en la memoria EEPROM (2816); están formadas de hasta 8 dígitos. El número de claves se pueden incrementar hasta un total de 254 utilizando la computadora. La razón por la que no se pueden programar más de 9 claves de forma directa es porque se requiere mayor tiempo para hacerlo.

Características:
- El sistema mantiene activada la cerradura durante aproximadamente 5 segundos, suficientes para empujar la puerta y entrar.

- Si la clave es dada equivocadamente no es necesario esperar un determinado tiempo, basta con presionar la tecla marcada con "#" para que el reconocimiento de clave se vuelva a iniciar.

La forma de decodificar el teclado es el mismo que el descrito en la sección 3.1. para identificar el teclado utilizado para la programación del sistema. En la figura 3.4.1 se muestran los dispositivos utilizados para el control de acceso.

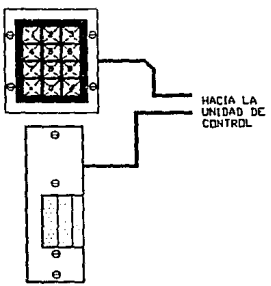


Fig. 3.4.1 Dispositivos para el control de acceso.

3.5 FUENTE Y RESPALDO DE BATERIAS

La fuente de alimentación es de tipo lineal que proporciona tres niveles de voltaje (5, 9 y 12 V), necesarios para los diferentes dispositivos. El nivel de 5 V, alimenta todos los circuitos lógicos y de control, como es el *microcontrolador* y sus diferentes periféricos. El nivel de 9 V, alimenta los sensores que requieran de energía eléctrica para operar; se encuentra dentro del rango de tensiones con los que trabajan la mayoría de sensores (6 a 12 V). El nivel restante (12 V), está previsto para proveer de energía a los dispositivos de salida que requieren mayor consumo de potencia, por ejemplo, las señales de alarma.

La fuente requiere un transformador de 12 V con capacidad de 4 A. Cuando la tensión de 12 V de c.a. llega a la tarjeta de alimentación, la corriente alterna queda rectificada en onda completa por los diodos D1 - D4; la corriente alterna queda rectificada con un componente de rizo significativo y se filtra con el capacitor C1.

La tensión de entrada es de aproximadamente 17 V que se aplica al regulador IC1, que es un regulador programable de 1.5 A. El voltaje de salida se fija por el arreglo de resistencias R5 y R6 que controlan la tensión en la terminal de *ADJ* del regulador, obteniendose aproximadamnente 9 V a la salida. Los 5 V se obtienen a través de IC2 que es un regulador de tensión fijo con capacidad de 1.5 A, la resistencia R7 ayuda en la disipación de potencia a IC2.

Como puede observarse en la figura 3.5.1, el sistema cuenta con una batería de ácido plomo recargable con capacidad de 7 AH. La batería se utiliza para alimentar al sistema cuando la energía de la línea desaparece; proporciona energía durante aproximadamente 12 horas sin activación de alarmas que son las que requieren un mayor consumo de potencia. En tales circunstancias es preciso que la tensión de salida de la batería no vaya hacia el transformador de alimentación, el diodo D5 realiza esta acción de bloqueo.

La batería se recarga automáticamente cuando la energía de la fuente principal se restablece, el diodo D5 se encarga también de rectificar en media onda la corriente alterna que carga a la batería y R1 limita la corriente de carga. Cuando la batería está totalmente sin carga la máxima corriente que permite R1 es de 1.5 A. De esta forma la batería se carga lentamente evitando calentamientos que reduzcan su vida útil.

Es muy importante conocer el estado de operación de la fuente, para ésto se cuenta con LEDS que indican las condiciones en que está operando. El arreglo formado por Q1, R1 y R3 controla el LED1 que indica cuando la batería se encuentra bajo carga. Cuando el voltaje en R1 es mayor que 0.6 V, el transistor Q1 se enciende haciendo circular corriente hacia el LED1 y R3 limita la corriente hacia el LED.

La batería toma la alimentación del sistema sólo cuando el voltaje de la línea a la salida del rectificador de onda completa cae por abajo de 9 V, con el fin de evitar que la batería alimente al sistema innecesariamente. Esta acción la realiza el SCR que es controlado por Q2, P1 y R4, si el voltaje es mayor que 9 V, Q1 se encuentra encendido (saturado), entonces el voltaje de compuerta (G) del SCR es de 0.2 V que evita se encienda, pero cuando la tensión cae abajo de 9 V Q1 se apaga reflejando el voltaje de la batería en la compuerta, encendiéndo el SCR, en ese momento la batería toma la alimentación del sistema.

El arreglo de transistores formados por Q3 y Q4 enciende el LED2 cuando la batería se encuentra baja o sin carga. Si el voltaje a la entrada de IC1 es mayor que 8.5 V, Q3 esta encendido (sat.) y Q4 apagado (corte) por lo tanto no hay circulación de corriente hacia el LED. Pero si el voltaje es menor que 8.5 V, entonces Q3 se apaga permitiendo que Q4 se encienda haciendo circular una corriente a través del LED2 para indicar que la batería se encuentra baja. El LED sólo indicará que la batería se encuentra baja cuando la carga este siendo alimentada por ésta.

FUENTE DE ALIEMTACION

VOLTAJE	CAPACIDAD	ALIMENTACION
5 V	1.5 A	Circuitos lógicos
9 V	1.5 A	Sensores
12 V	3.0 A	Alarmas y disp. de potencia

Tab. 3.5.1

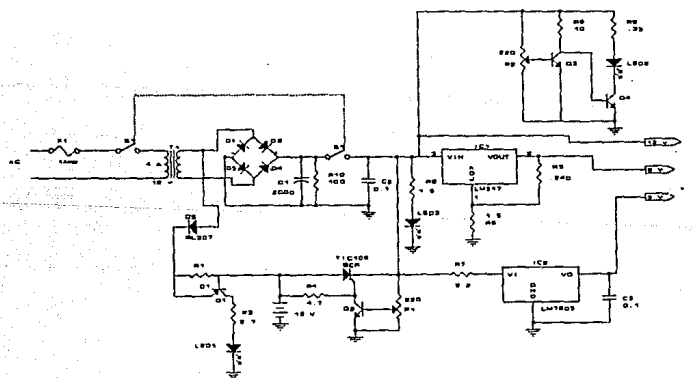


Fig 3.5.1 Diagrama electrónico de fuente de alimentación.

3.6 INTERFASE PC

El hacer uso de una computadora incrementa la capacidad de operación del sistema y además su manejo se vuelve más fácil. La comunicación entre el módulo de control y la computadora se realiza en forma serial, utilizando el estándar RS-232C de EIA.

El puerto serial asíncrono en la IBM-PC, PC/XT, PC/AT de la forma paralela a forma serial para la comunicación entre la computadora y un dispositivo externo, usualmente una impresora serie o un "modem". La transmisión asíncrona serie envía bits de datos individuales por una ruta de entrada/salida, la recepción de datos, también, es realizada ensamblando los bits recibidos para formar un carácter. Esto se logra a través de un controlador serial que empaqueta cada byte en una estructura aparte. Cada estructura consiste de un bit de inicio (start), el byte de datos, un bit de paridad (opcional) y un bit de paro (stop).

El bit de inicio es automáticamente insertado por el controlador antes de enviar el carácter. Este es el primer bit que recibe el dispositivo al que se le envían los datos, el byte de datos puede ser de 5, 6, 7 u 8 bits por carácter, que es enviado después del bit de inicio. Si el controlador se programa para detectar errores de paridad este bit vendría después de byte de datos. Finalmente, el bit de paro que completa la estructura, el controlador puede ser programado para insertar 1, 1.5 o 2 bits de paro.

3.6.1 INTERFASE RS-232C

El RS-232C es un interfase eléctrico estándar para la conexión de componentes del sistema como "modems", impresoras computadoras. El estándar fue establecido por Electronics Industries Association (EIA), una organización de comercio industrial. El RS-232C define un

camino de señal de 25 conductores que forman 18 circuitos con retorno por tierra. El estándar también define los rangos de voltaje de 0 y 1 lógicos usados en todos los circuitos. El puerto serie de la IBM PC utiliza cuando más 9 conductores, pero a menudo solo se utilizan 3. Las señales lógicas dentro de la PC están formadas por lo que se conoce como "niveles TTL". Estos niveles de señal no se utilizan fuera de la PC a causa de su insuficiente inmunidad al ruido eléctrico. En su lugar se utilizan los siguientes niveles, de -3 V a +3 V región de transición, de +3 V a +15 V es un 1 lógico y de -3 V a -15 V es un 0 lógico. En la PC y la mayoría del equipo de comunicaciones producen señales de salida de +- 12 V. Sin embargo una entrada de +- 3 V es suficiente para definir un estado lógico.

Los cables terminan en conectores, pero la constitución física del conector no está definida por el estándar. Los conectores tienen pines y los pines están numerados del 1 al 25. Los pines del RS-232 que utiliza un puerto serie típico de la PC se definen el la tabla 3.6.1.

En la mayoría de las tareas de comunicación de datos no se utilizan todas las señales mostradas. Los circuitos más importantes son el de transmisión de datos y el de recepción de datos. Estas son las líneas por los que los datos se envían y reciben simultáneamente. El resto de los pines, con excepción de la tierra son señales de control.

La interfase RS-232 tiene dos "sexos electrónicos" equipamiento de terminal de datos (Data terminal Equipment - DTE) y equipamiento de comunicación de datos (Data Communication Equipment -DCE). La computadora se considera un DTE y transmite por la línea 2 y algún otro dispositivo como por ejemplo un "modem" se considera DCE y entonces transmite por la línea 3. El estándar RS-232C especifica el número de los pines, no especifica el conector y se hace referencia comúnmente a un conector DB-25. El género de los conectores no tiene nada que ver con que el dispositivo sea DTE o DCE. El puerto serie (com 1) de casi todas las PC tienen conectores DB-25 macho.

Pines	Señal	Descripción
2	TX	Transmisión de datos.
3	RX	Recepción de datos.
4	RTS	Request to Send, establecida por la PC cuando quiere transmitir.
5	CTS	Clear to Send, recibido por la PC cuando el dispositivo está listo para recibir datos.
6	DSR	Data Set Ready, Recibido por la PC cuando el "modem" esta conectado y encendido.
7	GND	Señal de Tierra.
8	CD	Carrier Detected, recibida por la PC cuando el "modem" detecta una portadora.
20	DTR	Data Terminal Ready, establecida por la PC siempre que la comunicación de datos está activa.
22	RI	Ring Indicator, recibida por la PC cuando el "modem" esta recibiendo una señal de timbre.

Tabla 3.6.1

3.6.2 FUNCIONAMIENTO INTERNO DEL PUERTO SERIE

El adaptador asíncrono de comunicaciones de una PC esta formado por un circuito integrado conocido como UART (Universal Asynchronous Receiver/Transmitter). El CPU de la computadora enlaza al UART mediante el bus de datos y siete direcciones de puertos con estas ultimas puede acceder a los registros del UART (8250). En la tabla 3.6.2 se muestran las dirección de los registros del UART.

En el sistema, el módulo de control se toma como un dispositivo DCE y la computadora por sus características como un DTE. La comunicación con el modulo de control se realiza por el puerto serial 1 de la PC. El software para el manejo del puerto se desarrolló en el lenguaje de programación "C".

En enlace físico entre el sistema y la computadora sólo consta de tres conductores, para esto se utiliza un comector de "modem" nulo, el cual se muestra en la figura 3.6.1.

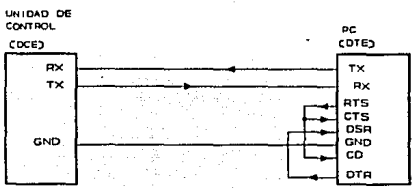


Fig. 3.6.1 "Modem" nulo.

Los niveles de voltaje necesarios pora la interfase RS-232C en la unidad de control, se obtienen a partir de un Circuito Integrado (CI) fabricado por *Maxims* MAX232 que a partir de una sola fuente de 5 V y 4 capacitores es suficiente para generar los niveles establecidos para el estándar RS-232C. La señal proporcionada por el puerto serie del 8031 a nivel "TTL" se hace pasar por el circuito MAX232 para proporcionar los niveles adecuados para su transmisión +9 para 1 lógico y -9 V para el 0 aproximadamente.

DIR. DE ENT/SAL	ESTADO LEC/ESC	DESCRIPCION
03F8H	ESC	Transmiter holding Register, registro que contiene el carácter a transmitir.
03F8H	LEC	Receiver Buffer Register, contiene el carácter recibido.
03F8H	ESC/LEC	Registro divisor byte bajo, define la velocidad de transmisión.
03F9H	ESC/LEC	Registro divisor byte alto, define la velocidad de transmisión.
03F9H	ESC/LEC	Registro habilitador de interrupción.
03FAH	LEC	Registro de prioridades de interrupción.
03FBH	ESC/LEC	Registro de control de línea.
03FCH	ESC/LEC	Registro de control de "modem".
03FDH	LEC	Registro de estado de línea.
03FEH	LEC	Registro de estado de "modem".

Tabla 3.1.2

4.0 CONSTRUCCION FISICA DEL SISTEMA

En esta parte del trabajo, se hace una descripción física del sistema desarrollado. Describiendo primero las partes que lo forman, posteriormente, se hacen algunas recomendaciones respecto a su instalación, y al final se muestran los circuitos impresos y electrónicos.

4.1 BLOQUES QUE INTEGRAN EL SISTEMA

El sistema se encuentra formado por cinco bloques principalmente, los cuales se describen a continuación:

1) Unidad de Control;
2) Sensores;
3) Alarmas;
4) Control de acceso, e
5) Interfase PC.

En la figura 4.1.1 se muestra la organización de los diferentes bloques, donde se puede observar que la parte fundamental es la unidad de control.

Ahora para la descripción física, la unidad de control está formada por la tarjeta principal que integra los componentes de la lógica de control y por la tarjeta de la fuente de alimentación; en el capítulo anterior se trataron de manera separada. También, como parte de dicha unidad se considera al "display" de cristal líquido y en general a el tablero de control.

El tablero de control tiene dimensiones de 26x8 cm y está formado por una tarjeta de circuito impreso que sostiene el teclado de programa, el módulo de cristal líquido AND 771 y tres "Leds". Esta parte de la unidad de control es la interfase entre el usuario y el sistema, ya que

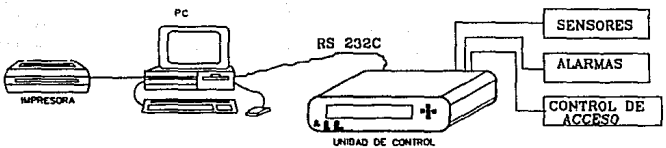


Fig. 4.1.1 Bloques del sistema.

es a través de éste como se visualizan y cambian parámetros de operación. La conexión con la tarjeta principal se realiza utilizando un par de conectores de cable plano.

El bloque referente a sensores está formado como su nombre lo indica por sensores, éstos son básicamente de humo e intrusos. Se tiene capacidad para recibir desde uno o hasta 15 sensores conectados a las diferentes líneas destinadas a éste propósito, pero si no es suficiente, pueden formarse mallas para incrementar el área de cobertura haciendo arreglos de sensores.

Como podrá observarse en la figura 4.1.1, el sistema incluye un bloque de alarmas, éste puede ser en un momento dado el más importante referente a seguridad, ya que es a través de señales de alarma como el sistema comunica que se activó algún sensor, delatando un posible incendio o un intruso. Las características que definen el bloque están en función de la aplicación específica y de los requerimientos de seguridad.

El control de acceso es también uno de los bloques importantes, ya que de éste depende que el acceso a la zona restringida sea eficiente o no. Físicamente, se encuentra formado por un

teclado con 12 teclas y un mecanismo de apertura de la puerta (contra eléctrica), los cuales se sitúan particularmente en la puerta que se desea controlar el acceso. Ambos dispositivos, cuentan con cables que terminan en conectores fácilmente enchufables en la unidad de control. El teclado tiene un conector DB9 - macho el cual permite que su conexión sea única, evitando errores de posición que podrían ocasionar problemas. La longitud máxima de los cables recomendada es de 15 m y el calibre de los conductores de AWG (American Wire Gage) 22, para evitar que su resistencia afecte la identificación de los niveles lógicos.

La interfase con la PC está formada por un cable blindado con conectores DB-25 macho y DB-9 también se "sexo electrónico" macho. El conector DB-25 se conecta al puerto serie No. 1 de la PC (COM 1) y la contraparte a la unidad de control. El cable no debe tener una longitud mayor que 15 m con calibre AWG 22.

Fig. 4.1.2

4.2 INSTALACION DEL SISTEMA

La instalación contempla en un principio, la planeación y ubicación de las zonas de detección de intrusos e incendios, con el fin de establecer el tipo de sensores adecuados para cada zona, así como, el número de éstos si es que se requiere.

Cuando el área que se pretende cubrir es amplio, es necesario formar mallas de detección que dividan a la zona en regiones más pequeñas, estas mallas se forman conectando sensores en serie o paralelo dependiendo del tipo de sensor con que se disponga. Las mallas instaladas deberán cubrir toda la zona por lo que se dedicará, por ejemplo, una al perímetro de la zona a proteger, que delatará al intruso en el momento de intentar colarse. Otra puede destinarse al perímetro de la construcción propiamente dicha, para delatar al intruso cuando se haya introducido al edificio y así sucesivamente destinar una malla para cada zona de interés. Tener cuidado de instalar los sensores adecuados para cada zona.

Se deberán instalar sensores de incendio en las áreas más propensas a sufrir este problema, se debe tomar en cuenta que los sensores detectan generalmente las partículas de humo del ambiente, por lo que deberán instalarse en áreas cerradas para evitar falsas alarmas.

Es muy importante que la ubicación de cada malla o sensor este perfectamente definida, para que cuando se active se conozca el sitio exacto del problema. Además, nos ayudará a saber si se trata de un intruso o incendio, por lo que tampoco deberán mezclarse en las mallas sensores de intrusos con los de incendio. Una observación importante, es que los sensores de intrusos no deberán estar a la vista para evitar que sean saboteados fácilmente.

Con respecto a la instalación de alarmas, una recomendación que en general se hace a la instalación de los sistemas de seguridad, es que por lo menos se deben instalar dos dispositivos sonoros. Uno en la parte externa del edificio que se brinda protección y otro de forma interna. También, se debe tener cuidado de que los dispositivos de alarma que se instalen cuenten con relevador para detectar cuando los hilos conductores sean cortados. Además, todos los dispositivos de alarma deben tener una instalación adecuada para dificultar el sabotaje.

Instalar la Unidad de Control es una labor que requiere de ciertas consideraciones, ya que de la operación de ésta depende todo el sistema. Se debe tener cuidado de que no este a la vista de

cualquier persona, es decir deberá estar colocada en un sitio estratégico. Se debe asegurar que la batería de respaldo este perfectamente instalada y lista para entrar en operación cuando la energía de la línea desaparezca producto de la acción de un intruso o falla en la línea. Los cables y conectores deben estar perfectamente conectados, además, evitar que éstos se muevan de tal manera que pudieran desconectarse o provocar falsos contactos que conducirían a fallas.

Para instalar el control de acceso no existe mucho problema, pues aquí todos los parámetros ya se han establecido, simplemente se colocan los dispositivos en su sitio adecuadamente.

4.3 CIRCUITOS IMPRESOS Y ELECTRONICOS

Los componentes de la unidad de control se encuentran distribuidos en dos tarjetas de circuito impreso, la tarjeta principal y la tajeta de fuente, con el propósito de tener por separado los componentes de control y los de alimentación.

4.3.1 Tarjeta principal

En la tarjeta principal se encuentra toda la circuitería de control que necesita el sistema; tiene dimensiones de 16x13 cm que incluye 3 conectores para cable plano y dos DB9 "hembra", los conectores DB9 se utilizan para enlazar la comunicación con la PC y para el manejo del teclado para el control de acceso. En la figura 4.3.1 se muestra el circuito impreso de esta tarjeta junto con el diagrama electrónico.

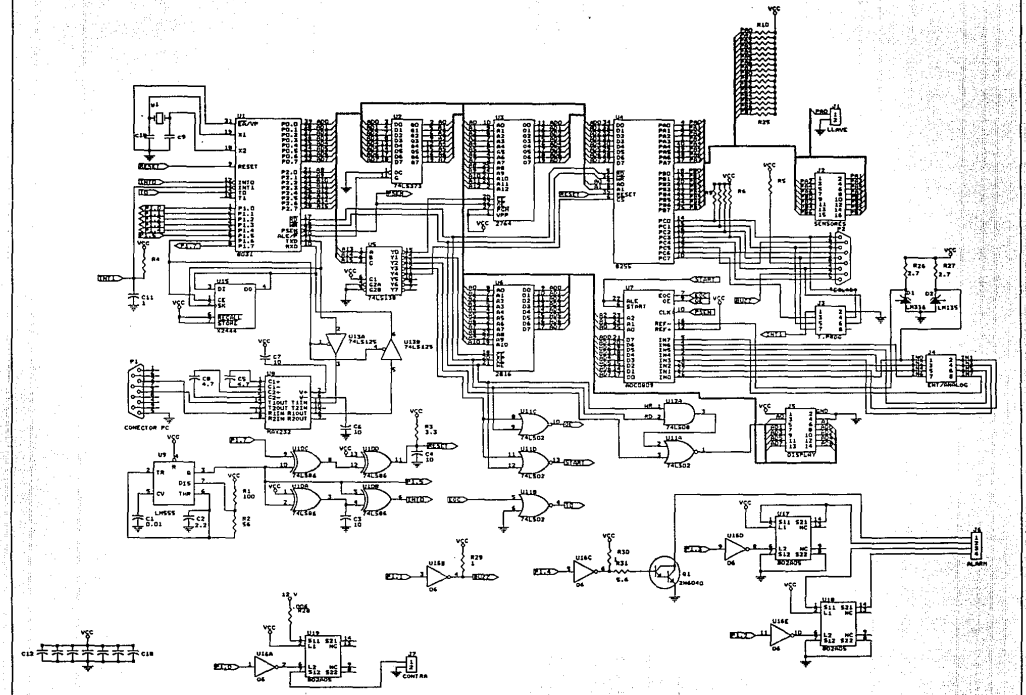


Fig. 4.3.1.a DIAGRAMA ELECTRONICO DE LA TARGETA PRINCIPAL.

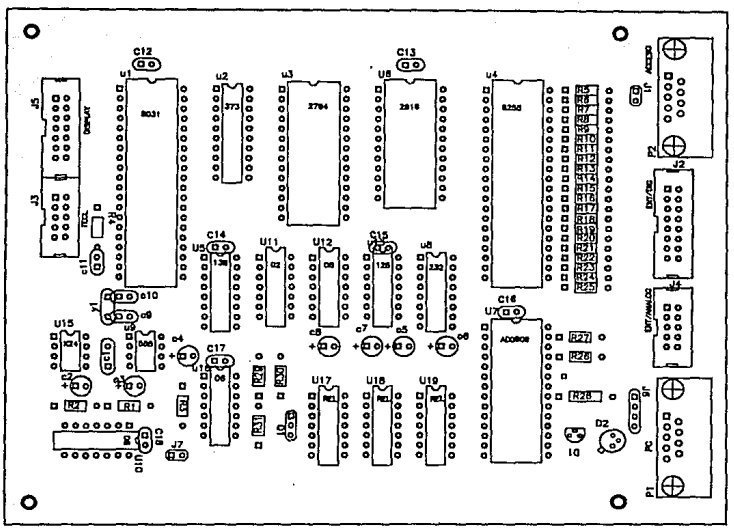


Fig. 4.3.1.b Distribución de componentes de la tarjeta principal.

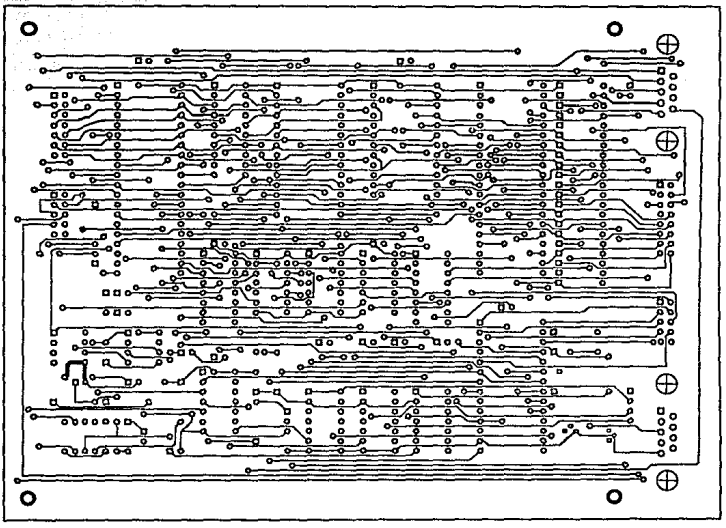


Fig. 4.3.1.c Lado de componentes.

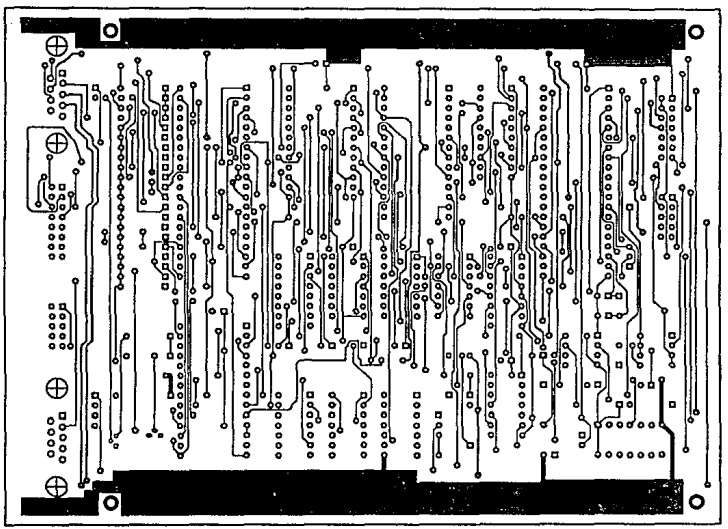


Fig 4.3.1.d Lado de soldadura.

Como parte del desarrollo del sistema, se construyó una tarjeta que contiene las teclas del tablero de control, la figura 4.3.1.d muestra esta tarjeta cuyas dimensiones de 5x5 cm; el diagrama eléctrico se ilustra en la figura 3.1.16.

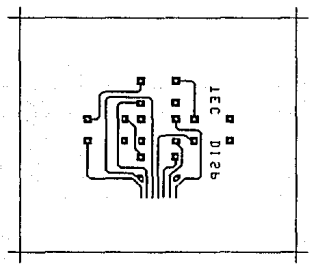


Fig. 4.3.1.d Tarjeta del teclado en el tablero de control.

4.3.2 Tarjeta de fuente de alimentación

La tarjeta de la fuente tiene dimensiones de 8x15 cm, la figura 4.3.2 muestra dicha tarjeta; el diagrama electrónico correspondiente se encuentra en la figura 3.5.1. El transformador se ubica dentro del módulo de control, y la batería del respaldo debido a sus dimensiones hace necesario situarlo fuera.

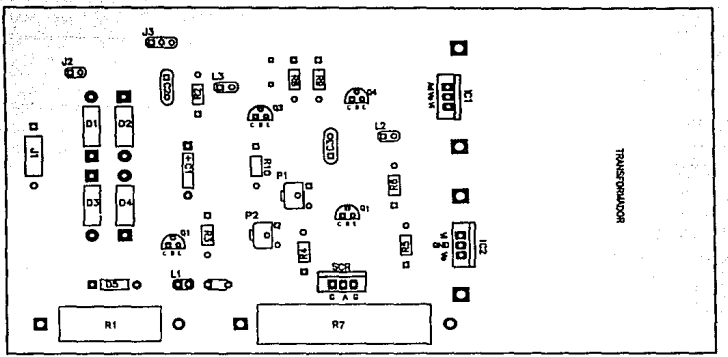


Fig. 4.3.2.a Distribución de componentes, fuente de alimentación.

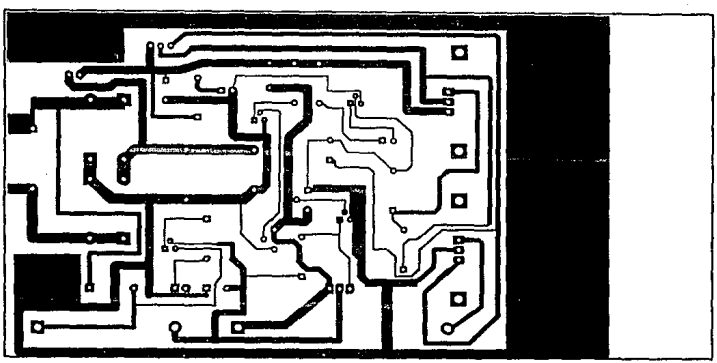


Fig. 4.3.2.b Fuente de alimentación, lado de soldadura.

5.0 DESCRIPCION GENERAL DE FUNCIONAMIENTO

El sistema controlador de acceso y seguridad, como ya se mencionó en la sección 3.1, tiene la capacidad de manejar todas sus funciones de manera autónoma o a través de una IBM PC o compatible.

En esta sección, se describe su funcionamiento, se habla primero de la operación autónoma y después se explica la operación cuando se tiene conectada la PC.

Antes de empezar a realizar sus funciones, se requiere programar los parámetros de configuración de acuerdo a las necesidades de cada usuario. Para arrancar el sistema sólo se activa el interruptor de la parte posterior de la unidad de control e inmediatamente inicia sus funciones de acuerdo a sus parámetros establecidos. Cuando entra en operación en el "display" aparece el mensaje que se muestra en la figura 5.1.

Fig. 5.1

Como podrá verse, se indica la temperatura actual del ambiente, este dato se obtiene a través de muestrear la señal del sensor de temperatura LM 135. Este mensaje estará presente siempre y cuando no se active algún sensor o se programe algún parámetro, cabe aclarar que este mensaje es desplegado cuando el sistema se encuentra trabajando de manera autónoma, de lo contrario, cuando se tiene conectada la PC, el mensaje que se despliega es el que aparece en la figura 5.2. Como puede observarse en esta última figura, se indica además de la temperatura, la hora actual; este dato se obtiene del reloj interno de la computadora, se debe tener cuidado de que este actualizado para no generar confusiones.

Fig.5.2

En operación normal las únicas modificaciones que se hacen a los datos presentados en el "display", es la actualización de la temperatura y la hora en caso de que se encuentre la PC operando con el sistema.

5.1 SISTEMA DE SEGURIDAD

En el sistema, las 15 mallas de sensores que se pueden manejar, son configuradas para sensores cuyos contactos son normalmente cerrados o abiertos. Además, de que pueden estar activadas o desactivadas dependiendo de las necesidades del usuario, es decir cuando una malla se encuentra desactivada a pesar de que se detecte una anormalidad en la zona cubierta por ésta no se genera una señal de alarma.

Cuando en alguna de las mallas se activa un sensor y si ésta se encuentra activada, el sistema envía al "display" un mensaje indicando el número de la malla o las mallas en que se detectó sensores activados, el mensaje es como se muestra en la figura 5.3. En donde se proporciona la ubicación del sensor que se activó, o sea, el problema que puede ser algún intruso o incendio. Al mismo tiempo que aparece el mensaje se inician las alarmas en el caso de que se encuentren activadas.

Las alarmas son continuas y la única forma de desactivarlas es introduciendo la *clave de seguridad* ("password"), con el fin de evitar que cualquiera persona pueda silenciar al sistema sólo descubriendo su ubicación y presionando alguna tecla. Como se podrá observar, no se proporciona más datos acerca de la ubicación de la malla, debido a que sólo se tienen dos líneas para el despliegue de mensajes, pero cuando se dispone de una PC, esto si es posible, pues previamente se programa el mensaje a desplegar; proporcionando la zona que es cubierta e información adicional que puede ser gran de utilidad en un momento dado.

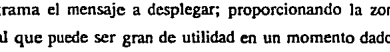


Fig 5.3

5.2 CONTROL DE ACCESO

El acceso al área restringida se lleva a cabo introduciendo una clave que identifica a la persona a través de un teclado. La clave está formada por una secuencia de hasta 8 dígitos, cada vez que se presiona una tecla se genera un sonido con el que se indica que la tecla se ha reconocida.

El control de acceso está diseñado para funcionar en áreas de mucha concurrencia principalmente, y que no requieren un nivel de seguridad elevado. Por ejemplo, la biblioteca de algún centro de investigación, en donde se requiere tener un control de los departamentos que utilizan los servicios, para lo cual a cada departamento se le asigna una clave que utilizará posteriormente para tener acceso a los servicios que ofrece dicha biblioteca. Otro caso, puede ser aquel en el que se requiere controlar el acceso del personal de algún departamento, en esta situación a cada integrante se le asigna una clave con la cual podrá introducirse más tarde.

Los ocho dígitos que puede contener una clave son del 1 al 9, como puede verse, no se incluye el cero debido a que el sistema cuando lee el teclado reconoce a la tecla marcada con cero como la número 11, que no concuerda con el cero que se programa en la definición de claves de acceso.

La persona que desea el acceso introduce su clave y cuando termina debe presionar la tecla inferior derecha (#), con la que se le indica al sistema que la clave a sido introducida completamente y que ya puede proceder a la comparación de los datos en memoria con los datos proporcionados.

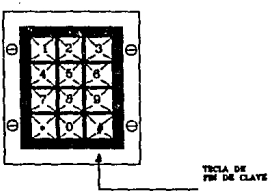


Fig. 5.2.1 Teclado del control de acceso.

Si por alguna razón el usuario se equivoca al introducir su clave, es necesario volver a presionar esta tecla para indicar que se va a enviar una nueva clave. De hecho, cada vez que se presiona esta tecla, el sistema se pasa a esperar una nueva clave por lo que no es necesario esperar un determinado tiempo como generalmente lo hacen los sistemas controladores de acceso. A partir del momento en que se termina de dar la clave se tienen aproximadamente 5 segundos para empujar la puerta y que ésta se abra totalmente para permitir el acceso.

Si el sistema cuenta con la PC, entonces en el momento en que el acceso es concedido, la unidad de control envía la clave correspondiente y la PC de acuerdo a su base de datos registra

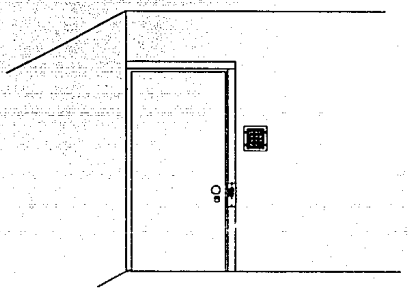


Fig. 5.2.2 Control de acceso.

la información de la persona que logró el acceso, así como la hora y la fecha; de lo contrario este paso se omite.

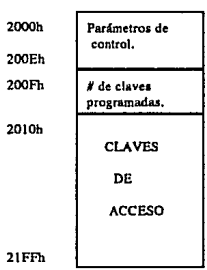


Fig. 5.2.3 Organización de la memoria 2816.

El sistema puede soportar un total de 254 claves distintas, las cuales están organizadas en bloques de 8 bytes en la memoria EEPROM (2816), empezando en la localidad 10h, o sea, en la dirección 2010h del sistema. En la figura 5.2.3 se muestra la organización de la memoria EEPROM 2816.

5.3 SISTEMA AUTONOMO

Trabajando de manera autónoma, el sistema realiza todas sus funciones de seguridad y control de acceso. Existen algunas limitantes, por ejemplo, la falta de un registro estádistico, también, el número de claves de acceso que se manejan es mínimo (9), además, la programación de parámetros requiere de cierto conocimiento, ya que la información para guiar al usuario es mínima. Pero de esta forma se tiene un sistema funcional y económico.

5.3.1 PROGRAMACION DE LOS PARAMETROS DE OPERACION

Programar al sistema a través de sus propios recursos, implica un mayor conociminento de la operación de éste, pero en terminos generales es sencillo.

A continuación se describe en forma detallada los pasos necesarios para llevar a cabo la configuración del sistema:

Cuando se requiere programar o verificar los parámetros de operación, en primer lugar se introduce la *clave de seguridad*. Para este propósito, se tiene el teclado de la unidad de control. Las teclas cuyas flechas señalan derecha o izquierda (***D/I***), únicamente recorren el cursor una posición hacia un lado o hacia el otro dependiendo de la tecla que se oprime. Aquellas con las flechas arriba/abajo (***S/B***), tienen diferentes funciones que dependen de la opción seleccionada.

La tecla central se utiliza para reconocimiento de datos, es decir, cuando es presionada el *microcontrolador* toma los datos escritos en el "display" y los procesa; es más o menos equivalente a la tecla ENTER en la PC. En la descripción de este trabajo se hace referencia a ella como la *tecla de entrada de datos (TED).*

Para iniciar la programación, se presiona la tecla TED que el sistema reconoce inicialmente como petición de programación y responde con el mensaje de la figura 5.3.1.

Fig. 5.3.1

A partir de este momento el sistema está en espera de la *clave de seguridad*, la cual es dada por medio de las teclas S/B. Cuando alguna de estas teclas se oprime aparece un dígito y si se vuelve a oprimir aparece otro en orden ascendente o descendente dependiendo de cual se oprime, es decir S/B sólo rota los dígitos entre el 0 y 9. Cuando el número deseado es puesto, entonces se recorre el cursor una posición y se realiza el mismo procedimiento hasta introducir toda la clave. Cuando se termina, se presione nuevamente la tecla TED y si la clave es correcta, en el "display" aparece el mensaje de la figura 5.3.2., de lo contrario se regresa al mensaje inicial.

Fig. 5.3.2.

El mensaje, presenta el inicio del menú principal, el cual contiene las diferentes opciones de programación, el menú completo es el siguiente:

M E N U
1.- CLAVES DE ACCESO
2.- DEFINIR MALLAS
3.- ACTIVAR MALLAS
4.- PRUEBA DEL SISTEMA
5.- ACTIVAR ALARMAS
6.- SALIR

Para seleccionar alguna de estas opciones, se recorre la lista por medio de las flechas hasta que aparezca la opción deseada, posteriormente, se presiona la tecla TED. A continuación de describe cada una de estas opciones.

1) CLAVES DE ACCESO

Esta opción permite observar o modificar los dígitos de las claves de acceso que se encuentran actualmente programadas. Cuando esta opción es seleccionada en el "display" aparece el mensaje de la figura 5.3.3.

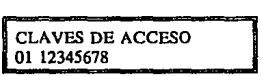


Fig. 5.3.3.

Como puede observarse, en el extremo izquierdo aparece el número de la clave que está siendo mostrada, seguido de los dígitos que lo forman. Con las flechas se puede ir observando cada una de las claves programadas.

Para modificar la clave que se está observando, se presiona la tecla TED para que el cursor se posicione en el primer dígito, con la flechas D/I se posiciona en el dígito que se desea modificar, posteriormente, con las teclas S/B, se pone el dígito requerido. Cuando se termina de efectuar las modificaciones pertinentes, se presiona nuevamente la tecla TED para que la clave sea actualizada y pueda continuarse con las siguientes.

Como ya se dijo, las claves pueden estar formadas de hasta de 8 dígitos. Cuando la clave no es de 8 dígitos exactamente, es decir contiene un número menor que 8, entonces al final de los dígitos que la forman se inserta el siguiente caracter "**>**", para evitar que el sistema compare los 8 dígitos; con esta marca el mC sólo toma como parte de la clave los dígitos que existen hasta antes del caracter "**>**", en el cuadro de la figura 5.3.4 puede observarse esta situación.

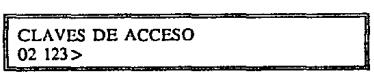


Fig. 5.3.4

El cero no puede formar parte de una clave, ya que en su lugar se reconoce el número 58 que en ASCII equivale a dos puntos (:).

Para salir de esta opción se recorren todas las claves hacia adelante, hasta que aparezca el mensaje de *salir*, entonces se presiona la tecla TED para retornar al menu principal. Esta es la razón por la que de manera autónoma no se manejan más de 9 claves de acceso.

2) DEFINIR MALLAS

A través de esta opción se define el tipo de sensores que forman una malla. Cuando se selecciona esta opción se presenta en el "display" el mensaje de la figura 5.3.5.

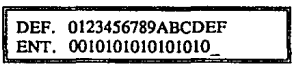


Fig. 5.3.5

En el renglón superior se indica el número de malla y en el de abajo su condición, es decir si la malla se identifica con un cero "0", quiere decir que está formada por sensores cuyos contactos son normalmente cerrados; si es con un "1", entonces indica que son normalmente abiertos. Si se esta de acuerdo se presiona la tecla (↓) y aparece el mensaje de salir que permite regresar al menú principal presionando la tecla TED, de lo contrario se procede la modificación que es similar a la de la programación de claves de acceso, sólo que en este caso, únicamente se rotan los dígitos 0 y 1.

En el "display" se muestran 16 mallas, pero en realidad sólo existen 15, ya que la malla cero tiene un propósito especial, dedicada para modificar el cambio de la clave de seguridad.

3) ACTIVACION DE SENSORES

La opción de activación de sensores permite especificar la(s) malla(s) que se desee supervisar, dicha opción se incluyó debido a que en algunos casos, por ejemplo, sólo se requiere vigilar

algunas zonas durante la el día o durante la ausencia del usuario. Con esta opción, si los sensores de una malla deshabilitada se activan las alarmas se ignoran. Cuando se selecciona esta opción aparece el mensaje de la figura 5.3.6 en el "display".

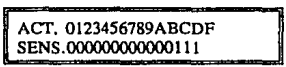


Fig. 5.3.6

El renglón superior muestra el número de malla y en la parte inferior su condición. Cuando en la malla aparece un 1, entoces se encuentra activada, pero si aparece un cero dicha malla está desactivada. Si desea cambiar la configuración actual se presiona la tecla TED y el cursor se posiciona en la malla cero, mediante las flechas se efectuan las modificaciones correspondientes. Terminada la operación se vuelve a presionar la tecla TED, con lo cual los parámetros quedan programados y retorna al menu principal.

4) PRUEBA DEL SISTEMA

Es recomendable efectuar una prueba de operación cada mes para verificar el correcto funcionamiento del sistema. Esta opción permite efectuar esta prueba y también es muy útil en la instalación. Cuando se selecciona aparece en el "display" el mensaje mostrado en la figura 5.3.7.

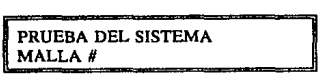


Fig. 5.3.7

Con esta opción, la unidad de control se va a monitorear las condiciones de todas las mallas independientemente que estén activadas o no. Si se detectan sensores activados, se despliega el número de la malla en que se encuentran dichos sensores. Como se mencionaba anteriormente, esta opción es muy útil cuando se realiza la instalación del sistema y se prueba su correcto funcionamiento sin necesidad de sonar las alarmas. Simplemente se forza a activar algún sensor y se observa si la unidad de control detecta dicho sensor activado.

5) ACTIVAR ALARMAS

Esta opción que permite activar o desactivar las señales de alarma, presenta el mensaje de la figura 5.3.8.

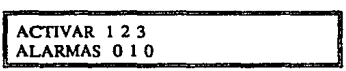


Fig. 5.3.8

El sistema cuenta con tres señales de alarma las cuales pueden programarse mediante esta opción, en la primera línea aparece la señal de alarma y en el renglón inferior su condición. Si la condición es cero, entonces la señal esta deshabilitada; de lo contrario, si aparece un uno, está activado. La forma de programar es similar a las anteriores, si se desea efectuar cambios sobre las condiciones actuales de las alarmas, se presiona la tecla TED y se realizan. Terminados los ajustes se vuelve a presionar la tecla TED para salir de esta opción.

Cuando no se tienen tres dispositivos de alarma instalados, no es necesario tener las tres señales habilitadas. Esta opción, permite desconectar la señal de alarma de una determinada

zona, es decir, cuando en una área se quiere que el dispositivo sonoro no se active, simplemente se deshabilita la señal correspondiente.

6) SALIR

La opción de SALIR del menú principal se selecciona cuando se desea terminar con la programación del sistema; nos regresa al mensaje inicial de cuando se arranca el sistema.

7) CAMBIO DE LA CLAVE DE SEGURIDAD ("PASSWORD").

En el capítulo 3 se describe la clave de seguridad a nivel de "hardware" básicamente, en esta sección se muestra la manera de efectuar el cambio de dicha la clave.

La forma de realizar el cambio de la *clave de seguridad* es como sigue: Se introduce la llave en la cerradura que se encuentra en la parte superior de la unidad de control, se gira para que abra los contactos respectivos; realizada esta operación se presiona la tecla TED en el tablero del control y en ese momento el mC reconoce la petición del cambio de clave enviando al "display" el mensaje que aparece en la figura 5.3.9, para que el usuario se entere que ya puede introducir la nueva clave de seguridad.

Fig.5.3.9

Los dígitos que forman la nueva clave, se ponen utilizando las teclas del tablero de control de igual manera como se programa una clave de acceso. También, si la clave de seguridad es de menos de 8 dígitos se debe introducir el caracter ">" al final de ésta. Cuando la clave se termina de introducir, se regresa la llave a su posición original; posteriormente, se presiona la tecla TED con la que la nueva clave es almacenada en memoria terminando así la operación.

5.4 EL SISTEMA OPERANDO CON LA PC.

La capacidad de interactuar con una PC permite al sistema llevar llevar un registro estadístico de los principales eventos ocurridos durante el día. Además, la computadora hace que el despliegue de información sea mucho más clara, ya que se pueden desplegar mensajes más largos y formateados.

Para efectuar el enlace entre la PC y la Unidad de Control se requiere ejecutar el archivo SISTEMA.EXE. Este archivo contiene el "software" que la computadora necesita para poder interactuar con el el sistema.

La computadora, prueba en primer lugar, que exista enlace de comunicación entre ésta y la Unidad de control. Si la comunicación entre ambos dispositivos es correcta, entonces la unidad de control envía sus parámetros de operación y configuración actuales. En el monitor de la computadora se despliegan las condiciones de las principales variables del sistema como son: el estado de las alarmas, de las mallas, además de un menú para ejecutar funciones específicas:

CONFIGURACION	F1
REPORTES	F2

PRUEBA DEL SISTEMA	F3
SALIR	F4

La opción de reporte se utiliza cuando se desea generar un informe, ya sea impreso o en pantalla de los datos almacenados durante el último mes. El reporte es generado sólo a petición del usuario y puede ser completo o únicamente de la información mostrada en pantalla. La prueba de operación del sistema se puede realizar desde la PC, a través de la opción F4.

Cuando se selecciona SALIR, la computadora almacena los datos más recientes en la unidad de disco y regresa al Sistema Operativo (MS-DOS). La configuración del sistema se describe en la sección siguiente. En el apendice A se muestran los diagramas correspondientes de las diferentes opciones que se muestran en la pantalla de la computadora.

5.4.1 CONFIGURACION A TRAVES DE LA PC.

La programación de los parámetros de operación a través de la computadora se realiza de una manera muy sencilla, ya que se establecen diálogos interactivos con el usuario facilitando esta tarea.

Cuando en el menú principal se selecciona la opción F1 para la configuración, la computadora solicita la Clave de Seguridad. Para introducirla se permiten sólo tres intentos, si después éstos la clave no ha sido dada correctamente, se cancela la operación reanudándose 5 minutos más tarde, generando un sonido de alarma en la PC. Si la clave es correcta, en la pantalla aparecen varias opciones que permiten modificar por separado los bloques del sistema. Cuando se termina la configuración, la computadora envía los nuevos parámetros a la Unidad de control, terminado así la programación del sistema.

5.5 "SOFTWARE" DEL SISTEMA

El "software" del sistema se encuentra implementado físicamente en dos sitios, uno es la PC y el otro el módulo de control; el primero se desarrolló en *lenguaje de progración "C"* y el segundo en *lenguaje ensamblador* del propio *8051*.

Los algoritmos implantados en la Unidad de Control son responsables de las funciones de seguidad y control de acceso; se encuentran cargados en la memoria EPROM 2764 que es la memoria de programa; el espacio utilizado es de aproximadamente 5 "Kbytes". En la figura 5.5.3 se ilustra el diagrama de flujo correspondiente al "software" de la Unidad de Control. Para generar el *código objeto* se utilizó el *ensamblador* de Avocet Systems (AVA51 ver 1.2).

Para el amacenamiento temporal de variables y datos del sistema, la memoria interna del *microcontrolador* se organizó de la forma que se muestra en la figura 5.5.1.

7Fh 50h	Zona de "Stack"
4Fh 40h	Datos de control
3Fh 38h	Clave de seguridad predefinida
37h 30h	Clave de seguridad a comparar
2Fh 28h	Clave de acceso predefinida
27h 20h	Clave de acceso a comparar
1Fh 00h	Bancos de registros

Fig 5.5.2

El "software" de la PC se encarga del registro estadístico de eventos detectados por la unidad de control y puede ser ejecutado por una IBM PC o compatible XT/AT. En la figura 5.3.2 se muestra el diagrama de flujo básico del algoritmo que se ejecuta por la PC.

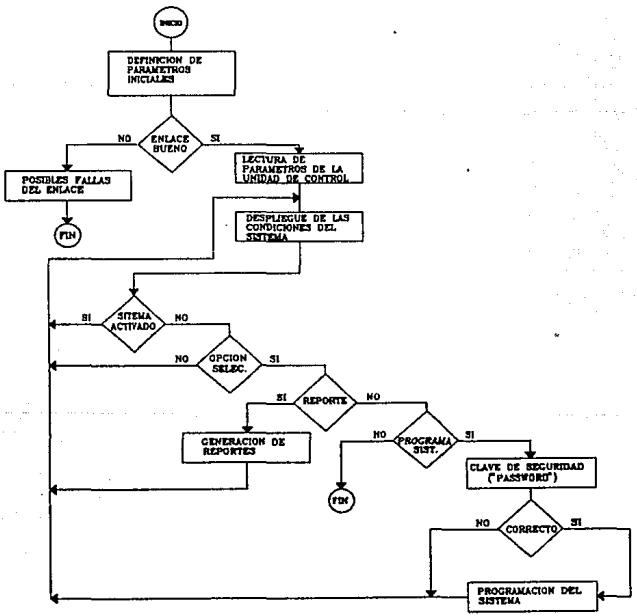


Fig.5.5.2 Diagrama de flujo de los algoritmos en "C".

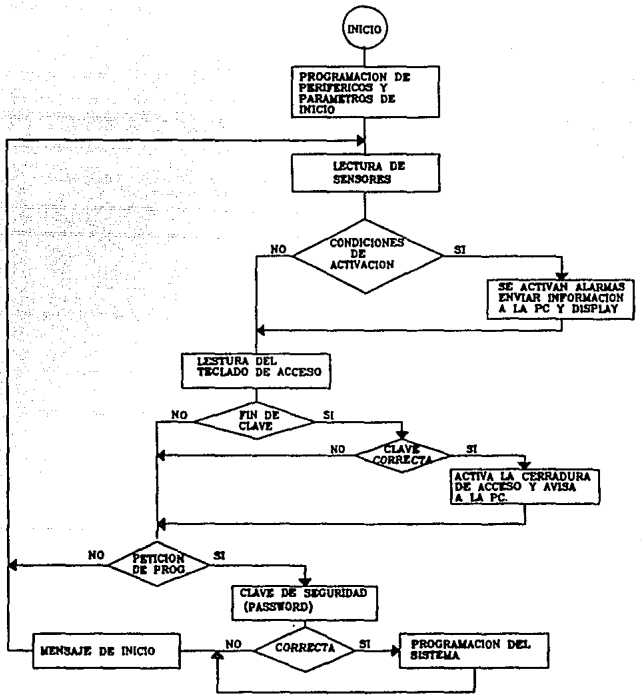


Fig. 5.5.3 Diagrama de flujo de los algoritmos en *ensamblador.*

6.0 CONCLUSIONES Y ALTERNATIVAS

El sistema descrito en el presente documento, controla el acceso de una área restringida y realiza funciones de supervisión de sensores para brindar protección contra posibles intrusos e incendios. Es en general un sistema económico y cumple con el alcance previsto al inicio de este trabajo.

El control de acceso se realiza con un nivel bajo de seguridad, ya que sólo se utiliza, para este propósito, un teclado y claves en memoria. Esta característica debe ser tomada en cuenta en su instalación, con el fin de darle un uso adecuado. El nivel de seguridad se puede incrementar, si así se desea, dedicando sensores a supervisar la puerta que se controla el acceso.

El motivo por cual sólo se controla un acceso, es principalmente, porque se desea un sistema económico y de fácil instalación. Sin embargo, utilizando esta misma arquitectura y como una alternativa, se puede incrementar el número de accesos controlados, sólo que para este caso se requiere de **instrumentar** el teclado de cada acceso, es decir, dedicar un *microcontrolador* a la decodificación del teclado y al control de la cerradura de la puerta.

Cada teclado se conecta a con la Unidad de Control a través de un enlace de comunicaciones RS-232, es decir, las líneas del puerto serie de la Unidad de Control conectan a cada teclado por loque se requiere que cada uno disponga de un puerto serie de para la comunicación. Entre cada teclado y la Unidad de Control existirá una relación maestro esclavo donde la Unidad de Control funge como maestro y el teclado como esclavo.

La operación de esta alternativa es como sigue: La unidad de control interroga secuencialmente a cada teclado, el cual responde con un mensaje que el maestro evalúa. Cuando el teclado detecta que una clave se ha terminado de introducir, entonces cuando es interrogado por el maestro, envía la información de la clave correspondiente a través del enlace de comunicaciones, el dispositvo central evalua dicha clave y regresa una orden al esclavo que

ejecuta la acción correspondiente. Con esta alternatiava se pueden manejar un gran número de accesos de una manera relativamente sencilla. Además, se reduce el problema del cableado, ya que sólo se requieren tres hilos conductores (Rx, Tx y Gnd), la figura 6.1 ilustra la alternativa propuesta.

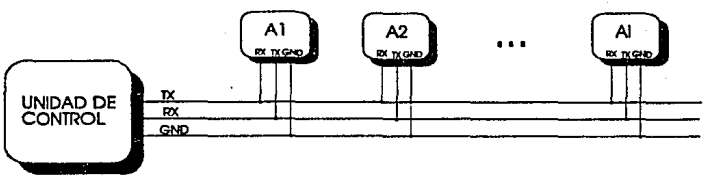


Fig. 6.1 Alternativa propuesta para incrementar el No. accesos controlados.

El desarrollo de un sistema económico implica por una parte la utilización de componentes de bajo costo y por otra su fácil adquisición en el mercado nacional. Ambos aspectos, se tuvieron presentes durante todo el desarrollo del sistema.

De los componentes utilizados, el microcontrolador en el que está basado el sistema, es uno de los más económicos y fáciles de adquirir. El módulo de "display" (AND 771) es el que tiene mayor costo dentro de la Unidad de Control, pero desempeña un papel muy importante, ya que

es a través de éste como la unidad de control se comunica con el usuario. Auque debido a la demanda de éste por su sencillo manejo es también fácil de adquirir.

Tal vez, el C.I. un poco menos accesible y mas caro sea el MAX 232, se utilizó este circuito porque reduce el "hardware" de la interfase entre la Unidad de Control y la PC, ya que sólo requiere de una fuente de 5 V y 4 capacitores para generar señales compatibles con el estándar RS-232C (± 12 V) a partir de señales "TTL" , aunque actualmente está siendo muy comercializado.

La tarjeta principal se diseñó de tal forma que pueda ser usada para otras aplicaciones, ya que en términos generales es una tarjeta de entradas y salidas. En el sistema que se describe, sólo se utiliza un canal del convertidor A/D, se tenía planeado sensar algunas otras variables como humedad relativa y sensores de intrusos con salida analógica, pero incorporar al sistema un sensor de humedad, incrementaría el costo y la humedad no es una variable que resulte de mucho interés en un sistema de seguridad. Respecto a sensores de intrusos estos generalmente entregan un nivel lógico o su salida final es un relevador.

Con respecto a seguridad, el sistema está previsto para manejar sensores de intrusos y de incendio. Considerando que éstos tienen como elemento final de salida un relevador, cuyos contactos pueden ser normalmente abiertos o cerrados. En éste tipo de sensores todo lo relacionado a transducción, amplificación y ajuste de sensibilidad se realiza dentro del propio módulo sensor. Esta característica permite que se puedan formar arreglos serie o paralelo de sensores para incrementar el área de cobertura. El sistema, también, tiene la capacidad sensar niveles lógicos y analógicos, sólo que para este caso es necesario incrementar el "hardware" para el acondicionamiento de las señales.

Como una alternativa, en caso que así se requiera, se pueden incorporar al sistema más circuitos PPI 8255 para incrementar el número de líneas de entrada/salida destinadas a supervisar sensores.

El proyecto se inicio utilizando el *microntrolador* 8748 que es también de Intel, pero a medida que se adicionaban funciones al sistema, se llegó a la necesidad de utilizar la configuración expandida, que incrementaba el costo y las dimensiones del "hardware", fue entonces que se cambio a la familia MCS-51.

Con el sistema desarrollado se comprueba que la tecnología de *microcontroladores* reduce bastante el "hardware" y por lo tanto las dimensiones físicas de las tarjetas electrónicas, ya que con pocos circuitos se tiene un sistema bastante útil y fácil de usar.

Una característica muy importante del sistema desarrolado es que no necesita a la PC de manera permanente para realizar sus funciones, como ocurre con algunos otros. El incluir a la PC de manera opcional nos permite tener ***sistema de seguridad y control de acceso económico.*** Si se dispone de una computadora, se tendrá un sistema mucho más completo, ya que pueden registrarse todos los eventos que ocurren durante la operación de éste.

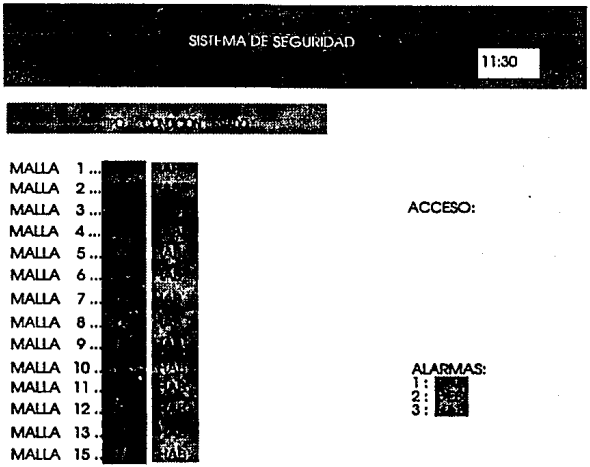
Muchos de los sistemas de seguridad existentes tienen sus parámetros ya definidos y sólo pueden ser cambiandos moviendo el "hardware", en el sistema que se describe todos son programables por "software".

Respecto a la computadora, la información que se maneja en ésta, es lo bastante clara para que usuario pueda conocer el estado de la zona que se está protegiendo. La información despligada no es muy dinámica, pero cumple con el alcance previstos para este trabajo.

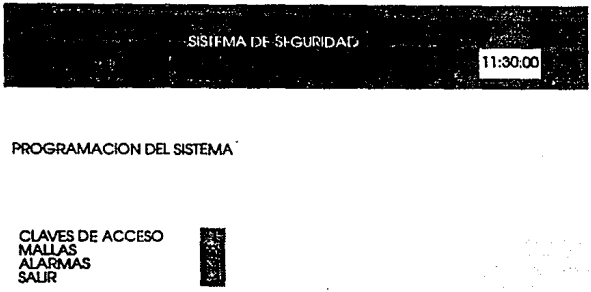
La interacción con el usuario se realiza por medio de presentaciones tipo texto; la selección de opciones por medio de teclas de funciones (Fn) o flechas, lo que hace que parezca un poco rudimentario comparado con los paquetes de "software" actuales. Una alternativa al respecto es el manejo de pantallas y opciones a través de "mouse" y en un ambiente de ventanas ("Windows"). Otra de las alternativas que se pueden considerar, es dejar el "software" residente en la memoria de la máquina y así poder utilizar a ésta para realizar cualquier otra labor.

El objetivo de esta tesis es el diseño y construcción de un sistema controlador de acceso y seguridad basado en un *microcontrolador*. Por lo que considero, a través de lo expuesto, que fue cumplido totalmente.

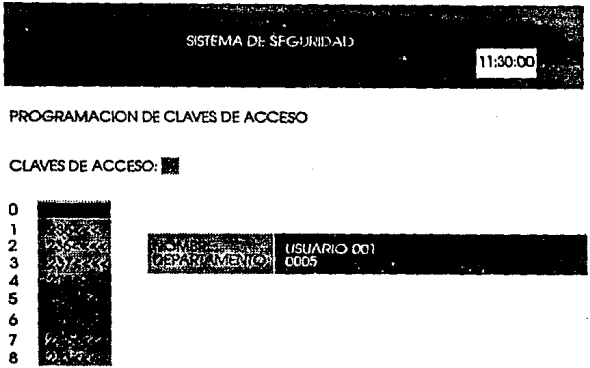
APENDICE A



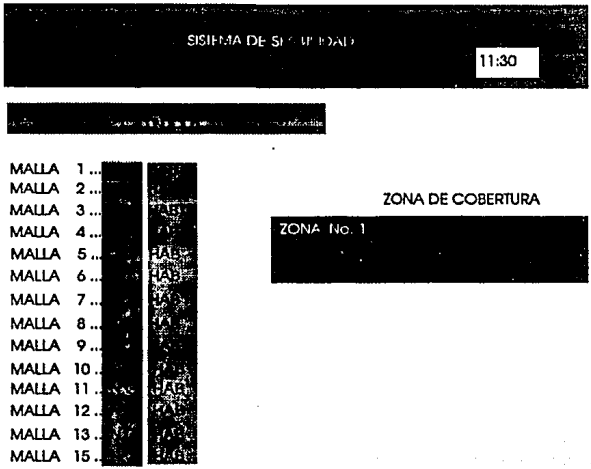
Despliegue en la pantalla de la PC cuando el enlace con la Unidad de Control se ha realizado.



Opciones presentadas para la configuración del sistema cuando se selecciona F1.

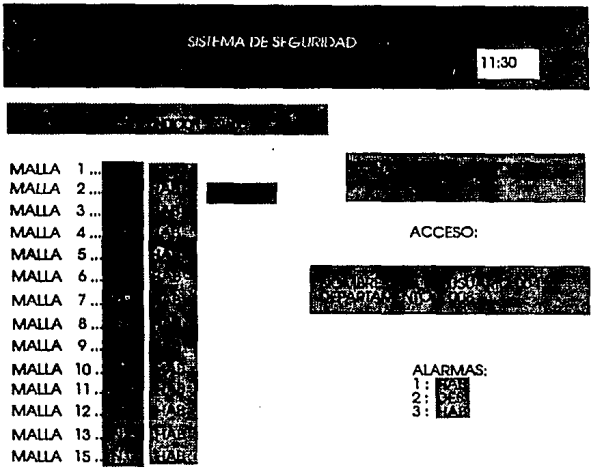


Despliegue para la programación de claves de acceso.



Pantalla presentada para definir el tipo de sensores y las areas de cobertura de cada malla.

Pantalla presentada cuando se desea configurar las alarmas del sistema.



Despliegue presentado cuando se ha concedido el acceso a un usuario y además se ha detectado un intruso en el area protegida.

APENDICE B

Listado del programa ejecutado por el *microcontrolador* para realizar la funciones de seguridad y control de acceso en la Unidad de Control.

```
;
;
        ORG 00H
        JMP 30H             ;

        ORG 03H             ; INTERRUP EXT0.
        JMP INT

        ORG 0BH             ; INTERRUP T/C 0.
        JMP ADC809

        ORG 13H             ; INTERRUP EXT1.
        JMP TECLADO

        ORG 23H             ;
        JMP RECIBE          ; INT. DEL PUERTO SERIE.

        ORG 30H             ;
        MOV SP,#48H
        MOV 1EH,#0FBH
        MOV 1FH,#00
        CLR PSW.1
        MOV 08,#00H
        SETB PX0            ;PRIORIDAD A LA INT EXTENA "0"
        CLR  PT0            ;
        SETB EA             ;HABILITA INTERRUPCIONES
        SETB EX0            ;HABILITA INTERRUPCION EXTERNA 0
        SETB IT0            ;ACTIVADA CON FLANCO
        SETB EX1            ; HABILITA INTERRUP EXT1
        CLR IT1             ; POR NIVEL

        MOV DPTR,#8003H      ;PROGRAMA 8255
        MOV A,#9AH
        MOVX @DPTR,A

        MOV THOD,#26H       ;PROGRAMA EL T/C 0 EN MODO CONTADOR
                            ;Y EL T/C 1 EN MODO TIMER
        ;************************************************
        MOV TL1,#0FEH       ; TIMER 1 EN MODO 2
        MOV TH1,#0FEH       ; PARA GENERAR LA FRECUENCIA
                            ; 9600 bps PARA LA COMUNICACION CON LA PC.
        SETB TR1            ; SE INICIA EL CONTEO (BPS)
        ;************************************************

        ;************************************************
        MOV TL0,#0FFH       ; MANEJO DEL ADC809
        MOV TH0,#0FFH       ; CONSTANTE
        SETB TR0            ; INICIA EL CONTEO (ADC)
        ;************************************************
;-----------------------------------------
        MOV R0,#10H         ;
        MOV R1,#3           ;
        MOV DPTR,#6000H     ;
FILTRO: MOVX @DPTR,A        ;
        JB P3.4, $          ; DATOS INICIALES DEL
        MOVX A,@DPTR        ; SENSOR DE TEMPERATURA
        CLR C               ;
        SUBB A,#29
        MOV @R0,A           ;
```

```
        INC RO              ;
        DJNZ R1,FILTRO      ;
;-------------------------------------------
        CALL DESACT         ; DESACTIVA LA MEMORIA X2444
        CALL SETDISP        ;INICIALIZA "DISPLAY"
        ;*****************************************************

        MOV DPTR,#200FH     ; SE ESTABLECE EL NUMERO DE
        MOV A,#09H          ; CLAVES DE ACCESO INICIALES.
        MOVX @DPTR,A        ;
;*************************************
;INICIO DEL PROGRAMA
;*************************************
        CALL SEGURI         ;DESPLIEGA EL MESAJE DE INICIO
        CALL MODO1          ;ESTABLECE EL MODO DE OPERACION DE
        CLR RI              ;DEL PUERTO SERIE.
        SETB REN
        SETB ES

        MOV R1,#28H         ;DIRECCION DE LA CLAVE "PASSWORD"
        MOV 09,#01          ;CONTADOR PARA FLECHA
        CLR F0
SERVI:  MOV R0,#20H         ;APUNTADOR DE CLAVE DE ACCESO
        MOV R2,#08H
SCAN:
        CALL CHECA          ;SUPERVISA LA CONDICION DE LOS SENSORES
        MOV DPTR,#8002H
        JB F0,PROGRA
        MOV R3,#0EEH
SIGUE:  MOV A,R3
        MOVX @DPTR,A
        CALL CONTROL        ;SUPERVISA LA CONDICION DE CADA TECLADO
        JNZ IDENTI
        MOV A,R3
        RL A
        MOV R3,A
        CJNE A,#0EEH,SIGUE
        MOV A,08
        JNZ NTEMP
        MOV A,17H
        CJNE A,#01,NTEMP
        CALL STADC
        JB P3.4,$
        CALL ADC809         ;LECTURA DEL SENSOR DE TEMPERATURA
NTEMP:  JMP SCAN
IDENTI: CALL NTECLA         ;DECODIFICACION DEL TECLADO
        JZ SCAN
        CJNE A,#3CH,OVER
        CALL COMPARA
        JMP SERVI
OVER:   DJNZ R2,SCAN
        JMP SERVI

;************************************
; SUBRUTINAS
;*******************************************
;EVALUACION DE LA CLAVE DE ACCESO PROPORCIONADA
;*******************************************
COMPARA:
        MOV DPTR,#200FH
        MOV R0,#20H
```

```
        MOVX A,@DPTR
        MOV R2,A
        MOV R5,A
        INC DPTR
ETI4:   PUSH DPL
        PUSH DPH
ETI3:   MOV 03,@RO
        MOVX A,@DPTR
        CJNE A,03H,ETI1
        CJNE A,#3CH,ETI2
        CLR C
        MOV A,R5
        SUBB A,R2
        CALL ABRE
        POP DPH
        POP DPL
        RET
ETI2:   INC DPTR
        INC RO
        JMP ETI3
ETI1:   POP DPH
        POP DPL
        MOV A,DPL
        CLR C
        ADD A,#08H
        MOV DPL,A
        MOV A,DPH
        ADDC A, #00H
        MOV DPH,A
        MOV RO,#20H
        DJNZ R2,ETI4
        RET
;*********************************
;DECODIFICACION DEL TECLADO, UTILIZADO PARA LA CONFIGURACION
;DEL SISTEMA
;*********************************
PROGRA: PUSH 01
        MOV A,08
        CJNE A,#22H,PROGRA1
        MOV A,#0C4H
        CALL CDISP
        MOV 01,#30H
SCAN2:  CALL SCAN1
        JNZ SPROG1
        JMP FPROGN
SPROG1: MOV A,R3
        CPL A
        JB ACC.0, ABAJ1
        JB ACC.1, IZQU1
        JB ACC.2, DERE1
        JB ACC.3, ARRI1
        JMP OPRI3
IZQU1:  CJNE R1,#30H,NREC1
        JMP OPRI3
NREC1:  MOV A,#00010000B
        CALL CDISP
        DEC R1
        JMP OPRI3
DERE1:  CJNE R1,#37H,NREC2
        JMP OPRI3
NREC2:  MOV A,#00010100B
```

```
        CALL CDISP
        INC R1
        JMP OPRI3
ABAJ1:
        DEC 9
        MOV A,09
        ANL A,#0FH
        ORL A,#30H
        MOV @R1,A
        CALL DISDAT
        MOV A,#00010000B
        CALL CDISP
        JMP OPRI3
ARRI1:  INC 09
        MOV A,09H
        ANL A,#0FH
        ORL A,#30H
        MOV @R1,A
        CALL DISDAT
        MOV A,#00010000B
        CALL CDISP
OPRI3:  CALL CONTROL
        JNZ OPRI3
        LJMP SCAN2
PROGRA1:CJNE A,#23H,PROGRA2
        JMP PROGRA3
PROGRA2:JMP PROGRA4
PROGRA3:MOV R4,#01
        MOV A,#0C5H
        CALL CDISP
SCAN3:  CALL SCAN1
        JNZ SPROG2
        JMP FPROGN
SPROG2: MOV A,R3
        CPL A
        JB ACC.0, ABAJ2
        JB ACC.1, IZQU2
        JB ACC.2, DERE2
        JB ACC.3, ARRI2
        JMP OPRI4
IZQU2:  CJNE R4,#01,NREC3
        JMP OPRI4
NREC3:  MOV A,#00010000B
        CALL CDISP
        DEC R4
        JMP OPRI4
DERE2:  CJNE R4,#10H,NREC4
        JMP OPRI4
NREC4:  MOV A,#00010100B
        CALL CDISP
        INC R4
        JMP OPRI4
ABAJ2:  CJNE R4,#08H,NEQ1
BYTE1:  MOV R5,04
        MOV A,#01111111B
SROT1:  RL A
        DJNE R5,SROT1
        MOV R1,#30H
        ANL A,@R1
        MOV @R1,A
        JMP LAB0
```

```
NEQ1:   JC BYTE1
        MOV A,R4
        SUBB A,#08
        MOV R5,A
        MOV A,#01111111B
SROT2:  RL A
        DJNZ R5,SROT2
        MOV R1,#31H
        ANL A,@R1
        MOV @R1,A

LAB0:   MOV A,#30H
        CALL DISDAT
        MOV A,#10H
        CALL CDISP
        JMP OPRI4

ARRI2:  CJNE R4,#08,NEQ2
BYTE2:  MOV R5,04H
        MOV A,#10000000B
SROT3:  RL A
        DJNZ R5,SROT3
        MOV R1,#30H
        ORL A,@R1
        MOV @R1,A
        JMP LAB1
NEQ2:   JC BYTE2
        MOV A,R4
        SUBB A,#08
        MOV R5,A
        MOV A,#10000000B
SROT4:  RL A
        DJNZ R5,SROT4
        MOV R1,#31H
        ORL A,@R1
        MOV @R1,A

LAB1:   MOV A,#31H
        CALL DISDAT
        MOV A,#10H
        CALL CDISP
OPRI4:  CALL CONTROL
        JNZ OPRI4
        JMP SCAN3
PROGRA4:CJNE A,#24H,PROGRA5
PROGRA5:CJNE A,#25H,PROGRA6
        JMP PROGRA7
PROGRA6:JMP PROGRA8
PROGRA7:MOV R4,#01
        MOV A,#0C5H
        CALL CDISP
SCAN4:  CALL SCAN1
        JNZ SPROG3
        JMP FPROGM
SPROG3: MOV A,R3
        CPL A
        JB ACC.0, ABAJ3
        JB ACC.1, IZQU3
        JB ACC.2, DERE3
        JB ACC.3, ARRI3
        JMP OPRI5
```

```
IZQU3:  CJNE R4,#01,NREC5
        JMP OPRI5
NREC5:  MOV A,#00010000B
        CALL CDISP
        DEC R4
        JMP OPRI5
DERE3:  CJNE R4,#10H,NREC6
        JMP OPRI4
NREC6:  MOV A,#00010100B
        CALL CDISP
        INC R4
        JMP OPRI5
ABAJ3:  CJNE R4,#08H,NEQ3
BYTE3:  MOV R5,04
        MOV A,#01111111B
SROT5:  RL A
        DJNZ R5,SROT5
        MOV R1,#30H
        ANL A,@R1
        MOV @R1,A
        JMP LAB2
NEQ3:   JC BYTE3
        MOV A,R4
        SUBB A,#08
        MOV R5,A
        MOV A,#01111111B
SROT6:  RL A
        DJNZ R5,SROT6
        MOV R1,#31H
        ANL A,@R1
        MOV @R1,A

LAB2:   MOV A,#30H
        CALL DISDAT
        MOV A,#10H
        CALL CDISP
        JMP OPRI5

ARRI3:  CJNE R4,#08,NEQ5
BYTE5:  MOV R5,04H
        MOV A,#10000000B
SROT7:  RL A
        DJNZ R5,SROT7
        MOV R1,#30H
        ORL A,@R1
        MOV @R1,A
        JMP LAB3
NEQ5:   JC BYTE5
        MOV A,R4
        SUBB A,#08
        MOV R5,A
        MOV A,#10000000B
SROT8:  RL A
        DJNZ R5,SROT8
        MOV R1,#31H
        ORL A,@R1
        MOV @R1,A

LAB3:   MOV A,#31H
        CALL DISDAT
        MOV A,#10H
```

```
        CALL CDISP
OPRI5:  CALL CONTROL
        JNZ OPRI5
        JMP SCAN4
PROGRAB:CJNE A,#26H,FPROGN
        MOV R4,#01
        MOV A,#0C9H
        CALL CDISP
SCAN5:  CALL SCAN1
        JNZ SPROG4
        JMP FPROGN
SPROG4: MOV A,R3
        CPL A
        JB ACC.0, ABAJ4
        JB ACC.1, IZQU4
        JB ACC.2, DERE4
        JB ACC.3, ARRI4
        JMP OPRI6
IZQU4:  CJNE R4,#01,NREC7
        JMP OPRI6
NREC7:  MOV A,#00010000B
        CALL CDISP
        DEC R4
        JMP OPRI6
DERE4:  CJNE R4,#08,NREC8
        JMP OPRI4
NREC8:  MOV A,#00010100B
        CALL CDISP
        INC R4
        JMP OPRI6
ABAJ4:
        MOV R5,04
        MOV A,#01111111B
SROT9:  RL A
        DJNZ R5,SROT9
        MOV R1,#34H
        ANL A,@R1
        MOV @R1,A
        MOV A,#30H
        CALL DISDAT
        MOV A,#10H
        CALL CDISP
        JMP OPRI6

ARRI4:
        MOV R5,04H
        MOV A,#10000000B
SROT10: RL A
        DJNZ R5,SROT10
        MOV R1,#34H
        ORL A,@R1
        MOV @R1,A
        MOV A,#31H
        CALL DISDAT
        MOV A,#10H
        CALL CDISP
OPRI6:  CALL CONTROL
        JNZ OPRI6
        JMP SCAN5

FPROGN: POP 01
```

```
        JMP SERVI

;*****************************************
;ATENCION A LA INTERRUPCION DE LA TECLA "TED"
;*****************************************
TECLADO:
        CLR EX1
        PUSH DPL
        PUSH DPH
        PUSH ACC
        PUSH 00
        PUSH 01
        PUSH 02
        PUSH 03
        MOV A,08H
        JNZ MENU
        MOV DPTR,#8000H
        MOVX A,@DPTR
        JNB ACC.0,CMBPASS
        CALL PASS
        JMP SALIR
CMBPASS:CALL PASSC
        JMP SALIR
MENU:
        CJNE A,#01H,ETI5
        MOV DPTR,#8000H
        MOVX A,@DPTR
        JNB ACC.0,NEWPASS
        MOV R2,#08H
        MOV DPTR,#2000H
        MOV R0,#28H
ETI7:   MOV 01,@R0
        MOVX A,@DPTR
        CJNE A,01H,ETI6
        CJNE A,#3CH,ETI8
        CALL SUBR1
        CALL CERO
        CALL DESALAR
        JMP SALIR
ETI8:   INC R0
        INC DPTR
        DJNZ R2,ETI7
        CALL SUBR1
        CALL CERO
        JMP SALIR
ETI6:   MOV 08,#00
        CALL SEGURI
        JMP SALIR
NEWPASS:
        MOV R0,#28H
        MOV DPTR,#2000H
        MOV R1,#08H
CAMP:   MOV A,@R0
        MOVX @DPTR,A
        INC R0
        INC DPTR
        DJNZ R2,CAMP
        CALL SEGURI
        MOV 08,#00
        JMP SALIR
ETI5:   CJNE A,#07H,SELEC
```

```
        MOV 08,#00
        CALL SEGURI
        JMP SALIR
SELEC:  CJNE A,#08,NSALIR
        CALL CERO
        JMP SALIR
NSALIR:
        CJNE A,#02H,VER1
        MOV A,#01H
        CALL CDISP
        CALL UNO
        MOV A,#0C0H
        CALL CDISP
        MOV DPTR,#200FH
        MOVX A,@DPTR        ;nUMERO DE CLAVES
        MOV 0CH,A

        MOV 0DH,#00
        MOV A,0DH
        MOV B,#100
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,#10H
        XCH A,B
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        MOV A,#20H
        CALL DISDAT

        MOV R2,#08
        MOV R0,#30H
        INC DPTR
        MOV 0AH,DPL
        MOV 0BH,DPH
DCLAVE:
        MOVX A,@DPTR
        MOV @R0,A
        CALL DISDAT
        INC DPTR
        INC R0
        DJNZ R2,DCLAVE
        MOV A,08
        SETB ACC.4
        MOV 08,A
        JMP SALIR

VER1:   CJNE A,#03,VER2
        CALL DEFENT
        MOV A,#0C5H
        CALL CDISP
        MOV R0,#30H
        MOV DPTR,#2008H
        MOVX A,@DPTR
        MOV @R0,A
        CALL DESPE
        INC DPTR
```

```
        INC R0
        MOVX A,@DPTR
        MOV @R0,A
        CALL DESPE
        MOV A,08
        SETB ACC.4
        MOV 08,A
        JMP SALIR
VER2:   CJNE A,#04H,VER3
        MOV A,#80H
        CALL CDISP
        CALL TRES
        MOV A,#0C0H
        CALL CDISP
        MOV 08,#24H
        JMP SALIR
VER3:   CJNE A,#05,VER4
        CALL ACTSEN
        MOV A,#0C5H
        CALL CDISP
        MOV R0,#30H
        MOV DPTR,#200AH
        MOVX A,@DPTR
        MOV @R0,A
        CALL DESPE
        INC DPTR
        MOVX A,@DPTR
        MOV @R0,A
        CALL DESPE
        MOV A,08
        SETB ACC.4
        MOV 08,A
        JMP SALIR
VER4:   CJNE A,#06H,VER5
        CALL ACTALAR
        MOV A,#0C9H
        CALL CDISP
        MOV R0,#34H
        MOV DPTR,#200CH
        MOVX A,@DPTR
        MOV @R0,A
        CALL DESPE
        MOV A,08
        SETB ACC.4
        MOV 08,A
        JMP SALIR
VER5:
        MOV A,08H
        ANL A,#0F0H
        SWAP A
        CJNE A,#01,SMODIF
        MOV A,08
        ANL A,#0FH
        SETB ACC.5
        MOV 08,A
        SETB F0
        JMP SALIR
SMODIF: MOV A,08H
        CJNE A,#22H,MODIF1

        MOV R0,#30H
```

```
        MOV DPL,0AH
        MOV DPH,0BH
        MOV R2,#08
FMODIF: MOV A,@R0
        MOVX @DPTR,A
        INC DPTR
        INC R0
        DJNZ R2,FMODIF
        MOV 08,#12H
        CLR F0
        JMP SALIR
MODIF1: CJNE A,#23H,MODIF2
        MOV R0,#30H
        MOV DPTR,#2008H
        MOV A,@R0
        MOVX @DPTR,A
        INC DPTR
        INC R0
        MOV A,@R0
        MOVX @DPTR,A
        CALL CERO
        CLR F0
        JMP SALIR
MODIF2: CJNE A,#24H,MODIF3
        CALL CERO
        JMP SALIR
MODIF3: CJNE A,#25H,MODIF4
        MOV R0,#30H
        MOV DPTR,#200AH
        MOV A,@R0
        MOVX @DPTR,A
        INC DPTR
        INC R0
        MOV A,@R0
        MOVX @DPTR,A
        CALL CERO
        CLR F0
        JMP SALIR
MODIF4: CJNE A,#26H,SALIR
        MOV R0,#34H
        MOV DPTR,#200CH
        MOV A,@R0
        MOVX @DPTR,A
        CALL CERO
        CLR F0
SALIR:  CALL RETARDO
        POP 03
        POP 02
        POP 01
        POP 00
        POP ACC
        POP DPH
        POP DPL
        JNB P3.3,$
        SETB EX1
        CALL RETARDO
        CALL RETARDO
        MOV R1,#28H
        MOV R4,#00H
        RETI
;*********************************************************
```

```
; IDENTIFICA LA TECLA OPRIMRDA
;************************************************
NTECLA:
        PUSH DPL
        PUSH DPH
        PUSH 02
        PUSH 04
        JNB ACC.7,AUXF
        MOV A,08H
        JZ OPRIAUX
        MOV A,R3
        CPL A
        JB ACC.0, ABAJO
        JB ACC.1, IZQUI
        JB ACC.2, DEREC
        JB ACC.3, ARRIB
ARRIB:  JMP ARRIBA
OPRIAUX:JMP OPRI
AUXF:   JMP NTEC5
IZQUI:  CJNE R1,#28H,NO8D
        JMP OPRI
NO8D:   MOV A,#00010000B
        CALL CDISP
        DEC R1
        MOV A,#00H
        JMP OPRI
DEREC:  CJNE R1,#30H,NOD8
        JMP OPRI

NOD8:   MOV A,#00010100B
        CALL CDISP
        INC R1
        MOV A,#00H
        JMP OPRI

ABAJO:  MOV A,08
        CJNE A,#01,LINSG1
        DEC 09
        MOV A,09H
        ANL A,#0FH
        ORL A,#30H
        MOV @R1,A
        CALL DISDAT
        MOV A,#00010000B
        CALL CDISP
        MOV A,#00H
        JMP OPRI
LINSG1: CJNE A,#02H,LSIG1
        CALL DOS
        JMP OPRI
LSIG1:  CJNE A,#03,LSIG2
        CALL TRES
        JMP OPRI
LSIG2:  CJNE A,#04,LSIG3
        CALL CUATRO
        JMP OPRI
LSIG3:  CJNE A,#05,LSIG4
        CALL CINCO
        JMP OPRI
LSIG4:  CJNE A,#06,LSIG5
        CALL SEIS
```

```
        JMP OPRI
LSIG5:
        CJNE A,#12H,SELE2
        MOV A,#0C0H
        CALL CDISP
        DEC 0CH
        INC 0DH
        MOV A,0CH
        CJNE A,#00H,SVER
        CALL SIETE
        JMP OPRI
SVER:
        MOV A,0DH
        MOV B,#100
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,#10
        XCH A,B
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        MOV A,#20H
        CALL DISDAT

        CLR C
        MOV A,0AH
        ADD A,#08
        MOV DPL,A
        MOV 0AH,A
        MOV A,0BH
        ADDC A,#00
        MOV DPH,A
        MOV 0BH,A

        PUSH 01
        MOV R2,#08
        MOV R1,#30H
OTDIG:  MOVX A,@DPTR
        MOV @R1,A
        CALL DISDAT
        INC DPTR
        INC R1
        DJNZ R2,OTDIG
        POP 01
SELE2:
        JMP OPRI
ARRIBA: MOV A,08
        CJNE A,#01,LINANT
        INC 09
        MOV A,09H
        ANL A,#0FH
        ORL A,#30H
        MOV @R1,A
        CALL DISDAT
        MOV A,#00010000B
        CALL CDISP
        MOV A,#00H
```

```
        JMP OPRI
LINANT: CJNE A,#02,LANT1    ;
        JMP OPRI            ;
LANT1:  CJNE A,#03,LANT2    ;
        CALL UNO            ;DESPLIEGUE DE OPCIONES
        JMP OPRI            ;DURANTE LA CONFIGURACION
LANT2:  CJNE A,#04,LANT3    ;DEL SISTEMA
        CALL DOS            ;
        JMP OPRI            ;
LANT3:  CJNE A,#05,LANT4    ;
        CALL TRES           ;
        JMP OPRI            ;
LANT4:  CJNE A,#06H,LANT5   ;
        CALL CUATRO         ;
        JMP OPRI            ;
LANT5:  CJNE A,#07H,OPRI    ;
        CALL CINCO          ;
        JMP OPRI            ;

NTEC5:                      ;
        MOV R4, #05H        ;
NOBIT:  RLC A               ;
        DEC R4              ;
        JC RENGLON          ;
        JMP NOBIT           ;
RENGLON:MOV A,R3            ;DECODIFICACION DEL
        CPL A               ;TECLADO DE ACCESO
        MOV R3,#09H         ;
NOACT:  RLC A               ;
        JC FIN              ;
        DEC R3              ;
        DEC R3              ;
        DEC R3              ;
        JMP NOACT           ;
FIN:    MOV A,R3            ;
        ADD A,R4            ;
        ORL A,#30H          ;
        MOV @R0,A           ;
        CALL BEEP           ; SONIDO QUE INDICA CUANDO LA
                            ; TECLA SE HA RECONOCIDO
        CALL RETARDO
        MOV A,@R0           ;
        INC R0              ;
        JMP TERMI           ;
OPRI:   CALL CONTROL        ;
        JNZ OPRI            ;
        MOV A,#00           ;
TERMI:  POP 04              ;
        POP 02              ;
        POP DPH             ;
        POP DPL             ;
        RET                 ;
;*****************************************
;SUBRUTINA ENCARGADA DE RECIBIR LOS DATOS TRANSMITIDOS
;POR LA PC
;*****************************************
RECIBE:
        PUSH PSW            ;
        PUSH DPL            ;SALVA EL "STACK"
        PUSH DPH            ;
        PUSH ACC            ;
```

```
        SETB RS0            ;BANCO DE REGISTROS #3
        SETB RS1            ;
        JNB RI,INT_E
        JMP I_CORR
INT_E:  JMP FINAL
I_CORR: CLR RI
        MOV A,SBUF
        CJNE R7,#00H,ENLACE1  ;
        MOV A,SBUF            ;PRUEBA DEL ENLACE PC/UNIDAD DE
        CJNE A,1EH,ENLACE2    ;CONTROL
        CJNE R6,#0FFH,ENLACE3 ;
        MOV SBUF,#0FBH      ;
        CALL TX1            ;
        MOV SBUF,#0FCH      ;RESPUESTA DE LA U. DE CONTROL
        CALL TX1            ;PARA INDICAR QUE EL ENLACE SE HA
        MOV SBUF,#0FDH      ;CONCRETADO
        CALL TX1            ;
        MOV SBUF,#0FEH      ;
        CALL TX1            ;
        MOV SBUF,#0FFH      ;
        CALL TX1            ;
        MOV R7,#01
        JMP FINAL
ENLACE2:MOV R6,#0FBH
        JMP FINAL
ENLACE3:INC R6
        JMP FINAL
ENLACE1:CJNE R7,#02,CONFIG
        MOV DPL,R0            ; RECIBE DATOS DE CONFIGURACION
        MOV DPH,R1
        MOV A,SBUF
        MOVX @DPTR,A
        INC DPTR
        MOV R0,DPL
        MOV R1,DPH
        MOV A,DPH
        CJNE A,#28H,T_CONF
        MOV R7,#01
T_CONF:
        JMP FINAL
CONFIG: MOV A,SBUF
        CJNE A,#218,DATOS
        MOV SBUF,#0FFH
        CALL TX1
        MOV SBUF,#0FEH
        CALL TX1
        MOV SBUF,#0FDH
        CALL TX1
        MOV SBUF,#0FCH
        CALL TX1
        MOV SBUF,#218
        CALL TX1
        MOV DPTR,#2000H     ;
STX:    MOVX A,@DPTR
        MOV SBUF,A
        CALL TX1
        INC DPTR
        MOV A,DPH           ;ENVIA LOS PARAMETROS DE CONFIGURACION
        CJNE A,#28H,STX     ;A LA PC
        JMP FINAL
DATOS:  CJNE A,#219,RELOJ
```

```
        MOV A,#0C0H
        CALL CDISP
        CALL RX1
        MOV A,SBUF          ;RECIBE HORA
        MOV B,#10
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        MOV A,#3AH
        CALL DISDAT
        CALL RX1
        MOV A,SBUF
        MOV B,#10
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        JMP FINAL
RELOJ:  CJNE A,#220,NCONFIG
        MOV DPTR,#2000H
        MOV R7,#02
        MOV R0,DPL
        MOV R1,DPH
        JMP FINAL
NCONFIG:CJNE A,#221,DES_ALR   ;RECIBE LA ORDEN PARA DESHABILITAR
        CALL DESALAR           ;LAS SENALES DE LARMA.
        JMP FINAL
DES_ALR:CJNE A,#222,PRUEBAS   ;SE VAN HA EFECTUAR PRUEBAS DE OPERACION
        MOV 08,#24H            ;AL SISTEMA
        JMP FINAL
PRUEBAS:CJNE A,#223,F_PRUE     ; TERMINARON LAS PRUEBAS
        MOV 08,#00
        CALL SEGURI
F_PRUE: CJNE A,#224,FINAL
        MOV R7,#00
FINAL:
        POP ACC
        POP DPH
        POP DPL
        POP PSW
        RETI
;*******************************************
;RETARDO
;*******************************************
RETARDO:PUSH 07
        MOV R6, #35H        ;
CTE:    MOV R7, #40H        ;
TIEMPO: NOP                 ;
        NOP
        DJNZ R7,TIEMPO
        DJNZ R6,CTE
        POP 07
        RET                 ;
;*******************************************
;RUTINA DE AUTO PRUEBA
;*******************************************
```

```
INT:    PUSH DPL
        PUSH DPH
        PUSH ACC
        JB P1.5, LABELO     ;
        SETB P1.7           ;
        JMP END_AUT         ;
LABELO: CLR P1.7            ;
END_AUT:
        MOV DPTR,#8003H     ;PROGRAMA 8255
        MOV A,#9AH
        MOVX @DPTR,A
        POP ACC
        POP DPH
        POP DPL
        RETI
;*******************************************
;INICIALIZA "DIPLAY" AND 771
;*******************************************
SETDISP:MOV DPTR,#4000H     ;
        MOV A,#00111011B    ; SET  (8 BITS)
        MOVX @DPTR,A
        CALL RETARDO
        CALL RETARDO
        MOV A,#00111011B    ; SET  (8 BITS)
        MOVX @DPTR,A
        CALL RETARDO
        MOV A,#00111011B    ; SET  (8 BITS)
        MOVX @DPTR,A
        CALL RETARDO
        MOV A,#00111011B    ; SET  (8 BITS)
        MOVX @DPTR,A

        CALL BUSYF
        MOV DPTR,#4000H
        MOV A,#00001000B    ;DIP OFF
        MOVX @DPTR,A
        CALL BUSYF
        MOV DPTR,#4000H
        MOV A,#01H          ;DISP CLEAR
        MOVX @DPTR,A
        CALL BUSYF
        MOV DPTR,#4000H
        MOV A,#00000110B    ;ENT MOD SET
        MOVX @DPTR,A
        CALL BUSYF
        MOV DPTR,#4000H
        MOV A,#00001110B    ;DISP ON
        MOVX @DPTR,A
        CALL BUSYF
        RET
;*************************************************
;ESCRIBE UN CARACTER EN EL "DISPLAY
;*************************************************
DISDAT: PUSH DPL
        PUSH DPH
        MOV DPTR,#4001H
        MOVX @DPTR,A
        CALL BUSYF
        POP DPH
        POP DPL
        RET
```

```
;************************************************
;ENVIA UN COMANDO AL "DISPLAY"
;************************************************
CDISP:  PUSH DPL
        PUSH DPH
        MOV DPTR,#4000H
        MOVX @DPTR,A
        CALL BUSYF
        POP DPH
        POP DPL
        RET
;*********************************************
;PREGUNTA AL "DISPLAY" SI ESTA LISTO PARA RECIBIR DATOS
;*********************************************
BUSYF:
        MOV DPTR,#4002H     ;R/W=1,RS=0
LABEL1: MOVX A,@DPTR
        JB ACC.7,LABEL1
        RET
;************************************************
; SUPERVISA LA TRANSMISION DE UN DATO
;************************************************
TX1:    JNB TI, $
        CLR TI
        RET
;************************************************
;SUPERVISA LA RECEPCION DE UN DATO
;************************************************
RX1:    CLR RI
        JNB RI, $
        RET
;************************************************
;SUPERVISA A CADA TECLADO
;************************************************
CONTROL:PUSH DPL
        PUSH DPH
        MOV DPTR,#8002H
        MOVX A,@DPTR        ;
        ORL A,#0FH          ;
        CPL A               ;
        JZ NOTEC
        CALL RETARDO
        MOVX A,@DPTR
        CPL A
NOTEC:  POP DPH
        POP DPL
        RET                 ;
;************************************************
;ACTIVA Y DEDACTIVA A LA MEMORIA X2444
;************************************************
DESACT: CLR P1.6
        RET
ACTIVA: SETB P1.6
        RET

;************************************************
;PROGRAMA EL PUERTO SERIE PARA LA COMUNICACION
;CON LA PC A 9600 BAUDS
;************************************************
MODO1:  CLR SM0             ; PUERTO SERIE
        SETB SM1            ; MODO 1
```

```
        CLR RI
        RET
;************************************************
;INICIA AL CONVERTIDOR A/D
;************************************************
STADC:  PUSH DPL
        PUSH DPH
        MOV DPTR,#6000H
        MOVX @DPTR,A
        POP DPH
        POP DPL             ;
        RET
;************************************************
;PROCESA LA INFORMACION QUE SE OBTIENE DEL CONVERTIDOR A/D
;************************************************
ADC809: PUSH DPL
        PUSH DPH
        PUSH ACC
        PUSH PSW
        MOV A,#0CH
        CALL CDISP
        SETB RS1            ; BANCO DE REGISROS
        CLR RS0             ; #2
        MOV A,#0D5H
        CALL CDISP
        MOV DPTR, #6000H    ;
        MOVX A, @DPTR       ;
        CLR C
        SUBB A,#29
        MOV R0,A
        ADD A,R1
        ADDC A,R2
        MOV B,#3
        DIV AB
        MOV B,#100
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,#10
        XCH A,B
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        XCH A,R0
        XCH A,R1
        XCH A,R2
        MOV A,#0C0H
        CALL CDISP
        MOV A,#0EH
        CALL CDISP
        POP PSW
        POP ACC
        POP DPH
        POP DPL
        RET
;************************************************
;ACTIVA LA CERRADURA DE LA PUERTA PARA PERMITIR EL ACCESO
;************************************************
```

```
ABRE:    CLR P1.0
         MOV A,1FH      ;CHECA SI EXISTE ENLACE
         JZ NO_PC       ;DE COMUNICACION
         MOV SBUF,#0FFH
         CALL TX1
         NOP
         MOV SBUF,#0FEH
         CALL TX1
         NOP
         MOV SBUF,#0FDH
         CALL TX1
         NOP
         MOV SBUF,#0FCH
         CALL TX1
         NOP
         MOV SBUF,#220
         CALL TX1
         MOV SBUF,A
         CALL TX1
NO_PC:   MOV R5,#45H
ABIERT:  CALL RETARDO
         CALL CHECA
         DJNZ R5,ABIERT
         SETB P1.0
         RET
;************************************************
;CODIFICACCION DE LOS MENSAJES DESPLEGADOS
;************************************************
DESPE:   MOV R1,#08H
         MOV R3,A
SWRX:    RRC A
         MOV R3,A
         JC WR1
         MOV A,#30H
         CALL DISDAT
         MOV A,R3
         DJNZ R1,SWRX
         JMP FWRX
WR1:     MOV A,#31H
         CALL DISDAT
         MOV A,R3
         DJNZ R1,SWRX
FWRX:    RET

PASS:    MOV A,#01H          ;
         CALL CDISP
         MOV R7,#01H
ETPASS:  MOV A,R7
         CALL TPASS
         CALL DISDAT
         INC R7
         CJNE R7,#19,ETPASS
         MOV A,#0C0H
         CALL CDISP
         MOV 08H,#01H
         RET

TPASS:   MOVC A,@A+PC
         RET
         DB "CLAVE DE SEGURIDAD"
```

```
PASSC:  MOV A,#01H          ;
        CALL CDISP
        MOV R7,#01H
CAMPASS:MOV A,R7
        CALL TCPASS
        CALL DISDAT
        INC R7
        CJNE R7,#19,CAMPASS
        MOV A,#0C0H
        CALL CDISP
        MOV 08H,#01
        RET

TCPASS: MOVC A,@A+PC
        RET
        DB "NUEVA CLAVE (PASS)"

CERO:
        MOV A,#01H              ; LIMPIA DISPLAY
        CALL CDISP
        MOV A,#10001000B        ;POCICION DEL CURSOR
        CALL CDISP
        MOV R7,#01H
ETIQ0:  MOV A,R7
        CALL TABLA0
        CALL DISDAT
        INC R7
        CJNE R7,#08H,ETIQ0
        MOV A,#11000000B
        CALL CDISP
        CALL UNO
        RET

TABLA0: MOVC A,@A+PC
        RET
        DB "M E N U"

UNO:    MOV R7,#01H
ETIQ1:  MOV A,R7
        CALL TABLA1
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ1
        MOV 08H,#02H
        MOV A,#0C0H
        CALL CDISP
        RET

TABLA1: MOVC A,@A+PC
        RET
        DB "1.- Cambio de clave      "

DOS:    MOV R7,#01H
ETIQ2:  MOV A,R7
        CALL TABLA2
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ2
        MOV 08H,#03H
        MOV A,#0C0H
        CALL CDISP
```

```
        RET

TABLA2: MOVC A,@A+PC
        RET
        DB "2.- Definir entradas     "

TRES:   MOV R7,#01H
ETIQ3:  MOV A,R7
        CALL TABLA3
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ3
        MOV 08H,#04H
        MOV A,#0C0H
        CALL CDISP
        RET

TABLA3: MOVC A,@A+PC
        RET
        DB "3.- Prueba del sistema   "

CUATRO: MOV R7,#01H
ETIQ4:  MOV A,R7
        CALL TABLA4
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ4
        MOV 08H,#05H
        MOV A,#0C0H
        CALL CDISP
        RET

TABLA4: MOVC A,@A+PC
        RET
        DB "4.- ACTIVAR SENSORES     "

CINCO:  MOV R7,#01H
ETIQ5:  MOV A,R7
        CALL TABLA5
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ5
        MOV 08H,#06H
        MOV A,#0C0H
        CALL CDISP
        RET

TABLA5: MOVC A,@A+PC
        RET
        DB "5.- ACTIVAR ALARMAS      "

SEIS:   MOV R7,#01H
ETIQ6:  MOV A,R7
        CALL TABLA6
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ6
        MOV 08H,#07H
        MOV A,#0C0H
        CALL CDISP
        RET
```

```
TABLA6: MOVC A,@A+PC
        RET
        DB "4.- Salir                "

SIETE:  MOV R7,#01H
ETIQ7:  MOV A,R7
        CALL TABLA7
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQ7
        MOV 08H,#08H
        MOV A,#0C0H
        CALL CDISP
        RET

TABLA7: MOVC A,@A+PC
        RET
        DB "SALIR                    "

SEGURI: MOV A,#01
        CALL CDISP
        MOV R7,#01H
ETIQS:  MOV A,R7
        CALL TABLA8
        CALL DISDAT
        INC R7
        CJNE R7,#15H,ETIQS
        MOV A,#0D0H             ; POCISION DEL CURSOR
        CALL CDISP
ETISE:  MOV A,R7
        CALL TABLA8
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETISE
        RET

TABLA8: MOVC A,@A+PC
        RET
        DB "SISTEMA DE SEGURIDADTEMP:  C"

DEFENT: MOV A,#01
        CALL CDISP
        MOV R7,#01H
ETIQE:  MOV A,R7
        CALL TENT
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIQE
        MOV A,#0C0H             ; POCISION DEL CURSOR
        CALL CDISP
ETIQE1: MOV A,R7
        CALL TENT
        CALL DISDAT
        INC R7
        CJNE R7,#1DH,ETIQE1
        RET

TENT:   MOVC A,@A+PC
        RET
        DB "DEF. 0123456789ABCDEF    ENT. "
```

```
ACTSEN: MOV A,#01
        CALL CDISP
        MOV R7,#01H
ETIAS:  MOV A,R7
        CALL TSEN
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETIAS
        MOV A,#0C0H             ; POCISION DEL CURSOR
        CALL CDISP
ETIAS1: MOV A,R7
        CALL TSEN
        CALL DISDAT
        INC R7
        CJNE R7,#1DH,ETIAS1
        RET

TSEN:   MOVC A,@A+PC
        RET
        DB "ACT. 0123456789ABCDEF    SEN. "

ACTALAR:MOV A,#01
        CALL CDISP
        MOV R7,#01H
ETALAR: MOV A,R7
        CALL TALAR
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETALAR
        MOV A,#0C0H             ; POCISION DEL CURSOR
        CALL CDISP
ETALAR1:MOV A,R7
        CALL TALAR
        CALL DISDAT
        INC R7
        CJNE R7,#20H,ETALAR1
        RET

TALAR:  MOVC A,@A+PC
        RET
        DB "ACTIVAR  01234567        ALARMAS"


SIST_AC:MOV A,#80H
        CALL CDISP
        MOV R7,#01H
ETISI:  MOV A,R7
        CALL TSSI
        CALL DISDAT
        INC R7
        CJNE R7,#1AH,ETISI
        MOV A,#0C0H             ; POCISION DEL CURSOR
        CALL CDISP
ETISI1: MOV A,R7
        CALL TSSI
        CALL DISDAT
        INC R7
        CJNE R7,#21H,ETISI1
        RET
TSSI:   MOVC A,@A+PC
        RET
```

```
        DB "   SISTEMA ACTIVADO      SENSOR "
;************************************************
;SUPERVISA AL TECLADO DE PROGRAMACION DURANTE ESTA
;************************************************
SUBR1:  MOV A,#08
        MOV R0,#28H
SABOT:  MOV @R0,#00
        INC R0
        DJNZ R2,SABOT
        RET
SCAN1:  MOV R3,#0EEH
SCANC:  MOV A,R3
        JNB F0,FSCAN
        MOVX @DPTR,A
        CALL CONTROL
        JB ACC.7,SIPROG
        MOV A,R3
        RL A
        MOV R3,A
        JMP SCANC
FSCAN:  MOV A,#00H
SIPROG: RET
;************************************************
;SUPERVISION DE SENSORES Y ACTIVACION DE ALARMAS
;************************************************
CHECA:
        MOV DPTR,#2008H
        MOVX A,@DPTR
        ANL A,#0FEH
        MOV R3,A
        MOV DPTR,#8000H
        MOVX A,@DPTR
        ANL A,#0FEH
        XRL A,R3
        JZ NACT_PA
        MOV R3,A
        MOV DPTR,#200AH      ;SENSOR ACTIVO
        MOVX A,@DPTR
        ANL A,#0FEH
        ANL A,R3
        JZ NACT_PA
        CALL ALAR_PA

NACT_PA:
        MOV DPTR,#2009H
        MOVX A,@DPTR
        MOV R3,A
        MOV DPTR,#8001H
        MOVX A,@DPTR
        XRL A,R3
        JZ NACT_PB
        MOV R3,A
        MOV DPTR,#200AH
        MOVX A,@DPTR
        ANL A,R3
        JZ NACT_PB
        CALL ALAR_PB
NACT_PB:RET

ALAR_PA:MOV R3,A
        MOV A,08
```

```
        JNZ NDSP
        CALL SIST_AC
NDSP:   MOV R4,#08H
        MOV A,R3
        CLR C
ALA1:   RRC A
        JC NUMSE
ALA2:   DJNZ R4,ALA1
        JMP SBLOC
NUMSE:  MOV R3,A
        MOV SBUF,#0FFH
        CALL TX1
        NOP
        MOV SBUF,#0FEH
        CALL TX1
        NOP
        MOV SBUF,#0FDH
        CALL TX1
        NOP
        MOV SBUF,#0FCH
        CALL TX1
        NOP
        MOV SBUF,#219
        CALL TX1
        MOV A,#08H
        CLR C
        SUBB A,R4
        MOV SBUF,A
        CALL TX1
        MOV A,08
        JNZ ALA2
        MOV A,#08H
        CLR C
        SUBB A,R4
        ORL A,#30H
        MOV SBUF,A
        CALL DISDAT
        MOV A,#2CH
        CALL DISDAT
        MOV A,R3
        JMP ALA2
SBLOC:  MOV DPTR,#2009H        ;CHECA PB
        MOVX A,@DPTR
        MOV R3,A
        MOV DPTR,#8001H
        MOVX A,@DPTR
        XRL A,R3
        JZ AC_ALRM
        MOV R3,A
        MOV DPTR,#200AH
        MOVX A,@DPTR
        ANL A,R3
        JZ AC_ALRM
        MOV R3,A
        MOV R4,#08H
        MOV A,R3
        CLR C
ALA3:   RRC A
        JC NUMSE1
ALA4:   DJNZ R4,ALA3
        JMP AC_ALRM
```

```
NUMSE1: MOV R3,A
        MOV SBUF,#0FFH
        CALL TX1
        NOP
        MOV SBUF,#0FEH
        CALL TX1
        NOP
        MOV SBUF,#0FDH
        CALL TX1
        NOP
        MOV SBUF,#0FCH
        CALL TX1
        NOP
        MOV SBUF,#219
        CALL TX1
        MOV A,#08H
        CLR C
        SUBB A,R4
        ADD A,#08H
        MOV SBUF,A
        CALL TX1
        MOV A,08
        JNZ ALA4
        MOV A,#08H
        CLR C
        SUBB A,R4
        ADD A,#08H
        MOV B,#10
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        MOV A,#2CH
        CALL DISDAT
        MOV A,R3
        JMP ALA4

AC_ALRM:MOV DPTR,#200CH
        MOVX A,@DPTR
        JNB ACC.0,AL1
        SETB P1.2
AL1:    JB ACC.1,AL2
        SETB P1.3
AL2:    JB ACC.2,AL3
        SETB P1.4
AL3:    RET

ALAR_PB:MOV R3,A
        MOV A,08
        JNZ NDSP1
        CALL SIST_AC
NDSP1:  MOV R4,#08H
        MOV A,R3
        CLR C
ALA6:   RRC A
        JC NUMSE3
ALA7:   DJNZ R4,ALA6
        JMP AC_ALRM
NUMSE3: MOV R3,A
```

```
        MOV SBUF,#0FFH
        CALL TX1
        NOP
        MOV SBUF,#0FEH
        CALL TX1
        NOP
        MOV SBUF,#0FDH
        CALL TX1
        NOP
        MOV SBUF,#0FCH
        CALL TX1
        NOP
        MOV SBUF,#219
        CALL TX1
        MOV A,#08H
        CLR C
        SUBB A,R4
        ADD A,#08H
        MOV SBUF,A
        MOV A,08
        JNZ ALA7
        MOV A,#08H
        CLR C
        SUBB A,R4
        ADD A,#08H
        MOV B,#10
        DIV AB
        ORL A,#30H
        CALL DISDAT
        MOV A,B
        ORL A,#30H
        CALL DISDAT
        MOV A,#2CH
        CALL DISDAT
        MOV A,R3
        JMP ALA7

DESALAR:CLR P1.2
        CLR P1.3
        CLR P1.4
        RET
;************************************************
;GENERA EL SONIDO CUANDO ALGUNA TECLA ES PRESIONADA
;************************************************
BEEP:
        CLR P1.1
        MOV R6,#20H
        NOP
        MOV R5,#10001011B
        CALL CHECA
BEEP1:  MOV A,R5
        RLC A
        MOV R5,A
        NOP
        NOP
        MOV P1.1,C
        DJNZ R6,BEEP1
        MOV DPTR,#8002H
        MOVX A,@DPTR        ;
        ORL A,#0FH          ;
        CPL A               ;
```

```
JNZ BEEP
RET
END
```

APENDICE C

Listado de programa en lenguaje "C" ejecutado por la computadora, utilizado para el registro estadistico de los eventos detectados por la Unidad de Control.

```
#include <dos.h>
#include <string.h>
#include <conio.h>
#include <stdio.h>
#include <time.h>
#include <bios.h>

typedef unsigned char byte;
int prueb_enl=0;
int estado=0;
int cont_alarm1=0;
int cont_alarm2=0;
int cont_acces1=0;
int cont_acces2=0;
byte mint=0xff;
byte seg;
int b=1,sel=1;
int inicio_men=0;
int fin_men=0;
int i=0;
int j=0;
int cont=0;
FILE *arch1;
FILE *arch2;
FILE *base_acceso;
FILE *base_alarma;
byte eeprom[2048],alarm[15],acceso[15];
struct cobert
{
char zona[100];
};
struct datos_acceso
{
 char nombre[30];
 char depart[30];
};
struct cobert loc[16];
struct datos_acceso base_usuario[254];

struct alarmas
{
byte mallas;
char locali[50];
byte horas;
byte minutos;
byte segundos;
char dia;
char mes;
int  ano;
}est_alarma;

struct accesos
{
byte claves;
char nombres[50];
char dto[30];
```

```
byte horas;
byte minutos;
byte segundos;
char dia;
char mes;
int  ano;
}est_acceso;
void inicializa();
void Enable_IRQ1();
void Desable_IRQ1();
void interrupt recibe();
void interrupt (* funcion2) ();

#define     BR_9600_hi  0x00
#define     BR_9600_lo  0x0C
#define     RBR   0x03F8     /*  RECEIVER BUFFER REGISTER              */
#define     THR   0x03F8     /*  TRANSMITTER HOLDING REGISTER          */
#define      IER   0x03F9     /*  INTERRUPT ENABLE REGISTER             */
#define      IIR   0x03FA     /*  INTERRUPT IDENTIFICATION REGISTER     */
#define     LCR   0x03FB     /*  LINE CONTROL REGISTER                 */
#define     MCR   0x03FC     /*  MODEM CONTROL REGISTER                */
#define     LSR   0x03FD     /*  LINE STATUS REGISTER                  */
#define     MSR   0x03FE     /*  MODEM STATUS REGISTER                 */
#define     DLL   0x03F8     /*  DIVISOR LATCH (LSB)                   */
#define     DLM   0x03F9     /*  DIVISOR LATCH (MSB)                   */
#define     ERDAI    0x01    /*  ENABLE RECEIVED DATA AVAILABLE        */
#define     ETHEI    0x02    /*  ENABLE TRANSMITTER HOLDING REGISTER   */
#define     ERLSI    0x04    /*  ENABLE RECEIVER LINE STATUS           */
#define     EMSI     0x08    /*  ENABLE MODEM STATUS                   */
#define     b_datos           0x03
#define     b_paro            0x00
#define ACCESS_DL_ON    0X80
#define ACCESS_DL_OFF   0X00
#define OUT_2           0X08
#define     DR           0x01
#define     OE           0x02
#define     PE           0x04
#define     FE           0x08
#define     BREAK        0x10
#define     THRE         0x20
#define     TSRE         0x40
#define     DCTS         0x01
#define     DDSR         0x02
#define     TERI         0x04
#define     DSLSD        0x08
#define     CTS          0x10
#define     DSR          0x20
#define     RI           0x40
#define     RLSD         0x80

#define F1    315
#define F2    316
#define F3    317
#define F4    318
#define F5    319
#define ESC    27
#define F_arriba    328
#define F_abajo     336
#define F_derecha   333
#define F_izqui     331
```

```
/* ====================================*/
void comunicacion()
{
   unsigned char data_line;
   data_line = 0x03;
   outportb(LCR,data_line);
}
/* ====================================*/
void val_trans() /* Define la velocidad de comunicación a 9600 bauds*/
{
      outportb(LCR,ACCESS_DL_ON);
      outportb(DLM,BR_9600_hI);
      outportb(DLL,BR_9600_lo);
      outportb(LCR,ACCESS_DL_OFF);
}
/*  ====================================*/
void EOI_8259()
{
      outportb(0x20,0x20);
}
/* ====================================*/
void inicializa() /*Establece los parámetros de configuración al UART*/
{
      desable_IRQ4();
      val_trans();
      comunicacion();
      outportb(IER,ERDAI);
      outportb(MCR,OUT_2|0x02);
      set_vects();
      Enable_IRQ4();
}
/* ====================================*/
set_vects()
{
      funcion2 = getvect (0x0C);
      setvect(0x0C,recibe);
}
/* ====================================*/
Enable_IRQ4()
{
      char imr;
      imr = inportb(0x021);
      imr = imr&(0x0ef);
      outportb(0x021,imr);
}
/* ====================================*/
desable_IRQ4()
{
      char imr;
      imr = inportb(0x021);
      imr = imr&(0x10);
      outportb(0x021,imr);

}
/* ====================================*/
void final()
{
   desable_IRQ4();
   setvect(0x0C,funcion2);
  /*  EOI_8259();*/
}
```

```
/* ================================*/
void interrupt recibe()
{
byte rdato;
rdato = inport(RBR);
if (prueb_en1<5)
    {
     switch(prueb_en1)
     {
      case 0: /*  fb */
       if (rdato==0xfb)
         prueb_en1++;
       else
         prueb_en1=0;
      break;
      case 1: /*  fc */
       if (rdato==0xfc)
         prueb_en1++;
       else
         prueb_en1=0;
      break;
      case 2: /*  fd */
       if (rdato==0xfd)
         prueb_en1++;
       else
         prueb_en1=0;
      break;
      case 3: /* fe */
       if (rdato==0xfe)
         prueb_en1++;
       else
         prueb_en1=0;
      break;
      case 4: /* ff */
       if (rdato==0xff)
         prueb_en1++;
       else
         prueb_en1=0;
      break;
     }

    }

else
    {
     switch(estado)
      {
       case 0:/* espera FF*/
       if (rdato==0xFF)
         estado++;
       else
         estado=0;
       break;
       case 1:/* espera FE*/
       if (rdato==0xFE)
         estado++;
       else
         estado=0;
       break;
       case 2:/* espera FD*/
       if (rdato==0xFD)
```

```
        estado++;
      else
        estado=0;
      break;
      case 3:/* espera FC*/
      if (rdato==0xFC)
        estado++;
      else
        estado=0;
      break;
      case 4:/*identificador */
      switch(rdato)
         {
           case 0xda: /* llegan los datos de la eeprom*/
             estado=5;
           break;
           case 0xdb; /*  se activó una alarma*/
             estado=6;
           break;
           case 0xdc; /* se concedió el acceso*/
             estado=7;
           break;
           }
      break;
      case 5:
      eeprom[cont]=rdato;
      cont++;
      if (cont==2047)
        {
          estado=0;
          cont=0;
        }
      break;
      case 6:
      alarm[cont_alarm1]=rdato;
      cont_alarm1++;
      estado=0;
      if(cont_alarm1==14)
        {
           cont_alarm1=0;
        }
      break;
      case 7:
      acceso[cont_acces1]=rdato;
      cont_acces1++;
      estado=0;
      if(cont_acces1==14)
        {
         cont_acces1=0;
        }
      break;
    }
   }
EOI_8259();
}

/* ================================*/
trans(dato)
      byte dato;
{
    while (!(THRE&inportb(LSR)));
```

```
    outportb(THR,dato);
}
/* ================================*/
int tecla(void)
{
int key,lo,hi;
key= bioskey(0);
lo= key & 0x00FF;
hi= (key & 0xFF00)>>8;
return((lo==0)? hi+256:lo);
}
/*================================*/
reloj()
  {
  struct time hora;
  struct date fecha;
  gettime(&hora);
  getdate(&fecha);
  est_alarma.horas=hora.ti_hour;
  est_alarma.minutos=hora.ti_min;
  est_alarma.segundos=hora.ti_sec;
  est_alarma.dia=fecha.da_day;
  est_alarma.mes=fecha.da_mon;
  est_alarma.ano=fecha.da_year;
  }
reloj1()
  {
  struct time hora;
  struct date fecha;
  gettime(&hora);
  getdate(&fecha);
  est_acceso.horas=hora.ti_hour;
  est_acceso.minutos=hora.ti_min;
  est_acceso.segundos=hora.ti_sec;
  est_acceso.dia=fecha.da_day;
  est_acceso.mes=fecha.da_mon;
  est_acceso.ano=fecha.da_year;
  }
/*================================*
tiempo()
{
  int x1,y1;
  byte h,m;
  struct time hora;
  gettime(&hora);
  if (mint!=0xff)
    {
      if (mint!=hora.ti_min)
      {
        x1=wherex();
        y1=wherey();
        gotoxy(70,1);
        textbackground(BLACK);
        cprintf("\n %2d:%2d:%2d",hora.ti_hour, hora.ti_min, hora.ti_sec);
        gotoxy(x1,y1);
        mint=hora.ti_min;
        trans(0xdb);
        trans(hora.ti_hour);
        trans(hora.ti_min);
      }
      else
```

```
      {
        if (seg!=hora.ti_sec)
          {
            x1=wherex();
            y1=wherey();
            gotoxy(70,1);
            textbackground(BLACK);
            cprintf("\n %2d:%2d:%2d",hora.ti_hour, hora.ti_min, hora.ti_sec);
            gotoxy(x1,y1);
            seg=hora.ti_sec;
          }
      }
    }
else
   {
     x1=wherex();
     y1=wherey();
     gotoxy(70,1);
     textbackground(BLACK);
     cprintf("\n %2d:%2d:%2d",hora.ti_hour, hora.ti_min, hora.ti_sec);
     gotoxy(x1,y1);
     h=hora.ti_hour;
     m=hora.ti_min;
     trans(0x0db);
     trans(h);
     trans(m);
     mint=hora.ti_min;
     seg=hora.ti_sec;
   }
}
/*=============================================================*/
enlace()
{
   byte dat;
   dat=0xfa;
   dat++;
   while (dat!=0)
       {
       printf(".%c",dat);
       trans(dat);
       dat++;
       }
   delay(1000);
   if (prueb_enl!=5)
       {
         dat=0xfa;
         dat++;
         while (dat!=0)
           {
             printf(".%c",dat);
             trans(dat);
             dat++;
           }
       }
    delay(1000);
    if (prueb_enl!=5)
      {
      sound(450);
      delay(2000);
      printf("\n VERIFICAR LAS CONEXIONES DEL PUERTO SERIE\n");
      printf(" EL ENLACE NO FUE POSIBLE");
```

```
      nosound();
      final();
      getch();
      exit();
      }
   else
   pinta();
}
/*=========================================================*/
pantalla1()
{
int x1,y1;
gotoxy(0,3);
textbackground(5);
cprintf("          TIPO  CONDICION  ESTADO");
printf("\nMALLA  1...\n");
printf("MALLA  2...\n");
printf("MALLA  3...\n");
printf("MALLA  4...\n");
printf("MALLA  5...\n");
printf("MALLA  6...\n");
printf("MALLA  7...\n");
printf("MALLA  8...\n");
printf("MALLA  9...\n");
printf("MALLA 10...\n");
printf("MALLA 11...\n");
printf("MALLA 12...\n");
printf("MALLA 13...\n");
printf("MALLA 14...\n");
printf("MALLA 15...\n");
x1=wherex();
y1=wherey();
gotoxy(50,18);
printf("ALARMAS:");
gotoxy(50,19);
printf("1 :");
gotoxy(50,20);
printf("2 :");
gotoxy(50,21);
printf("3 :");
gotoxy(50,12);
printf("ACCESO :");
gotoxy(x1,y1);
printf("\n\n\n\n CONFIGURACION");
textbackground(BLUE);
cprintf(" F1 ");
printf("   REPORTES  ");
cprintf(" F2 ");
printf("   PRUEBA DEL SISTEMA ");
cprintf(" F3 ");
printf("  SALIR  ");
cprintf(" F4 ");

}
/*=========================================================*/
pantalla2()
{
  int x1,y1;
  textbackground(BLUE);
  printf("PROGRAMACION DEL SISTEMA\n\n\n\n");
  printf("CLAVES DE ACCESO ....");
```

```
  cprintf(" F1");
  printf("\n\nHALLAS ..............");
  cprintf(" F2");
  printf("\n\nALARMAS .............");
  cprintf(" F3");
  printf("\n\nSALIR ...............");
  cprintf(" F4");
}
/*=================================================*/
atributos()
{
byte atr1,atr2,atr3;
int x1=12,y1=6;
textbackground(6);
atr1=2;
for (atr2=0;atr2<7;atr2++)
     {
       if ((eeprom[8]&atr1)==0)
       {
       gotoxy(x1,y1);
       cprintf("N.C");
       y1++;
       atr1=atr1<<1;
       }
       else
       {
         gotoxy(x1,y1);
       cprintf("N.A");
       y1++;
       atr1=atr1<<1;
       }
     }

atr1=1;
for (atr2=0;atr2<8;atr2++)
     {
       if ((eeprom[9]&atr1)==0)
       {
       gotoxy(x1,y1);
       cprintf("N.C");
       y1++;
       atr1=atr1<<1;
       }
       else
       {
       gotoxy(x1,y1);
       cprintf("N.A");
       y1++;
       atr1=atr1<<1;
       }
     }
x1=20;
y1=6;
eeprom[0x0a]=0x22;
textbackground(3);
atr1=2;
for (atr2=0;atr2<7;atr2++)
     {
       if ((eeprom[10]&atr1)==0)
       {
       gotoxy(x1,y1);
```

```
        cprintf("DES");
        yl++;
        atr1=atr1<<1;
        }
        else
        {
          gotoxy(x1,y1);
        cprintf("HAB");
        yl++;
        atr1=atr1<<1;
        }
      }
      atr1=1;
for (atr2=0;atr2<8;atr2++)
      {
        if ((eeprom[11]&atr1)==0)
         {
           gotoxy(x1,y1);
           cprintf("DES");
           yl++;
           atr1=atr1<<1;
         }
        else
         {
           gotoxy(x1,y1);
           cprintf("HAB");
           yl++;
           atr1=atr1<<1;
         }
      }
        x1=54;
        y1=19;
        atr1=1;
for (atr2=0;atr2<3;atr2++)
      {
        if ((eeprom[0x0c]&atr1)==0)
        {
          gotoxy(x1,y1);
          cprintf("DES");
          yl++;
          atr1=atr1<<1;
        }
        else
        {
          gotoxy(x1,y1);
          cprintf("HAB");
          yl++;
          atr1=atr1<<1;
        }
        }
}
/*************************************************************/
act_alarm(no_alarm)
byte no_alarm;
{
   int x1,y1;
   textbackground(RED);
   textcolor(WHITE+BLINK);
   x1=wherex();
   y1=wherey();
   gotoxy(30,5+no_alarm);
```

```
   cprintf("alarma");
   textcolor(WHITE);
   gotoxy(45,6);
   printf("UBICACION DE LA ALARMA");
   ventana(40,7,80,8,5);
   cprintf("%-20s",loc[no_alarm].zona);
   textmode(-1);
   gotoxy(x1,y1);
}
/*****************************************************/
identificacion(no_clave)
byte no_clave;
{
int x,y;
x=wherex();
y=wherey();
ventana(40,13,80,14,6);
cprintf("NOMBRE: %-20s\n",base_usuario[no_clave].nombre);
gotoxy(1,wherey());
cprintf("DEPART: %-20s",base_usuario[no_clave].depart);
textmode(-1);
gotoxy(x,y);
}
/*=====================================================*/
pinta()
{
clrscr();
textbackground(1);
textcolor(15);
cprintf("                S I S T E M A   D E  S E G U R I D A D
          ");
cprintf("
          ");
cprintf("
 \n");
textbackground(BLACK);
textcolor(WHITE);
}
pintal(char stitulo[20])
{
int x,y;
pinta();
textbackground(BLUE);
textcolor(2);
x=wherex();
y=wherey();
gotoxy(30,2);
cprintf(stitulo);
gotoxy(x,y);
textcolor(WHITE);
}
/*=====================================================*/
password()
{
byte pass[8],pas,ceep,cpass=0;
cprintf("CLAVE DE ACCESO (PASSWORD):");
while((pas=getch())!=13)
   {
    pass[cpass]=pas;
    cpass++;
   }
```

```
pas=0;
for (ceep=0;ceep<cpass;ceep++)
   {
    if (eeprom[ceep]==pass[pas])
     pas++;
    else
     {
     if (eeprom[ceep]=='<')
       ceep=1;
      else
        ceep=0;
      break;
      }
   }
return(ceep);
}
/*=========================================================*/
listaclave(n_clave,color,nombre,departamento)
char nombre[30],departamento[30];
int color;
byte n_clave;
{
byte cont_byte;
int x,y;
x=wherex();
y=wherey();
ventana(45,10,80,11,BLUE);
cprintf("%s",nombre);
gotoxy(1,2);
cprintf("%s",departamento);
textmode(-1);
textbackground(BLACK);
gotoxy(x,y);
cprintf("\n");
gotoxy(1,wherey());
cprintf("%3d    ",n_clave);
n_clave=n_clave * 8 + 0x10;
textbackground(color);
textcolor(WHITE);
for (cont_byte=0;cont_byte<8;cont_byte++)
  {
   cprintf("%c",eeprom[n_clave]);
   n_clave++;
  }
}
/*=========================================================*/

numero_de_claves()
{
  byte No_clave=0;
  pintal("CONFIGURACION");
  printf("PROGRAMACION DE CLAVES DE ACCESO\n");
  textbackground(3);
  printf("\n No. de claves programadas=  ");
  cprintf("%d",eeprom[0x0f]);
  if (eeprom[0x0f]>16)
   {
     for (No_clave=0;No_clave<18;No_clave++)
     listaclave(No_clave,3,NULL,NULL);
   }
  else
```

```
   {
     for (No_clave=0;No_clave<eeprom[0x0f];No_clave++)
     listaclave(No_clave,3,NULL,NULL);
   }
gotoxy(30,10);
cprintf("USUARIO      :");
gotoxy(30,11);
cprintf("DEPARTAMENTO:");
textbackground(BLUE);
gotoxy(25,25);
printf("No de claves ");
cprintf(" F1 ");
printf("   USUARIO ");
cprintf(" F2 ");
printf("  SALIR ");
cprintf(" F3 ");
}
/*************************************************************/
modifica_claves()
{
byte opcion_clave=1,No_clave=0,c_lineas=0,c_actual;
int opcion,x,y,x2,y2;
char *fin;
numero_de_claves();
y2=7;
No_clave=0;
opcion_clave=1;
while(opcion_clave)
 {
 if (!kbhit())
  {
   tiempo();
   if (No_clave==0)
    {
    gotoxy(1,y2);

listaclave(No_clave,6,base_usuario[No_clave].nombre,base_usuario[No_clave].dep
art);
     No_clave++;
     c_lineas++;
    }
  }
  else
  {
    switch(opcion=tecla())
     {
      case F_arriba:
      if (No_clave>1)
       {
         if((c_lineas==1)&(No_clave>1))
           {
            scroll_aba();
            c_lineas++;
            gotoxy(1,7);

listaclave(No_clave-1,3,base_usuario[No_clave-1].nombre,base_usuario[No_clave-
1].depart);
            gotoxy(1,7);

listaclave(No_clave-2,6,base_usuario[No_clave-2].nombre,base_usuario[No_clave-
2].depart);
```

```
            No_clave--;
            c_lineas--;
            }
           else
            {
            gotoxy(1,wherey()-1);

listaclave(No_clave-1,3,base_usuario[No_clave-1].nombre,base_usuario[No_clave-
1].depart);
            gotoxy(1,wherey()-2);

listaclave(No_clave-2,6,base_usuario[No_clave-2].nombre,base_usuario[No_clave-
2].depart);
            No_clave--;
            c_lineas--;
            }
         }
        break;
        case F_abajo:
         if ((No_clave)!=0&(No_clave<eeprom[0x0f]))
           {
           if (c_lineas==18)
             {
             scroll_arr();
             c_lineas--;
             gotoxy(1,23);
             textbackground(3);

listaclave(No_clave-1,3,base_usuario[No_clave-1].nombre,base_usuario[No_clave-
1].depart);
             gotoxy(1,24);
             textbackground(6);
             gotoxy(1,24);

listaclave(No_clave,6,base_usuario[No_clave].nombre,base_usuario[No_clave].dep
art);
             No_clave++;
             c_lineas++;
             }
             else
             {
             gotoxy(1,wherey()-1);
             textbackground(3);

listaclave(No_clave-1,3,base_usuario[No_clave-1].nombre,base_usuario[No_clave-
1].depart);
             gotoxy(1,y2+No_clave);
             textbackground(6);

listaclave(No_clave,6,base_usuario[No_clave].nombre,base_usuario[No_clave].dep
art);
             No_clave++;
             c_lineas++;
             }
           }
         break;
         case 13:
           x=wherex();
           y=wherey();
           gotoxy(8,wherey());
           modifica(No_clave);
```

```
          gotoxy(x,y);
        break;
        case F1:
          numero_claves();
          No_clave=0;
        break;
        case F2:
          x=wherex();
          y=wherey();
          ventana(45,10,80,11,BLUE);
          fin=gets(base_usuario[No_clave-1].nombre);
          gotoxy(1,2);
          fin=gets(base_usuario[No_clave-1].depart);
          textmode(-1);
          gotoxy(x,y);
        break;
        case F3:
          opcion_clave=0;
          pinta();
          pantalla2();
        break;
       }
      }
     }

}
/*************************************************************/
ventana(x_der,y_der,x_izq,y_izq,color)
int x_der,y_der,x_izq,y_izq,color;
{
 byte cvent;
 gotoxy(x_der-1,y_der-1);
 textbackground(color);
 textcolor(WHITE);
 window(x_der,y_der,x_izq,y_izq);
 clrscr();
}
/*************************************************************/
modifica(no_clave)
byte no_clave;
{
byte nclav[8]="<<<<<<<<",cl,csep,ccl=0;
while(((cl=getche())!=13)&&(ccl<7))
    {
        nclav[ccl]=cl;
        ccl++;
    }
no_clave--;
no_clave=no_clave*8+0x10;
cl=0;
for (csep=0;csep<8;csep++)
    {
        eeprom[no_clave+csep]=nclav[cl];
        cl++;
    }
}
/*************************************************************/
numero_claves()
{
int No_claves;
byte correcto=1;
```

```
while (correcto)
 {
  gotoxy(30,7);
  scanf("%d",&No_claves);
  if((No_claves<254)&(No_claves>0))
      {
      correcto=0;
      eeprom[0x0f]=No_claves;
      }
 }
 numero_de_claves();
}
/*=========================================================*/
scroll_arr()
{
union REGS in,out;
in.h.ah=6;
in.h.al=1;
in.h.ch=7;
in.h.cl=1;
in.h.dh=24;
in.h.dl=15;
in.h.bh='';
int86(0x10,&in,&out);
}
/*=========================================================*/
scroll_aba()
{
union REGS in,out;
in.h.ah=7;
in.h.al=1;
in.h.ch=7;
in.h.cl=1;
in.h.dh=24;
in.h.dl=15;
in.h.bh='';
int86(0x10,&in,&out);
}
/*=========================================================*/
complemento(n_sensor,col)
byte n_sensor,col;
{
byte puerto,bit,cr1;
 if (col==0)
 {
  if (n_sensor<8)
   puerto=0x08;
  else
   puerto=9;
 }
else
 {
  if (n_sensor<8)
   puerto=10;
  else
   puerto=11;
 }
if (n_sensor<8)
       {
        bit=1;
        for (cr1=0;cr1<n_sensor;cr1++)
```

```
        bit=bit<<1;
       }
      else
       {
        bit=1;
        for (crl=0;crl<(n_sensor-8);crl++)
         bit=bit<<1;
        }
if ((eeprom[puerto]&bit)==0)
  eeprom[puerto]=eeprom[puerto]|bit;
  else
  {
  bit=~bit;
  eeprom[puerto]=eeprom[puerto]&bit;
  }
}
/*=====================================================*/
listasensor(n_sensor,color,col,area)
char area[80];
int color;
byte n_sensor;
byte col;
{
byte crl,bit,puerto;
int x=12,y=5;
ventana(35,10,80,12,BLUE);
cprintf("%s",area);
textmode(-1);
y=y+n_sensor;
if (col==0)
 {
  x=12;
  if (n_sensor<8)
   puerto=0x08;
  else
   puerto=9;
 }
else
 {
  x=20;
  if (n_sensor<8)
   puerto=10;
  else
   puerto=11;
 }
      textbackground(color);
      textcolor(WHITE);
      if (n_sensor<8)
       {
        bit=1;
        for (crl=0;crl<n_sensor;crl++)
         bit=bit<<1;
       }
      else
       {
        bit=1;
        for (crl=0;crl<(n_sensor-8);crl++)
         bit=bit<<1;
        }
```

```
if (col==0)
   {
   if ((eeprom[puerto]&bit)==0)
            {
            gotoxy(x,y);
            cprintf("N.C");
            }
   else
            {
            gotoxy(x,y);
            cprintf("N.A");
            }
   }
else
   {
   if ((eeprom[puerto]&bit)==0)
            {
            gotoxy(x,y);
            cprintf("DES");
            }
   else
            {
            gotoxy(x,y);
            cprintf("HAB");
            }
   }
}
/*=============================================================*/
condicion()
{
byte atr1,atr2,atr3;
int x1=12,y1=6;
textbackground(3);
atr1=2;
for (atr2=0;atr2<7;atr2++)
     {
       if ((eeprom[8]&atr1)==0)
       {
       gotoxy(x1,y1);
       cprintf("N.C");
       y1++;
       atr1=atr1<<1;
       }
       else
       {
         gotoxy(x1,y1);
       cprintf("N.A");
       y1++;
       atr1=atr1<<1;
       }
     }

atr1=1;
for (atr2=0;atr2<8;atr2++)
     {
       if ((eeprom[9]&atr1)==0)
       {
       gotoxy(x1,y1);
       cprintf("N.C");
       y1++;
       atr1=atr1<<1;
```

```
        }
        else
        {
        gotoxy(x1,y1);
        cprintf("N.A");
        y1++;
        atr1=atr1<<1;
        }
      }
x1=20;
y1=6;
eeprom[0x0a]=0x22;
textbackground(3);
atr1=2;
for (atr2=0;atr2<7;atr2++)
      {
        if ((eeprom[10]&atr1)==0)
        {
        gotoxy(x1,y1);
        cprintf("DES");
        y1++;
        atr1=atr1<<1;
        }
        else
        {
          gotoxy(x1,y1);
        cprintf("HAB");
        y1++;
        atr1=atr1<<1;
        }
      }
     atr1=1;
for (atr2=0;atr2<8;atr2++)
      {
        if ((eeprom[11]&atr1)==0)
                {
                     gotoxy(x1,y1);
                     cprintf("DES");
                     y1++;
                     atr1=atr1<<1;
                }
        else
                {
                     gotoxy(x1,y1);
                     cprintf("HAB");
                     y1++;
                     atr1=atr1<<1;
                 }
        }
}
/*=============================================================*/
prog_sensores()
{
byte No_sensor=1,lado=0,opcion_sensor=1;
int x=12,y=6,seleccion;
char *fin;
pintal("CONFIGURACION");
pantalla3();
condicion();
while(opcion_sensor)
 {
```

```
if (!kbhit())
 {
 tiempo();
 if (No_sensor==1)
  {
  listasensor(No_sensor,6,lado,loc[No_sensor].zona);
  No_sensor++;
  }
 }
else
 {
 switch(seleccion=tecla())
  {
  case F_arriba:
  if (No_sensor>2)
   {
   listasensor(No_sensor-1,3,lado,loc[No_sensor-1].zona);
   listasensor(No_sensor-2,6,lado,loc[No_sensor-2].zona);
   No_sensor--;
   }
  break;
  case F_abajo:
  if ((No_sensor)!=16&(No_sensor<16))
   {
   listasensor(No_sensor-1,3,lado,loc[No_sensor-1].zona);
   listasensor(No_sensor,6,lado,loc[No_sensor].zona);
   No_sensor++;
   }
   break;
  case F_derecha:
   listasensor(No_sensor-1,3,lado,loc[No_sensor-1].zona);
   lado=1;
   listasensor(No_sensor-1,6,lado,loc[No_sensor-1].zona);
  break;
  case F_izqui:
   listasensor(No_sensor-1,3,lado,loc[No_sensor-1].zona);
   lado=0;
   listasensor(No_sensor-1,6,lado,loc[No_sensor-1].zona);
  break;
  case 13:
   complemento(No_sensor-1,lado);
   listasensor(No_sensor-1,6,lado,loc[No_sensor-1].zona);
  break;
  case F1:
   x=wherex();
   y=wherey();
   ventana(35,10,80,12,BLUE);
   fin=gets(loc[No_sensor-1].zona);
   textmode(-1);
   gotoxy(x,y);
  break;
  case F2:
   printf("salir");
   opcion_sensor=0;
   pintal("CONFIGURACION");
   pantalla2();
  break;
  }
 }
}
```

```
}
/*=========================================================*/
pantalla3()
{
textbackground(5);
cprintf("          TIPO  CONDICION  ");
printf("\nMALLA  1...\n");
printf("MALLA  2...\n");
printf("MALLA  3...\n");
printf("MALLA  4...\n");
printf("MALLA  5...\n");
printf("MALLA  6...\n");
printf("MALLA  7...\n");
printf("MALLA  8...\n");
printf("MALLA  9...\n");
printf("MALLA 10...\n");
printf("MALLA 11...\n");
printf("MALLA 12...\n");
printf("MALLA 13...\n");
printf("MALLA 14...\n");
printf("MALLA 15...\n");
textbackground(BLUE);
gotoxy(1,25);
printf("AREA DE COBERTURA");
cprintf(" F1 ");
printf("   SALIR ");
cprintf(" F2 ");
textbackground(BLACK);
gotoxy(55,9);
cprintf("ZONA DE COBERTURA");
}
/*=========================================================*/
pantalla4()
{
textbackground(5);
gotoxy(1,5);
cprintf("              CONDICION  ");
printf("\nALARMA   I...\n");
printf("ALARMA  II...\n");
printf("ALARMA III...\n");
textbackground(BLUE);
gotoxy(1,25);
printf("   SALIR ");
cprintf(" F1 ");
}
/*=========================================================*/
condicion2()
{
byte atr1,atr2;
int x1=18,y1=6;
textbackground(3);
atr1=1;
for (atr2=0;atr2<3;atr2++)
    {
        if ((eeprom[12]&atr1)==0)
        {
            gotoxy(x1,y1);
            cprintf("DESCONECTADO");
            y1++;
            atr1=atr1<<1;
          }
```

```
      else
            {
          gotoxy(x1,y1);
              cprintf("CONECTADO    ");
              y1++;
              atr1=atr1<<1;
             }
       }
}
/*================================================*/
listalarma(n_alarma,color)
int color;
byte n_alarma;
{
byte cr1,bit;
int x=18,y=6;
y=y+n_alarma;
textbackground(color);
bit=1;
for (cr1=0;cr1<n_alarma;cr1++)
  bit=bit<<1;
   if ((eeprom[12]&bit)==0)
             {
             gotoxy(x,y);
             cprintf("DESCONECTADO");
             }
    else
             {
             gotoxy(x,y);
             cprintf("CONECTADO    ");
             }
}
/*================================================*/
prog_alarmas()
{
byte No_alarma=0,opcion_alarma=1;
int x,y,seleccion;
pintal("CONFIGURACION");
pantalla4();
condicion2();

while(opcion_alarma)
  {
   if (!kbhit())
       {
        tiempo();
        if (No_alarma==0)
         {
             listalarma(No_alarma,6);
             No_alarma++;
         }
       }
   else
       {
        switch(seleccion=tecla())
         {
             case F_arriba:
              if (No_alarma>1)
                    {
                      listalarma(No_alarma-1,3);
                      listalarma(No_alarma-2,6);
```

```
                No_alarma--;
                }
            break;
            case F_abajo:
             if ((No_alarma)!=0&(No_alarma<3))
                {
                 listalarma(No_alarma-1,3);
                 listalarma(No_alarma,6);
                 No_alarma++;
                }
             break;
             case 13:
              complementol(No_alarma-1);
              listalarma(No_alarma-1,6);
             break;
             case F1:
              opcion_alarma=0;
              pinta();
              pantalla2();
             break;
           }
       }
   }
}
/*********************************************************/
complementol(n_alarma)
byte n_alarma;
{
byte bit,cr1;
        bit=1;
        for (cr1=0;cr1<n_alarma;cr1++)
         bit=bit<<1;
if ((eeprom[12]&bit)==0)
  eeprom[12]=eeprom[12]|bit;
  else
  {
  bit=~bit;
  eeprom[12]=eeprom[12]&bit;
  }
}

/*********************************************************/
pantalla5()
{
pintal("REPORTES");
textbackground(BLUE);
textcolor(WHITE);
printf("R E P O R T E S\n\n");
printf("claves de acceso     ");
cprintf(" F1 ");
printf("\nalarmas              ");
cprintf(" F2 ");
printf("\nsalir                ");
cprintf(" F3 ");
}
/*********************************************************/
reportes()
{
int rep_op=1,seleccion;

pantalla5();
```

```
while(rep_op)
 {
  if (!kbhit())
       {
        tiempo();
       }
   else
       {
        switch(seleccion=tecla())
         {
               case F1:
               rep_acceso();
               pantalla5();
               break;
               case F2:
               rep_alarma();
               pantalla5();
               break;
               case F3:
               rep_op=0;
               break;
         }
       }
  }

pinta();
pantalla1();
atributos();
}
/*********************************************************/
rep_acceso()
{
struct accesos reporte_accesos[25];
int x,y,report_op=1,rep_selec,i1,i2,i3;
long int farch;
pinta1("REPORTES");
printf("No. clave       nombre                  departamento            hora
fecha\n");
x=wherex();
y=wherey();
textbackground(BLUE);
textcolor(WHITE);
gotoxy(1,25);
printf("IMPRIMIR:  PANTALLA");
cprintf(" F1 ");
printf("    COMPLETO ");
cprintf(" F2 ");
printf("    SALIR  ");
cprintf("  F3 ");
printf("    ");
cprintf("%c",18);
gotoxy(x,y);
textcolor(WHITE);
archi=fopen("a:acceso.seg","r");
for (i1=0;i1<20;i1++)
{
if (!feof(archi))
  fread(&reporte_accesos[i1],sizeof(struct accesos),1,archi);
else
  break;
}
```

```
  for (i2=0;i2<i1-1;i2++)
  {

printf("%3d|%3d|%-25s|%-20s",i2,reporte_accesos[i2].claves,reporte_accesos[i2]
.nombres,reporte_accesos[i2].dto);

printf("|%2d:%2d:%2d",reporte_accesos[i2].horas,reporte_accesos[i2].minutos,re
porte_accesos[i2].segundos);

printf("|%2d/%2d/%4d\n",reporte_accesos[i2].dia,reporte_accesos[i2].mes,report
e_accesos[i2].ano);
  }
while(report_op)
 {
  if (!kbhit())
      tiempo();
  else
      {
       switch(rep_selec=tecla())
        {
              case F1:

              break;
              case F_abajo:

              break;
              case F2:

              break;
              case F3:
              report_op=0;
              break;
        }
      }
 }
fclose(arch1);
}
/*******************************************/
rep_alarma()
{
struct alarmas reporte_alarmas[20];
int x,y,report_op=1,rep_selec,i1,i2,i3;
long int farch;
arch2=fopen("a:alarm.seg","r");
printal("REPORTES");
printf("No. malla        zona                              hora          fecha\n");
x=wherex();
y=wherey();
textbackground(BLUE);
textcolor(WHITE);
gotoxy(1,25);
printf("IMPRIMIR:  PANTALLA");
cprintf(" F1 ");
printf("    COMPLETO ");
cprintf(" F2 ");
printf("    SALIR  ");
cprintf("  F3 ");
printf("    ");
cprintf("%c",18);
gotoxy(x,y);
textbackground(WHITE);
```

```
for (i1=0;i1<20;i1++)
{
if (!feof(arch2))
  fread(&reporte_alarmas[i1],sizeof(struct alarmas),1,arch2);
else
  break;
}
  for (i2=0;i2<i1-1;i2++)
  {

printf("%3d|%3d|%-40s",i2,reporte_alarmas[i2].mallas,reporte_alarmas[i2].locali);

printf("|%2d:%2d:%2d",reporte_alarmas[i2].horas,reporte_alarmas[i2].minutos,re
porte_alarmas[i2].segundos);

printf("|%2d/%2d/%4d\n",reporte_alarmas[i2].dia,reporte_alarmas[i2].mes,report
e_alarmas[i2].ano);
  }
while(report_op)
 {
  if (!kbhit())
      {
       tiempo();
      }
  else
      {
       switch(rep_selec=tecla())
        {
              case F1:
              break;
              case F_abajo:

              break;
              case F_arriba:
              break;
              case F2:
              break;
              case F3:
              report_op=0;
              break;
        }
      }
 }
/*fseek(arch1,farch,SEEK_SET);*/
fclose(arch2);
}
/*****************************************************/

/*****************************************************/
pantalla6()
{
int x1,y1;
gotoxy(0,3);
textbackground(5);
cprintf("          TIPO  CONDICION  ESTADO");
printf("\nMALLA  1...\n");
printf("MALLA  2...\n");
printf("MALLA  3...\n");
printf("MALLA  4...\n");
printf("MALLA  5...\n");
printf("MALLA  6...\n");
```

```
printf("MALLA  7...\n");
printf("MALLA  8...\n");
printf("MALLA  9...\n");
printf("MALLA 10...\n");
printf("MALLA 11...\n");
printf("MALLA 12...\n");
printf("MALLA 13...\n");
printf("MALLA 14...\n");
printf("MALLA 15...\n");
x1=wherex();
y1=wherey();
gotoxy(50,18);
printf("ALARMAS:");
gotoxy(50,19);
printf("1 :");
gotoxy(50,20);
printf("2 :");
gotoxy(50,21);
printf("3 :");
gotoxy(50,12);
printf("ACCESO :");

gotoxy(x1,y1);
printf("\n\n\n\n LIMPIAR");
textbackground(BLUE);
cprintf(" F1 ");
printf("   SALIR  ");
cprintf(" F2 ");
}
/*************************************************************/
prueba()
{
int prueb_op=1,seleccion,y,x;
pintal("PRUEBA DEL SISTEMA");
pantalla6();
atributos();
arch2=fopen("a:alarm.seg","a+");
est_alarma.mallas=00;
strcpy(est_alarma.locali,"PRUEBA DEL SISTEMA");
reloj();
fwrite(&est_alarma,sizeof(struct alarmas),1,arch2);
fclose(arch2);
while(prueb_op)
 {
  if (!kbhit())
   {
     tiempo();
     if (cont_alarm1!=cont_alarm2)
       {
       act_alarm(alarm[cont_alarm2]);
       cont_alarm2=14;
       if (cont_alarm2==14)
          cont_alarm2=0;
       else
          cont_alarm2++;
       }
     if (cont_acces1!=cont_acces2)
       {
       identificacion(acceso[cont_acces2]);
       cont_acces2=14;
       if (cont_acces2==14)
```

```
        cont_acces2=0;
      else
        cont_acces2++;
     }
    }
  else
      {
       switch(seleccion=tecla())
        {

              case F1:
              pintal("PRUEBA DEL SISTEMA");
              pantalla6();
              atributos();
              break;
              case F2:
              prueb_op=0;
              break;
        }
      }
  }

pinta();
pantallal();
atributos();
}
/*************************************************************/
main ()
{
unsigned int w;
int c,meop,x,y,flecha,x2,y2,x3,y3,i1,i2;
byte svpas,cclaves,nclave,sell=1;
pinta();
inicializa();
for (w=0;w<30000;w++)
inport(RBR);
enlace();
base_alarma=fopen("a:base2.lib","r");
for (x=0;x<16;x++)
fread(&loc[x],sizeof(struct cobart),1,base_alarma);
fclose(base_alarma);
base_acceso=fopen("a:base1.lib","r");
for (x=0;x<254;x++)
fread(&base_usuario[x],sizeof(struct datos_acceso),1,base_acceso);
fclose(base_acceso);

pantallal();
atributos();
trans(0xda);       /* peticion de parametros*/
delay(5000);
/*cont_alarm2=1;
cont_acces2=1;
alarm[1]=0x02;
acceso[1]=0x02;*/
while(b)
 {
  if (!kbhit())
   {
      tiempo();
      if (cont_alarm1!=cont_alarm2)
        {
```

```
        arch2=fopen("a:alarm.seg","a+");
        est_alarma.mallas=alarm[cont_alarm2];
        strcpy(est_alarma.locali,loc[alarm[cont_alarm2]].zona);
        reloj();
        fwrite(&est_alarma,sizeof(struct alarmas),1,arch2);
        fclose(arch2);
        act_alarm(alarm[cont_alarm2]);
        cont_alarm2++;
        if (cont_alarm2==14)
           cont_alarm2=0;
      }
    if (cont_acces1!=cont_acces2)
      {
        arch1=fopen("a:acceso.seg","a+");
        est_acceso.claves=acceso[cont_acces2];
        strcpy(est_acceso.nombres,base_usuario[acceso[cont_acces2]].nombre);
        strcpy(est_acceso.dto,base_usuario[acceso[cont_acces2]].depart);
        reloj1();
        fwrite(&est_acceso,sizeof(struct accesos),1,arch1);
        fclose(arch1);
        identificacion(acceso[cont_acces2]);
        /*cont_acces2=14;*/
        cont_acces2++;
        if (cont_acces2==14)
          cont_acces2=0;
      }

    }
else
    {
     switch(c=tecla())
      {
       case F1: /* programación del sistema*/
          pinta();
          evpas=password();
          if (evpas==0)
            {
             pinta();
             pantalla1();
             atributos();
            }
          else
            {
             pinta();
             pintal("CONFIGURACION");
             pantalla2();
             sel=1;
             while(sel)
                {
                 if (!kbhit())
                   {
                      tiempo();
                   }
                 else
                      {
                       switch(c=tecla())
                            {
                             case F1: /* claves de acceso*/
                                modifica_claves();
                             break;
                             case F2: /*mallas*/
```

```
                                        prog_sensores();
                                        break;
                                case F3: /*alarmas*/
                                        prog_alarmas();
                                        break;
                                case F4: /*salir*/
                                        pinta();
                                        pantalla1();
                                        atributos();
                                        sel=0;
                                        break;
                                }
                        }
                }
        base_acceso=fopen("a:base1.lib","w");
        for (x=0;x<254;x++)
        fwrite(&base_usuario[x],sizeof(struct  datos_acceso),1,base_acceso);
        fclose(base_acceso);
        base_alarma=fopen("a:base2.lib","w");
        for (x=0;x<16;x++)
        fwrite(&loc[x],sizeof(struct cobert),1,base_alarma);
        fclose(base_alarma);
         arch2=fopen("a:alarm.seg","a+");
        est_alarma.mallas=00;
        strcpy(est_alarma.local1,"CONFIGURACION DEL SISTEMA");
        reloj();
        fwrite(&est_alarma,sizeof(struct alarmas),1,arch2);
        fclose(arch2);
        trans(0xdc);
        for (cont=0;cont<2048;cont++)
        {
        trans(eeprom[cont]);
        delay(5);
        }
        }
      break;
      case F2: /* reportes*/
           reportes();
      break;
      case F3:
           prueba();
      break;
      case F4: /* salir*/
           b=0;
           break;
      }
     }
  }
  clrscr();
  trans(0xe0);
  final();
}
```

BIBLIOGRAFIA

BARNART, ROBERT L.
"Intrusion Detection system".
Butterworths, 1988.

CEBALLOS, FCO. JAVIER
Curso de programacion con "C"
Macrobit, 1990.

GOODWIN, MARK
"Serial Communications Programming in C and C++".
Mis:pres, 1992.

GRAHAM, L. [Y] T.FIELD
Guía del IBM PC.
Mc Graw-Hill, México,1990.

KARNIGAN, BRIAN W. [Y] DENNIS M.
"The C programing language".
Prentice Hall, New Jersey, 1988.

INTEL CORPORATION
"Embbeded Controller Handbook".
Intel, 1986.

KRUGLISKI, DAVID
Guía de las Comunicaciones del IBM PC.
Mc Graw-hill, México, 1986.

LATHI,B.P.
Sistemas de Comunicación.
Interamericana, 1986.

PHOENIX TECHNOLOGIES LTD.
"System Bios For IBM PC/XT/AT Computers and compatibles".
Addison-Wesley, 1986.

SCHILLING, DONALD [Y] CHARLES BELOVE
Circuitos Electrónicos
Marcombo, 1987.

TAUB, HEBERT [Y] DONALD SCHILLING
"Digital Integrated Electronics".
McGraw-Hill, 1977.

VAUGHN, MARTIN [Y] DEAN DAVIS.
Proyectos de Seguridad.
CEAC, 1989.

NATIONAL SEMICONDUCTOR
"Data Conversion/Adquisition".
National Semiconductor, 1984.