

6



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA
DE ALARMA Y SEGURIDAD BASADO EN EL
MICROCONTROLADOR COP8780.**

299089

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECÁNICO ELECTRICISTA
(AREA ELÉCTRICA Y ELECTRÓNICA)

P R E S E N T A N :

**ANTÚNEZ ROMERO ALFONSO
ROMERO ANGUIANO ROBERTO**

Director de tesis: Ing. Miguel Angel Cruz León



Ciudad Universitaria

2001



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

ÍNDICE

INTRODUCCIÓN.....	I
CAPÍTULO I.....	7
SISTEMAS DE SEGURIDAD TÍPICOS	7
1.1 DESCRIPCIÓN GENERAL	9
1.2 COMPONENTES DE UN SISTEMA DE SEGURIDAD.....	9
1.2.1 ELEMENTOS DE CONMUTACIÓN	10
1.2.2 UNIDAD DE CONTROL	10
1.2.3 LA UNIDAD DE CONTROL DEL SISTEMA DESEMPEÑA CINCO FUNCIONES BÁSICAS:.....	10
1.3 CIRCUITO ANTI-SABOTAJE	11
1.4 BOTÓN AUXILIAR.....	11
1.5 INDICADORES	11
1.6 PRUEBA DE PREAMBIVACIÓN	11
1.7 ACTIVACIÓN Y DESACTIVACIÓN DEL SISTEMA.....	12
1.8 FUENTE DE PODER.....	13
1.9 "RESET".....	14
1.10 SECCIÓN DE RESPUESTA DEL SISTEMA.....	14
1.11 APARATOS DE ALARMA ACÚSTICA	14
1.12 APARATOS DE ALARMA ÓPTICA.....	15
1.13 APARATO DE ALARMA "SILENCIOSA".....	15
1.14 LÍNEAS DE AVISO	15
1.15 CIRCUITOS DE SEGURIDAD BÁSICOS.....	18
1.16 SISTEMAS DE SEGURIDAD INALÁMBRICOS Y DE CIRCUITO CERRADO POR TV 19	
CAPÍTULO 2.	21
SENSORES PARA SISTEMAS DE SEGURIDAD	21
2.1 SENSORES PARA SISTEMAS DE SEGURIDAD.....	23
2.2 A) SENSORES TIPO INTERRUPTOR	23
2.2.1 SENSORES DE CONTACTO.....	23
2.2.2 MICROINTERRUPTORES	24

B.1	DIAGRAMA A BLOQUES.....	98
B.3	ORGANIZACION DE LA MEMORIA.....	99
B.3.1	MEMORIA DE PROGRAMA.....	99
B.3.2	MEMORIA DE DATOS.....	99
B.4	MAPEO DE LA MEMORIA DE REGISTROS DE I/O.....	101
B.4.1	PUERTOS ENTRADA/SALIDA CONFIGURABLES.....	102
B.4.2	SALIDAS DEDICADAS.....	102
B.4.3	ENTRADAS DEDICADAS.....	102
B.5	EL NÚCLEO DE LOS REGISTROS.....	103
B.5.1	ACUMULADOR.....	103
B.5.2	CONTADOR DE PROGRAMA.....	103
B.5.3	REGISTRO PSW (DIRECCIÓN 00EF Hex).....	103
B.5.4	REGISTRO DE CONTROL CNTRL (DIRECCIÓN 00EE Hex).....	104
B.5.5	REGISTRO DE DATOS.....	104
B.5.6	APUNTADOR DE PILA.....	105
B.5.7	APUNTADORES DE MEMORIA DE DATOS (REGISTROS ÍNDICE).....	105
B.5.8	REGISTRO "MICROWIRE/PLUS".....	106
B.5.9	REGISTROS TEMPORIZADORES (TIMER).....	106
B.6	FUNCIONAMIENTO DEL CPU.....	106
B.6.1	LÓGICA DE CONTROL.....	106
B.6.2	UNIDAD LÓGICA ARITMÉTICA (ALU).....	106
B.7	"RESET".....	108
B.8	OPCIONES DE RELOJ.....	108
B.8.1	OSCILADOR DE CRISTAL.....	108
B.8.2	CIRCUITO OSCILADOR RC.....	108
B.8.3	CIRCUITO OSCILADOR RC SOLO PARA EL COP840CJ.....	109
B.8.4	OSCILADOR EXTERNO.....	109
B.9	MODOS DE DIRECCIONAMIENTO.....	109
B.9.1	MODO DE DIRECCIONAMIENTO DE OPERANDO.....	109
B.9.2	MODOS DE TRANSFERENCIA DE CONTROL.....	112
B.10	MODO DE "HALT".....	114
B.11	INTERRUPCIONES.....	114
B.11.1	CONTROL DE INTERRUPCIÓN.....	115
B.11.2	PROCESAMIENTO DE UNA INTERRUPCIÓN.....	116
B.11.3	DETECCIÓN DE CONDICIONES ILEGALES.....	116
B.12	TIMER/COUNTER.....	116
B.12.1	MODO 1.- TEMPORIZADOR CON REGISTRO DE AUTO CARGA.....	117
B.12.2	MODO 2.- CONTADOR DE EVENTOS EXTERNO.....	117
B.12.3	MODO 3.- TEMPORIZADOR CON REGISTRO DE ENTRADA DE CAPTURA.....	118
	APÉNDICE C.....	119
	CONJUNTO DE INSTRUCCIONES DE LA FAMILIA COP8.....	119

E.9	CORRECCION DECIMAL BCD	166
E.10	SELECCION DEL MICROCONTROLADOR	167
APÉNDICE F	169
ESPECIFICACIONES	169
APÉNDICE G	173
FUNDAMENTOS PARA LA EVALUACIÓN DE PROYECTOS	173
BIBLIOGRAFÍA	177
BIBLIOGRAFÍA	179

INTRODUCCIÓN

Introducción

EL sistema de alarma y seguridad fue desarrollado en base al sistema de desarrollo COP8 del laboratorio de Electrónica.

Una de las necesidades del sistema de alarma y seguridad son la cantidad de sensores que se colocaron en la casa y las características de configuración de acuerdo a las especificaciones del usuario de la casa habitación

De acuerdo con la información del usuario de la casa se requiere en la planta baja sensores en ventanas, puertas y un sensor de movimiento, en la planta alta se requiere sensores en puertas, ventanas, domos y un sensor de movimiento que debe ser configurado por el usuario dado que tiene niños y suelen andar por la casa de día o de noche.

El sistema requiere suficientes sensores así como un teclado y un "display" para mostrar el estado del sistema.

El microcontrolador COP8780 de 8 bits, cuenta con 40 terminales, de las cuales ya se tienen ocupados Vcc, GND, Reset, G7/CKO y CKI, tiene 5 puertos de los cuales 3 son de entrada y salida (dos de ocho terminales y uno de cuatro terminales), un puerto de solo entrada y otro puerto de solo salida.

Por lo anterior se decide utilizar puertos multiplexados en el teclado para poder colocar mas sensores en el sistema dado que el número de terminales es limitado.

El desarrollo de la tesis esta organizada en los siguientes capítulos:

El Capítulo 1, muestra los componentes básicos de un sistema de seguridad. sensores, unidad de control, circuitos anti-sabotaje, botón auxiliar, indicadores, pruebas de activación y desactivación, "reset", fuente de poder, etc

El Capítulo 2, muestra los principales sensores utilizados en los sistemas de seguridad divididos en dos categorías: sensores tipo interruptor y sensores volumétricos (Área protegida).

El Capítulo 3, describe el desarrollo del sistema de seguridad, tomando como base los requerimientos del usuario de la casa, se muestran croquis de la planta baja y alta de la casa habitación.

También se muestran los diagramas de flujo del programa del sistema de alarma y seguridad, una explicación de las rutinas y subrutinas, diagramas y circuitos del sistema, así como una descripción de la forma de operar el sistema y el programa final del sistema de seguridad y alarma con comentarios del microcontrolador COP8780.

El Capítulo 4, corresponde a los resultados y conclusiones del sistema de seguridad y alarma en base al microcontrolador COP8780

Los Apéndices A, B, C y D, tienen información de la familia básica del COP8, diagrama a bloques de la arquitectura, mapeo de memoria, registros, modos de direccionamiento, conjunto de instrucciones básicas, el sistema de desarrollo, ensamblador, etc.

CAPÍTULO 1

SISTEMAS DE SEGURIDAD TÍPICOS

1.1 DESCRIPCIÓN GENERAL

Un sistema de seguridad es un conjunto de elementos que interactúan entre sí para brindar protección a una zona en particular; al momento en que alguno de los conmutadores de seguridad cambie su estado original, se generará una señal de alarma y se procede a ejercer una acción de seguridad o de alerta.

La función principal de un sistema de seguridad es dar aviso del allanamiento de una zona protegida por dispositivos de detección o sensores. Esto quiere decir que un sistema de seguridad, no necesariamente impide que un intruso entre a una área protegida, si no que el sistema sólo da aviso de que se ha detectado a dicho intruso.

Existen dos tipos de instalaciones en los sistemas de seguridad:

CLASE A: En este tipo de instalación sólo existe la vigilancia en los puntos más importantes mediante detectores de movimiento, humo, fuego, etc.

CLASE B: La instalación debe abarcar los puntos de la instalación CLASE A, así como puertas, ventanas y otros puntos de entrada.

1.2 Componentes de un sistema de seguridad

Un sistema de seguridad está formado por tres componentes básicos:

- Elementos de conmutación (sensores y detectores)
- La unidad de control
- Sección de respuesta del sistema (Fig. 6.1).

Los sensores y detectores registran variaciones físicas en su zona de operación y envía una señal a la unidad de control. La unidad de control reacciona de una manera programada a la recepción de la señal emitiendo inmediatamente, o con un retardo preestablecido, una señal de alarma óptica, acústica o pidiendo ayuda mediante un aparato telefónico automático, dependiendo del tipo de aparato que se tenga como elemento de aviso de violación del sistema.

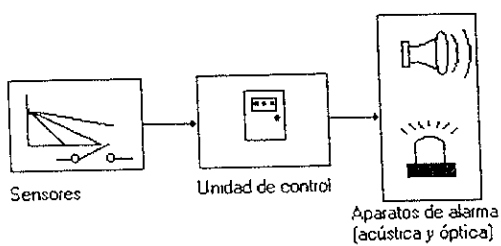


Fig. 1.1 Componentes básicos de un Sistema de Seguridad

1.2.1 Elementos de conmutación

Los sensores y detectores forman los elementos de conmutación de un sistema de seguridad, actuando como contactos de circuitos eléctricos, los cuales se conectan a la unidad de control del sistema por medio de las líneas de aviso. La modificación del estado original de un elemento de conmutación es transmitida a la unidad de control del sistema a través de la línea de aviso. La descripción y función de los diferentes sensores y detectores se describen en el capítulo 7.

1.2.2 Unidad de control

La unidad de control es el cerebro del sistema, ya que es la que controla todos los sensores, líneas de aviso y aparatos de salida. A sus entradas van conectados los cables de todas las líneas de aviso, y también las líneas de salida. Normalmente las zonas protegidas se representan con diodos emisores de luz (LED's) que indican directamente, en la unidad de control, en qué zona se ha detectado la violación de alguno de los dispositivos de seguridad. Hay líneas de aviso que disparan la alarma instantáneamente, mientras que otras lo hacen después de un determinado retardo de tiempo.

El diseño de la unidad de control depende de las particularidades de los objetos y áreas a proteger y de las necesidades del usuario. Por lo que el equipamiento de una instalación de seguridad depende de lo que se quiera de ella.

Una unidad de control sencilla debe tener las siguientes características:

- Contar con conexiones para varias líneas de aviso.
- Tener como mínimo una línea de acción de seguridad.
- Tener una batería para alimentación de emergencia.
- Permitir la conexión de aparatos de alarma exteriores ópticos y acústicos.
- Indicar claramente, en el panel de control, el estado de la instalación.

1.2.3 La unidad de control del sistema desempeña cinco funciones básicas:

- 1) Captación de las señales emitidas por los sensores de cada una de las líneas de aviso.
- 2) Activación del aparato de alarma.
- 3) Evaluación de las señales e información de los diferentes dispositivos de conmutación situados fuera de la unidad de control.
- 4) Vigilancia de las instalaciones con indicadores de averías y sabotajes.
- 5) Ejercer una acción de seguridad al momento de la activación de determinados detectores o elementos de conmutación.

1.3 Circuito anti-sabotaje

Por regla general, al menos una línea de aviso de la unidad de control permanecerá activa las 24 horas. Esta línea, por lo general, es la que vigila cualquier intento de sabotaje y reacciona cuando se manipulan componentes y líneas de aviso. Estamos hablando, por lo tanto, de una línea de corriente de reposo a la que se conectan, en cada sensor, unos microinterruptores que se activan cuando, por ejemplo, se levanta la tapa de la carcasa. Puede ser también conveniente disponer de otra línea de aviso de 24 horas con un circuito de corriente de trabajo, para conectar sensores. Todas las líneas de aviso que operan las 24 horas dan la alarma inmediatamente después de que se activo alguno de los contactos.

1.4 Botón auxiliar

Este botón auxiliar sirve para disparar la alarma, a pesar de que todo el sistema esté desactivado es utilizado para pedir ayuda en caso de emergencia, o bien para ahuyentar a algún intruso. El botón auxiliar puede formar parte de uno de los circuitos de las líneas de aviso, o también algunos sistemas mantienen este botón en forma "independiente" del sistema, de tal forma que si se restablece el sistema de seguridad la alarma sigue sonando y sólo es posible desactivarlo mediante una llave de seguridad o con un código independiente al del sistema de control.

1.5 Indicadores

Los indicadores sirven para saber en que estado de operación se encuentra nuestro sistema, así como para darnos a conocer la zona donde ha sido activado alguno de los sensores cuando se active la alarma. Hay diferentes tipos de indicadores, como son: botones de luz, LED, flechas y display LCD. Los indicadores usados en la unidad de control deben estar asociados con cada uno de los circuitos: anti-sabotaje, auxiliar y de cada una de las zonas. En algunos casos los indicadores se encienden momentáneamente para indicar el correcto funcionamiento de los circuitos.

Algunas unidades de control que usan microcontroladores manejan un display LCD, por medio del cual dan a conocer el estado de los circuitos de detección para que el usuario tome las medidas necesarias.

1.6 Prueba de preactivación

La mayoría de las unidades de control cuentan con un modo de prueba para comprobar el estado de los sensores antes de poner en operación el sistema de seguridad. En una prueba básica la unidad de control hace sonar una bocina interna, o enciende una luz indicadora, en lugar del aparato de alarma principal para indicar que alguno de los sensores está actuando. Cuando se tiene una unidad de control que indique las zonas por medio de luces, ésta mostrara cual es la zona que está activa. Si el sistema detecta una puerta abierta, o algún otro conmutador operando, se debe corregir la posición del conmutador antes de poner en operación el sistema. En caso de que todos los conmutadores estén en posición correcta la unidad debe encender una luz indicando que todo está en orden para encender el sistema.

En algunos modelos el interruptor principal debe tomar primero la posición de prueba antes de alcanzar la posición de operación. Lo que significa que el sistema no se

podrá encender si no pasa primero por la posición de prueba. Otros modelos no tienen una posición de prueba del interruptor, puesto que lo hacen automáticamente cuando se pone en operación el sistema y cualquier anomalía en el sistema es dada a conocer por el indicador, sin que suene el aparato de alarma principal.

Este tipo de prueba sólo verifica que la malla cerrada no se encuentre abierta en ningún punto y que los sensores estén en la posición indicada; no prueba el funcionamiento correcto de cada uno de los sensores. Y tampoco, en la mayoría de los casos, la unidad de control se checa a sí misma; el indicador de "todo en orden" no puede ser activado por que el circuito de control falla al momento de revisar el estado de los sensores y podrá fallar también en el caso de que realmente se active alguno de los dispositivos de seguridad.

Pero no es el mismo caso para un detector de presencia que puede fallar sin ser percibido, o cualquier otro dispositivo de seguridad que use el conmutador NO (normalmente abierto), como un tapete de detección de presencia. Aunque no es posible revisar el funcionamiento de cada uno de los sensores mediante la unidad de control, es necesario hacerles una prueba de funcionamiento regularmente. Para realizar esta prueba se cambia, mediante un interruptor, la alarma principal por un zumbador para que cada sensor a probar sea activado y el sistema sólo haga sonar el zumbador y no la alarma principal.

El elemento de alarma es probado ocasionalmente por un periodo corto. Algunas unidades de control pueden probar el elemento de alarma haciéndolo sonar suavemente. Esto es posible haciendo pasar una corriente pequeña por el elemento de alarma.

El circuito de protección contra sabotaje para el elemento de alarma sólo puede ser probada cortando los cables que la unen a la unidad de control. El elemento de alarma se activa, pero no se puede detener desde la unidad de control ya que al momento de cortar los cables el elemento de alarma queda automáticamente conectado a la batería de protección, la cual debe ubicarse dentro de la misma caja de protección que resguarda al elemento de alarma.

1.7 Activación y desactivación del sistema

Un sistema de seguridad puede activarse y desactivarse desde la unidad de control y puede ser con un circuito de retardo el cual permite al usuario retirarse del lugar protegido después de la activación del sistema, sin rebasar el lapso de tiempo preajustado. También se dispone de un lapso de tiempo para desactivar el sistema después de accionar alguno de los elementos de protección. Se puede adaptar la unidad de control para que, si se rebasa el lapso de tiempo, se dispare primero una alarma previa interior para advertir al usuario de la activación de un sensor.

Este tipo de conmutación para el encendido tiene sus ventajas y desventajas. Como ventajas tenemos: su bajo costo de instalación. Y como desventajas: el riesgo de olvidar conectar el sistema, el intruso tiene la posibilidad de anular completamente la instalación después de vencer la seguridad mecánica puesto que el sistema actúa con retardo.

Puede ser más cómodo y seguro la activación del sistema de seguridad mediante dispositivos de conexión externos instalados fuera de la zona protegida. La forma de operación de algunos de estos dispositivos se describe a continuación:

Contacto electromagnético:

Una de las soluciones más sencilla es colocar un contacto electromagnético en la puerta de la casa. El contacto, colocado detrás del cerradero, se acciona al cerrar la puerta y despiázarse el pasador de la cerradura, activando el sistema. El

inconveniente del contacto electromagnético es que una apertura forzada de la cerradura provocara, tal vez, la desconexión del sistema.

Conmutador de llave:

El conmutador de llave puede colocarse en cualquier parte, ya sea en montaje saliente o empotrado. Cuentan con protección contra sabotajes y muestran, con diodos luminosos, el estado de activación del sistema.

Conmutador por código:

Es posible teclear un código numérico en lugar de accionar el conmutador exterior con una llave.

Cerradura de conexión por pasador:

Para activar el sistema de seguridad al salir de la casa, desde la última puerta, se puede montar, además de la cerradura existente, un cerradura de conexión por pasador. Estamos hablando de una cerradura con conmutador integrado que por un lado ofrece seguridad mecánica y por otro está bien protegida contra cualquier sabotaje del sistema.

Cerradura de bloqueo:

Esta cerradura se puede montar como complemento de la cerradura normal en su puerta. De igual forma que la cerradura de conexión por pasador, activa la instalación del sistema al cerrarse la puerta, pero también cuenta con un dispositivo obligatorio, ligado a la cerradura, que sólo libera el pasador de ésta si la instalación del sistema está correctamente preparada para su conexión. Con este dispositivo, todas las puertas y ventanas protegidas tienen que estar cerradas, ya que de lo contrario un electroimán bloquea la cerradura y avisa al usuario de su olvido.

Ningún sistema de seguridad puede compensar los defectos de seguridad mecánica. Y es por esto que un sistema de seguridad debe ir precedido de dispositivos de seguridad mecánica para brindar una mayor protección.

1.8 Fuente de poder

La corriente para que funcione el sistema de seguridad, es tomada de la fuente principal, la cual se deriva de un toma corriente de 127 Vac, además de tener una batería de emergencia la cual se activa automáticamente si la fuente principal llega a fallar por algún motivo. La fuente de poder es probablemente el punto más débil de todo el sistema. Si por algún motivo no llega energía eléctrica a la casa, que se encuentra protegida por el sistema de seguridad, este quedara a disposición de las baterías de emergencia. Las baterías usadas para esta tarea deben de tener una duración de funcionamiento como mínimo de 2½ días. El tiempo de duración de una batería depende de variantes, tales como: cuantas veces se active el sistema en el tiempo de servicio de la batería, del número y tipo de dispositivos de protección que se están usando en las zonas protegidas, del tipo de indicadores y de la capacidad de la batería.

1.9 "Reset"

La unidad de control se puede inicializar de dos maneras: ya sea por medio de un botón de "reset", o apagando todo el sistema por medio de un interruptor de llave. El botón de "reset" no debe estar al alcance del usuario, ya que el sistema debe ser sólo inicializado por personal capacitado. Esto se hace para prevenir falsas alarmas. Si el disparo de la alarma fue debido al mal funcionamiento del sistema, un simple "reset" del sistema sin encontrar el problema sólo provocara otra falsa alarma. Cuando se llama a un ingeniero para atender una falsa alarma, él revisa todo el sistema antes de usar el "reset".

Cuando se agota el tiempo de la alarma después de una activación, algunos sistemas se autoreinician pero dejan de tomar en cuenta el circuito que se encuentra en estado de alarma; esto hace que el sistema no se vuelva a activar y el resto de los sensores continúan funcionando.

1.10 Sección de respuesta del sistema

La sección de respuesta de un sistema incluye campanas, zumbadores, luces, unidades de marcación telefónica, registradores gráficos, cámaras e indicadores de medida.

Cuando se modifica el estado original de uno de los contactos de los sensores, llega un aviso hasta la unidad de control por medio de la línea de aviso, ésta reacciona según el circuito, o la programación, activando la salida del aparato de alarma, ya sea inmediatamente o después de un retardo.

Dado que los aparatos de alarma son los más importantes del sistema, éstos deben estar bien protegidos contra intentos de sabotaje e inclemencias del tiempo. Así que para una máxima seguridad del aparato de alarma es conveniente incluir una batería recargable en su caja de protección, de ésta manera la alarma no podrá ser desactivada al cortan los cables que la comunican a la unidad de control. También para duplicar la seguridad se puede colocar un segundo aparato de alarma acústica en un lugar separado del primero.

Es necesario diseñar un circuito que mantenga la batería recargada, mediante la fuente de alimentación de la unidad de control y que el aparato de alarma sólo haga uso de la batería cuando se desconecte el aparato de alarma de la unidad de control, de ésta manera la batería se conservara para casos de emergencia.

1.11 Aparatos de alarma acústica

Los aparatos de alarma acústica tienen dos misiones: la primera es ahuyentar al intruso, y al mismo tiempo llamar la atención de los vecinos o de alguna patrulla. Para una alarma acústica se utiliza un timbre o sirena, en caso de utilizar un timbre electromecánico es necesario usar un triac, o un relé, debido a su alto consumo de corriente. Las sirenas electrónicas se pueden conectar directamente a la salida de la unidad de control, gracias al bajo consumo de corriente. También se utilizan como aparatos de alarma internos pequeñas sirenas piezoeléctricas.

1.12 Aparatos de alarma óptica

Este tipo de alarmas pueden ser luces giratorias o intermitentes de bajo consumo y acopladas, necesariamente, a alarmas acústicas. La combinación se hace para llamar la atención en el entorno, dado que la duración de la alarma acústica está limitada a tres minutos por las disposiciones contra ruidos vigentes en la actualidad, mientras que la alarma óptica puede permanecer activada sin ninguna limitación.

1.13 Aparato de Alarma "Silenciosa"

Este tipo de aparatos son muy convenientes ya que retransmiten una señal de alarma por medio de una línea telefónica. Hay aparatos que transmiten la señal en intervalos que el destinatario debe conocer para saber que acciono la alarma. Se pueden almacenar varios números telefónicos para que el aparato marque uno tras otro y se tenga una mayor probabilidad de que el usuario reciba el mensaje.

Para no alertar inmediatamente al ladrón, y capturarlo, se puede hacer uso de la alarma "silenciosa". Sin que el intruso se percate de ello, la unidad de control, una vez activado alguno de los sensores, pone en marcha un teléfono automático que avisa, por ejemplo a un servicio de vigilancia o de seguridad.

Los teléfonos automáticos se utilizan más bien como complemento de alarmas ópticas y acústicas, para llamar la atención de los alrededores y de una persona de confianza de las proximidades.

1.14 Líneas de aviso

Se denomina línea de aviso o bucle de alarma a la conexión que hay entre el sensor y la unidad de control. Debido a que cada sensor actúa como un contacto normalmente cerrado (NC) o normalmente abierto (NO), las líneas de aviso se pueden clasificar de la siguiente forma:

Líneas de aviso de corriente de trabajo:

Una línea de aviso de corriente de trabajo es aquella que contiene uno o varios sensores con contactos normalmente abiertos. Al cerrarse alguno de los contactos circulará una corriente, la cual es detectada por la unidad de control para que active la alarma. En una línea de aviso de corriente de trabajo se pueden conectar todos los sensores de alarma que se desee, siempre y cuando se conecten en paralelo. En un circuito de esta clase, la unidad de control no puede reconocer en qué lugar de la línea de aviso se ha cerrado el circuito, por lo que también reacciona en caso de corto circuito en los cables que unen los sensores con la unidad de control.

Cabe señalar que este tipo de línea de aviso tiene un grave inconveniente: se puede sabotear fácilmente cortando cualquier hilo de la línea como se muestra en la Figura 6.2, por esta razón las líneas de corriente de trabajo sólo se deben emplear en los lugares donde no se espere sabotaje, por ejemplo para sensores que disparen dispositivos técnicos (detectores térmicos, de humo o de gas), así como en circuitos de asalto

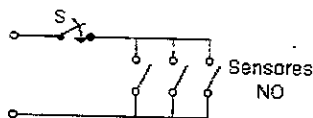


Figura 1.14.1 Circuito de corriente de trabajo con desconexión
(Con el interruptor "S" abierto el circuito de corriente está inactivo).

Líneas de aviso de corriente de reposo:

Este tipo de líneas de aviso, al contrario de uno de corriente de trabajo, hace uso de sensores con contactos normalmente cerrados y conectados en serie. Por la línea circula una corriente de manera que al activarse alguno de los sensores se abre el circuito, la unidad de control detecta esta variación y dispara la alarma. El inconveniente en este tipo de líneas es que si se cortocircuita el circuito de corriente antes de los sensores estos quedan fuera de servicio como se muestra en la figura 6.3. Al igual que en la línea de aviso de corriente de trabajo, también se pueden conectar los sensores que se desee, pero hay que conectarlos en serie.

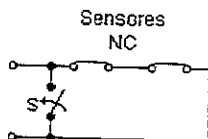


Figura 1.14. 2 Circuito de corriente de reposo.
(Con el interruptor "S" cerrado el circuito de corriente está inactivo)

Líneas de aviso diferenciales:

Existe una solución para el punto débil del principio de la corriente de reposo: en el circuito de corriente de reposo se conecta una resistencia para regular el paso de corriente en el circuito (Fig. 6.4). La unidad de control está ajustada al valor de la resistencia y al valor de la corriente, según el principio de la línea de aviso diferencial. Por lo tanto, si varía la resistencia eléctrica de la línea de aviso en un porcentaje determinado, la unidad de control reacciona disparando la alarma. Es así como se reconoce la señal del sensor y además cualquier sabotaje por cortocircuito o corte de la línea. Las líneas de aviso diferencial, además, permiten emplear no sólo sensores de alarma por apertura de contactos, sino también por cierre de contacto (Fig. 6.5). Incluso, hay circuitos en los que se emplean ambos tipos de sensores (Fig. 6.6). Algunos sistemas de seguridad manejan circuitos diferenciales en los que se pueden conectar sensores, con conectores normalmente cerrados en paralelo con una resistencia individual en cada caso (Fig. 6.7).

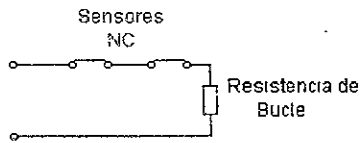


Figura 1.14.3 Circuito diferencial con sensores de corriente de reposo

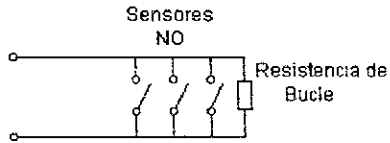


Figura 1.14.4 Circuito diferencial con sensores de corriente de trabajo

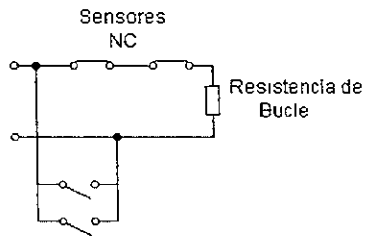


Figura 1.14.5 Circuito diferencial con sensores combinados.

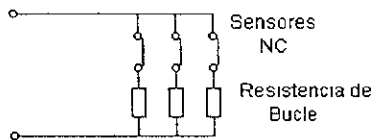


Figura 1.14.6 Circuito diferencial con sensores NC en paralelo

En el caso de los sistemas inalámbricos la unidad de control hace uso de un conmutador de código, que tienen los transmisores, para saber de que zona proviene la señal del transmisor. Las instalaciones de este tipo de sistemas de seguridad trabajan en el margen de los 430 MHz para evitar perturbaciones con emisoras externas. Antes de la puesta en servicio se realiza una modulación de frecuencias individual ajustando el código de transmisión, libremente seleccionable.

Sistemas de seguridad de circuito cerrado por TV

El sistema de seguridad de circuito cerrado por TV, o sistema de vídeo ofrece ventajas adicionales que los sistemas alámbricos e inalámbricos no poseen, por ejemplo, la superación de las limitaciones que tiene un detector de sonido en un sistema de seguridad alámbrico. La confianza y flexibilidad son dos características ofrecidas por el método de vídeo. Y para evitarse el inconveniente de estar continuamente vigilando los monitores del sistema, se pueden adaptar al sistema otros dispositivos de detección que indiquen el momento preciso en que dicha vigilancia es necesaria.

CAPÍTULO 2

SENSORES PARA SISTEMAS DE SEGURIDAD

2.1 SENSORES PARA SISTEMAS DE SEGURIDAD

Los sensores son una parte muy importante en cualquier sistema de alarma, estos son los guardianes que responden a cualquier disturbio causado por un intruso y disparan el circuito principal de la alarma.

Transductor:

Es un instrumento que transforma energía de una clase en energía de otra clase que guarda una relación conocida con la primera, a fin de poderla medir mas facilmente, o procesarla o transmitirla a distancia.

Existen dos clases de sensores:

A.- Consiste esencialmente de interruptores de varios tipos.

Usualmente tienen la capacidad de defender el perímetro, como son ventanas y puertas.

B.- La segunda clase protege el espacio por detección de movimiento.

Usualmente llamados sensores volumétricos.

2.2 A) SENSORES TIPO INTERRUPTOR

2.2.1 Sensores de contacto

Lo primero acerca de los interruptores es que funcionan como contactos Normalmente Abiertos (NO) y cierran cuando actúan o son Normalmente Cerrados (NC) y abren cuando actúan. Algunos sensores tienen dos pares de contactos, un par abre mientras el otro cierra; otros tienen tres contactos, uno es el común y el interruptor cambia entre los otros dos, estos son llamados "Changeover" (CO) o un-polo doble-tiro, (single-pole double-throw, SPDT), algunos tienen seis contactos que en realidad son dos interruptores DPDT, estos no se encuentran normalmente en sistemas de seguridad.

Los contactos normalmente cerrados son utilizados en circuitos en donde la corriente circula constantemente y cuando abre se suspende la corriente disparando la alarma; como se puede ver éste es el método principal de detección.

Un segundo sistema corre por todos los sensores en sistemas comerciales que funcionan las 24 horas del día. La mayoría de los sensores tienen cuatro terminales, dos para el interruptor y dos para el anti-sabotaje (Anti-tamper).

Los sensores normalmente abiertos disparan la alarma cuando están cerrados, estos generalmente son de presión, sin embargo, un circuito anti-sabotaje siempre está cerrado en sistemas comerciales, por lo que siempre están protegidos.

Los sensores tipo un-polo doble-tiro ofrecen gran seguridad por que pueden operar "normalmente abiertos" o "normalmente cerrados".

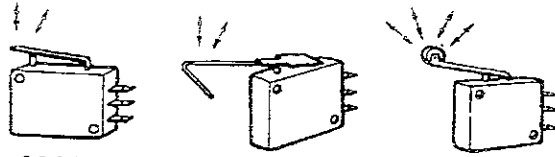


Figura 2.2.1.1 Microinterruptores: diferentes tipos de direcciones de operación.

2.2.2 Microinterruptores

Un tipo sencillo de interruptor es el microinterruptor (ver figura 2.1), como su nombre lo indica es un interruptor pequeño que es fácil de fijar en las ventanas, puertas, cajas fuertes, armarios y cajones en conjunto con el circuito anti-sabotaje.

El microinterruptor básico actúa cuando se presiona y regresa a su posición original cuando se deja de presionar (ver figura 2.2).

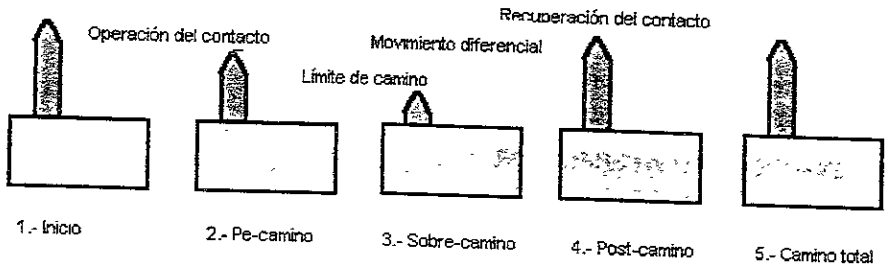


Figura 2.2.2.1 Operación del Microinterruptor mostrando sus puntos de operación.

A pesar de su tamaño los microinterruptores tienen un amplio rango de corriente; un porcentaje de 5 A es lo adecuado, pero algunos pueden llegar hasta 20 A.

La vida de operación es normalmente alta y usualmente especificada por el fabricante; la más baja es de 100 000 operaciones, pero típicamente se extiende a más de 10 millones.

2.2.3 Interruptores magnéticos

Una desventaja de los microinterruptores es el daño físico. Los contactos o palancas pueden sufrir deformaciones, atorarse o romperse por un impacto físico.

El interruptor Magnético (ver figuras 2.3 y 2.4) se describe como normalmente cerrado, (en el momento de su instalación el sistema esta en guardia) esto debido a que está influenciado por un campo magnético, sin embargo cuando el contacto está libre el circuito está normalmente abierto, por que no se encuentra cerca del campo magnético.

La distancia a la cual el imán puede influenciar al interruptor varía de acuerdo al

tipo de interruptor y fuerza del imán; típicamente las distancias pueden ser de 10 a 35 mm, estas representan las distancias en una ventana o puerta para que se puedan abrir y disparar la alarma.

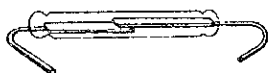


Figura 2.2.3.1 Interruptor magnético en un tubo de vidrio.

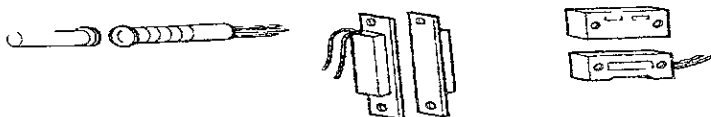


Figura 2.2.3.2 Interruptores magnéticos superficiales con imán.

2.2.4 Contactos resistencia-conectada

Circuito de doble propósito; se utiliza una resistencia a través de cada par de contactos. Una resistencia al final de la línea del circuito. Por ejemplo: cuando el sistema se encuentra en el "modo de día", una corriente circula a través del circuito, monitoreando que se encuentra intacto, si el contacto se abre cuando se abre la puerta, la resistencia mantiene la continuidad. Cuando el sistema cambia al "modo de noche", todos los contactos son cerrados, el panel de control monitorea éste valor de resistencia, por lo que si un contacto es abierto y la resistencia cambia, se dispara una alarma. Esto proporciona un circuito separado anti-robó, por lo tanto, permite que sensores normalmente abiertos puedan ser conectados en el mismo circuito y de esta forma se economiza una instalación eléctrica extra.

2.2.5 Sensores tipo radio

Otro sistema de alarma es el radio-control, el cual consiste de sensores y una unidad de control. Los sensores son interruptores convencionales, que pueden ser de montaje superficial para puertas y ventanas, sin embargo, en lugar de ser alambrado a la unidad de mando, cada sensor tiene un pequeño transmisor de radio con su propia pila interna. Cuando está funcionando el sensor, éste puede transmitir una señal a la unidad de control para activar la alarma.

Las ventajas de los sistemas de radio-control es que no necesitan una línea de control y, por lo tanto, los sensores se pueden trasladar de un lugar a otro. La desventaja es que no tienen circuito anti-robó y el tiempo de funcionamiento depende de la capacidad de la pila interna.

2.2.6 Tapete de presión

Existen tapetes rectangulares flexibles que vienen en una gran variedad de tamaños, aunque usualmente tienen un área de 0.2 a 0.6 m².

El método usual de construcción es un "sandwich" de dos hojas de metal (laminillas) separadas por una capa de hule espuma perforada.

La cantidad de presión que se necesita para actuar sobre un tapete es raramente dada por el fabricante, está se encuentra normalmente sobre 0.18 - 0.21 Kg/cm².

2.2.7 Cintas para ventanas

Existen varios métodos para proteger ventanas de vidrio, uno de los más comunes son cintas de lamina de metal fijas en la superficie del vidrio. Estas usualmente son hechas de aluminio, esto es útil donde hay alta contaminación atmosférica.

Las cintas se proporcionan en rollos de varios anchos: 3, 6 y 9 mm, que son las más comunes, pueden tener pegamento o no, la cinta con pegamento es la más conveniente de instalar.

Se pone la cinta laminada a lo largo de una área vulnerable de vidrio y se puede conectar al sistema de alarma, al circuito del botón de pánico las 24 horas o al circuito anti-robo. El método normal es poner dos cintas horizontales.

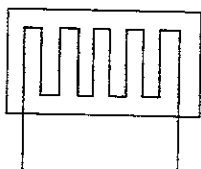


Figura 2.2.7.1 El cableado del vidrio consiste de dos bandas separadas aproximadamente unos 50 mm entre ellas.

2.2.8 Detectores de impacto

Los detectores de impacto responden a vibraciones o impactos, como los de vibración pero son más sofisticados. Se utiliza un cristal piezoeléctrico y cualquier vibración causa deformación lo cual genera un voltaje o corriente; un circuito analiza la naturaleza de la señal, esto hace que actúe un relevador si excede del nivel de configuración.

2.2.9 Detectores de inercia

Estos sensan perturbaciones y vibraciones, pero son particularmente sensitivos a las bajas frecuencias; como su nombre lo indica, el componente móvil tiene una masa que proporciona la inercia.

La corriente máxima que maneja el dispositivo es baja, debido a que el área de la esfera en contacto es mínima; la corriente especificada es de 0.2 mA, cuando el voltaje aplicado no excede de 2 Volts.

Los detectores de inercia son muy sensibles, pero, cuando en una cerca o estructura es sujeto a vibraciones de pequeña escala, como la exposición al viento, se pueden utilizar sensores magnéticos.

Los sensores de inercia son el mejor método para proteger una cerca, pero es se deben colocar aproximadamente cada 3 m.

2.3 B) SENSORES VOLUMÉTRICOS (ÁREA PROTEGIDA)

Los dispositivos de detección que se describieron anteriormente en donde todos (con la excepción del tapete de presión) operan por un intento de penetración y movimiento de alguna de sus partes. Por esta razón son descritos como defensa del perímetro, esto es esencial para mantener lejos a los intrusos hasta donde sea posible y no detectarlos dentro del área protegida.

En algunos casos la protección total de todas las partes del perímetro es impráctico. Siempre que se proporciona una alta seguridad, siempre existe la posibilidad de que un intruso inteligente se haga camino, como respaldo, los detectores de Protección de Espacio, a menudo llamados Sensores Volumétricos, se pueden añadir al sistema para cubrir estas necesidades.

Los tipos principales de protectores de espacios son: Ultrasonido, Microondas, Infrarrojo Activo y el Infrarrojo Pasivo.

Los detectores: ultrasonido y microondas se basan en el efecto "Doppler". El efecto "Doppler" es muy familiar en sí cuando el sonido de sirena de una ambulancia se acerca se empieza a escuchar algo agudo, cuando el vehículo pasa cerca de nosotros se escucha mas fuerte y cuando se aleja el sonido de la sirena es más grave

2.3.1 Detectores Ultrasónicos

El termino ultrasónico denota frecuencias de sonido que están arriba del rango de las capacidades de escuchar del ser humano. El límite máximo de 16 KHz es lo que se tienen promedio en gente sana.

Las frecuencias usadas en estos detectores varían dependiendo de los modelos, que pueden ser desde 23 hasta 40 KHz. La frecuencia es generada por un oscilador electrónico y alimenta a uno o más altavoces. Estos últimos deben ser demasiado pequeños, más pequeños que un "tweeter" en su sistema de alta fidelidad, ya que las partes deben moverse sumamente rápido y por lo tanto su masa debe mantenerse en un mínimo. Esta es una ventaja para un sistema de alarma ya que cada unidad puede ser hecha de forma pequeña y que no obstruye. La misma unidad mantiene un micrófono con circuitos de procesamiento y de amplificación.

El sonido de alta frecuencia es de esta manera protegido dentro del área. Los sonidos son captado directamente por un micrófono desde el altavoz; después de haber sido reflejado desde las paredes y objetos en el espacio de resguardo o protección. Cuando no hay algún movimiento, tanto los sonidos captados directos y reflejados mantienen la misma frecuencia, en el momento en que ocurre un movimiento, el sonido

reflejado desde el nuevo objeto en cuestión, produce un cambio de frecuencia perteneciente al efecto DOPPLER. En este sentido el micrófono ahora ha captado dos frecuencias: la original captada directamente desde el altavoz y el cambio de frecuencia reflejada desde el objeto en movimiento.

Las dos frecuencias distintas se mezclan y el resultado es una tercera cuyo valor es la diferencia de ambas. Cuando las dos frecuencias son mezcladas la tercera es frecuentemente llamada Nota de Ritmo (Beat Note).

Los circuitos de procesamiento detectan la presencia de este "Beat Note" y actúan como un relevador (puede considerarse como un relevador normalmente cerrado).

Los detectores ultrasónicos usualmente tienen un rango de hasta 9 m (30 pies), pero pueden tener un control de sensibilidad, que puede reducir el rango cerca de 3m (10 pies). El patrón polar tiene la forma de un lóbulo ancho por lo que el dispositivo proporciona mínima protección hacia los lados. Este puede ser enfocado hacia los puntos de entrada. De acuerdo a que el efecto Doppler se produce movimientos hacia adelante y hacia atrás, el sensor es más sensible a estos movimientos, que hacia los movimientos que son de lado, esta es la otra razón para fijarlo enfocado hacia los posibles puntos de entrada.

Las conexiones anti-robo son usuales y actúan como si fueran un micro-interruptor, que se activa cuando las cubiertas (o carcazas) son removidas. Estas conexiones no son usadas en situaciones domésticas.

La principal desventaja de los detectores ultrasónicos es la probabilidad de falsas alarmas, estas pueden ser originadas por un vehículo que se descompone, corrientes de gas o de agua, equipo de T.V., en ambientes industriales es posible que fugas o salidas de aire comprimido y la mayoría de las máquinas generan estas frecuencias como armónicas de su salida normal de ruido. Estos sonidos pueden ser detectados junto con la frecuencia del sensor y al producir una "Beat Note" podrían interpretarse como producto del efecto "Doppler". Otra fuente de falsas alarmas es la turbulencia del aire que puede distorsionar la propagación del sonido y producir cambios en el efecto "Doppler". Estas turbulencias pueden ser causadas por centrales de aire y calefacción, ventiladores, cortinas en movimiento. Los controles de sensibilidad ayudan a que se adapten los dispositivos a ciertos valores o rangos esperados.

2.3.2 Detector de Microondas

Estos detectores utilizan el efecto "Doppler" y trabajan de manera muy parecida al sensor Ultrasónico. La diferencia es que emplean radio ondas que oscilan a altas frecuencias. La frecuencia más utilizada es de 10.7 GHz, pero también se emplea 1.48 GHz. Los sensores de microondas no son tan susceptibles a falsas alarmas, como los detectores ultrasónicos; las turbulencias en el aire no tienen efecto, ni cualquier onda de sonido. Las lámparas fluorescentes y de neón pueden generar señales de radio que pueden ser aceptadas por el receptor, pero usualmente pueden ser filtradas y removidas electrónicamente.

2.3.3 Detector Infrarrojo Activo

Su forma de operar es muy similar a los dispositivos de microondas en donde un generador radia un haz que es recogido por el receptor. Si el haz es interrumpido por un intruso la alarma es disparada. El rayo Infrarrojo es producido por un cristal de galio

arsénico. Algunas lámparas de bulbos, calentadores o cerillos pueden generar radiación infrarroja; esto ofrece la posibilidad de burlar al sistema infrarrojo dirigiendo una fuente portátil al receptor y moviendo la fuente original.

2.3.4 Detector Infrarrojo Pasivo

El detector Infrarrojo Pasivo (PIR); se fabrica con el desarrollo de detectores infrarrojos cerámicos muy sensibles. El dispositivo no tiene un proyector, pero opera por detección de la radiación infrarroja emitida por el cuerpo humano.

La radiación recibida se enfoca en el sensor infrarrojo por un espejo curvo detrás él o unos lentes de plástico curvos colocados en el frente.

Una salida eléctrica se genera solamente cuando hay una diferencia entre ellas, nada pasa cuando ninguna o ambas reciben radiación constante; así sólo un objeto en movimiento produce una salida, una fuente estacionaria de calor no tiene efecto. Éste sirve para eliminar muchas posibles falsas alarmas.





CAPÍTULO 3

PRUEBAS DEL SISTEMA DE SEGURIDAD Y ALARMA

CAPÍTULO 3

PRUEBAS DEL SISTEMA DE SEGURIDAD Y ALARMA

3.1 INTRODUCCIÓN

El sistema de seguridad y alarma básicamente está formado por un "Display" de Cristal Líquido (LCD 2 x 16), un teclado 5 x 3, 15 sensores Normalmente Cerrados (NC), opcionalmente se pueden conectar 3 sensores que operen en modo Normalmente Abierto (NO) y la Unidad Central de Proceso (CPU). El Sistema de Seguridad es un sistema diseñado en base al microcontrolador COP8780CJ, que pertenece a la familia básica del microcontrolador COP8, de esta manera se logra tener un sistema de seguridad flexible en lo que se refiere a la adaptación de éste en diferentes áreas a cubrir, despliegue de información y modos de operación.

3.2 DISEÑO DEL SISTEMA DE SEGURIDAD

Las características de diseño del sistema de seguridad deben estar acorde con el área a proteger y de las necesidades de los habitantes de ella, por tal motivo se realizó un croquis de la casa habitación, la fig. 3.2.1 muestra el croquis de la planta baja y la figura 3.2.2 muestra el croquis de la planta alta de la casa habitación.

En la planta baja, tomando en cuenta el número y la localización de las ventanas y puertas se decidió colocar cuatro sensores magnéticos (NC), un sensor de vibración en la ventana de la recámara y un sensor de movimiento en el área del comedor, cocina, sala y escaleras, el cual tiene integrado un sistema antisabotaje.

En la planta alta se colocaron nueve sensores magnéticos (NC) en puertas y ventanas, dos sensores de ruptura en los domos, uno de movimiento en el área de la sala y dos líneas antisabotaje colocadas en la sirena y en el sensor de movimiento. La unidad de control (cerebro) se encuentra ubicada en la recámara principal desde la cual se comanda todo el sistema.

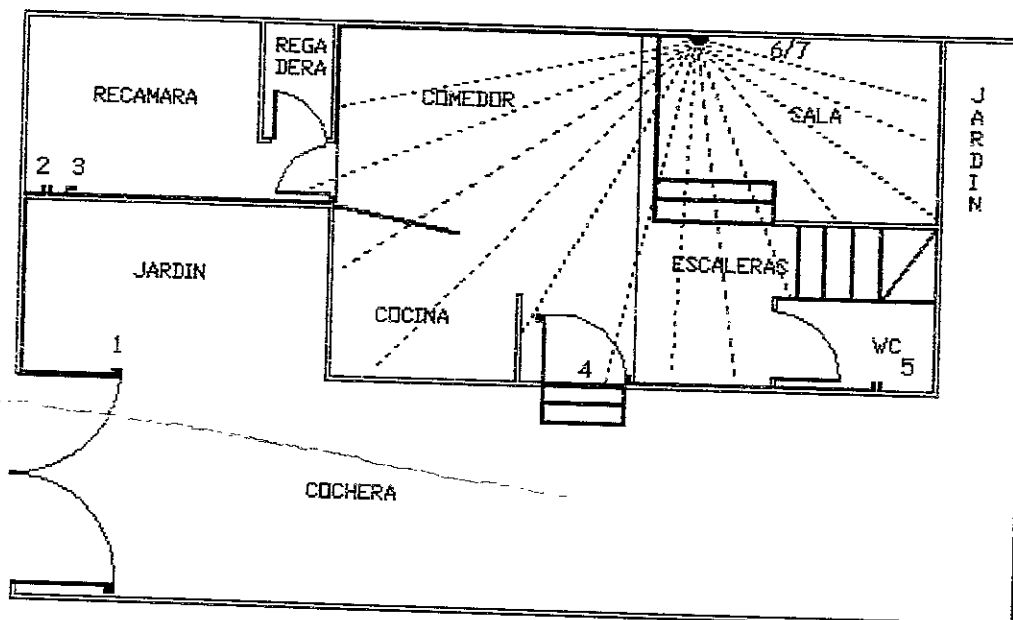


Figura 3.2.1 Croquis de la Planta Baja de la casa habitación.

SENSOR	TIPO DE SENSOR	LETRERO QUE DESPLIEGA
1	MAGNETICO	"PUERTA DE CALLE ABIERTA"
2	MAGNETICO	"VENT. DE RECAMARA PLANTA BAJA"
3	VIBRACION	"VENT. DE RECAMARA PLANTA BAJA"
4	MAGNETICO	"PUERTA DE PATIO ABIERTA"
5	MAGNETICO	"VENTANA DE BANO PLANTA BAJA"
6	INT. MEC. INTERNO	"SABOTAJE SENSOR-PRESENCIA P.BAJA"
7	INFRAROJO	"INTRUSO EN AREA PRINCIPAL BAJA"

Tabla 3.2.1 Descripción de sensores en la planta baja.

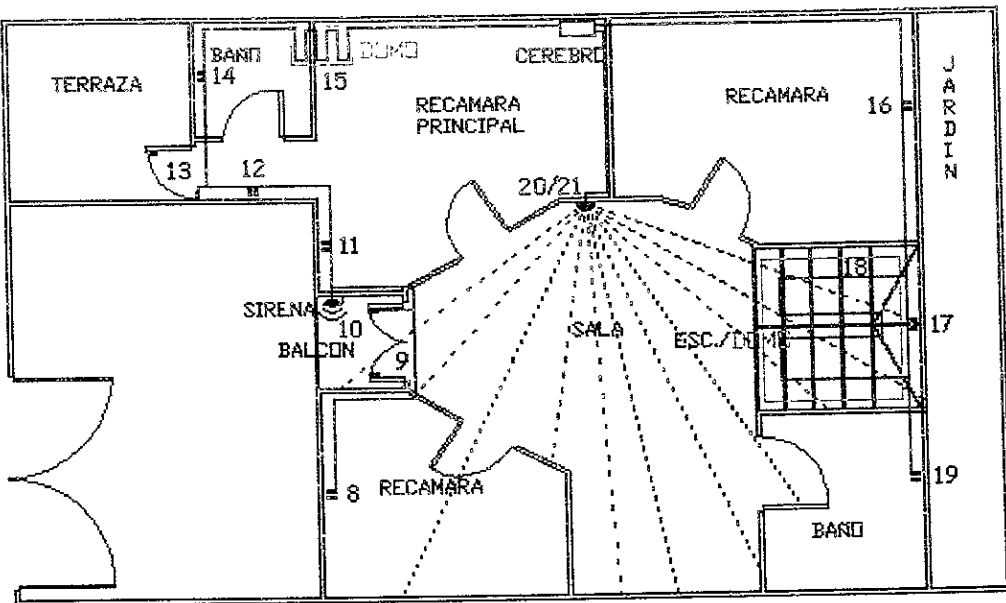


Figura 3.2.2 Croquis de la planta alta de la casa habitación.

SENSOR	TIPO DE SENSOR	LETRERO QUE DESPLIEGA
8	MAGNETICO	"PTA. BALCON ABIERTA, VENT.REC-PEQ"
9	MAGNETICO	"PTA. BALCON ABIERTA, VENT.REC-PEQ"
10	INTERRUPTOR MEC.	"LINEA ANTISABOTAJE SIRENA"
11	MAGNETICO	"PTA-TARRAZA, VENT.REC-PRINCIPAL"
12	MAGNETICO	"PTA-TARRAZA, VENT.REC-PRINCIPAL"
13	MAGNETICO	"PTA-TARRAZA, VENT.REC-PRINCIPAL"
14	MAGNETICO	"VENTANA DE BAÑO PLANTA ALTA"
15	HILO DE COBRE	"DOMO DE RECAMARA PRINCIPAL ROTO"
16	MAGNETICO	"PROTEC.TRASERA; RECA, DOMO-ESC, BA"
17	MAGNETICO	"PROTEC.TRASERA; RECA, DOMO-ESC, BA"
18	HILO DE COBRE	"PROTEC.TRASERA; RECA, DOMO-ESC, BA"
19	MAGNETICO	"PROTEC.TRASERA; RECA, DOMO-ESC, BA"
20	INT. MEC. INTERNO	"SABOTAJE SENSOR-PRESENCIA P. ALTA"
21	INFRAROJO	"INTRUSO EN SALA PLANTA ALTA"

Tabla 3.2.2 Descripción de sensores en planta alta.

En las tablas 3.2.1 y 3.2.2 se describen los diferentes tipos de sensores que se usaron en la planta baja y alta respectivamente, también se indica el letrero que despliega el sistema al momento de activar alguno de estos sensores.

Para el diseño del sistema de seguridad se usó un microcontrolador COP8780CJ (40-pin) para procesar la información, un "Display" de Cristal Líquido (LCD) el cual facilita la comunicación del usuario con el sistema, un teclado numérico-multifuncional para interactuar y comandar al sistema y sensores con lo que se detecta la presencia en una área determinada.

3.2.1 Adaptación del Microcontrolador al Sistema de Seguridad y Alarma

El microcontrolador de 8-bits COP8780CJ cuenta con una memoria RAM de 128 bytes y una UVEPROM de 4096 bytes, suficiente memoria como para crear un programa en él y que controle un número de sensores considerable y un LCD. Otras de las características que hacen que este microcontrolador sea ideal para el sistema de seguridad son: ejecución de instrucciones en $1\mu\text{s}$ (reloj de 10 MHz), bajo consumo de corriente 21 mA, 3 fuentes de interrupciones, un contador de carrera libre (TIMER) de 16-bits y 5 puertos.

Se puede decir que el cerebro del sistema de seguridad es el microcontrolador el cual toma decisiones predeterminadas para cada uno de los diferentes acontecimientos que sucedan alrededor de su ambiente de trabajo.

Funciones del microcontrolador

Las principales funciones del microcontrolador son las siguientes:

- Inicialización de LCD al momento de una transición APAGADO-ENCENDIDO del sistema, o también si ocurre una interrupción por software.
- Detectar si se oprimió alguna de las teclas, si es así despliega en pantalla los caracteres correspondientes a la tecla oprimida y realiza la función correspondiente a dicha tecla.
- Verificar si existe una transición de estado en los sensores, si es así, activa la sirena e indica en pantalla cual de los sensores se activó.
- Mediante el reloj de carrera libre genera sonido de alerta.

Descripción y funcionamiento del programa

A continuación se muestra el programa en diagramas de flujo para un mejor entendimiento de éste. En la figura 3.2.3 se observa de manera muy general como se conforma todo el programa y en las figuras 3.2.4, 3.2.5, 3.2.6 y 3.2.7 se puede analizar de una manera más clara como procesa la información el microcontrolador y además se puede ver también todas las funciones que se tienen en el sistema.

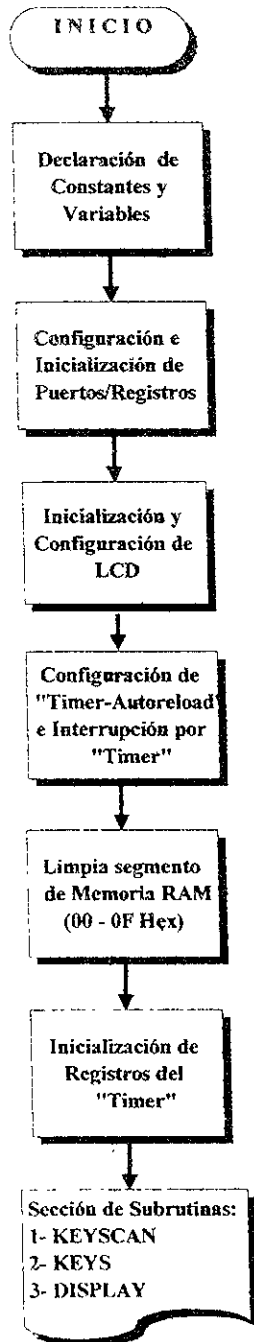


Figura 3.2.3 Diagrama de flujo del programa en forma general

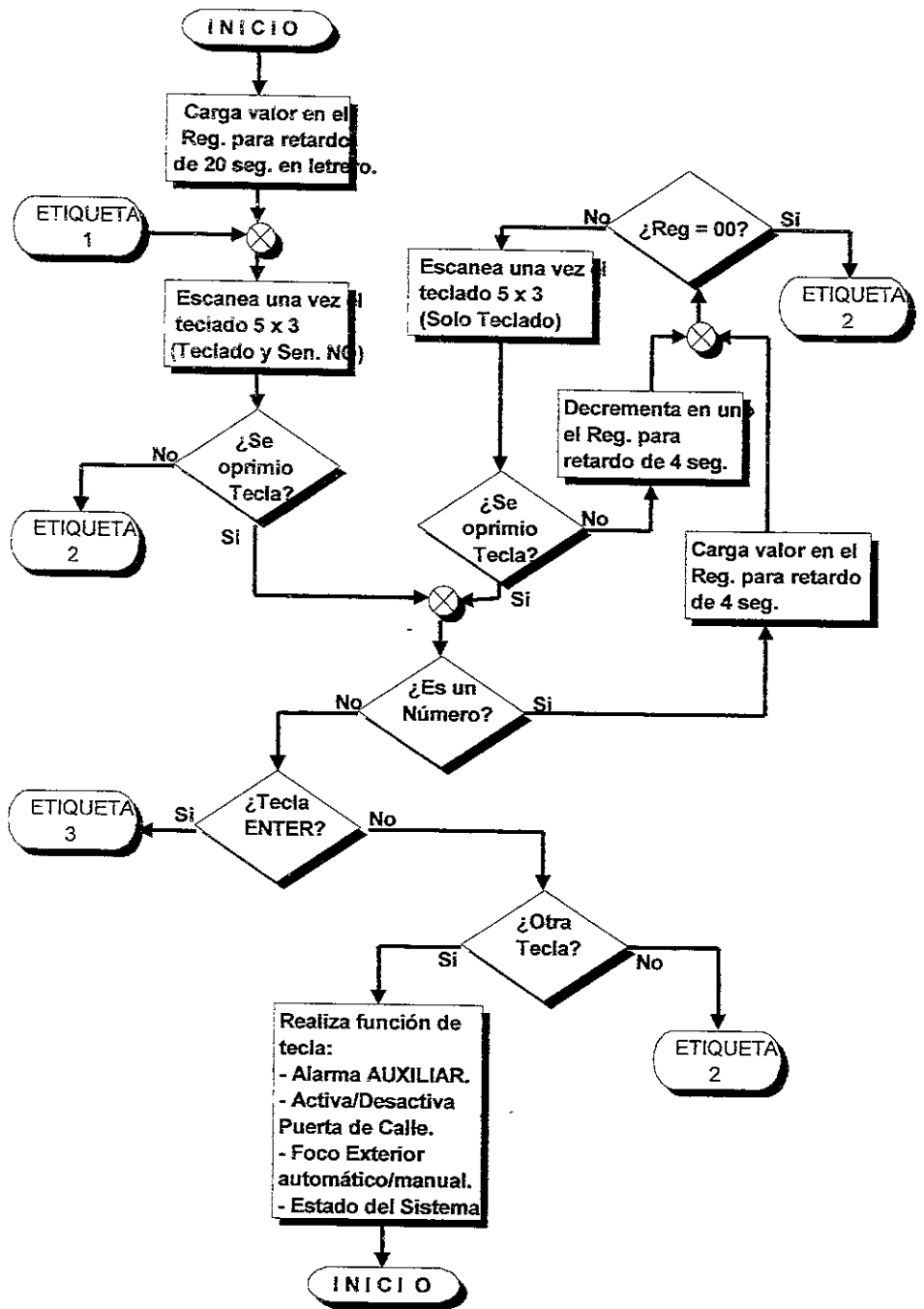


Figura 3.2.4 Primera parte del diagrama de flujo del programa.

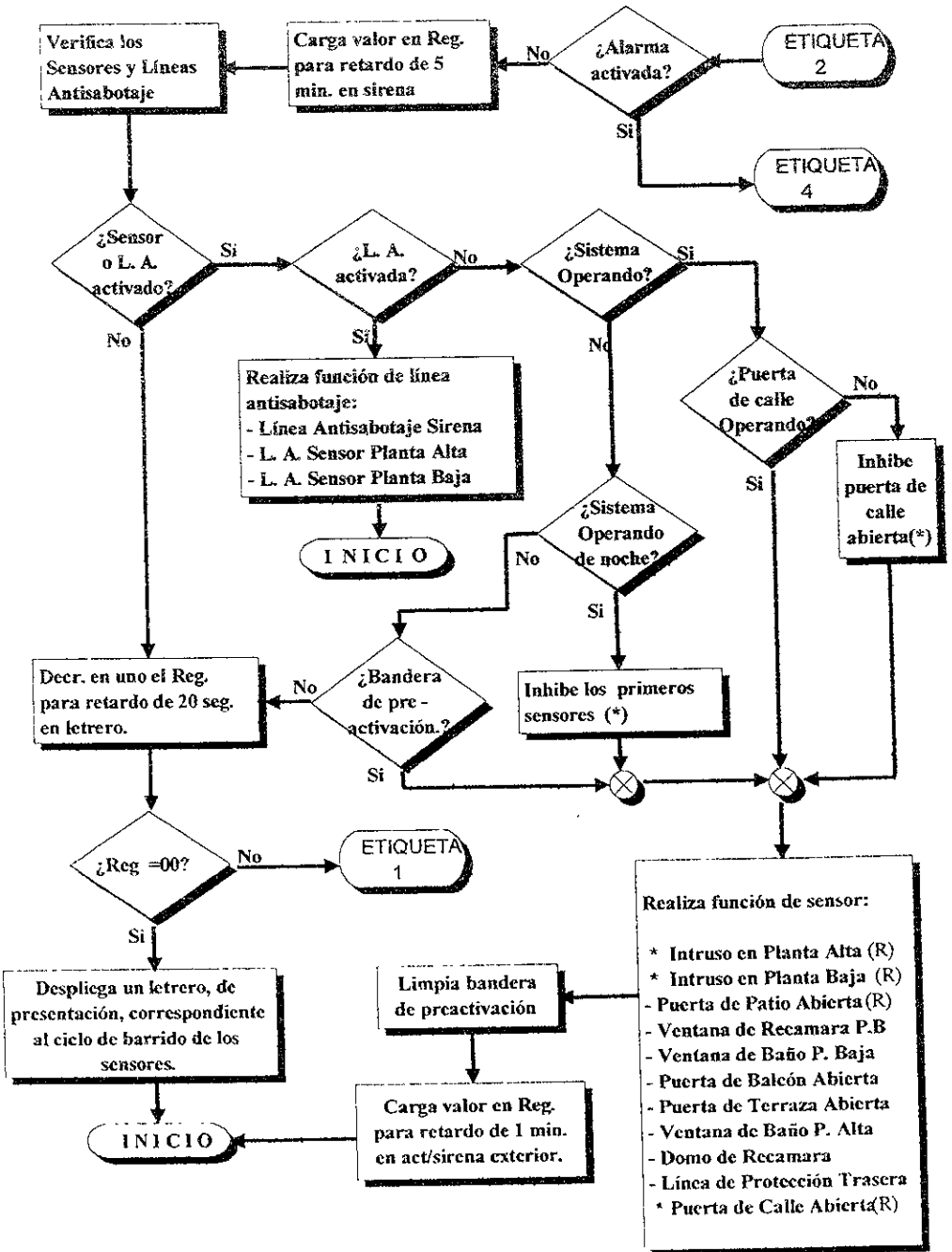


Figura 3.2.5 Segunda parte del diagrama de flujo del programa

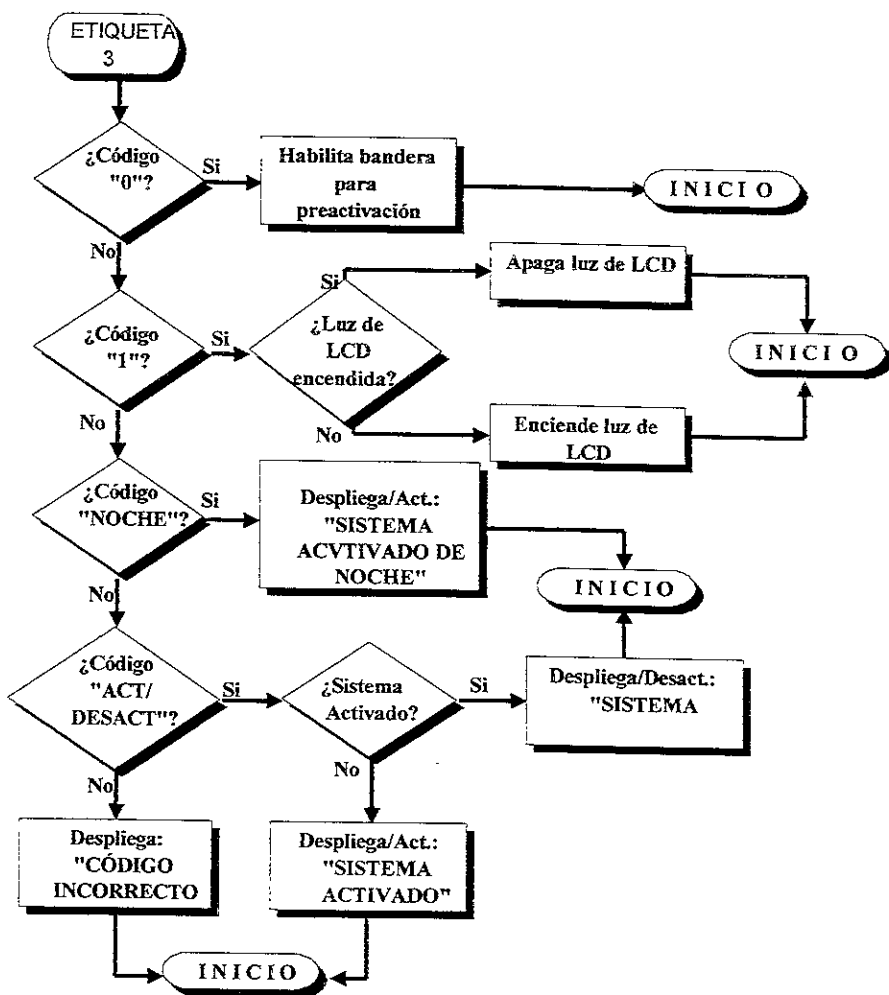


Figura 3.2.6 Tercera parte del diagrama de flujo del programa

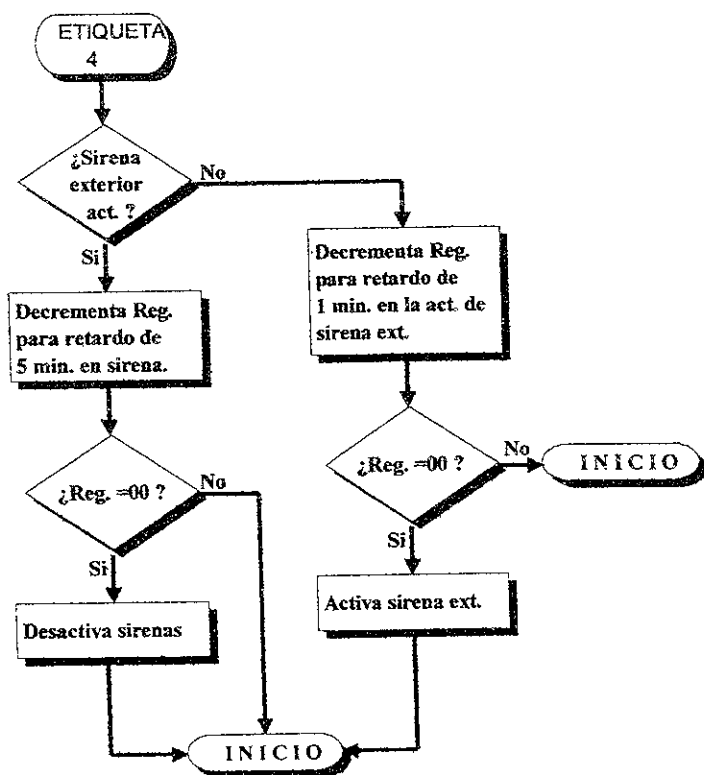


Figura 3.2.7 Cuarta parte del diagrama de flujo del programa

Como podemos observar en la fig. 3.2.3 el programa lo podemos dividir en siete etapas principales:

- Declaración de constantes y variables: consiste en la asignación de una cadena de caracteres a las localidades de memoria RAM que se usan en el programa. Esto se hace para llevar un mejor control y ordenamiento de dicha memoria.
- Configuración e inicialización de puertos y registros: los puertos bidireccionales se configuran como entrada o como salida según su uso en el programa. Para evitar conflictos con datos no deseados en puertos y algunos registros, estos se inicializan con valores requeridos.
- Inicialización y configuración del LCD: el "display" requiere de ciertos pasos y tiempos de espera antes de entrar en operación. La inicialización del LCD se describe en la sección 3.2.2.
- Configuración del "Timer-Autoreload" e interrupción por "Timer": el registro de carrera libre está configurado para activar una interrupción cuando se decrementa hasta cero, lo cual genera un pulso en Pin G3 (PG3) y un salto al vector de interrupciones (0FF H:hex), en donde se atiende dicha interrupción para a su vez

generar el sonido de alerta. El sonido de alerta se obtiene variando el valor que se almacena en el registro de carrera libre provocando que se tenga un pulso en TIO (PG3) a diferentes frecuencias. En esta misma sección del programa se controlan tiempos de retardo para activación de alarma exterior de un minuto y el tiempo de duración del sonido de alerta, así como la atención a interrupción por software la cual hace que se reinicie el programa.

- Limpia segmento de memoria RAM (00 - 0F Hex): es necesario limpiar la memoria RAM a partir de la localidad 00 Hex hasta 0F Hex, ya que el programa al momento de correr inicialmente requiere que este bloque de memoria no tenga datos almacenados. Lo anterior es debido a que se usan localidades de la RAM como registros de banderas para que en determinadas secciones del programa se ejecuten saltos o no se ejecuten, todo depende si determinada bandera en estos registros se encuentra activada o no.
- Inicialización de registros del "Timer": existe un registro de carrera libre de 16-bits y un registro de autocarga de 16-bits, este último tiene la función de autocargar al registro de carrera libre cuando se decremente hasta cero. Inicialmente se almacenan valores en ambos registros para que al momento de correr el registro de carrera libre se genere un primer pulso en PG3/TIO en un intervalo de tiempo proporcional al valor cargado en el registro de carrera libre, posteriormente se modifica constantemente el valor almacenado en el registro de autocarga con el fin de variar la frecuencia del pulso en PG3/TIO y de esta manera generar el sonido de alerta.
- Sección de subrutinas: el programa se compone de ocho subrutinas, tres principales, KEYSKAN, KEYS y DISPLAY; contenidas dentro de estas otras cinco subrutinas, BUSLCD, VERIF, RUTILIN, NOTECL y NOTECL1.

KEYSCAN:

Es la subrutina que se encarga de detectar las teclas oprimidas y leer el estado de los sensores, así como desplegar uno de los doce letreros cada veinte segundos si el microcontrolador no es interrumpido por teclas o sensores en un lapso de tiempo de veinte segundos. Los doce letreros que se despliegan están listados en la Tabla 3.2.3. La forma en que el microcontrolador lee el teclado y los sensores en los "puertos multiplexados", recorriendo un "0" en cada una de las terminales del puerto L para enseguida leer el puerto I verificando si existe un valor diferente a 0FF Hex, si es así quiere decir que una de las teclas o alguno de los sensores se accionó por lo que en ese momento se almacena en memoria el valor enviado por el puerto L y el valor recibido en el puerto I. Estos dos valores almacenados en memoria sirven para ejecutar dos saltos a diferentes direcciones en la subrutina KEYS lo cual da como resultado una dirección única perteneciente a la tecla o sensor que se accionó, de esta forma se llevan a cabo las diferentes funciones que tienen cada una de las teclas y sensores. En caso de que el sistema de seguridad sea activado mediante uno de los sensores y éste quede permanentemente accionado provoca que al teclear una secuencia numérica, refiriéndonos a un código de desactivación/activación, sea interrumpida, para evitar este suceso el programa sólo barre el teclado durante quince segundos continuos después de que se oprimió una de las teclas numéricas, de esta manera se tiene el tiempo suficiente como para teclear la contraseña y desactivar el sistema. Debido a que se manejan sensores NO y NC es necesario proteger al microcontrolador con diodos y una resistencia para impedir un corto circuito en las líneas que forman la matriz, o bien que se tengan lecturas erróneas en el puerto I. En la fig. 3.2.8 se muestra el diagrama de conexiones del

LCD, teclas y sensores con los puertos correspondientes. Al utilizar los Puertos Multiplexados (Puertos I y L) con sensores NC sólo se utilizan ocho, por lo que se tuvieron que usar seis líneas del puerto G para conectar más sensores NC y completar la cantidad necesaria para proteger la casa. Puesto que dos de los pines del puerto G se encuentran funcionando, en este caso, como salida de reloj (PG7/CKO) y como contador del "TIMER" (PG3/TIO), no se puede leer el puerto G en forma de "byte" por lo tanto los seis sensores colocados en el puerto G son leídos de forma independiente, preguntando por cada una de las líneas que se encuentran conectadas a los sensores. Cuando el programa se encuentra corriendo en esta sección de revisión del puerto G carga automáticamente un primer valor predeterminado por el programador en memoria RAM y si alguno de los sensores conectados al puerto G es activado entonces un segundo valor, predeterminado por el programador, es almacenado en memoria RAM; de esta manera se adquieren los dos valores necesarios para ejecutar los dos saltos en la subrutina KEYS. El diagrama de conexión de los seis sensores NC con el puerto G se muestran en la fig. 3.2.9.

El orden en que se realiza la revisión de teclas y sensores es el siguiente: teclas y sensores NO, sensores del puerto multiplexado NC y por último sensores colocados en el puerto G. Si no se detecta alguna tecla oprimida o algún sensor activado por un lapso de tiempo de veinte segundos, entonces se despliega uno de los doce letreros que se muestran en la Tabla 3.2.3 y así, cada veinte segundos si no ocurre la interrupción del microcontrolador por teclas o sensores se seguirá dando el ciclo de despliegue de letreros en forma continúa, aún si el sistema no se encuentra operando. En caso de que el sistema haya sido activado por uno de los sensores y se desactive la sirena en forma automática a los cinco minutos, no se desplegara ninguno de los doce letreros y permanecerá el "display" indicando cual de los sensores fue accionado hasta que el sistema sea desactivado.

Lapso de Tiempo (Seg.)	Letrero Desplegado en LCD
20	"UNAM, FACULTAD DE INGENIERIA"
40	"TESIS PROFESIONAL QUE PRESENTO"
60	"ROBERTO ROMERO Y ALFONSO ANTUNEZ"
80	"EL SISTEMA DETECTA LA VIOLACION"
100	"D'CADADA DISP. D'SEG. INSTALADO"
120	"UN SINCERO AGRADECIMIENTO A LAS"
140	"PERSONAS QUE COLABORARON PARA"
160	"QUE ESTE PROYECTO SE LLEVARA"
180	"A CABO, ING. MIGUEL A. CRUZ LEON"
200	"Y EL SR. RICARDO ROMERO ANGUIANO"
220	"**MUCHAS GRACIAS* A TODOS Y ESPEC-
240	"IALMENTE A MI MADRE, GRACIAS"

Tabla 3.2.3 Lista de letreros desplegados en la subrutina KEYSKAN

KEYS:

La subrutina KEYS es la que hace más uso de los registros de control, la que ordena al microcontrolador las funciones que debe tomar por medio de los registros, por lo tanto indica a las demás subrutinas que es lo que procede.

Con el primer valor almacenado en memoria RAM en la subrutina KEYS CAN se realiza un salto a una determinada sección del programa la cual es compartida con algunas otras teclas y sensores, el siguiente salto realizado mediante el segundo valor almacenado en memoria proporciona la dirección final en la que se realizan una serie de instrucciones que caracterizan a la tecla o sensor activado. Algunas de las instrucciones que se realizan en esta parte del programa son tales como: preguntar por banderas, activar o desactivar banderas, activar o desactivar el contador de carrera libre, poner o quitar el modo de operación del sistema, generar el retardo de un minuto para dar tiempo a que el usuario salga de la casa antes de que se active el sistema, indicar la dirección de inicio de la cadena de caracteres en ASCII para alguno de los letreros y hacer un salto a la página correspondiente (D-0900 a D-0E00 Hex) donde se encuentra dicha cadena de caracteres, o "letrero", para almacenar los datos en memoria RAM y que posteriormente sean desplegados por la subrutina "DISPLAY".

La forma en que se almacenan los caracteres ASCII en memoria RAM es la siguiente:

Primero con la directiva .BYTE se escribe con comillas el letrero que se quiere desplegar en el "display", esta directiva almacena los valores en hexadecimal de cada uno de los caracteres ASCII en localidades de memoria EPROM consecutivas. Considerando que una página está formada por 256 localidades de memoria EPROM (000-0FF Hex) y que el algoritmo para almacenar los caracteres ASCII en memoria RAM es máximo de 13 localidades de memoria EPROM, se deduce entonces que podemos contar con espacio de memoria EPROM por página de 243 localidades para los letreros. Cada una de las páginas asignadas para guardar los datos de los letreros contienen su propio algoritmo, el cual cumple con la función de direccionar el letrero seleccionado y almacenarlo en memoria RAM. Fue necesario repetir el mismo algoritmo en cada página por que la instrucción LAID que se usó para copiar los valores de los caracteres ASCII a la memoria RAM, carga los valores indirectamente pero sólo los correspondientes a la página de memoria EPROM a la que corresponde esta instrucción. Las páginas que se usaron para almacenar los letreros son la 0900 Hex, 0A00 Hex, 0B00 Hex, 0C00 Hex, 0D00 Hex y 0E00 Hex, incluyendo las dos subrutinas NOTECL y NOTECL1.

DISPLAY:

Se encarga de desplegar en la Pantalla de Cristal Líquido (LCD) la información almacenada en la memoria RAM, hecha por las subrutinas KEYS, NOTECL o NOTECL1, en forma de caracteres ASCII así como de limpiar o no el LCD según la información detectada en los registros de control.

BUSLCD:

Esta subrutina es la más sencilla de todas debido a que su única función es la de preguntar por el estado de la bandera BUSY que utiliza el LCD para indicar que no está listo para ejecutar el próximo comando hasta que la bandera BUSY esté en cero, es entonces cuando el programa sale de esta subrutina y continúa con la tarea que estaba haciendo.

VERIF:

Se encuentra dentro de la subrutina KEYSKAN y su función es desplegar un letrero, "ADELANTE PUEDE ACTIVAR EL SISTEMA", si es que el microcontrolador no detectó ninguno de los sensores activados después de haber oprimido cero y después ENTER, que es la clave para verificar que ninguno de los sensores se encuentre accionado antes de poner en operación el sistema de seguridad. En caso de que se detecte la activación de algún sensor, se acciona sólo la bocina interna del sistema y se despliega un letrero indicando cual de los sensores se encuentra operando.

RUTILIN:

Esta subrutina también se encuentra dentro de la KEYSKAN y la función que tiene es detectar sólo teclas que se oprimen en el tiempo que está activada la sirena e inhibir a todos los sensores por lo que no tiene sentido continuar revisando los sensores, si la sirena está activada. Otra de las funciones que tiene la subrutina RUTILIN es evitar tener dificultades al teclear una serie de números, como es el caso de la contraseña, ya que si alguno de los sensores que se encuentran en la matriz queda accionado puede ocasionar que se tengan lecturas de datos erróneas en el puerto I (Fig. 3.2.8). La manera en que detecta la tecla oprimida es preguntando bit por bit, en las localidades de memoria RAM donde se almacena el valor mandado por el puerto L y recibido por el puerto I.

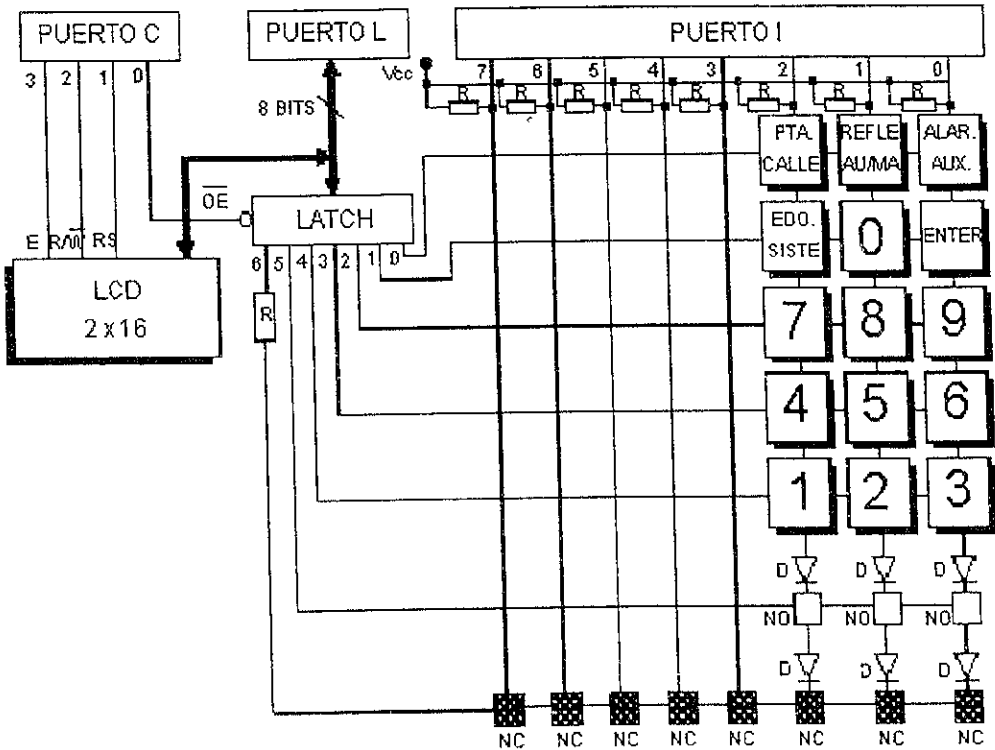


Figura 3.2.8 Diagrama de alambrado de teclado, LCD y algunos de los sensores

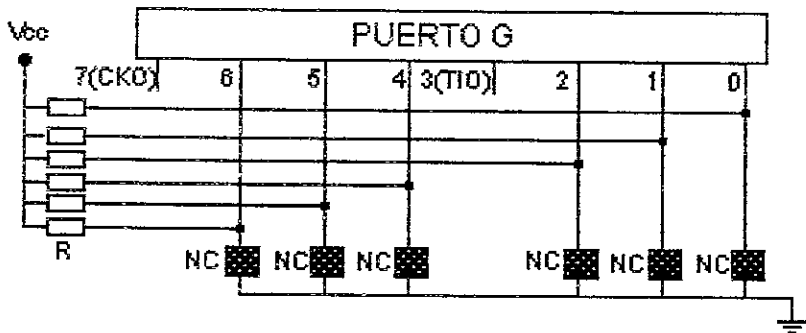


Figura 3.2.9 Diagrama del alambrado de los sensores en el puerto G

NOTECL y NOTECL1:

Estas dos subrutinas sirven para almacenar en memoria los caracteres ASCII de los doce letreros que se despliegan cuando el microcontrolador no es interrumpido por teclas o sensores. Si el microcontrolador no detecta variación alguna en las teclas o sensores, desde la subrutina KEYSKAN hace un salto a cualquiera de estas subrutinas para almacenar los caracteres en memoria RAM y posteriormente desplegarlos en el LCD.

3.2.2 Circuitos y elementos que forman el sistema de seguridad y alarma

Los circuitos que conforman el sistema de seguridad son: una fuente de poder de 12 V y 5V a 1.5 A, un autocargador para la batería de emergencia, una etapa de potencia para los altavoces, un circuito para el encendido automático de un foco y un circuito para iluminar el LCD. Los elementos que forman el sistema son 13 sensores magnéticos, 2 sensores de presencia, 1 sensor de vibración, un "display" de 16x2 caracteres, un teclado de 5x3, un conector de 60-pines, 90 m de cable plano telefónico de 6 hilos y 90 m de 4 hilos.

Autocargador de batería auxiliar

En la fig. 3.2.10 se muestra la fuente de poder de 15 V a 1.5 A y el autocargador de batería. Por medio del diodo D1 se suministra toda la corriente que demande el regulador LM7805, independientemente de que la batería demande poca o mucha corriente. Y por otro lado tenemos una fuente de corriente controlada por voltaje, la cual provee a la batería de una corriente de mantenimiento para que cuando el sistema carezca de su principal fuente de energía eléctrica, la batería se encuentre cargada y

pueda reemplazar a la fuente de poder principal (12 V y 1.5 A de la fuente por los 12 V y 1200 mAh de la batería).

El circuito que suministra la corriente de mantenimiento para la batería funciona de la siguiente manera: el transistor de potencia TIP30 se encuentra polarizado permanentemente mediante R1 de tal forma que mantenga una corriente I_{ec1} constante, para que la batería conserve su nivel de voltaje. Supongamos que la energía eléctrica de la casa, a la cual se encuentra conectado nuestro sistema, se desconecta por un instante y que el nivel de voltaje de la batería llega a ser menor a los 11 V, segundos antes de que vuelva la energía eléctrica a la casa, entonces en el momento en que el sistema recibe nuevamente energía de su fuente principal se le comienza a suministrar a la batería una corriente I_{ec1} máxima de 240 mA, la cual corresponde al 20 % de la corriente nominal de la batería, que se encuentra controlada por R2. En el momento en que el nivel de voltaje de la batería llegue a ser un poco mayor a los 11.7 V, de tal forma que polarice al diodo zener (11 V) y al transistor Q2, se hace circular una corriente I_{ec2} en el transistor Q2, logrando que se aumente el voltaje en el punto A y por lo tanto se llegue a tener una corriente I_{ec1} de aproximadamente 24 mA, la cual corresponde al 2% de la corriente nominal de la batería que ésta necesita para mantener su voltaje de 12 V.

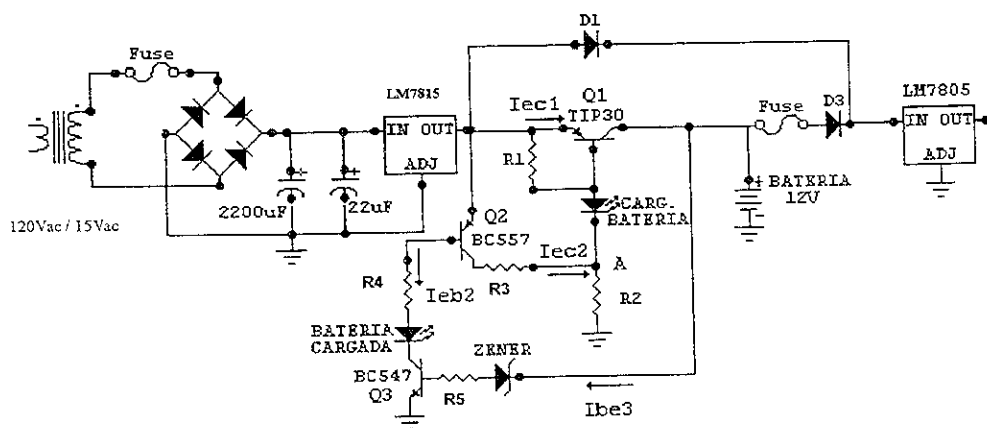


Figura 3.2.10 Circuito de fuente de poder y de autocargador de batería

Análisis de corrientes del circuito autocargador de batería

Para el análisis de corrientes de este circuito se considera un primer caso cuando la batería se encuentra descargada y un segundo caso cuando la batería se encuentra recargada

1er caso, batería sin carga:

Q3 APAC Q2 APAG. $V_{ee} = 15 \text{ V}$

$I_{cs1} = 20\%$ de corriente nominal de batería

$$I_{csc1} = (1200) * (0.2) = 240 \text{ mA}$$

; Donde I_{csc1} es la corriente de colector ;cuando la batería se encuentra sin carga.

$$I_{csc1} \sim I_{bs1} = (I_{bs1}) * (\beta_1) \gg I_{bs1} = I_{csc1} / \beta_1$$

; Donde $\beta_1 = 16$ para Q_1 .

$$I_{bs1} = 240 \text{ mA} / 16 = 15 \text{ mA}$$

; I_{bs1} es la corriente de base cuando la :batería se encuentra sin carga.

De malla 1 tenemos

$$V_{ee} = V_{R1} + V_{led} + (I_{bs1} * R_2)$$

; Donde $V_{ee} = 15V$, $V_{R1} = V_{eb1} = 0.7V$

$$R_2 = (V_{ee} - 1.4V) / I_{bs1} = 0.906 \text{ K}\Omega$$

$$R_2 \sim 1.0 \text{ K}\Omega$$

; Valor comercial de Resistencia ; 0.680 k Ω .

$$\gg V_A = V_{ee} - V_{eb1} - V_{LED}$$

$$V_A = 13.6 \text{ V}$$

; Voltaje en el nodo "A"

2do caso, batería con carga completa:

Para este caso se

Q3 ENC. Q2 ENC.

$I_{cc1} = 2\%$ de corriente nominal de batería

; Donde I_{cc1} es la corriente de colector ;cuando la batería se encuentra cargada

$$I_{cc1} = (1200) * (0.02) = 24 \text{ mA}$$

$$I_{cc1} \sim I_{bc1} = (I_{bc1}) * (\beta_1) \gg I_{bc1} = I_{cc1} / \beta_1$$

; Donde $\beta_1 = 16$ para Q_1

$$I_{bc1} = 24 \text{ mA} / 16 = 1.5 \text{ mA}$$

; Donde I_{bc} es la corriente de base cuando ;la batería se encuentra con carga.

En nodo "A " tenemos:

$$I_{c2} = I_{bs1} - I_{bc1}$$

; I_{c2} es la corriente de colector ; de Q_2 .

$$I_{c2} = 15 - 1.5 = 13.50 \text{ mA}$$

$$I_{b2} = I_{c2} / \beta_2$$

; I_{b2} es la corriente de base de ; Q_2 , $\beta_2 = 200$

$$I_{b2} = 13.50 / 200 = 67.50 \mu\text{A}$$

$$I_{b2} = I_{c3}$$

; Donde I_{c3} es la corriente de
; colector de Q_3

$$\gg I_{b3} = I_{c3} / \beta_3$$

; $\beta_3 = 150$ para Q_3

$$I_{b3} = 67.50 \mu\text{A} / 150 = 0.45 \mu\text{A}$$

; Es la corriente de base de Q_3

De malla II tenemos.

$$V_{ee} = V_{ec2} + I_{c2} \cdot R_3 + V_A$$

; Donde el $V_{ec2} = V_{sat} = 0.2 \text{ V}$

$$\gg R_3 = (15 \text{ V} - 0.2 \text{ V} - 13.6\text{V}) / 13.50 \text{ mA} = 0.088 \text{ K}\Omega$$

$$R_3 \sim 100 \Omega$$

De malla III tenemos:

$$V_{ee} = V_{eb2} + I_{b2} \cdot R_4 + V_{LED} + V_{ce3}$$

; Donde el $V_{ec3} = V_{sat} = 0.2 \text{ V}$
; $I_{b2} = 0.675 \text{ mA}$

$$R_4 = 13.4 \text{ V} / 0.0675 \text{ mA} = 198.5 \text{ K}\Omega$$

$$R_4 \sim 200 \text{ K}\Omega$$

De malla IV tenemos:

$$V_{BAT} = V_Z + I_{b3} R_5 + V_{be3}$$

$$15 \text{ V} = 11 \text{ V} + (0.45 \cdot R_4) + 0.7 \text{ V}$$

$$\gg R_5 = 3.3 \text{ V} / 0.45 \mu\text{A} = 7.3 \text{ M}\Omega$$

$$R_5 \sim 7.5 \text{ M}\Omega$$

Amplificador de audio para bocina y alta voz, así como control de altavoz

Al inicio de este capítulo se comentó algo a cerca del pulso que genera el contador de carrera libre (Registro CNTRL) cuando se decremента hasta cero. Este pulso es inyectado en la base del transistor Q_1 el cual trabaja en la región de corte y saturación, de V_{cc} al colector del transistor se encuentra conectada una pequeña bocina con la función de amplificar los trenes de pulsos que manda el microcontrolador al transistor con el fin de generar la señal de alerta o sirena. Con el potenciómetro P_1 se ajusta el nivel de volumen en la bocina, la cual se encuentra en el interior de la unidad de control

Para amplificar esta misma señal pero a un nivel de volumen más elevado en el altavoz exterior de la casa, es necesario utilizar un transistor tipo "darlington" y conectar el altavoz entre 12 V y el emisor de Q_4 para proporcionar una corriente alta. El bit 0 del puerto D (PDC) es usado por el microcontrolador para hacer sonar o no el altavoz, por

medio del tren de pulsos del TIO, según lo requiera el sistema de seguridad. Con el potenciómetro P2 se puede controlar el volumen del altavoz.

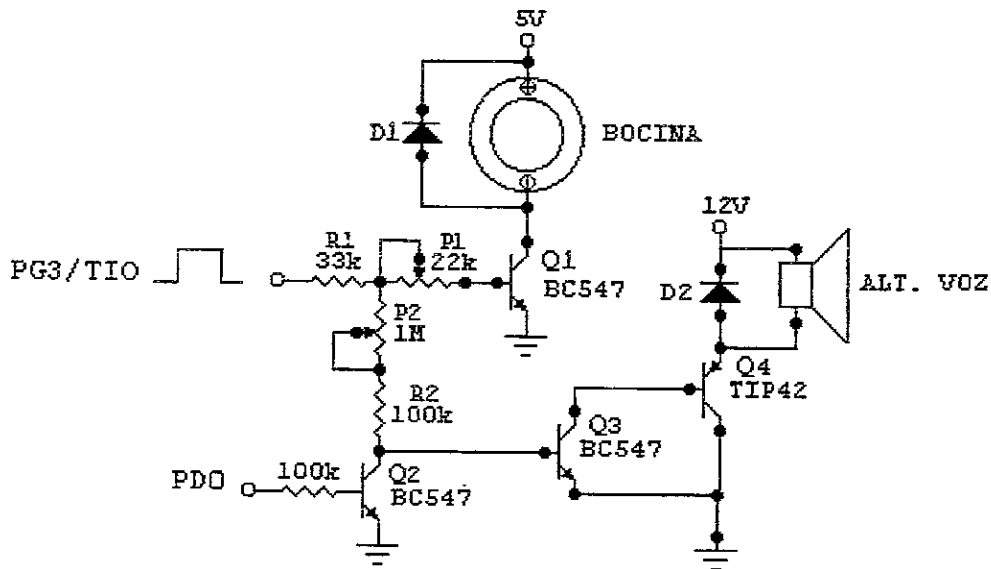


Figura 3.2.11 Circuito amplificador de audio para bocina y altavoz

Control de luz en LCD

Por medio del Pin D4 el microcontrolador realiza el encendido y apagado de la luz en el "display", de esta manera se puede visualizar lo que indica el "display", si se llega a activar el sistema por la noche o cuando se vaya la energía eléctrica de la casa. Con el potenciómetro se ajusta la intensidad de iluminación en el "display"; si el sistema se encuentra operando mediante la batería auxiliar se recomienda no dejar la luz del LCD encendida permanentemente ya que disminuirá rápidamente la carga de la batería.

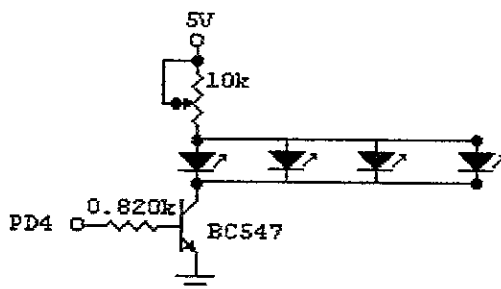


Figura 3.2.12 Circuito para iluminar LCD

Encendido automático de foco

El sistema cuenta con la función para hacer que un foco encienda automáticamente en el exterior de la casa cuando oscurece. Si esta opción se encuentra habilitada el foco puede encender automáticamente o de lo contrario sólo encenderá manualmente mediante su propio interruptor. El circuito está formado por un fototransistor que mide la intensidad de luz solar, un potenciómetro para ajustar la sensibilidad del fototransistor, un fotoacoplador (MOC3010) para separar la corriente alterna de la demás circuitería y para alimentar el foco un triac (TIC216). El colector del fototransistor es alimentado mediante el Pin D1 del microcontrolador con el propósito de usar este circuito cuando se le indique al sistema de seguridad.

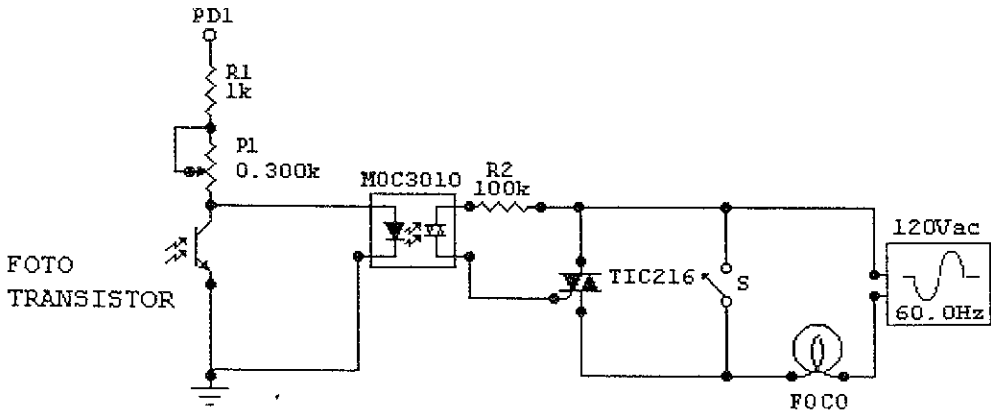


Figura 3.2.13 Circuito para encender el foco en forma automática al anochecer

Pantalla de Cristal Líquido (LCD)

En la Tabla 3.2.4 se muestran las funciones de los 14 pines con los que cuenta el LCD y en la fig. 3.2.14 se muestra, de izquierda a derecha respectivamente, como se ejecuta una instrucción (Limpiar pantalla), como se lee la bandera de ocupado (DB7) y la dirección donde se coloca el cursor (DB6-DB0) y por último la manera en que se escribe un carácter en pantalla

Pin No.	Conexión	Función
1	GND	0 V
2	Vcc	5 V
3	V _o	Contraste (0 a 5 V)
4	RS	"H" Entrada de Datos "L" Entrada de Comandos
5	R/W	"H" Lectura de Datos (Mod. » CPU) "L" Escritura de Datos (CPU » Mod.)
6	E	Señal para Habilitar la Esc. o Lect.
7	DB0	Bus de Datos
8	DB1	Bus de Datos
9	DB2	Bus de Datos
10	DB3	Bus de Datos
11	DB4	Bus de Datos
12	DB5	Bus de Datos
13	DB6	Bus de Datos
14	DB7	Bus de Datos

Tabla 3.2.4 Función de los pines del LCD

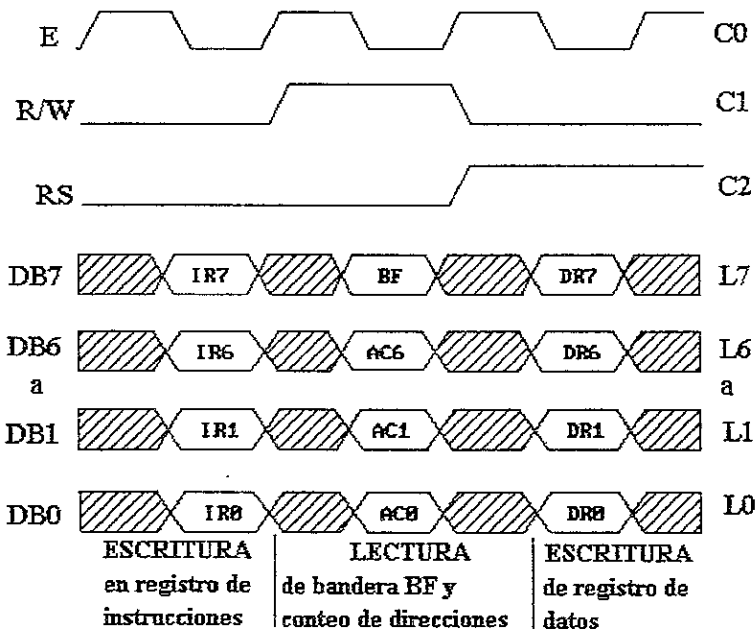


Figura 3.2.14 Diagrama de ciclos en la transferencia de datos

Existe un procedimiento de inicialización para el LCD, el cual sirve para darle un tiempo de arranque y de estabilidad al LCD antes de ejecutar alguna instrucción; también sirve para configurar la longitud de los datos en la interfaz (4-bits o 8-bits), para encender el "display", para limpiar el "display" y para seleccionar el desplazamiento de los caracteres o el desplazamiento del cursor. En la fig. 3.2.15 se muestran los pasos que hay que seguir para poner en operación el "display" de cristal líquido. Cabe mencionar que después de poner los modos en los que operara el "display" en esta etapa de inicialización, no podrán ser cambiados más adelante al menos que se vuelva a seguir un procedimiento semejante al de inicialización descrito en la figura antes citada.

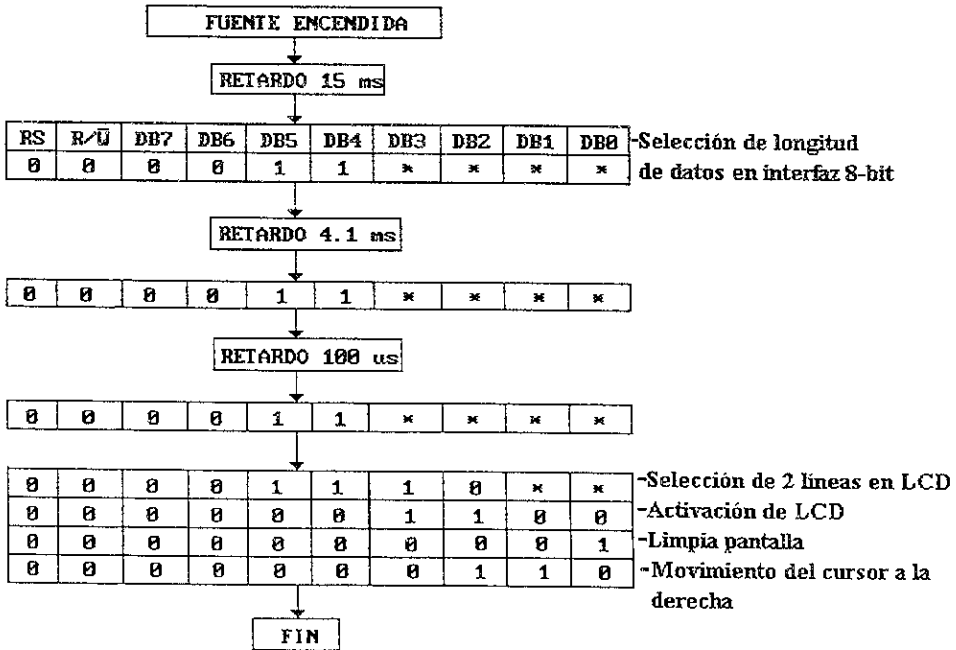


Figura 3.2.15 Procedimiento de inicialización del LCD

3.3 FORMA DE OPERAR el sistema DE SEGURIDAD

3.3.1 Introducción

El sistema de seguridad protege una casa hogar que consta de tres recámaras, dos salas, una cocina, tres baños y un estudio, todas estas áreas se encuentran protegidas mediante sensores magnéticos, sensores de vibración, sensores de ruptura y sensores de presencia. Por lo que se refiere a la unidad de control del sistema es muy fácil de operar ya que los comandos de operación que el usuario debe conocer para controlar el sistema son de oprimir una o dos teclas, excepto cuando se teclea la contraseña de activación/desactivación.

3.3.2 DESCRIPCIÓN DEL ÁREA A PROTEGER

En las figuras 3.2.1 y 3.2.2 se muestran los planos de la casa habitación para la planta baja y planta alta respectivamente. En la planta baja se tiene un sensor magnético, en la puerta de la calle un sensor magnético y un sensor de vibración, por si abren la ventana o rompen el vidrio del estudio, un sensor magnético en la puerta de entrada a la casa, otro más en la ventana del baño y por último un sensor de presencia que registra el movimiento del área formada por el comedor, sala y cocina.

Para una mejor cobertura del sensor de movimiento, se debe seleccionar el punto de montaje de tal manera que la dirección probable de movimiento del intruso atraviese el patrón de protección, además el sensor de movimiento cuenta con un ajuste vertical, con el cual se puede seleccionar el alcance de protección deseado (Tabla 3.3.1). En la figura 3.3.1 se muestra el patrón de protección que usa el sensor para detectar el movimiento, para un óptimo funcionamiento del sensor no se debe instalar donde quede expuesto a la luz solar directa, o por encima de fuentes de calor; en zonas donde existan objetos que puedan producir cambios rápidos de temperatura como calefactores.

Altura de Montaje	ALCANCE DE LA PROTECCION			
	6m	9m	12m	15m
2.5m	7.5	5.5	3	2.5
2.4m	7	5	3	2.5
2.1m	5.5	4	2.5	2
1.8m	4.5	3.5	2	1.5

Ajuste del patrón vertical

Tabla 3.3.1 Guía de instalación.

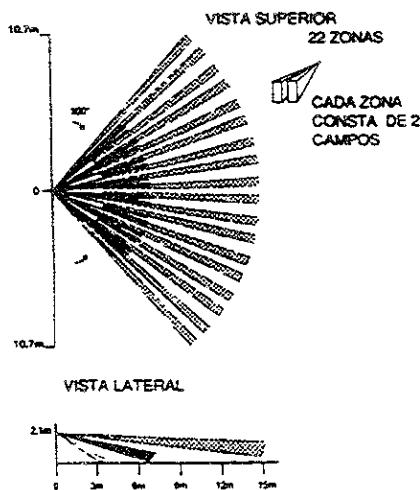


Figura 3.3.1 Patrón de protección, lente estándar

La planta alta está protegida por nueve sensores magnéticos colocados entre las diferentes puertas y ventanas, dos sensores de ruptura colocados en los domos, un sensor antisabotaje colocado en la sirena y un sensor de presencia colocado en la sala, con el cual se protege el área de la sala y una parte de las escaleras.

Cabe mencionar que para los cuatro sensores utilizados en la parte trasera de la casa se usó una sola línea, es decir los tres sensores magnéticos y el de ruptura se encuentran en serie desplegándose por lo tanto un sólo letrero por los cuatro sensores, esto se debió a que la arquitectura de diseño del sistema no permitió usar más de 14 líneas de control para sensores NC, un sensor por línea y un letrero por línea.

Se pudieron haber usado las tres líneas que controlan los sensores NO, para registrar de forma independiente los cuatro sensores de la parte trasera de la casa y desplegar un letrero de aviso por cada sensor. No se hizo de esta manera puesto que no se pudieron conseguir los sensores magnéticos NO debido a que son menos comerciales que los sensores NC, dado que tienen la desventaja de ser desactivados con tan sólo cortar los cables que los conectan a la unidad de control.

3.3.3 Operación de la unidad de control

La forma en la que se controla el sistema de seguridad es muy sencilla ya que la mayoría de las funciones son ejecutadas con tan sólo oprimir la tecla indicada además de que la función se describe abajo de cada una de las teclas, tales como:

♦ PUERTA DE CALLE OPERANDO/NO-OPERANDO:

Debido a que el usuario del sistema de seguridad sugirió que en ocasiones se requería que la puerta de la calle quedara desactivada mientras que todos los demás sensores debían de operar en forma normal, se creó esta tecla para controlar la operación de dicha puerta. Si se requiere que la puerta opere, entonces se oprime una vez la tecla y en el "display" se observa el letrero

"PUERTA DE CALLE OPERANDO", por lo tanto, si el sistema se pone en operación en este momento y si se abre la puerta de la calle entonces se activará la alarma. Volviendo a oprimir la tecla se despliega el letrero "PUERTA DE CALLE NO-OPERANDO", entonces no importa si el sistema está operando la sirena no sonará si se abre la puerta de la calle. Esta puerta se puede activar o desactivar esté operando o no el sistema.

◆ **REFLECTOR AUTO/MANUAL:**

Con esta tecla se selecciona si se quiere que el foco del balcón encienda de manera automática cuando se oculte el sol, o de una forma manual con el interruptor.

◆ **ALARMA AUXILIAR:**

El sistema cuenta con una tecla auxiliar cuya única función es la de hacer sonar la sirena de forma inmediata sin tomar en cuenta el estado de operación del sistema. Generalmente se usa esta tecla para ahuyentar a algún intruso que ronde la casa, para casos de emergencia o bien para probar la sirena.

◆ **ESTADO DE OPERACION DEL SISTEMA:**

Al oprimir esta tecla nos muestra en el "display" el estado actual del sistema de seguridad, es decir, si el sistema se encuentra operando en ese momento o no se encuentra operando, también nos indica si el sistema se encuentra operando en modo de noche o normal.

◆ **ENTER:**

Esta tecla sirve para indicarle al sistema que se ha tecleado un número o una serie de números y que los debe comparar para dar continuidad a una acción, por ejemplo: "1" y después "ENTER" » "enciende luz del display".

◆ **0,1,2,3,4,5,6,7,8,9:**

El sistema cuenta con un teclado numérico el cual sirve para teclear una serie de números que le indican al sistema la acción que debe ejecutar, como son:

"0" y después "ENTER"; este código se tecldea generalmente antes de poner en operación el sistema, para evitar falsas alarmas y lo que hace el sistema es verificar que todos los sensores se encuentren cerrados, si es así se despliega un letrero en el "display" "ADELANTE, PUEDE ACTIVAR EL SISTEMA".

y después "ENTER"; después de cerciorarse de que los sensores no están accionados con el código "0" y después "ENTER", se tecldean los cinco dígitos de seguridad para poner en operación el sistema. Los mismos cinco dígitos son usados para desactivar el sistema.

"1" y después "ENTER"; este código sirve para encender la luz del "display", cuando sea necesario y poder leerlo sin problema. Oprimiendo nuevamente el mismo código se apagará la luz.

3.3.4 Modos de operación del sistema

El sistema cuenta con tres modos de operación:

A) Modo no-operando

En este modo el sistema sólo detecta las líneas de antisabotaje que son tres, una línea antisabotaje para cada uno de los sensores de presencia y otra línea antisabotaje para la sirena exterior.

B) Modo operando

para que el sistema opere en este modo, la casa debe de estar vacía dado que todos los sensores se encuentran en estado de alerta. En este modo de operación el sistema hace uso de un retardo de un minuto para darle tiempo al usuario de que salga de la casa después de programar el sistema de seguridad.

C) Modo de noche

este modo de operación generalmente se usa en la noche, cuando todos se van a dormir, y cierran todas las puertas y ventanas, puesto que el sistema desactiva los dos sensores de presencia y deja a todos los demás sensores en estado de alerta, de tal forma que las personas que habitan la casa pueden pasear por el interior de ésta sin que suene la sirena. Para que el sistema de seguridad opere en este modo, se teclea un código de seguridad de cinco dígitos diferentes al Modo operando.

Es importante señalar que cuando se activa alguno de los que se encuentran en: la puerta de calle, puerta de patio, sensor de presencia planta baja y planta alta, primero se activa la bocina interna, que se encuentra en el interior de la unidad de control, y después de un minuto, si no se ha desactivado el sistema, comienza a sonar el altavoz exterior. Esto se hizo con el propósito de que el usuario tenga un lapso de tiempo para desactivar el sistema de seguridad cuando llegue a su casa.

3.3.5 Características eléctricas del sistema.

El sistema de seguridad consume una corriente mínima de 60 mA cuando no se encuentran activadas ninguna de las dos sirenas, cuando la batería auxiliar se encuentra cargada y cuando no está encendida la luz del LCD. El consumo de corriente debido al LCD, sirena interior y sirena exterior es el siguiente.

La luz del LCD consume una corriente de 105 mA con una intensidad de luz regulada a la que se pueden leer los letreros en el LCD, la bocina interior consume una corriente de 23 mA cuando se activa por el sistema y la bocina exterior consume una corriente de 158 mA cuando se activa por el sistema.

Conociendo la corriente de consumo para cada uno de los casos podemos calcular el tiempo que puede permanecer el sistema funcionando usando sólo la energía de la batería auxiliar de 12 V a 1200 mAh.

* Para el caso en que la sirena no llega a activarse, el tiempo en horas es:

$$(1200 \text{ mAh}) / (60 \text{ mA}) = \underline{20 \text{ h}}$$

* Para este otro caso se consideran tres activaciones de la sirena por X razones, por lo tanto la duración de la batería es el siguiente:

Considerando que la sirena exterior tiene un tiempo de duración de 5 min. por activación, entonces tenemos un total 15 min., el tiempo que permanece la sirena encendida y tomando en cuenta también que el consumo de corriente por las dos bocinas este tiempo es de 181 mA, tenemos que

$$(181 \text{ mA})(0.25 \text{ h}) = 45.25 \text{ mAh} ; \text{ lo que consume el sistema en los 15 min.}$$

$$1200 \text{ mAh} - 45.25 \text{ mAh} = 1154.75 \text{ mAh} ; \text{ la carga restante en la batería.}$$

$$(1154.75 \text{ mAh}) / (60 \text{ mA}) = 19.24 \text{ h} ; \text{ el tiempo que le resta a la batería después de los 15 min. de carga.}$$

$$19.24 \text{ h} + 0.25 \text{ h} = \underline{19.5 \text{ h}} ; \text{ es el tiempo de duración de la batería después de mantener una carga de 181 mA durante 15 min.}$$

Por lo que se refiere a la corriente con la que se carga la batería, después de ser descargada casi en su totalidad por el sistema (3 V en los polos de la batería), inicialmente la batería demanda una corriente aproximada de 100 mA durante las primeras seis horas, para las siguientes seis horas la corriente disminuye hasta los 50 mA, y por último, durante otras doce horas la corriente tiende a disminuir hasta los 25 mA, manteniendo la batería esta corriente de carga en forma constante.

3.4 PROGRAMA FINAL DEL SISTEMA DE SEGURIDAD

NSC ASMCOP, Versión 4.6 (Nov 29 09:47 1995) ALARMA

26-Mar-99 14:37

```

1
2
3
4
5
6
7
8
9
10
11 0000
12
13
14
15
16
17 0000
18 0001
19 0002
20 0003
21 0004
22 0005
23 0006
24 0007
25 0008
26 0009
27 000A
28 000B
29 000C
30 000D
31 000E
32 000F
33 0010
34 0062
35 0063
36 0064
37 0065
38
39
40
41 00F0
42 00F2
43 00F3
44 00F4
45 00F6
46 00F9
47 00FA
48 00FB
49 00FF
50
51 0000
52 0003
53 000D
54 000B
55 0009
56
57 0000
58 0001
59
60
61 0000
62 0001
63 0002
64 0003
65 0004
66 0005
67 0006
68 0007

```

* ESTE PROGRAMA BARRE UNA MATRIZ DE 7X8 Y DESPLIEGA UN LETRERO *
 * CORRESPONDIENTE A LA TECLA QUE SE OPRIMIO, O SENSOR ACTIVADO., EN UN LCD *
 * AL MISMO TIEMPO, UN VALOR ES CARGADO DENTRO DEL REGISTRO DE *
 * CORRIMIENTO ("TIMER") PARA GENERAR UNA SEÑAL "PWM" LA CUAL ES USADA *
 * POR UN ALTAVOZ. TAMBIEN ES USADO EL PUERTO G PARA COLOCAR 6 SENSORES *

```

TITLE ALARMA
INCLD COP880.INC
SECT MAIN,ROM,ABS=0

RAM, DECLARACION N DE LOCALIDADES

066 - 06F espacio reservado para "Stack Pointer".
010 - 051 espacio reservado para caracteres de letreros.
DIGIT = 000

ILINE = 001
LLINE = 002
ILANT = 003
LLANT = 004
RAMSON = 005
DIG0 = 006
DIG1 = 007
DIG2 = 008
DIG3 = 009
DIG4 = 00A
RAMBIT1 = 00B
RAMBIT = 00C
ANTER = 00D
TEMP = 00E
DATO = 00F
LETRERO = 010
RECODI = 062
RAMGEN = 063
GUARDA = 064
GUARDB = 065

DECLARACION DE REGISTROS Y CONSTANTES

RETAR = 0F0 , y 0F1 usados para generar retardos
CONT = 0F2
RET60SG = 0F3 ,para retardo de 60 seg
RETINICI = 0F4 ,para retardo inicial, y 0F5
NOACCI = 0F6 ,para retardo de letreros.(0F6-0F8 Hex)
REG5MIN1 = 0F9 ,para retardo de 5 m. n.
REG5MIN2 = 0FA
RETTCL1 = 0FB
RETTCL2 = 0FF

E = 000 ,bit para generar pulsos de reloj para LCD
OE = 003 ,bit para controlar el "latch" HAB /DESHAB.
ESCR1 = B'1101 ,PORTC. OE, RS, R/W, y E respectivamente
LEER = B'1011 ,dato para leer el LCD
NUDIR = B'1001 ,dato para direccionar cursor en LCD

SIRENA = 0 ,PD0 control de altavoz para sirena
REFLEC = 1 ,PD1 control de encendido automático de foco

RAMBIT, 1 Registro de condiciones

PTA = 0 ,activa y desactiva la acción de PTA-calle
CODI = 1 ,para uso en teclas de n meros
REPETI = 2 ,para oprimir varias veces una tecla
CINCMIN = 3 ,para conteo de 5 min en act-alarma
LIMLCD = 4 ,para limpiar LCD
DESALAR = 5 ,desactivación de alarma a los 5 m n
NOTFCLA = 6 ,despliega letreros "no-tecla y no sensor
INDIACI = 7 ,para generar un tono poque o

```

```

59                                     ;para guardar el estado de la bandera de acarreo.
70                                     ;*
71                                     RAMBIT1, 2 Registro de condiciones
72 0000 ACARR1 = 0
73 0001 NOIDENT = 1 ;indica que no se detecto tecla, C/TRUN=1.
74 0002 PARAREL = 2 ;indica el paro de "TIMER" x corto tiempo.
75 0003 DESXTIE = 3 ;indica la desactivación de la alarma a los 5 min.
76 0004 PREXSEN = 4 ;si detecta sens. abierto sólo activa sirena interior.
77 0005 DESBIT4 = 5 ;lo habilitan los sensores NO.
78 0006 DENOCHHE = 6 ;anuncia alarma de noche
79 0007 PRACTI = 7 ;se activa cuando se hace el primer ciclo
80 0007 BUFL = 7 ;de barrido de los sensores.
81 00FC MINFREQ = 0FC ;para leer bandera "BUSY" de LCD
82                                     ;
83                                     ;* CONFIGURACION E INICIALIZACION DE PUERTOS I/O Y REGISTROS
84                                     ;
85 0000 INICIO:
86 0000 DD6F 3 LD SP, #06F ;inicialización de "STACK POINTER"
87 0002 BCD1FF 3 LD PORTLC, #0FF ;PORTL-SALIDA, parcial
88 0005 BCD90F 3 LD PORTCC, #00F ;PORTC-SALIDA, permanente
89 0008 BCD508 3 LD PORTGC, #008 ;PORTG BIT 0,1,2,4,5,6, entradas permanentemente.
90 0008 BCD0C1 3 LD PORTD, #B'00000001 ;Luz LCD apagada., Sirena ext. desconectada.
91 000E BDD46B 4 RBIT TIO, PORTGD ;inicialización de pin-TIO para sirena.
92                                     ;
93                                     ;* INICIALIZACION DE "DISPLAY" LCD CON UN RETARDO TOTAL DE 4.2 ms.
94                                     ;
95 0011 DED8 3 LD B, #PORTCD
96 0013 9E09 2 LD [B], #NUDIR
97 0015 BCD038 3 LD PORTLD, #038 ;longitud de datos de 8-bits.
98 0018 68 1 RBIT E, [B] ;ejecuta 1/2 ciclo de reloj para LCD.
99 0019 78 1 SBIT E, [B] ;ejecuta el otro 1/2 ciclo de reloj.
100 001A D0FF 3 LD RETAR, #0FF ;***** RETARDO DE 4.1 ms. *****
101 001C D104 3 LD RETAR+1, #004
102 001E C0 3 DECRE1: DRSZ RETAR
103 001F FE 3 JP DECRE1
104 0020 C1 3 DRSZ RETAR+1
105 0021 FC 3 JP DECRE1
106 0022 BCD038 3 LD PORTLD, #038
107 0025 68 1 RBIT E, [B]
108 0026 78 1 SBIT E, [B]
109 0027 D015 3 LD RETAR, #015 ;***** RETARDO DE 100 us *****
110 0029 C0 3 DECRE2: DRSZ RETAR
111 002A FE 3 JP DECRE2
112 002B BCD038 3 LD PORTLD, #038
113 002E 68 1 RBIT E, [B]
114 002F 78 1 SBIT E, [B]
115 0030 3072 5 JSR BUSLCD
116 0032 9E09 2 LD [B], #NUDIR
117 0034 BCD038 3 LD PORTLD, #038
118 0037 68 1 RBIT E, [B]
119 0038 78 1 SBIT E, [B]
120 0039 3072 5 JSR BUSLCD
121 003B 9E09 2 LD [B], #NUDIR
122 003D BCD00F 3 LD PORTLD, #00F ;encendido de LCD
123 0040 68 1 RBIT E, [B]
124 0041 78 1 SBIT E, [B]
125 0042 3072 5 JSR BUSLCD
126 0044 9E09 2 LD [B], #NUDIR
127 0046 BCD001 3 LD PORTLD, #001 ;limpia LCD
128 0049 68 1 RBIT E, [B]
129 004A 78 1 SBIT E, [B]
130 004B 3072 5 JSR BUSLCD
131 004D 9E09 2 LD [B], #NUDIR
132 004F BCD006 3 LD PORTLD, #006 ;config. de movimiento de cursor a la der.
133 0052 68 1 RBIT E, [B] ;en el LCD.
134 0053 78 1 SBIT E, [B]
135 0054 3072 5 JSR BUSLCD
136 ;
137 ;* CONFIG. DEL "TIMER-AUTORELOAD" CON FLANCO EN TIO(PG3) Y HAB. INT. POR "TIMER"
138 ;
139 0056 BCEEA0 3 LD CNTRL, #0AD
140 0059 BCEF11 3 LD PSW, #011
141 ;
142 ;* INICIALIZACION DE LOCALIDADES DE LA RAM CORRESPONDIENTES A LOS BITS

```

```

143                                     DE COMPARACION
144
145 005C 5C      1      LD  B, #LANT
146 005D                CLEAR
147 005D 9A00    2      LD  [B+], #00
148 005F 40      1      IFBNE #00
149 0060 FC      3      JP   CLEAR
150
151                *      INICIALIZACION DE SECCION DEL "TIMER"      *
152
153 0061 DEEA    3      LD  B, #TMRLO
154 0063 9A1E    2      LD  [B+], #01E
155 0065 9A00    2      LD  [B+], #000
156 0067 9A1E    2      LD  [B+], #01E
157 0069 9E00    2      LD  [B], #000
158
159                ***** SECCION DE SUBROUTINAS *****
160
161 006B                TOP
162 006B 31AE    5      JSR  KEYSKAN      ;detecta la tecla oprimida.
163 006D 33E7    5      JSR  KEYS      ;identifica tecla oprimida y ejecuta función
164 006F 3081    5      JSR  DISPLAY    ;despliega letrero correspondiente a tec. o sensor
165 0071 F9      3      JP   TOP
166
167                *      SUBROUTINA DE CHEQUEO DE BANDERA "BUSY-DISPLAY"      *
168
169 0072                BUSLCD:
170 0072 9E0B    2      LD  [B], #LEER
171 0074 BCD100  3      LD  PORTLC, #00
172 0077 68      1      ESPERA: RBIT E, [B]
173 0078 78      1      SBIT E, [B]
174 0079 BDD277  4      IFBIT BUFL, PORTLP
175 007C FA      3      JP   ESPERA
176 007D BCD1FF  3      LD  PORTLC, #0FF
177 0080 8E      5      RET
178
179                *      SECCION DE ACTUALIZACION DEL "DISPLAY"      *
180
181 0081                DISPLAY
182 0081 BD0C76  4      IFBIT NOTECLA, RAMBIT
183 0084 13      3      JP   SALT
184 0085 BD0C72  4      IFBIT REPETI, RAMBIT
185 0088 09      3      JP   NTEMP
186 0089 9D0E    3      LD  A, TEMP      ;pregunta si fue el mismo sensor, o
187 008B BD0F82  4      IFEQ A, DATO      ;tecla, la que se acciono.
188 008E 8E      5      RET
189
190 008F 9C0F    3      X   A, DATO      ;almacena etiqueta de tec o sen en RAM
191 0091 06      3      JP   SALT
192
193 0092                NTEMP      ;limpia reg donde se almacena etiqueta
194 0092 BC0F00  3      LD  DATO, #00      ;de sensor o tecla
195 0095 BD0C6A  4      RBIT REPETI, RAMBIT
196 0098 BDD87B  4 SALT      SBIT OE, PORTCD      ;latch desconectado
197 009B DC10    3      LD  X, #LETRERO      ;el reg X es direccionado en donde
198 009D DED8    3      LD  B, #PORTCD      ;inicia el letrero próximo a desplegar,
199 009F BD0C71  4      IFBIT CODI, RAMBIT      ;el cual se encuentra almacenado en la RAM
200 00A2 D1      3      JP   DIG0?
201 00A3 05      3      JP   LIMPIA
202 00A4                DIG0?
203 00A4 BD0C74  4      IFBIT LIMLCD, RAMBIT
204 00A7 01      3      JP   LIMPIA
205 00A8 0C      3      JP   OTROCA
206 00A9 BD0C6C  4 LIMPIA  RBIT LIMLCD, RAMBIT      ;limpia LCD
207 00AC 9E09    2      LD  [B], #NUDIR
208 00AE BCD001  3      LD  PORTLD, #001
209 00B1 68      1      RBIT E, [B]
210 00B2 78      1      SBIT E, [B]
211 00B3 3072    5      JSR  BUSLCD
212 00B5                OTROCA      ;despliega de carácter
213 00B5 BA      3      LD  A, [X+]
214 00B6 977C    2      IFEQ A, #07C      ;si el carácter a desplegar es '
215 00B8 8E      5      RET      ;(07C Hex), entonces finaliza letrero
216 00B9 9E0D    2      LD  [B], #ESCR1

```

```

217 00BB 9CD0 3 X A, PORTLD
218 00BD 68 1 RBIT E, [B]
219 00BE 78 1 SBIT E, [B]
220 00BF 3072 5 JSR BUSLCD
221 00C1 9E08 2 LD [B], #LEER
222 00C3 BCD100 3 LD PORTLC, #00
223 00C6 BC0000 3 LD DIGIT, #00
224 00C9 9DD2 3 LD A, PORTLP
225 00CB 9210 2 IFEQ A, #D10
226 00CD BC00C0 3 LD DIGIT, #0C0
227 00D0 9D00 3 LD A, DIGIT
228 00D2 BCD1FF 3 LD PORTLC, #0FF
229 00D5 9200 2 IFEQ A, #00
230 00D7 20B5 3 JMP OTROCA
231 00D9 9E09 2 NUEVD1: LD [B], #NUDIR
232 00DB 9CD0 3 X A, PORTLD
233 00DD 68 1 RBIT E, [B]
234 00DE 78 1 SBIT E, [B]
235 00DF 3072 5 JSR BUSLCD
236 00E1 20B5 3 JMP OTROCA
237 ;
238 ; ***** SECCION DE SERVICIO A INTERRUPCION *****
239 ;
240 00FF ;=00FF
241 00FF BDEF75 4 IFBIT TPND, PSW
242 0102 3138 5 JSR TIMER ;int. por "TIMER"
243 0104 31A7 5 JSR SOFINT ;int. por Software
244 0106 9D65 3 LD A, GUARDB ;retorno de subrutina "TIMER"
245 0108 9CFE 3 X A, B
246 010A 9D64 3 LD A, GUARDA
247 010C BD0C75 4 IFBIT DESALAR, RAMBIT
248 010F 13 3 JP DESACTI
249 0110 BD0C77 4 IFBIT INDIACT, RAMBIT
250 0113 04 3 JP UNAVEZ
251 0114 BDEE7C 4 SBIT TRUN, CNTRL
252 0117 8F 5 RETI
253 0118 9DF1 3 UNAVEZ: LD A, RETAR+1 ;genera un tono peque que sólo indica
254 011A 9201 2 IFEQ A, #001 ;la activación o desactivación del
255 011C 0F 3 JP SENAL ;sistema
256 011D 9D64 3 LD A, GUARDA
257 011F BDEE7C 4 SBIT TRUN, CNTRL
258 0122 8F 5 RETI
259 0123 BD0C6D 4 DESACTI: RBIT DESALAR, RAMBIT ;inicia la desactivación del "TIMER"
260 0126 BD0B7B 4 SBIT DESXIE, RAMBIT
261 0129 BD0B6D 4 RBIT DESBIT4, RAMBIT
262 012C 9D64 3 SENAL: LD A, GUARDA
263 012E BDDC78 4 SBIT SIRENA, PORTD ;desactiva sirena
264 0131 BDEE6C 4 RBIT TRUN, CNTRL
265 0134 BD0C6F 4 RBIT INDIACT, RAMBIT
266 0137 8F 5 RETI
267 0138 ;TIMER:
268 0138 BDEE6C 4 RBIT TRUN, CNTRL
269 013B BDEF6D 4 RBIT TPND, PSW ;deshabilitación de int. por "TIMER"
270 013E 9C64 3 X A, GUARDA
271 0140 9DFE 3 LD A, B
272 0142 9C65 3 X A, GUARDB
273 0144 BDDC70 4 IFBIT SIRENA, PORTD
274 0147 09 3 JP COMPARA
275 0148 BD0C73 4 IFBIT CINCMIN, RAMBIT
276 014B 0F 3 JP PROSI
277 014C D9BF 3 LD REG5MIN1, #0BF
278 014E DA01 3 LD REG5MIN2, #001
279 0150 0A 3 JP PROSI
280 0151 9D0E 3 COMPARA: LD A, TEMP
281 0153 BD0D82 4 IFEQ A, ANTER ;verifica si es el mismo sensor accionado.
282 0156 04 3 JP PROSI
283 0157 9CD0 3 X A, ANTER
284 0159 D330 3 LD RET60SG, #030
285 015B ;PROSI:
286 015B 5A 1 LD B, #RAMSON ;sección para generar tono de sirena.
287 015C 9DED 3 LD A, TAUHI
288 015E 9303 2 IFTG A, #00
289 0160 06 3 JP MASRAP
290 0161 77 1 IFBIT 7, [B]

```

```

291 0162 0A      3      JP   TM
292 0163 AE      1      LD   A, [B]
293 0164 8A      1      INC  A
294 0165 A6      1      X    A, [B]
295 0166 8D      5      RETSK
296 0167 71      1  MASRAP  IFBIT 1, [B]
297 0168 04      3      JP   TM
298 0169 AE      1      LD   A, [B]
299 016A 8A      1      INC  A
300 016B A6      1      X    A, [B]
301 016C 8D      5      RETSK
302 016D 9E00    2  TM:   LD   [B], #00          ;inicia el siguiente ciclo del tono
303 016F DEEC    3      LD   B, #TAULO
304 0171 AE      1      LD   A, [B]
305 0172 93FE    2      IFGT A, #0FE
306 0174 04      3      JP   DONE1
307 0175 9409    2      ADD  A, #009
308 0177 A6      1      X    A, [B]
309 0178 8D      5      RETSK
310 0179 9A1E    2  DONE1: LD   [B+], #01E
311 017B BDDC77  4      IFBIT INDIACT, RAMBIT
312 017E 01      3      JP   INDIACT
313 017F 02      3      JP   ALMOME
314 0180 C1      3  INDICA: DRSZ  RETAR+1
315 0181 8D      5      RETSK
316 0182 AE      1  ALMOME: LD   A, [B]
317 0183 9401    2      ADD  A, #001
318 0185 A6      1      X    A, [B]
319 0186 9306    2      IFGT A, #006
320 0188 01      3      JP   DONE2
321 0189 8D      5      RETSK
322 018A 9B00    2  DONE2: LD   [B-], #00
323 018C 9E1E    2      LD   [B], #01E
324 018E BDDC70  4      IFBIT SIRENA, PORTD
325 0191 0F      3      JP   DECRE
326 0192 C9      3      DRSZ  REG5MIN1          ;genera retardo para contar los 5 min
327 0193 09      3      JP   SETBAN          ;que debe durar como máximo el "TIMER".
328 0194 CA      3      DRSZ  REG5MIN2
329 0195 07      3      JP   SETBAN
330 0196 BDDC7D  4      SBIT  DESALAR, RAMBIT ;indica la finalización del conteo de los 5 min
331 0199 BDDC6B  4      RBIT  CINCMIN, RAMBIT
332 019C 8D      5      RETSK
333 019D BDDC7B  4  SETBAN  SBIT  CINCMIN, RAMBIT
334 01A0 8D      5      RETSK
335 01A1 C3      3  DECRE   DRSZ  RET60SG          ;genera retardo de 1 min. para después
336 01A2 8D      5      RETSK          ;activar la sirena exterior
337 01A3 BDDC68  4      RBIT  SIRENA, PORTD
338 01A6 8D      5      RETSK
339
340 01A7          SOFINT
341 01A7 BCEF00   3      LD   PSW, #00
342 01AA DD6F     3      LD   SP, #06F
343 01AC 2000     3      JMP  INICIO
344
345          *  ESTA SECCION BARRE LA MATRIZ DE 7x8 EL PUERTO LES RECORRIDO, *
346          *  EN FORMA SECUENCIAL, POR UN CERO Y EL PUERTO DE ENTRADA I ES *
347          *  REGISTRADO PARA DETECTAR SI SE ACTIVO ALGUN INTERRUPTOR SI SE *
348          *  ACTIVO ALGUNO DE LOS INTERRUPTORES SE ALMACENA EL DATO MANDADO POR *
349          *  EL PUERTO L Y EL DATO RECIBIDO POR EL PUERTO I PARA POSTERIORMENTE *
350          *  USARLOS Y ACCESAR A LA TECLA CORRESPONDIENTE TAMBIEN SE VERIFICAN *
351          *  6 LINEAS DEL PUERTO G EN LAS CUALES SE ENCUENTRAN CONECTADOS OTROS *
352          *  SENSORES *
353
354 01AE          KEYSKAN
355 01AE BCD800   3      LD   PORTCD, #00          ;latch activado
356 01B1 D6DF     3      LD   NOACCI, #0DF
357 01B3 D70F     3      LD   NOACCI+1, #00F
358 01B5 DB0F     3      LD   RETTCL1, #00F
359 01B7 DF0F     3      LD   RETTCL2, #00F
360 01B9 83      1      JFC
361 01BA 04      3      JP   CC
362 01BB BDD688   4      RBIT  ACARRI, RAMBIT
363 01BE 03      3      JP   PROSO

```



```

364 01BF BD0B78 4 CC: SBIT ACARRI, RAMBITI
365 01C2 D80F 3 PROSO: LD NOACCI+2, #00F
366 01C4 A1 1 PROS: SC
367 01C5 5F 1 LD B, #DIGIT
368 01C6 9EDF 2 LD [B], #B'11011111
369 01C8 BD0B75 4 IFBIT DESBIT4, RAMBITI
370 01CB 9EEF 2 LD [B], #B'110101111
371 01CD BD0C71 4 IFBIT CODI, RAMBITI
372 01D0 9EEF 2 LD [B], #B'11011111
373 01D2 3DEE74 4 IFBIT TRUN, CNTRL
374 01D5 9EEF 2 LD [B], #B'110101111
375 01D7 BD0B70 4 IFBIT ACARRI, RAMBITI
376 01DA 02 3 JP SCAN
377 01DB 9EEF 2 LD [B], #B'11011111
378 01DD SCAN: ;inicio de barrido
379 01DD AE 1 LD A, [B]
380 01DE 9CD0 3 X A, PORTLD
381 01E0 B8 1 NOP
382 01E1 B8 1 NOP
383 01E2 B8 1 NOP
384 01E3 9DD7 3 LD A, PORTI
385 01E5 95FF 2 AND A, #OFF
386 01E7 92FF 2 IFEQ A, #OFF
387 01E9 02 3 JP NEXDIG
388 01EA 22FB 3 JMP DECOD
389 01EC NEXDIG: ;recorre el cero hacia la derecha para
390 01EC BD0B69 4 RBIT NOIDENT, RAMBITI ;verificar los sensores de la siguiente
391 01EF AE 1 LD A, [B] ;línea.
392 01F0 B0 1 RRC A
393 01F1 A6 1 X A, [B]
394 01F2 9CF2 3 X A, CONT
395 01F4 BDF270 4 IFBIT 0, CONT
396 01F7 E5 3 JP SCAN
397 01F8 8DEE74 4 IFBIT TRUN, CNTRL ;si el "TIMER" está corriendo sólo verifica
398 01FB 21C4 3 JMP PROS ;la matriz de teclas.
399 01FD BD0C71 4 IFBIT CODI, RAMBITI
400 0200 01 3 JP PROXTEC
401 0201 06 3 JP CLOSENS
402 0202 C6 3 PROXTEC:DRSZ NOACCI ;decrementa en uno el reg. para retardo
403 0203 21C4 3 JMP PROS ;de letras no-tecla y no-sensor.
404 0205 C7 3 DRSZ NOACCH+1
405 0206 21C4 3 JMP PROS
406 0208 BCD000 3 CLOSENS: LD PORTLD, #00 ;verifica de los 8 sensores NC que se
407 020B B8 1 NOP ;encuentran conectados a la línea 6 del
408 020C B8 1 NOP ;puerto L
409 020D 9DD7 3 LD A, PORTI
410 020F 95FF 2 AND A, #OFF
411 0211 9200 2 IFEQ A, #000
412 0213 2242 > 3 JP REVPGS
413 0215 BD0B77 4 IFBIT PRACTI, RAMBITI ;indica que se debe revisar sensor por
414 0218 07 3 JP BITxBIT ;sensor debido a que se puede tener m s
415 0219 9C01 3 X A, ILINE ;de un sensor activado al mismo tiempo
416 021B BC0240 3 RETBIT: LD ILINE, #040
417 021E 2397 3 JMP BIEN
418 0220 BC0100 3 BITxBIT: LD ILINE, #00 ;se revisan sólo los sensores que no
419 0223 BDD770 4 IFBIT 0, PORTI ;proporcionan un retardo en la activación
420 0226 BC0101 3 LD ILINE, #001 ;de la sirena exterior.
421 0229 BDD771 4 IFBIT 1, PORTI
422 022C BC0102 3 LD ILINE, #002
423 022F BDD774 4 IFBIT 4, PORTI
424 0232 BC0110 3 LD ILINE, #010
425 0235 BDD776 4 IFBIT 6, PORTI
426 0238 BC0140 3 LD ILINE, #040
427 023B 9D01 3 LD A, ILINE
428 023D 9200 2 IFEQ A, #00
429 023F 02 3 JP REVPGS
430 0240 221B > 3 JP RETBIT
431 0242 DED6 3 REVPGS: LD B, #PORTGP ;revisión de sensores conectados en PORTG
432 0244 BC0220 3 LD ILINE, #020
433 0247 70 1 IFBIT 0, [B]
434 0248 22DD 3 JMP G0
435 024A 71 1 IFBIT 1, [B]
436 024B 22E2 3 JMP G1
437 024D 72 1 IFBIT 2, [B]

```

438 024E 22E7	3	JMP	G2	
439 0250 74	1	IFBIT	4, [B]	
440 0251 22EC	3	JMP	G4	
441 0253 75	1	IFBIT	5, [B]	
442 0254 22F1	3	JMP	G5	
443 0256 76	1	IFBIT	6, [B]	
444 0257 22F6	3	JMP	G6	
445 0259 BD0B74	4	IFBIT	PRExSEN, RAMBITI	
446 025C 32C6	5	JSR	VERIF	
447 025E BDEE74	4	IFBIT	TRUN, CNTRL	,si está activada la alarma no despliega
448 0261 21C4	3	JMP	PROS	,letreros de no-interrupción
449 0263 BD0B73	4	IFBIT	DESxTIE, RAMBITI	
450 0266 21C4	3	JMP	PROS	
451 0268 C6	3	DRSZ	NOACCI	,esta parte del programa despliega 12
452 0269 21C4	3	JMP	PROS	,letreros, uno cada 20 seg., si el
453 026B C7	3	DRSZ	NOACCI+1	,microcontrolador no es interrumpido por
454 026C 21C4	3	JMP	PROS	,teclas o sensores en un lapso de tiempo
455 026E C8	3	DRSZ	NOACCI+2	,de 20 seg.
456 026F 02	3	JP	COPCION	
457 0270 21C2	3	JMP	PROS0	
458 0272 9DF8	3	COPCION: LD	A, NOACCI+2	
459 0274 BDEE74	4	IFBIT	TRUN, CNTRL	,si está activada la alarma, o si se
460 0277 21C4	3	JMP	PROS	,apago automáticamente después de 5 min.,
461 0279 BD0B73	4	IFBIT	DESxTIE, RAMBITI	,no se despliega letrero de
462 027C 21C4	3	JMP	PROS	,no-interrupción.
463 027E DE63	3	LD	B, #RAMGEN	
464 0280 9E00	2	LD	[B], #00	
465 0282 920C	2	IFEQ	A, #00C	
466 0284 9E0E	2	LD	[B], #00E	;"UNAM, FACULTAD DE INGENIERIA. "
467 0286 920B	2	IFEQ	A, #00B	
468 0288 9E30	2	LD	[B], #030	;"TESIS PROFESIONAL QUE PRESENTO"
469 028A 920A	2	IFEQ	A, #00A	
470 028C 9E52	2	LD	[B], #052	;"ROBERTO ROMERO Y ALFONSO ANTUNEZ"
471 028E 9209	2	IFEQ	A, #009	
472 0290 9E74	2	LD	[B], #074	;"EL SISTEMA DETECTA LA VIOLACION"
473 0292 9208	2	IFEQ	A, #008	
474 0294 9E96	2	LD	[B], #096	;"D"CADA DISP D"SEG INSTALADO"
475 0296 9207	2	IFEQ	A, #007	
476 0298 9E0E	2	LD	[B], #00E	;"UN SINCERO AGRADECIMIENTO A LAS"
477 029A 9206	2	IFEQ	A, #006	
478 029C 9E30	2	LD	[B], #030	;"PERSONAS QUE COLABORARON PARA "
479 029E 9205	2	IFEQ	A, #005	
480 02A0 9E52	2	LD	[B], #052	;"QUE ESTE PROYECTO SE LLEVARA "
481 02A2 9204	2	IFEQ	A, #004	
482 02A4 9E74	2	LD	[B], #074	;"A CABO,ING. MIGUEL A CRUZ LEON"
483 02A6 9203	2	IFEQ	A, #003	
484 02A8 9E96	2	LD	[B], #096	;"Y EL SR RICARDOROMERO ANGUIANO."
485 02AA 9202	2	IFEQ	A, #002	
486 02AC 9EB8	2	LD	[B], #0B8	;"MUCHAS GRACIAS*A TODOS Y ESPEC."
487 02AE 9201	2	IFEQ	A, #001	
488 02B0 9EDA	2	LD	[B], #0DA	;"IALMENTE A MI MADRE, GRACIAS "
489 02B2 AE	1	LD	A, [B]	
490 02B3 9200	2	IFEQ	A, #00	
491 02B5 21C4	3	JMP	PROS	
492 02B7 9DF8	3	LD	A, NOACCI+2	
493 02B9 9307	2	IFGT	A, #007	
494 02BB 3D00	5	JSR	NOTECL	
495 02BD 3E00	5	JSR	NOTECL1	
496 02BF 3081	5	JSR	DISPLAY	
497 02C1 BCD800	3	LD	PORTCD, #00	
498 02C4 21C4	3	JMP	PROS	
499 02C6		VERIF		,una vez que se reviso el estado normal
500 02C6 A0	1	RC		,de los sesores se procede a desplegar el
501 02C7 BD0B6C	4	RBIT	PRExSEN, RAMBITI	,siguiente letrero
502 02CA BD0B68	4	RBIT	ACARRI, RAMBITI	;"ADELANTE, PUEDE ACTIVAR EL SIST "
503 02CD BD0B6F	4	RBIT	PRACTI, RAMBITI	
504 02D0 DE63	3	LD	B, #RAMGEN	
505 02D2 9ED7	2	LD	[B], #0D7	
506 02D4 3D00	5	JSR	NOTECL	
507 02D6 B8	1	NOP		
508 02D7 3081	5	JSR	DISPLAY	
509 02D9 BCD800	3	LD	PORTCD, #00	
510 02DC 8F	5	RTT		

512 02DD BC0109	3	G0	LD	ILINE, #009	
513 02E0 2397	3		JMP	BIEN	
514 02E2 BC010A	3	G1	LD	ILINE, #00A	
515 02E5 2397	3		JMP	BIEN	
516 02E7 BC010B	3	G2	LD	ILINE, #00B	
517 02EA 2397	3		JMP	BIEN	
518 02EC BC010C	3	G4	LD	ILINE, #00C	
519 02EF 2397	3		JMP	BIEN	
520 02F1 BC010D	3	G5	LD	ILINE, #00D	
521 02F4 2397	3		JMP	BIEN	
522 02F6 BC010E	3	G6	LD	ILINE, #00E	
523 02F9 2397	3		JMP	BIEN	
524 02FB			DECOD:		;decodifica tecla oprimida
525 02FB BDDEE74	4		IFBIT	TRUN, CNTRL	
526 02FE 02	3		JP	ACORTA	
527 02FF CB	3		DRSZ	RETTCL1	
528 0300 FA	3		JP	DECOD	
529 0301 CF	3	ACORTA:	DRSZ	RETTCL2	
530 0302 F8	3		JP	DECOD	
531 0303 9DD7	3		LD	A, PORT1	
532 0305 95FF	2		AND	A, #0FF	
533 0307 92FF	2		IFEQ	A, #0FF	
534 0309 21EC	3		JMP	NEXDIG	
535 030B 96FF	2		XOR	A, #0FF	
536 030D 9C01	3		X	A, ILINE	
537 030F AE	1		LD	A, [B]	
538 0310 95FF	2		AND	A, #0FF	
539 0312 96FF	2		XOR	A, #0FF	
540 0314 9C02	3		X	A, LLINE	
541 0316 BDDEE74	4		IFBIT	TRUN, CNTRL	;si el "TIMER" se encuentra corriendo
542 0319 12	3		JP	RECICOD	;entonces revisa bit por bit del puerto
543 031A 9D02	3		LD	A, LLINE	;L y del puerto I para verificar la
544 031C 9220	2		IFEQ	A, #020	;tecla oprimida.
545 031E 2397	3		JMP	BIEN	
546 0320 BCD060	3		LD	PORTLD, #B'01100000	
547 0323 9DD7	3	SUELTA:	LD	A, PORT1	
548 0325 95FF	2		AND	A, #0FF	
549 0327 92FF	2		IFEQ	A, #0FF	
550 0329 2397	3		JMP	BIEN	
551 032B F7	3		JP	SUELTA	
552 032C			RECICOD:		;esta sección es usada sólo cuando el
553 032C BD0271	4		IFBIT	1, LLINE	; "TIMER" se encuentra corriendo, lo cual
554 032F 0F	3		JP	IENTRA1	;indica que se encuentra activa la alarma,
555 0330 BD0272	4		IFBIT	2, LLINE	;por lo que se espera que se teclé sólo
556 0333 17	3		JP	IENTRA2	;el código de desactivación, es decir,
557 0334 BD0273	4		IFBIT	3, LLINE	;todos los números y otras dos teclas.
558 0337 1F	3		JP	IENTRA3	;Lo que se hace en esta sección es inhibir
559 0338 BD0274	4		IFBIT	4, LLINE	;la detección de otros sensores y otras
560 033B 2363	3		JMP	IENTRA4	;teclas que no tiene sentido leerlas en
561 033D 21EC	3		JMP	NEXDIG	;este momento.
562 033F BC0202	3	IENTRA1:	LD	LLINE, #002	
563 0342 336F	5		JSR	RUTILIN	
564 0344 BD0B71	4		IFBIT	NOIDENT, RAMBIT1	
565 0347 21EC	3		JMP	NEXDIG	
566 0349 2397	3		JMP	BIEN	
567 034B BC0204	3	IENTRA2:	LD	LLINE, #004	
568 034E 336F	5		JSR	RUTILIN	
569 0350 BD0B71	4		IFBIT	NOIDENT, RAMBIT1	
570 0353 21EC	3		JMP	NEXDIG	
571 0355 2397	3		JMP	BIEN	
572 0357 BC0208	3	IENTRA3:	LD	LLINE, #008	
573 035A 336F	5		JSR	RUTILIN	
574 035C BD0B71	4		IFBIT	NOIDENT, RAMBIT1	
575 035F 21EC	3		JMP	NEXDIG	
576 0361 2397	3		JMP	BIEN	
577 0363 BC0210	3	IENTRA4:	LD	LLINE, #010	
578 0366 336F	5		JSR	RUTILIN	
579 0368 BD0B71	4		IFBIT	NOIDENT, RAMBIT1	
580 036B 21EC	3		JMP	NEXDIG	
581 036D 2397	3		JMP	BIEN	
582 036F			RUTILIN:		
583 036F BD0170	4		IFBIT	0, ILINE	
584 0372 0C	3		JP	GUARD1	
585 0373 BD0171	4		IFBIT	1, ILINE	

586 0376 10	3	JP	GUARD2	
587 0377 BD0172	4	IFBIT	2, ILINE	
588 037A 14	3	JP	GUARD3	
589 037B BD0B79	4	SBIT	NOIDENT, RAMBIT	
590 037E 8E	5	RET		
591 037F BC0101	3	GUARD1:	LD ILINE, #001	
592 0382 BDD770	4	NORET1:	IFBIT 0, PORT1	
593 0385 8E	5	RET		
594 0386 FB	3	JP	NORET1	
595 0387 BC0102	3	GUARD2:	LD ILINE, #002	
596 038A BDD771	4	NORET2:	IFBIT 1, PORT1	
597 038D 8E	5	RET		
598 038E FB	3	JP	NORET2	
599 038F BC0104	3	GUARD3:	LD ILINE, #004	
600 0392 BDD772	4	NORET3:	IFBIT 2, PORT1	
601 0395 8E	5	RET		
602 0396 FB	3	JP	NORET3	
603				
604 0397		BIEN:		,una vez que se tienen los dos valores
605 0397 BD0B73	4	IFBIT	DESXTIE, RAMBIT	;almacenados en RAM pertenecientes a una
606 039A 23CE	3	JMP	PREXLL	,tecla o sensor, se regresa el valor que
607 039C 9D02	3	RETORNO:	LD A, LLINE	;tenia la bandera de acarreo cuando recién
608 039E 9310	2	IFGT	A, #010	;entro a esta subrutina Si se trata de un
609 03A0 23DF	3	JMP	ALMACEN	,sensor quedan los valores en otra
610 03A2 BD0C6E	4	RETALMA:	RBIT NOTECLA, RAMBIT	;localidad de menor a para que no se
611 03A5 BDEE74	4	IFBIT	TRUN, CNTRL	;detecte el mismo sensor en forma
612 03A8 1D	3	JP	PARACK	,consecutiva
613 03A9 BD0B70	4	REGRESO	IFBIT ACARRI, RAMBIT	
614 03AC 0B	3	JP	PC	
615 03AD A0	1	RC		
616 03AE BD0B72	4	IFBIT	PARAREL, RAMBIT	
617 03B1 BDEE7C	4	SBIT	TRUN, CNTRL	
618 03B4 BD0B6A	4	RBIT	PARAREL, RAMBIT	
619 03B7 8E	5	RET		
620 03B8		PC		
621 03B8 BD0B68	4	RBIT	ACARRI, RAMBIT	
622 03BB A1	1	SC		
623 03BC BD0B72	4	IFBIT	PARAREL, RAMBIT	
624 03BF BDEE7C	4	SBIT	TRUN, CNTRL	
625 03C2 BD0B6A	4	RBIT	PARAREL, RAMBIT	
626 03C5 8E	5	RET		
627 03C6 BDEE6C	4	PARACK	RBIT TRUN, CNTRL	
628 03C9 BD0B7A	4	SBIT	PARAREL, RAMBIT	
629 03CC 23A9	3	JMP	REGRESO	
630 03CE 9D02	3	PREXLL	LD A, LLINE	
631 03D0 BD0482	4	IFEQ	A, LLANT	
632 03D3 02	3	JMP	PREXLL	
633 03D4 239C	3	JMP	RETORNO	
634 03D6 9D01	3	PREXLL	LD A, ILINE	
635 03D8 BD0382	4	IFEQ	A, ILANT	
636 03DB 21C4	3	JMP	PROS	
637 03DD 239C	3	JMP	RETORNO	
638 03DF 9C04	3	ALMACEN	X A, LLANT	
639 03E1 9D01	3	LD	A, ILINE	
640 03E3 9C03	3	X	A, ILANT	
641 03E5 23A2	3	JMP	RETALMA	
642				
643	*	ESTA SECCION DECODIFICA LA TECLA OPRIMIDA O SENSOR ACCIONADO		*
644	*	PARA REALIZAR UN SALTO A UNA RUTINA CORRESPONDIENTE A DICHA TECLA		*
645	*	O SENSOR LA DECODIFICACION DE LOCALIDADES SIN USO SON USADAS		*
646	*	COMO RETORNO A SUBROUTINA POR LO QUE NO SE REALIZA ACCION ALGUNA		*
647				
648 03E7		KEYS		
649 03E7 DC10	3	LD	X, #LETRERO	
650 03E9 BD0C71	4	IFBIT	CODI, RAMBIT	,indica que la tecla oprimida
651 03EC 03	3	JP	NCRB	,anteriormente fue un n micro
652 03ED DE63	3	LD	B, #RAMGEN	
653 03EF 04	3	JP	NOCODI	
654 03F0		NCRB		
655 03F0 9D62	3	LD	A, RECODI	
656 03F2 9CFE	3	X	A, B	
657 03F4 9D02	3	NOCODI:	LD A, LLINE	,realización de un primer salto
658 03F6 9201	?	IFLQ	A, #001	
659 03F8 19	3	JP	PARO	

660 03F9 9202	2	IFEQ A, #002
661 03FB 2421	3	JMP PAR1
662 03FD 9204	2	IFEQ A, #004
663 03FF 2430	3	JMP PAR2
664 0401 9208	2	IFEQ A, #008
665 0403 243F	3	JMP PAR3
666 0405 9210	2	IFEQ A, #010
667 0407 244E	3	JMP PAR4
668 0409 9220	2	IFEQ A, #020
669 040B 245D	3	JMP PAR5
670 040D 9240	2	IFEQ A, #040
671 040F 2484	3	JMP PAR6
672 0411 8D	5	RETSK
673		
674 0412		PAR0:
675 0412 9D01	3	LD A, ILINE
676 0414 9201	2	IFEQ A, #001
677 0416 24A7	3	JMP KEY0
678 0418 9202	2	IFEQ A, #002
679 041A 24BE	3	JMP KEY1
680 041C 9204	2	IFEQ A, #004
681 041E 24D7	3	JMP KEY2
682 0420 8D	5	RETSK
683		
684 0421		PAR1:
685 0421 9D01	3	LD A, ILINE
686 0423 9201	2	IFEQ A, #001
687 0425 24F0	3	JMP KEY8
688 0427 9202	2	IFEQ A, #002
689 0429 25C3	3	JMP KEY9
690 042B 9204	2	IFEQ A, #004
691 042D 25E5	3	JMP KEY10
692 042F 8D	5	RETSK
693		
694 0430		PAR2:
695 0430 9D01	3	LD A, ILINE
696 0432 9201	2	IFEQ A, #001
697 0434 25FD	3	JMP KEY16
698 0436 9202	2	IFEQ A, #002
699 0438 261F	3	JMP KEY17
700 043A 9204	2	IFEQ A, #004
701 043C 2641	3	JMP KEY18
702 043E 8D	5	RETSK
703		
704 043F		PAR3:
705 043F 9D01	3	LD A, ILINE
706 0441 9201	2	IFEQ A, #001
707 0443 2663	3	JMP KEY24
708 0445 9202	2	IFEQ A, #002
709 0447 2685	3	JMP KEY25
710 0449 9204	2	IFEQ A, #004
711 044B 26A7	3	JMP KEY26
712 044D 8D	5	RETSK
713		
714 044E		PAR4:
715 044E 9D01	3	LD A, ILINE
716 0450 9201	2	IFEQ A, #001
717 0452 26C9	3	JMP KEY32
718 0454 9202	2	IFEQ A, #002
719 0456 26EB	3	JMP KEY33
720 0458 9204	2	IFEQ A, #004
721 045A 270D	3	JMP KEY34
722 045C 8D	5	RETSK
723		
724 045D		PAR5:
725 045D 9D01	3	LD A, ILINE
726 045F 9201	2	IFEQ A, #001
727 0461 272F	3	JMP KEY40
728 0463 9202	2	IFEQ A, #002
729 0465 274C	3	JMP KEY41
730 0467 9204	2	IFEQ A, #004
731 0469 2769	3	JMP KEY42
732 046B 9209	2	IFEQ A, #009
733 046D 2841	3	JMP KEY64

;dirección usada por sensores colocados
;en puerto G y sensores NO.

;dirección usada por sensores colocados
;en la línea 6 del puerto L.

;dirección usada por sensores colocados
;en puerto G y sensores NO.

734 046F 920A	2	IFEQ	A, #00A	
735 0471 285B	3	JMP	KEY65	
736 0473 920B	2	IFEQ	A, #00B	
737 0475 2874	3	JMP	KEY66	
738 0477 920C	2	IFEQ	A, #00C	
739 0479 288E	3	JMP	KEY67	
740 047B 920D	2	IFEQ	A, #00D	
741 047D 28A8	3	JMP	KEY68	
742 047F 920E	2	IFEQ	A, #00E	
743 0481 28C2	3	JMP	KEY69	
744 0483 8D	5	RETSK		
745				
746 0484	PAR6			,dirección usada por sensores colocados
747 0484 9D01	3	LD	A, ILINE	en la línea 6 del puerto L.
748 0486 9201	2	IFEQ	A, #001	
749 0488 2786	3	JMP	KEY48	
750 048A 9202	2	IFEQ	A, #002	
751 048C 27A0	3	JMP	KEY49	
752 048E 9204	2	IFEQ	A, #004	
753 0490 27BA	3	JMP	KEY50	
754 0492 9208	2	IFEQ	A, #008	
755 0494 27D1	3	JMP	KEY51	
756 0496 9210	2	IFEQ	A, #010	
757 0498 27E3	3	JMP	KEY52	
758 049A 9220	2	IFEQ	A, #020	
759 049C 27FC	3	JMP	KEY53	
760 049E 9240	2	IFEQ	A, #040	
761 04A0 2812	3	JMP	KEY54	
762 04A2 9280	2	IFEQ	A, #080	
763 04A4 282B	3	JMP	KEY55	
764 04A6 8D	5	RETSK		
765				
766 04A7	KEY0.			, "ALARMA AUXILIAR-*** ACTIVADA ***"
767 04A7 BC0E01	3	LD	TEMP, #001	, código de tecla
768 04AA 9E09	2	LD	[B], #009	
769 04AC BD0C7B	4	SBIT	3, PORTD	, encendido de "LED" rojo en unidad de control
770 04AF BD0C69	4	RBIT	CODI, RAMBIT	
771 04B2 BD0C6B	4	RBIT	CINCMIN, RAMBIT	
772 04B5 BD0C68	4	RBIT	SIRENA, PORTD	, activa sirena exterior inmediatamente
773 04B8 A1	1	SC		, habilita todo el sistema
774 04B9 BD0E7C	4	SBIT	TRUN, CNTRL	, activa "TIMER" para generar la señal "PWM"
775 04BC 2900	3	JMP	CAPU1	, en TIO
776				
777 04BE	KEY1			
778 04BE BD0C71	4	IFBIT	CODI, RAMBIT	, si se oprimió un número anteriormente
779 04C1 8D	5	RETSK		, no ejecutes esta función
780 04C2 BD0C7A	4	SBIT	REPET1, RAMBIT	, indica que la tecla tiene dos funciones
781 04C5 BD0C71	4	IFBIT	REFLEC, PORTD	
782 04C8 07	3	JP	APAG	
783 04C9 BD0C79	4	SBIT	REFLEC, PORTD	
784 04CC 9E2B	2	LD	[B], #02B	, "ENCENDIDO D"REFLECTOR AUTOMATIC"
785 04CE 2900	3	JMP	CAPU1	
786 04D0 BD0C69	4	APAG	RBIT REFLEC, PORTD	
787 04D3 9E4D	2	LD	[B], #04D	, "ENCENDIDO D"REFLECTOR MANUAL"
788 04D5 2900	3	JMP	CAPU1	
789				
790 04D7	KEY2			
791 04D7 BD0C71	4	IFBIT	CODI, RAMBIT	
792 04DA 8D	5	RETSK		
793 04DB BD0C7A	4	SBIT	REPET1, RAMBIT	
794 04DE BD0C70	4	IFBIT	PTA, RAMBIT	
795 04E1 07	3	JP	DESC	
796 04E2 BD0C78	4	SBIT	PTA, RAMBIT	
797 04E5 9E6C	2	LD	[B], #06C	, "PUERTA DE CALLE OPERANDO"
798 04E7 2900	3	JMP	CAPU1	
799 04E9 BD0C68	4	DESC.	RBIT PTA, RAMBIT	
800 04EC 9E8A	2	LD	[B], #08A	, "PUERTA DE CALLE NO-OPERANDO"
801 04EE 2900	3	JMP	CAPU1	
802				
803 04F0	KEY8			comparación del código de operación
804 04F0 PD0C71	4	IFBIT	CODI, RAMBIT	, de preactivación y de encendido de luz
805 04F3 01	3	JP	ACEP	, en LCD
806 04F4 8D	5	RETSK		
807 04F5	ACCP			

```

808 04F5 BDDC7A 4 SBIT REPETI, RAMBIT
809 04F8 59 1 LD B, #DIG0
810 04F9 64 1 CLR A
811 04FA A2 2 X A, [B+]
812 04FB 9235 2 IFEQ A, #035 ;5 = 035 hex
813 04FD 09 3 JP NEX0
814 04FE 9230 2 IFEQ A, #030 ;tecla '0' para verificar que los sensores
815 0500 2599 3 JMP VERIFIC ;se encuentren en estado normal.
816 0502 9231 2 IFEQ A, #031 ;tecla '1' para encender luz de LCD.
817 0504 25A7 3 JMP LUZLCD
818 0506 16 3 JP RCOD
819 0507 A2 2 NEX0: X A, [B+]
820 0508 9232 2 IFEQ A, #032
821 050A 01 3 JP NEX1
822 050B 11 3 JP RCOD
823 050C A2 2 NEX1: X A, [B+]
824 050D 9230 2 IFEQ A, #030
825 050F 01 3 JP NEX2
826 0510 0C 3 JP RCOD
827 0511 A2 2 NEX2: X A, [B+]
828 0512 9230 2 IFEQ A, #030
829 0514 01 3 JP NEX3
830 0515 07 3 JP RCOD
831 0516 A2 2 NEX3: X A, [B+]
832 0517 9230 2 IFEQ A, #030
833 0519 12 3 JP NEX4
834 051A 9231 2 IFEQ A, #031
835 051C 09 3 JP NOCHE
836 051D BD0C69 4 RCOD: RBIT CODI, RAMBIT ;si el código de operación no es el
837 0520 DE63 3 LD B, #RAMGEN ;correcto se despliega el letrero
838 0522 9E09 2 LD [B], #009 ;"CONTRASEÑA NO IDENTIFICADA."
839 0524 2B00 3 JMP CAPTU3
840 0526 88 1 NOCHE: IFC
841 0527 F5 3 JP RCOD
842 0528 BD0B7E 4 SBIT DENOCHE, RAMBIT
843 052B 03 3 JP ACTNOCH
844 052C NEX4.
845 052C 88 1 IFC
846 052D 256C 3 JMP RCARR
847 052F A1 1ACTNOCH:SC ;el sistema operar en modo de noche.
848 0530 D4FF 3 LD RETINICI, #0FF
849 0532 D5FF 3 LD RETINICI+1, #0FF
850 0534 D095 3 LD RETAR, #095 ;para generar retardo de 60 seg.
851 0536 BD0C69 4 RBIT CODI, RAMBIT
852 0539 BD0B6F 4 RBIT PRACTI, RAMBIT
853 053C BDDC78 4 SBIT SIRENA, PORTD
854 053F DE63 3 LD B, #RAMGEN
855 0541 9EB5 2 LD [B], #0B5 ;"EL SIST SE ACTIVARA EN 60 SEG."
856 0543 3D00 5 JSR NOTECL
857 0545 B8 1 NOP
858 0546 3081 5 JSR DISPLAY
859 0548 C4 3 DISMINU: DRSZ RETINICI ;genera retardo de 60 seg.
860 0549 FE 3 JP DISMINU
861 054A C5 3 DRSZ RETINICI+1
862 054B FC 3 JP DISMINU
863 054C C0 3 DRSZ RETAR
864 054D FA 3 JP DISMINU
865 054E DE63 3 LD B, #RAMGEN
866 0550 DC10 3 LD X, #LETRERO
867 0552 BD0C7F 4 SBIT INDIACT, RAMBIT
868 0555 BDDC68 4 RBIT SIRENA, PORTD
869 0558 D102 3 LD RETAR+1, #002
870 055A BDEE7C 4 SBIT TRUN, CNTRL
871 055D BDDC7B 4 SBIT 3, PORTD ;enciende "LED" rojo
872 0560 BD0B7E 4 IFBIT DENOCHE, RAMBIT
873 0563 04 3 JP NOCTUR
874 0564 9E47 2 LD [B], #047 ;*** SISTEMA ***EN OPERACION TOTAL"
875 0566 2C00 3 JMP CAPTU4
876 0568 9E48 2 NOCTUR: LD [B], #048
877 056A 2B00 3 JMP CAPTU3 ;**** SISTEMA ****OPERANDO D"NOCHE"
878 056C RCARR:
879 056C A0 1 RC
880 056D BDEE6C 4 RBIT TRUN, CNTRL ;para "TIMER".
881 0570 BDDC78 4 SBIT SIRENA, PORTD ;desconecta sirena exterior

```

```

882 0573 BD0C69 4 RBIT CODI, RAMBIT ,reseteo de banderas en registro de
883 0576 BD0B6F 4 RBIT PRACTI, RAMBITI ,control
884 0579 BD0B6E 4 RBIT DENOCH, RAMBITI
885 057C BD0B6B 4 RBIT DESxTIE, RAMBITI
886 057F BD0B6D 4 RBIT DESBIT4, RAMBITI
887 0582 BD0B6C 4 RBIT PRExSEN, RAMBITI
888 0585 BD0C6B 4 RBIT 3, PORTD ,apaga "LED" rojo en unidad de control
889 0588 BD0C7F 4 SBIT INDIACT, RAMBIT
890 058B BD0C68 4 RBIT SIRENA, PORTD
891 058E D102 3 LD RETAR+1, #002
892 0590 BD0EE7C 4 SBIT TRUN, CNTRL
893 0593 DE63 3 LD B, #RAMGEN
894 0595 9E09 2 LD [B], #009 ,**** SISTEMA ****NO-OPERANDC****
895 0597 2A00 3 JMP CAPTU2
896 0599 VERIFIC
897 0599 A1 1 SC ,barre sólo una vez los sensores
898 059A BD0C69 4 RBIT CODI, RAMBIT
899 059D BD0B7C 4 SBIT PRExSEN, RAMBITI
900 05A0 BD0B7F 4 SBIT PRACTI, RAMBITI
901 05A3 BD0C78 4 SBIT SIRENA, PORTD
902 05A6 8D 5 RETSK
903 05A7 LUZLCD ,enciende/apaga luz de LCD.
904 05A7 BD0C74 4 IFBIT 4, PORTD
905 05AA 0C 3 JP APAGA
906 05AB BD0C7C 4 SBIT 4, PORTD
907 05AE BD0C69 4 RBIT CODI, RAMBIT
908 05B1 DE63 3 LD B, #RAMGEN
909 05B3 9EC5 2 LD [B], #0C5 ,*** LUZ DE LCD *** ENCENDIDA ***
910 05B5 2A00 3 JMP CAPTU2
911 05B7 APAGA:
912 05B7 BD0C6C 4 RBIT 4, PORTD
913 05BA BD0C69 4 RBIT CODI, RAMBIT
914 05BD DE63 3 LD B, #RAMGEN
915 05BF 9EC7 2 LD [B], #0C7 ,*** LUZ DE LCD *** APAGADA ***
916 05C1 2C00 3 JMP CAPTU4
917
918 05C3 KEY9 :tecla "0"
919 05C3 BD0C7A 4 SBIT REPETI, RAMBIT
920 05C6 BD0C71 4 IFBIT CODI, RAMBIT
921 05C9 04 3 JP SIGU0
922 05CA BD0C79 4 SBIT CODI, RAMBIT
923 05CD 59 1 LD B, #DIG0
924 05CE 9A30 2 SIGU0 LD [B+], #030
925 05D0 9DFE 3 LD A, B
926 05D2 9207 2 IFEQ A, #DIG1
927 05D4 BD0C7C 4 SBIT LIMLCD, RAMBIT
928 05D7 9C62 3 X A, RECODI
929 05D9 BC1023 3 LD LETRERO, #023 , # = 023 Hex
930 05DC BC117C 3 LD LETRERO+1, #07C , | = 07C Hex, indica que el letrero
931 05DF 920A 2 IFEQ A, #DIG4 ,finaliza cuando se encuentre este
932 05E1 BC6206 3 LD RECODI, #DIG0 ,carácter.
933 05E4 8E 5 RET
934
935 05E5 KEY10 ,indica el estado del sistema
936 05E5 BD0C71 4 IFBIT CODI, RAMBIT
937 05E8 8D 5 RETSK
938 05E9 BD0C7A 4 SBIT REPETI, RAMBIT
939 05EC 88 1 IFC
940 05ED 04 3 JP OPERA
941 05EE 9E09 2 LD [B], #009 ,**** SISTEMA ****NO-OPERANDO****
942 05F0 2A00 3 JMP CAPTU2
943 05F2 9E48 2 OPERA, LD [B], #048 ,**** SISTEMA ****OPERANDO D"NOCHE"
944 05F4 BD0B7E 4 IFBIT DENOCH, RAMBITI
945 05F7 2B00 3 JMP CAPTU3
946 05F9 9E47 2 LD [B], #047 ,*** SISTEMA ***EN OPERACION TOTAL*
947 05FB 2C00 3 JMP CAPTU4
948
949 05FD KEY16 :tecla "9"
950 05FD BD0C7A 4 SBIT REPETI, RAMBIT
951 0600 BD0C71 4 IFBIT CODI, RAMBIT
952 0603 04 3 JP SIGU9
953 0604 BD0C79 4 SBIT CODI, RAMBIT
954 0607 59 1 LD B, #DIG0
955 0608 9A39 2 SIGU9 LD [B+], #039

```


956 060A 9DFE	3		LD	A, B	
957 060C 9207	2		IFEQ	A, #DIG1	
958 060E BD0C7C	4		SBIT	LIMLCD, RAMBIT	
959 0611 9C62	3		X	A, RECODI	
960 0613 BC1023	3		LD	LETRERO, #023	
961 0616 BC117C	3		LD	LETRERO+1, #07C	
962 0619 920A	2		IFEQ	A, #DIG4	
963 061B BC6206	3		LD	RECODI, #DIG0	
964 061E 8E	5		RET		
965					
966 061F		KEY17:			;tecla "8"
967 061F BD0C7A	4		SBIT	REPETI, RAMBIT	
968 0622 BD0C71	4		IFBIT	CODI, RAMBIT	
969 0625 04	3		JP	SIGU8	
970 0626 BD0C79	4		SBIT	CODI, RAMBIT	
971 0629 59	1		LD	B, #DIG0	
972 062A 9A38	2	SIGU8:	LD	[B+], #038	
973 062C 9DFE	3		LD	A, B	
974 062E 9207	2		IFEQ	A, #DIG1	
975 0630 BD0C7C	4		SBIT	LIMLCD, RAMBIT	
976 0633 9C62	3		X	A, RECODI	
977 0635 BC1023	3		LD	LETRERO, #023	
978 0638 BC117C	3		LD	LETRERO+1, #07C	
979 063B 920A	2		IFEQ	A, #DIG4	
980 063D BC6206	3		LD	RECODI, #DIG0	
981 0640 8E	5		RET		
982					
983 0641		KEY18:			;tecla "7"
984 0641 BD0C7A	4		SBIT	REPETI, RAMBIT	
985 0644 BD0C71	4		IFBIT	CODI, RAMBIT	
986 0647 04	3		JP	SIGU7	
987 0648 BD0C79	4		SBIT	CODI, RAMBIT	
988 064B 59	1		LD	B, #DIG0	
989 064C 9A37	2	SIGU7:	LD	[B+], #037	
990 064E 9DFE	3		LD	A, B	
991 0650 9207	2		IFEQ	A, #DIG1	
992 0652 BD0C7C	4		SBIT	LIMLCD, RAMBIT	
993 0655 9C62	3		X	A, RECODI	
994 0657 BC1023	3		LD	LETRERO, #023	
995 065A BC117C	3		LD	LETRERO+1, #07C	
996 065D 920A	2		IFEQ	A, #DIG4	
997 065F BC6206	3		LD	RECODI, #DIG0	
998 0662 8E	5		RET		
999					
1000 0663		KEY24:			;tecla "6"
1001 0663 BD0C7A	4		SBIT	REPETI, RAMBIT	
1002 0666 BD0C71	4		IFBIT	CODI, RAMBIT	
1003 0669 04	3		JP	SIGU6	
1004 066A BD0C79	4		SBIT	CODI, RAMBIT	
1005 066D 59	1		LD	B, #DIG0	
1006 066E 9A36	2	SIGU6:	LD	[B+], #036	
1007 0670 9DFE	3		LD	A, B	
1008 0672 9207	2		IFEQ	A, #DIG1	
1009 0674 BD0C7C	4		SBIT	LIMLCD, RAMBIT	
1010 0677 9C62	3		X	A, RECODI	
1011 0679 BC1023	3		LD	LETRERO, #023	
1012 067C BC117C	3		LD	LETRERO+1, #07C	
1013 067F 920A	2		IFEQ	A, #DIG4	
1014 0681 BC6206	3		LD	RECODI, #DIG0	
1015 0684 8E	5		RET		
1016					
1017					
1018 0685		KEY25:			;tecla "5"
1019 0685 BD0C7A	4		SBIT	REPETI, RAMBIT	
1020 0688 BD0C71	4		IFBIT	CODI, RAMBIT	
1021 068B 04	3		JP	SIGU5	
1022 068C BD0C79	4		SBIT	CODI, RAMBIT	
1023 068F 59	1		LD	B, #DIG0	
1024 0690 9A35	2	SIGU5:	LD	[B+], #035	
1025 0692 9DFE	3		LD	A, B	
1026 0694 9207	2		IFEQ	A, #DIG1	
1027 0696 BD0C7C	4		SBIT	LIMLCD, RAMBIT	
1028 0699 9C62	3		X	A, RECODI	
1029 069B BC1023	3		LD	LETRERO, #023	

```

1030 069E BC117C 3 LD LETRERO+1, #07C
1031 06A1 920A 2 IFEQ A, #DIG4
1032 06A3 BC6206 3 LD RECODI, #DIG0
1033 06A6 8E 5 RET
1034
1035 06A7 KEY26 ,tecla "4"
1036 06A7 BD0C7A 4 SBIT REPETI, RAMBIT
1037 06AA BD0C71 4 IFBIT CODI, RAMBIT
1038 06AD 04 3 JP SIGU4
1039 06AE BD0C79 4 SBIT CODI, RAMBIT
1040 06B1 59 1 LD B, #DIG0
1041 06B2 9A34 2 SIGU4: LD [B+], #034
1042 06B4 9DFE 3 LD A, B
1043 06B6 9207 2 IFEQ A, #DIG1
1044 06B8 BD0C7C 4 SBIT LIMLCD, RAMBIT
1045 06BB 9C62 3 X A, RECODI
1046 06BD BC1023 3 LD LETRERO, #023
1047 06C0 BC117C 3 LD LETRERO+1, #07C
1048 06C3 920A 2 IFEQ A, #DIG4
1049 06C5 BC6206 3 LD RECODI, #DIG0
1050 06C8 8E 5 RET
1051
1052 06C9 KEY32 ,tecla "3"
1053 06C9 BD0C7A 4 SBIT REPETI, RAMBIT
1054 06CC BD0C71 4 IFBIT CODI, RAMBIT
1055 06CF 04 3 JP SIGU3
1056 06D0 BD0C79 4 SBIT CODI, RAMBIT
1057 06D3 59 1 LD B, #DIG0
1058 06D4 9A33 2 SIGU3 LD [B+], #033
1059 06D6 9DFE 3 LD A, B
1060 06D8 9207 2 IFEQ A, #DIG1
1061 06DA BD0C7C 4 SBIT LIMLCD, RAMBIT
1062 06DD 9C62 3 X A, RECODI
1063 06DF BC1023 3 LD LETRERO, #023
1064 06E2 BC117C 3 LD LETRERO+1, #07C
1065 06E5 920A 2 IFEQ A, #DIG4
1066 06E7 BC6206 3 LD RECODI, #DIG0
1067 06EA 8E 5 RET
1068
1069 06EB KEY33 ,tecla "2"
1070 06EB BD0C7A 4 SBIT REPETI, RAMBIT
1071 06EE BD0C71 4 IFBIT CODI, RAMBIT
1072 06F1 04 3 JP SIGU2
1073 06F2 BD0C79 4 SBIT CODI, RAMBIT
1074 06F5 59 1 LD B, #DIG0
1075 06F6 9A32 2 SIGU2: LD [B+], #032
1076 06F8 9DFE 3 LD A, B
1077 06FA 9207 2 IFEQ A, #DIG1
1078 06FC BD0C7C 4 SBIT LIMLCD, RAMBIT
1079 06FF 9C62 3 X A, RECODI
1080 0701 BC1023 3 LD LETRERO, #023
1081 0704 BC117C 3 LD LETRERO+1, #07C
1082 0707 920A 2 IFEQ A, #DIG4
1083 0709 BC6206 3 LD RECODI, #DIG0
1084 070C 8E 5 RET
1085
1086 070D KEY34 ,tecla "1"
1087 070D BD0C7A 4 SBIT REPETI, RAMBIT
1088 0710 BD0C71 4 IFBIT CODI, RAMBIT
1089 0713 04 3 JP SIGU1
1090 0714 BD0C79 4 SBIT CODI, RAMBIT
1091 0717 59 1 LD B, #DIG0
1092 0718 9A31 2 SIGU1: LD [B+], #031
1093 071A 9DFE 3 LD A, B
1094 071C 9207 2 IFEQ A, #DIG1
1095 071E BD0C7C 4 SBIT LIMLCD, RAMBIT
1096 0721 9C62 3 X A, RECODI
1097 0723 BC1023 3 LD LETRERO, #023
1098 0726 BC117C 3 LD LETRERO+1, #07C
1099 0729 920A 2 IFEQ A, #DIG4
1100 072B BC6206 3 LD RECODI, #DIG0
1101 072F 8E 5 RET
1102
1103 072F KEY40

```

pertenece a un sensor normalmente

1104 072F 89	1	IFNC	;abierto(NO).
1105 0730 8D	5	RETSK	
1106 0731 BC0E02	3	LD TEMP, #002	
1107 0734 9EC8	2	LD [B], #0C8	; "ALARMA AUXILIAR *** ACTIVADA ***"
1108 0736 BD0C69	4	RBIT CODI, RAMBIT	
1109 0739 BD0B7D	4	SBIT DESBIT4, RAMBITI	
1110 073C BD0B74	4	IFBIT PRExSEN, RAMBITI	
1111 073F 04	3	JP TEC40	
1112 0740 BDDC68	4	RBIT SIRENA, PORTD	
1113 0743 03	3	JP PROX40	
1114 0744 BDDC78	4 TEC40:	SBIT SIRENA, PORTD	
1115 0747 BDEE7C	4 PROX40:	SBIT TRUN, CNTRL	
1116 074A 2900	3	JMP CAPTU1	
1117			
1118 074C	KEY41:		; pertenece a un sensor normalmente
1119 074C 89	1	IFNC	;abierto(NO).
1120 074D 8D	5	RETSK	
1121 074E BC0E03	3	LD TEMP, #003	
1122 0751 9EB3	2	LD [B], #0B3	; "***LINEA 01-N.O***"
1123 0753 BD0C69	4	RBIT CODI, RAMBIT	
1124 0756 BD0B7D	4	SBIT DESBIT4, RAMBITI	
1125 0759 BD0B74	4	IFBIT PRExSEN, RAMBITI	
1126 075C 04	3	JP TEC41	
1127 075D BDDC68	4	RBIT SIRENA, PORTD	
1128 0760 03	3	JP PROX41	
1129 0761 BDDC78	4 TEC41:	SBIT SIRENA, PORTD	
1130 0764 BDEE7C	4 PROX41:	SBIT TRUN, CNTRL	
1131 0767 2A00	3	JMP CAPTU2	
1132			
1133 0769	KEY42:		; pertenece a un sensor normalmente
1134 0769 89	1	IFNC	;abierto(NO).
1135 076A 8D	5	RETSK	
1136 076B BC0E04	3	LD TEMP, #004	
1137 076E 9EC9	2	LD [B], #0C9	; "alinea 02-N.O***"
1138 0770 BD0C69	4	RBIT CODI, RAMBIT	
1139 0773 BD0B7D	4	SBIT DESBIT4, RAMBITI	
1140 0776 BD0B74	4	IFBIT PRExSEN, RAMBITI	
1141 0779 04	3	JP TEC42	
1142 077A BDDC68	4	RBIT SIRENA, PORTD	
1143 077D 03	3	JP PROX42	
1144 077E BDDC78	4 TEC42:	SBIT SIRENA, PORTD	
1145 0781 BDEE7C	4 PROX42:	SBIT TRUN, CNTRL	
1146 0784 2B00	3	JMP CAPTU3	
1147			
1148 0786	KEY48:		
1149 0786 89	1	IFNC	
1150 0787 8D	5	RETSK	
1151 0788 BC0E05	3	LD TEMP, #005	
1152 078B 9E88	2	LD [B], #0B8	; "VENTANA DE BANO PLANTA BAJA"
1153 078D BD0C69	4	RBIT CODI, RAMBIT	
1154 0790 BD0B74	4	IFBIT PRExSEN, RAMBITI	
1155 0793 04	3	JP TEC48	
1156 0794 BDDC68	4	RBIT SIRENA, PORTD	
1157 0797 03	3	JP PROX48	
1158 0798 BDDC78	4 TEC48:	SBIT SIRENA, PORTD	
1159 079B BDEE7C	4 PROX48:	SBIT TRUN, CNTRL	
1160 079E 2B00	3	JMP CAPTU3	
1161			
1162 07A0	KEY49		
1163 07A0 89	1	IFNC	
1164 07A1 8D	5	RETSK	
1165 07A2 BC0E06	3	LD TEMP, #006	
1166 07A5 9E09	2	LD [B], #009	; "VENT.DE RECAMARA PLANTA BAJA"
1167 07A7 BD0C69	4	RBIT CODI, RAMBIT	
1168 07AA BD0B74	4	IFBIT PRExSEN, RAMBITI	
1169 07AD 04	3	JP TEC49	
1170 07AE BDDC68	4	RBIT SIRENA, PORTD	
1171 07B1 03	3	JP PROX49	
1172 07B2 BDDC78	4 TEC49:	SBIT SIRENA, PORTD	
1173 07B5 BDEE7C	4 PROX49:	SBIT TRUN, CNTRL	
1174 07B8 2C00	3	JMP CAPTU4	
1175			
1176 07BA	KEY50:		
1177 07BA 89	1	IFNC	

1178 07BB 8D	5	RETSK	
1179 07BC BD0C70	4	IFBIT PTA, RAMBIT	
1180 07BF 01	3	JP ABIER	
1181 07C0 8D	5	RETSK	
1182 07C1 BC0E07	3	ABIER LD TEMP, #007	
1183 07C4 9E29	2	LD [B], #029	;"PUERTA DE CALLE ABIERTA"
1184 07C6 BD0C69	4	RBIT CODI, RAMBIT	
1185 07C9 BD0C78	4	SBIT SIRENA, PORTD	;genera retardo de 60 seg.
1186 07CC BDEE7C	4	SBIT TRUN, CNTRL	
1187 07CF 2C00	3	JMP CAPTU4	
1188			
1189 07D1		KEY51	
1190 07D1 89	1	IFNC	
1191 07D2 8D	5	RETSK	
1192 07D3 BC0E08	3	LD TEMP, #008	
1193 07D6 9EAC	2	LD [B], #0AC	;"PUERTA DE PATIO ABIERTA"
1194 07D8 BD0C69	4	RBIT CODI, RAMBIT	
1195 07DB BD0C78	4	SBIT SIRENA, PORTD	;genera retardo de 60 seg.
1196 07DE BDEE7C	4	SBIT TRUN, CNTRL	
1197 07E1 2B00	3	JMP CAPTU3	
1198			
1199 07E3		KEY52:	
1200 07E3 BC0E09	3	LD TEMP, #009	
1201 07E6 9E2B	2	LD [B], #02B	;"SABOTAJE SENSOR-PRESENCIA P.BAJA"
1202 07E8 BD0C69	4	RBIT CODI, RAMBIT	
1203 07EB BD0B74	4	IFBIT PRExSEN, RAMBITI	
1204 07EE 04	3	JP TEC52	
1205 07EF BD0C68	4	RBIT SIRENA, PORTD	
1206 07F2 03	3	JP PROX52	
1207 07F3 BD0C78	4	TEC52 SBIT SIRENA, PORTD	
1208 07F6 A1	1	PROX52: SC	
1209 07F7 BDEE7C	4	SBIT TRUN, CNTRL	
1210 07FA 2A00	3	JMP CAPTU2	
1211			
1212 07FC		KEY53	
1213 07FC 89	1	IFNC	
1214 07FD 8D	5	RETSK	
1215 07FE BD0B76	4	IFBIT DENOCH, RAMBITI	
1216 0801 8D	5	RETSK	
1217 0802 BC0E0A	3	LD TEMP, #00A	
1218 0805 9E6A	2	LD [B], #06A	;"INTRUSO EN AREA PRINCIPAL BAJA"
1219 0807 BD0C69	4	RBIT CODI, RAMBIT	
1220 080A BD0C78	4	SBIT SIRENA, PORTD	;genera retardo de 60 seg.
1221 080D BDEE7C	4	SBIT TRUN, CNTRL	
1222 0810 2B00	3	JMP CAPTU3	
1223			
1224 0812		KEY54:	
1225 0812 BC0E0B	3	LD TEMP, #00B	
1226 0815 9E4D	2	LD [B], #04D	;"SABOTAJE SENSOR-PRESENCIA P.ALTA"
1227 0817 BD0C69	4	RBIT CODI, RAMBIT	
1228 081A BD0B74	4	IFBIT PRExSEN, RAMBITI	
1229 081D 04	3	JP TEC54	
1230 081E BD0C68	4	RBIT SIRENA, PORTD	
1231 0821 03	3	JP PROX54	
1232 0822 BD0C78	4	TEC54 SBIT SIRENA, PORTD	
1233 0825 A1	1	PROX54 SC	
1234 0826 BDEE7C	4	SBIT TRUN, CNTRL	
1235 0829 2A00	3	JMP CAPTU2	
1236			
1237 082B		KEY55	
1238 082B 89	1	IFNC	
1239 082C 8D	5	RETSK	
1240 082D BD0B76	4	IFBIT DENOCH, RAMBITI	
1241 0830 8D	5	RETSK	
1242 0831 BC0E0C	3	LD TEMP, #00C	
1243 0834 9E88	2	LD [B], #088	;"INTRUSO EN SALA PLANTA ALTA"
1244 0836 BD0C69	4	RBIT CODI, RAMBIT	
1245 0839 BD0C78	4	SBIT SIRENA, PORTD	;genera rotardo de 60 seg
1246 083C BDEE7C	4	SBIT TRUN, CNTRL	
1247 083F 2C00	3	JMP CAPTU4	
1248			
1249 0841		KEYG4	
1250 0841 89	1	IFNC	
1251 0842 8D	5	RETSK	

1252 0843 BC0E0D	3	LD	TEMP, #00D	
1253 0846 9EA9	2	LD	[B], #0A9	;"VENTANA DE BANO PLANTA ALTA"
1254 0848 BD0C69	4	RBIT	CODI, RAMBIT	
1255 0848 BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1256 084E 04	3	JP	TEC64	
1257 084F BD0C68	4	RBIT	SIRENA, PORTD	
1258 0852 03	3	JP	PROX64	
1259 0853 BD0C78	4	TEC64:	SBIT SIRENA, PORTD	
1260 0856 BDEE7C	4	PROX64:	SBIT TRUN, CNTRL	
1261 0859 2900	3	JMP	CAPTU1	
1262				
1263 085B		KEY65:		
1264 085B BC0E0E	3	LD	TEMP, #00E	
1265 085E 9E2B	2	LD	[B], #02B	;"LINEA ANTISABOTAJE SIRENA"
1266 0860 BD0C69	4	RBIT	CODI, RAMBIT	
1267 0863 BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1268 0866 04	3	JP	TEC65	
1269 0867 BD0C68	4	RBIT	SIRENA, PORTD	
1270 086A 03	3	JP	PROX65	
1271 086B BD0C78	4	TEC65:	SBIT SIRENA, PORTD	
1272 086E A1	1	PROX65:	SC	
1273 086F BDEE7C	4	SBIT	TRUN, CNTRL	
1274 0872 2B00	3	JMP	CAPTU3	
1275				
1276 0874		KEY66:		
1277 0874 89	1	IFNC		
1278 0875 8D	5	RETSK		
1279 0876 BC0E0F	3	LD	TEMP, #00F	
1280 0879 9E68	2	LD	[B], #068	;"DOMO DE RECAMARA PRINCIPAL ROTO"
1281 087B BD0C69	4	RBIT	CODI, RAMBIT	
1282 087E BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1283 0881 04	3	JP	TEC66	
1284 0882 BD0C68	4	RBIT	SIRENA, PORTD	
1285 0885 03	3	JP	PROX66	
1286 0886 BD0C78	4	TEC66:	SBIT SIRENA, PORTD	
1287 0889 BDEE7C	4	PROX66:	SBIT TRUN, CNTRL	
1288 088C 2C00	3	JMP	CAPTU4	
1289				
1290 088E		KEY67:		
1291 088E 89	1	IFNC		
1292 088F 8D	5	RETSK		
1293 0890 BC0E10	3	LD	TEMP, #010	
1294 0893 9E6F	2	LD	[B], #06F	;"PTA.BALCON ABIERTA, VENT. REC-PEQ"
1295 0895 BD0C69	4	RBIT	CODI, RAMBIT	
1296 0898 BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1297 089B 04	3	JP	TEC67	
1298 089C BD0C68	4	RBIT	SIRENA, PORTD	
1299 089F 03	3	JP	PROX67	
1300 08A0 BD0C78	4	TEC67:	SBIT SIRENA, PORTD	
1301 08A3 BDEE7C	4	PROX67:	SBIT TRUN, CNTRL	
1302 08A6 2A00	3	JMP	CAPTU2	
1303				
1304 08A8		KEY68:		
1305 08A8 89	1	IFNC		
1306 08A9 8D	5	RETSK		
1307 08AA BC0E11	3	LD	TEMP, #011	
1308 08AD 9EA7	2	LD	[B], #0A7	;"PTA-TARRAZA, VENT. REC-PRINCIPAL"
1309 08AF BD0C69	4	RBIT	CODI, RAMBIT	
1310 08B2 BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1311 08B5 04	3	JP	TEC68	
1312 08B6 BD0C68	4	RBIT	SIRENA, PORTD	
1313 08B9 03	3	JP	PROX68	
1314 08BA BD0C78	4	TEC68:	SBIT SIRENA, PORTD	
1315 08BD BDEE7C	4	PROX68:	SBIT TRUN, CNTRL	
1316 08C0 2C00	3	JMP	CAPTU4	
1317				
1318 08C2		KEY69:		
1319 08C2 89	1	IFNC		
1320 08C3 8D	5	RETSK		
1321 08C4 BC0E12	3	LD	TEMP, #012	
1322 08C7 9E91	2	LD	[B], #091	;"PROTEC.TRASERA; RECA.DOMO-ESC, BA"
1323 08C9 BD0C69	4	RBIT	CODI, RAMBIT	
1324 08CC BD0B74	4	IFBIT	PRExSEN, RAMBITI	
1325 08CF 04	3	JP	TEC69	

```

1326 08D0 BDDC68 4 RBIT SIRENA, PORTD
1327 08D3 03 3 JP PROX69
1328 08D4 BDDC78 4 TEC69 SBIT SIRENA, PORTD
1329 08D7 BDEE7C 4 PROX69 SBIT TRUN, CNTRL
1330 08DA 2A00 3 JMP CAPTU2
1331
1332 0900 =0900
1333 0900 CAPTU1
1334 0900 AE 1 LD A, [B]
1335 0901 8A 1 INC A
1336 0902 A6 1 X A, [B]
1337 0903 A4 3 LAID
1338 0904 922F 2 IFEQ A, #02F ;J = 02F
1339 0906 8E 5 RET
1340 0907 B2 3 X A, [X+]
1341 0908 F7 3 JP CAPTU1
1342 0909 414C4152 .BYTE 'ALARMA AUXILIAR-*** ACTIVADA ***!'
090D 4D412041
0911 5558494C
0915 4941522D
0919 2A2A2A20
091D 41435449
0921 56414441
0925 202A2A2A
0929 7C2F
1343 092B 454E4345 .BYTE 'ENCENDIDO D"REF-LECTOR AUTOMATIC!'
092F 4E444944
0933 4F204427
0937 5245462D
093B 4C454354
093F 4F522041
0943 55544F4D
0947 41544943
094B 7C2F
1344 094D 454E4345 .BYTE 'ENCENDIDO D"REF-LECTOR MANUAL!'
0951 4E444944
0955 4F204427
0959 5245462D
095D 4C454354
0961 4F52204D
0965 414E5541
0969 4C7C2F
1345 096C 50554552 .BYTE 'PUERTA DE CALLE OPERANDO!'
0970 54412044
0974 45204341
0978 4C4C4520
097C 20202020
0980 4F504552
0984 414E444F
0988 7C2F
1346 098A 50554552 .BYTE 'PUERTA DE CALLE NO-OPERANDO!'
098E 54412044
0992 45204341
0996 4C4C4520
099A 20204E4F
099E 2D4F5045
09A2 52414E44
09A6 4F7C2F
1347 09A9 56454E54 .BYTE 'VENTANA DE BANO PLANTA ALTA!'
09AD 414E4120
09B1 44452042
09B5 414E4F20
09B9 2020504C
09BD 414E5441
09C1 20414C54
09C5 417C2F
1348 09C8 414C4152 .BYTE 'ALARMA**MOVIMI-ENTO** ACTIVADA!'
09CC 4D41782A
09D0 2A4D4F56
09D4 494D492D
09D8 454E544F
09DC 2A2A2041
09E0 43544956
09E4 4144417C

```

```

09E8 2F
1349
1350 0A00           ;=0A00
1351 0A00           CAPTU2:
1352 0A00 AE       1      LD   A, [B]
1353 0A01 8A       1      INC  A
1354 0A02 A6       1      X   A, [B]
1355 0A03 A4       3      LAID
1356 0A04 922F    2      IFEQ A, #02F           ; I = 02F
1357 0A06 8E       5      RET
1358 0A07 B2       3      X   A, [X+]
1359 0A08 F7       3      JP  CAPTU2
1360 0A09 2A2A2A2A .BYTE  "**** SISTEMA ****NO-OPERANDO****"
      0A0D 20534953
      0A11 54454D41
      0A15 202A2A2A
      0A19 2A2A4E4F
      0A1D 2D4F5045
      0A21 52414E44
      0A25 4F2A2A2A
      0A29 7C2F
1361 0A2B 5341424F .BYTE  'SABOTAJE SENSOR-PRESENCIA P.BAJA/'
      0A2F 54414A45
      0A33 2053454E
      0A37 534F522D
      0A3B 50524553
      0A3F 454E4349
      0A43 4120502E
      0A47 42414A41
      0A4B 7C2F
1362 0A4D 5341424F .BYTE  'SABOTAJE SENSOR-PRESENCIA P.ALTA/'
      0A51 54414A45
      0A55 2053454E
      0A59 534F522D
      0A5D 50524553
      0A61 454E4349
      0A65 4120502E
      0A69 414C5441
      0A6D 7C2F
1363 0A6F 5054412E .BYTE  'PTA.BALCON ABIE-RTA,VENT.REC-PEQ/'
      0A73 42414C43
      0A77 4F4E2041
      0A7B 4249452D
      0A7F 5254412C
      0A83 56454E54
      0A87 2E524543
      0A8B 2D504651
      0A8F 7C2F
1364 0A91 50524F54 .BYTE  'PROTEC.TRASERA; RECA,DOMO-ESC,BAJ/'
      0A95 45432E54
      0A99 52415345
      0A9D 52413B2D
      0AA1 52454341
      0AA5 2C444F4D
      0AA9 4F2D4553
      0AAD 432C4241
      0AB1 7C2F
1365 0AB3 2A2A4C49 .BYTE  'ALINEA 01-N O**/'
      0AB7 4E45412D
      0ABB 30312D4E
      0ABF 2E4F2A2A
      0AC3 7C2F
1366 0AC5 2A2A204C .BYTE  '*** LUZ DE LCD *** ENCENDIDA **/'
      0AC9 555A2044
      0ACD 45204C43
      0AD1 44202A2A
      0AD5 202A2A20
      0AD9 454E4345
      0ADD 4E444944
      0AE1 41202A2A
      0AE5 7C2F
1367
1368 0B00           ;=0B00
1369 0B00           CAPTU3:

```

```

1370 0B00 AE      1      LD  A,[B]
1371 0B01 BA      1      INC  A
1372 0B02 A6      1      X   A,[B]
1373 0B03 A4      3      LAID
1374 0B04 922F    2      IFEQ A,#02F          ,I = 02F
1375 0B06 8E      5      RET
1376 0B07 B2      3      X   A,[X+]
1377 0B08 F7      3      JP  CAPTU3
1378 0B09 434F4E54  BYTE 'CONTRASENA NO IDENTIFICADA.1&?/'
      0B0D 52415345
      0B11 4E41204E
      0B15 4F202020
      0B19 4944454E
      0B1D 54494649
      0B21 43414441
      0B25 2E21263F
      0B29 7C2F
1379 0B2B 4C494E45  BYTE 'LINEA ANTISABOT-AJE SIRENA!/'
      0B2F 4120414E
      0B33 54495341
      0B37 424F542D
      0B3B 414A4520
      0B3F 20534952
      0B43 454E417C
      0B47 2F
1380 0B48 2A2A2A20 .BYTE '*** SISTEMA ****OPERANDO D*NOCHE!/'
      0B4C 53495354
      0B50 454D4120
      0B54 2A2A2A2A
      0B58 4F504552
      0B5C 414E444F
      0B60 2044274E
      0B64 4F434845
      0B68 7C2F
1381 0B6A 494E5452  BYTE 'INTRUSO EN AREA PRINCIPAL BAJA!/'
      0B6E 55534F20
      0B72 454E2041
      0B76 52454120
      0B7A 20505249
      0B7E 4E434950
      0B82 414C2042
      0B86 414A417C
      0B8A 2F
1382 0B8B 56454E54  BYTE 'VENTANA DE BANO PLANTA BAJA !/'
      0B8F 414E4120
      0B93 44452042
      0B97 414E4F20
      0B9B 2020504C
      0B9F 414E5441
      0BA3 2042414A
      0BA7 4120207C
      0BAB 2F
1383 0BAC 50554552  BYTE 'PUERTA DE PATIO ABIERTA!/'
      0BB0 54412044
      0BB4 45205041
      0BB8 54494F20
      0BBC 20202020
      0BC0 41424945
      0BC4 5254417C
      0BC8 2F
1384 0BC9 2A2A4C49  BYTE 'ALINEA 02-N O**!/'
      0BCD 4E454120
      0BD1 3032204E
      0BD5 2E4F2A2A
      0BD9 7C2F
1385 ;
1386 0C00          = 0C00
1387 0C00          CAPTU4
1388 0C00 AE      1      LD  A [B]
1389 0C01 BA      1      INC  A
1390 0C02 A6      1      X   A [B]
1391 0C03 A4      3      LAID
1392 0C04 922F    2      IFEQ A #02F          I = 02F
1393 0C06 8E      5      RET

```



```

1394 0C07 B2      3      X   A, [X+];
1395 0C08 F7      3      JP  CAPTU4
1396 0C09 56454E54 .BYTE 'VENT.DE RECAMARA PLANTA BAJA/'
      0C0D 2E444520
      0C11 52454341
      0C15 4D415241
      0C19 20202050
      0C1D 4C414E54
      0C21 41204241
      0C25 4A417C2F
1397 0C29 50554552 .BYTE 'PUERTA DE CALLE ABIERTA/'
      0C2D 54412044
      0C31 45204341
      0C35 4C4C4520
      0C39 20202020
      0C3D 20414249
      0C41 45525441
      0C45 7C2F
1398 0C47 2A2A2053 .BYTE '*** SISTEMA ***ENOPERACION TOTAL/'
      0C4B 49535445
      0C4F 4D41202A
      0C53 2A2A454E
      0C57 4F504552
      0C5B 4143494F
      0C5F 4E20544F
      0C63 54414C7C
      0C67 2F
1399 0C68 444F4D4F .BYTE 'DOMO DE RECAMARAPRINCIPAL ROTO/'
      0C6C 20444520
      0C70 52454341
      0C74 4D415241
      0C78 5052494E
      0C7C 43495041
      0C80 4C20524F
      0C84 544F7C2F
1400 0C88 494E5452 .BYTE 'INTRUSO EN SALA PLANTA ALTA/'
      0C8C 55534F20
      0C90 454E2053
      0C94 414C4120
      0C98 2020504C
      0C9C 414E5441
      0CA0 20414C54
      0CA4 417C2F
1401 0CA7 5054412D .BYTE 'PTA-TARRAZA,VENT.REC.PRINCIPAL/'
      0CAB 54415252
      0CAF 415A412C
      0CB3 56454E54
      0CB7 2E524543
      0CBB 2D505249
      0CBF 4E434950
      0CC3 414C7C2F
1402 0CC7 2A2A204C .BYTE '*** LUZ DE LCD *** APAGADA ***/'
      0CCB 555A2044
      0CCF 45204C43
      0CD3 44202A2A
      0CD7 202A2A20
      0CDB 41504147
      0CDF 41444120
      0CE3 2A2A7C2F
1403      :
1404 0D00      =0D00
1405 0D00      NOTECL:
1406 0D00 DC10  3      LD  X, #LETRERO
1407 0D02 BD0C7E 4      SBIT NOTECLA, RAMBIT
1408 0D05 AE      1 NOTECL: LD  A, [B]
1409 0D06 8A      1      INC  A
1410 0D07 A5      1      X   A, [B]
1411 0D08 A4      3      LAID
1412 0D09 922F   2      IFEQ A, #02F ; / = 02F
1413 0D0B 8D      5      RETSK
1414 0D0C B2      3      X   A, [X+];
1415 0D0D F7      3      JP  NOTECL
1416 0D0E 554E414D .BYTE 'UNAM, FACULTAD- DE INGENIERIA. /'
      0D12 2C204641

```

```

0D16 43554C54
0D1A 41442D20
0D1E 44452049
0D22 4E47454E
0D26 49455249
0D2A 412E2E2E
0D2E 7C2F
1417 0D30 54455349      BYTE 'TESIS PROFESION-AL QUE PRESENTO I/'
0D34 53205052
0D38 4F464553
0D3C 494F4E2D
0D40 414C2051
0D44 55452050
0D48 52455345
0D4C 4E544F20
0D50 7C2F
1418 0D52 524F4245      BYTE 'ROBERTO ROMERO Y ALFONSO ANTUNEZ I/'
0D56 52544F20
0D5A 524F4D45
0D5E 524F2059
0D62 20414C46
0D66 4F4E534F
0D6A 20414E54
0D6E 554E455A
0D72 7C2F
1419 0D74 454C2053      BYTE 'EL SISTEMA DETE-CTA LA VIOLACION I/'
0D78 49535445
0D7C 4D412044
0D80 4554452D
0D84 43544120
0D88 4C412056
0D8C 494F4C41
0D90 43494F4E
0D94 7C2F
1420 0D96 44274341      BYTE 'D\CADA DISP D"SEG. INSTALADO I/'
0D9A 44412044
0D9E 4953502E
0DA2 20442753
0DA6 45472E20
0DAA 494E5354
0DAE 414C4144
0DB2 4F7C2F
1421 0DB5 454C2053      BYTE 'EL SIST SE ACT-IVARA EN 60 SEG I/'
0DB9 4953542E
0DBD 20534520
0DC1 4143542D
0DC5 49564152
0DC9 4120454E
0DCC 20363020
0DD1 5345472E
0DD5 7C2F
1422 0DD7 4144454C      BYTE 'ADELANTE, PUEDE ACTIVAR EL SIST I/'
0DDB 414E5445
0DDF 2C205055
0DE3 45444520
0DE7 41435449
0DEB 56415220
0DEF 454C2053
0DF3 4953542E
0DF7 7C2F
1423
1424 0E00      =0E00
1425 0E00      NOTECL1
1426 0E00 DC10 3      LD X, #LETRERO
1427 0E02 BD0C7E 4      SBIT NOTECLA, RAMBIT
1428 0E05 AE 1 NOTECL1 LD A, [B]
1429 0E06 8A 1      INC A
1430 0E07 A6 1      X A, [B]
1431 0E08 A4 3      LAID
1432 0E09 922F 2      IFEQ A, #02F , I - 02F
1433 0E0B 8E 5      RET
1434 0E0C B2 3      X A [X+]
1435 0E0D F1 3      JP NOTECL1
1436 0E0F 554F 2053      BYTE 'UN SINCERO AGRA-DECIMI ENTO A LAS I/'

```

```

0E12 494E4345
0E16 524F2041
0E1A 4752412D
0E1E 44454349
0E22 4D49454E
0E26 544F2041
0E2A 204C4153
0E2E 7C2F
1437 0E30 50455253 .BYTE 'PERSONAS QUE CO-LABORARON PARA,/'
0E34 4F4E4153
0E38 20515545
0E3C 20434F2D
0E40 4C41424F
0E44 5241524F
0E48 4E205041
0E4C 52412E2E
0E50 7C2F
1438 0E52 51554520 .BYTE 'QUE ESTE PROYEC-TO SE LLEVARA.../'
0E56 45535445
0E5A 2050524F
0E5E 5945432D
0E62 544F2053
0E66 45204C4C
0E6A 45564152
0E6E 412E2E2E
0E72 7C2F
1439 0E74 41204341 .BYTE 'A CABO,ING. MIG-UEL A. CRUZ LEON,/'
0E78 424F2C49
0E7C 4E472E20
0E80 4D49472D
0E84 55454C20
0E88 412E2043
0E8C 52555A20
0E90 4C454F4E
0E94 7C2F
1440 0E96 5920454C .BYTE 'Y EL SR. RICARDOROMERO ANGUIANO,/'
0E9A 2053522E
0E9E 20524943
0EA2 4152444F
0EA6 524F4D45
0EAA 524F2041
0EAE 4E475549
0EB2 414E4F2C
0EB6 7C2F
1441 0EB8 2A4D5543 .BYTE '*MUCHAS GRACIAS*A TODOS Y ESPEC-/'
0EBC 48415320
0EC0 47524143
0EC4 4941532A
0EC8 4120544F
0ECC 444F5320
0ED0 59204553
0ED4 5045432D
0ED8 7C2F
1442 0EDA 49414C4D .BYTE 'IALMENTE A MI- MADRE, GRACIAS../'
0EDE 454E5445
0EE2 2041204D
0EE6 492D2020
0EEA 4D414452
0EEE 452C2047
0EF2 52414349
0EF6 41532E2E
0EFA 7C2F
1443 0EFC .END INICIO

```

**** Errors: 0, Warnings: 0

```

Checksum: 0x68ED
Byte Count: 0x0E47 (3655)
Input File: testisf.asm
Output File: testisf.obj
Memory Model: Small
Chip: 880

```

Este archivo es invocado en el programa fuente, predeclara etiquetas para localidades de memoria frecuentemente usadas por el programador

CHIP 880 , Indica la última dirección de memoria UVEPROM (0FFF Hex)

DECLARACIÓN DE REGISTROS

PORTLD = 0D0 Hex , Puerto L salida de datos
PORTLC = 0D1 Hex , Puerto L configuración
PORTLP = 0D2 Hex , Puerto L entrada de datos

PORTGD = 0D4 Hex , Puerto G salida de datos
PORTGC = 0D5 Hex , Puerto G configuración
PORTGP = 0D6 Hex , Puerto G entrada de datos

PORTD = 0DC Hex , Puerto D
PORTI = 0D7 Hex , Puerto I

PORTCD = 0D8 Hex , Puerto C salida de datos
PORTCC = 0D9 Hex , Puerto C configuración
PORTCP = 0DA Hex , Puerto C entrada de datos

SIOR = 0E9 Hex , Registro de corrimiento
TMRLO = 0EA Hex , Byte bajo del "TIMER"
TMRHI = 0EB Hex , Byte alto del "TIMER"
TAULO = 0EC Hex , Registro del "TIMER" byte bajo
TAUHI = 0ED Hex , Registro del "TIMER" byte alto

CNTRL = 0EE Hex , Registro de Control
PSW = 0EF , Registro PSW ("Procesor Status Word")

DECLARACIÓN DE CONSTANTES Y BANDERAS.

Definición de las funciones alteras de los bits del Puerto G

INT = 0 , Entrada para interrupción externa
TIO = 3 , Entrada/Salida para el registro de corrimiento ("TIMER")
SO = 4 , Salida del puerto serial
SK = 5 , Reloj del puerto serial
SI = 6 , Entrada del puerto serial
CKO = 7 , Salida de reloj / Quitar modo "Halt" (paro de reloj), ahorro de energía

Definición de los bits del registro CNTRL

TC1 = 7 , Selección del modo de operación del "TIMER"
TC2 = 6 , Selección del modo de captura
TC3 = 5 , Selección de la polaridad del flanco de entrada
TRUN = 4 , Inicio/paro del "TIMER"
MSEL = 3 , Habilitación del puerto serial ("Microwire")
IEDG = 2 , Selección de la polaridad del flanco para la interrupción externa
S1 = 1 , Selección del divisor para el reloj del "Microwire"
S0 = 0 , Selección del divisor para el reloj del "Microwire"

Definición de los bits del registro PSW

HC = 7 , Bandera de medio acarreo "Half carry"
C = 6 , Bandera de acarreo
TPND = 5 , Interrupción por "TIMER" pendiente
ENT1 = 4 , Interrupción por "TIMER" habilitada
IPND = 3 , Interrupción externa pendiente
BUSY = 2 , "Microwire" ocupado
ENI = 1 , Interrupción externa habilitada
GIE = 0 , Habilitación de todas las interrupciones

CAPÍTULO 4

RESULTADOS Y CONCLUSIONES

El diseño y construcción del Sistema de Alarma y Seguridad se empezó a realizar estudiando las características del microcontrolador de la Familia COP8, el sistema de desarrollo y la Arquitectura RISC.

Dado que el microcontrolador cuenta con 40 terminales (pins), al ser utilizado en el Sistema de Alarma y Seguridad se requería de un "Display" de cristal líquido, un teclado y sensores. La cantidad de sensores depende del área a proteger y de las necesidades de los usuarios de la casa, por tal motivo se decidió multiplexar los puertos L y el puerto I de 8 bits cada uno, en los cuales se colocaron un teclado, sensores y las líneas antisabotaje.

Al empezar a diseñar el sistema de seguridad y alarma se consideró usar sensores normalmente abiertos dado que en los puertos multiplexados de 8 x 8 se pueden usar hasta 64 interruptores entre sensores y teclas. Los sensores de movimiento y el sensor de vibración que se consiguieron para este sistema de seguridad cuentan con interruptores normalmente cerrados. El inconveniente de usar los sensores NC es que sólo se pueden usar hasta ocho sensores de este tipo, por lo que se requirió de otras seis entradas del puerto G para completar el número de sensores requeridos de otras seis entradas del sistema de seguridad no está limitado por el número de sensores, tomando en cuenta que cada sensor se encuentra conectado a una línea de control y que cada línea de control despliega en el "Display" un sólo letrero de aviso. De cualquier manera el sistema de seguridad y alarma cuenta con otras tres líneas disponibles en las cuales se pueden conectar sensores con interruptor normalmente abierto, u otros dispositivos como: sensores de movimiento o sensores de humo que manejen este tipo de interruptores; se conecta este tipo de dispositivos en las líneas NO debido a que un intruso no se interesa por desconectar estos dispositivos de seguridad al momento de invadir una área protegida.

Es conveniente desarrollar un sistema de seguridad y alarma basado en las existencias de los dispositivos de seguridad en el mercado para posteriormente, si el sistema es requerido por otros usuarios, no tener problemas en la obtención de los sensores y tampoco en la modificación del programa.

El sistema de seguridad y alarma está hecho de acuerdo a las necesidades y requerimientos del usuario, para lograr esto siempre se frecuentó y se consultó a éste de tal forma que se logró construir un sistema de seguridad que cumple con su propósito y que dejó satisfecho al usuario.

Dado que el sistema de seguridad está construido en base a un microcontrolador, es posible modificar el programa, los datos, y en algunos casos el hardware si el sistema es instalado por ejemplo en un automóvil, o para satisfacer una necesidad de protección específica.

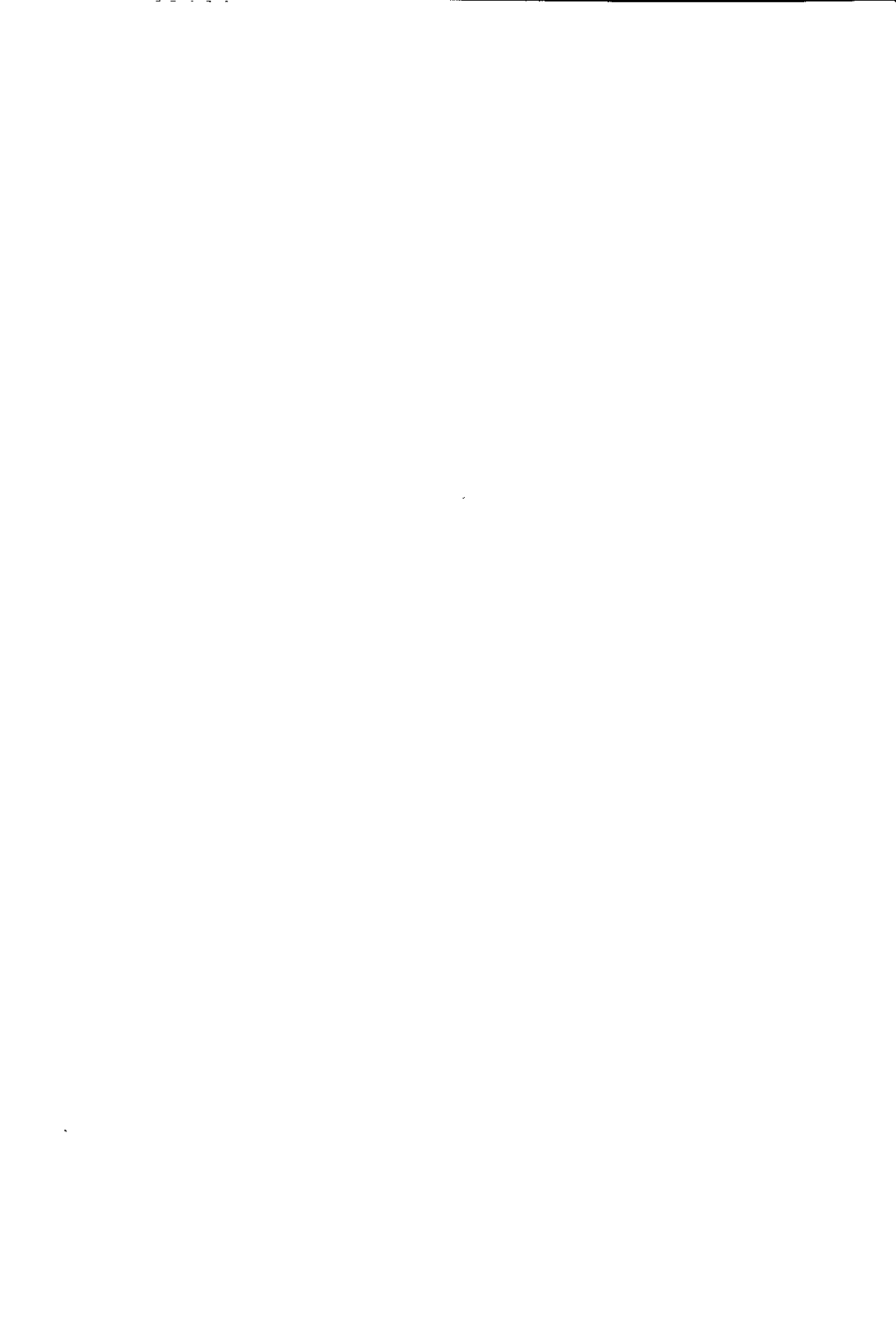
El hecho de que se tenga un teclado numérico interactuando con un microcontrolador en un sistema de seguridad y alarma permite que se puedan activar una serie de funciones específicas, previamente programadas, con tan sólo teclear el número correspondiente a la función, de esta manera el usuario puede participar también aportando ideas al programador sobre algunas acciones en particular que se quiere tenga el sistema de seguridad; esto hace que el sistema sea multifuncional.

Después de medir la corriente máxima de 100 mA que demanda la batería cuando se encuentra cargando, uno puede pensar que no tiene sentido diseñar el circuito autocargador de batería si ésta no demanda más de los 240 mA, que es la corriente máxima que controla el circuito autocargador, pero puede suceder que en algún momento

la batería se dañe y se ponga en corto, en este caso el circuito autocargador limita la corriente a los 240 mA y de esta manera se evita dañar los componentes. Otra de las razones por lo cual es necesario contar con este circuito, es que cuando la batería se dañe y se reemplace por otra con diferentes características puede ser que en esta ocasión la batería sí demande una corriente de inicio de carga hasta los 240 mA, lo cual provoca que la vida de la batería se acorte.

APÉNDICE A

INTRODUCCIÓN A LA FAMILIA COP8



APÉNDICE B

DESCRIPCIÓN GENERAL DEL MICROCONTROLADOR COP8780

B INTRODUCCIÓN

La Familia Básica del microcontrolador COP8 cuenta con Memoria de Programa, Memoria de Datos, Puertos I/O configurables, un Temporizador interno (Timer) y una interfase "MICROWIRE/PLUS". La presencia de la memoria interna y periféricos en este circuito proporciona soluciones a muchas aplicaciones.

La organización de memoria de la Familia Básica COP8 está basada en la arquitectura Harvard, en la cual la Memoria de Programa es diferente de la Memoria de Datos. Cada una de estos dos tipos de memoria, tiene su propio espacio y usa su propio canal interno de direcciones. En contraste con la arquitectura Von Neumann, la Memoria de Programa y la Memoria de Datos comparten el mismo canal de direcciones.

Excepto por el Acumulador (A) y el Contador de Programa (PC); todos los Registros: puertos de I/O y Memoria RAM, son mapeados, en el espacio de direcciones de la Memoria de Datos, entre éstos se encuentra: el registro B, el registro X, el Apuntador de Pila (SP), y los registros correspondientes a cada puerto I/O. Todos los registros mapeados pueden ser accedidos para leer o escribir en su dirección de memoria.

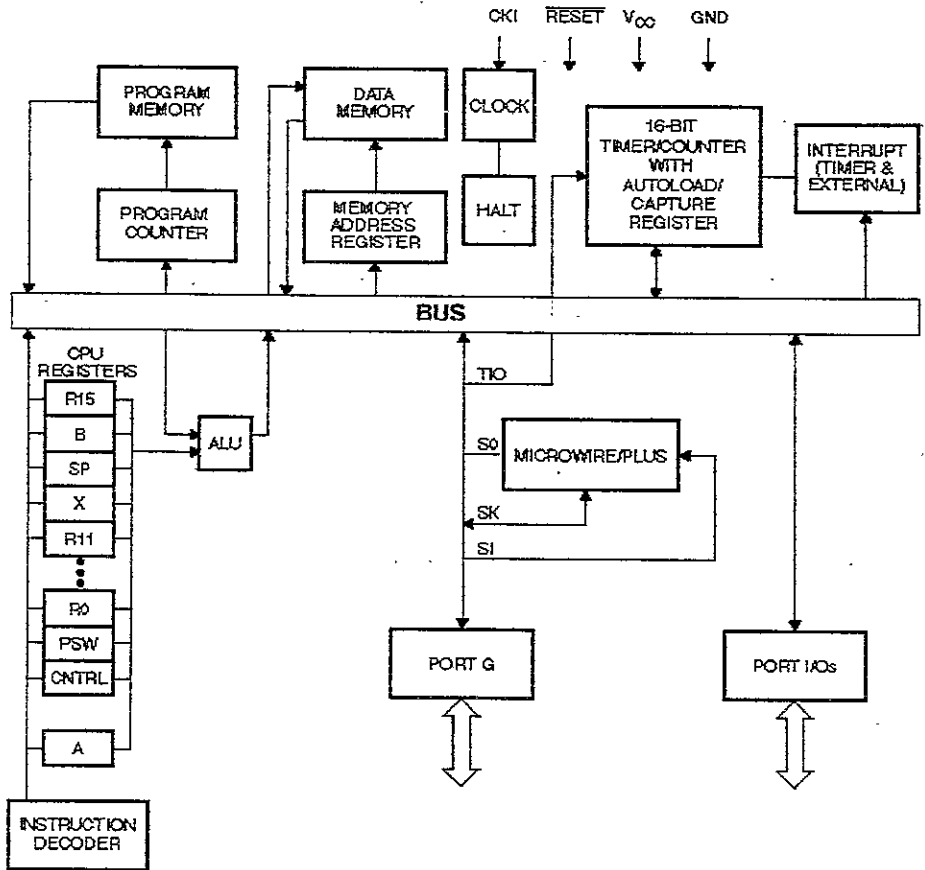
La arquitectura de la familia básica COP8 proporciona un perfeccionamiento a la arquitectura Harvard: Una instrucción llamada Carga Indirecta del Acumulador (Load Acumulador Indirect, LAID), que da acceso a tablas de datos almacenadas en la memoria del programa. Una arquitectura Harvard convencional no tiene esta característica. Es por esto que se le asignó el nombre de Arquitectura Harvard Modificada.

El dispositivo de la Familia Básica COP8 se comunica con otros dispositivos a través del puerto I/O configurable, o a través de la interfase serie "MICROWIRE/PLUS". Los puertos I/O están designados por nombres de letras tal como: Puerto C, Puerto D, Puerto G, Puerto Y y Puerto L. Un Temporizador de 16 bits de propósito general, asociado con un registro de Autocarga/Captura de 16 bits. El Temporizador puede ser configurado para operar en cualquiera de estos tres modos: Modulación por Ancho de Pulso (PWM), Contador de Eventos Externo y Modo de Entrada de Captura.

Tres diferentes interrupciones están disponibles en el dispositivo. la Interrupción Externa Mascarable, Interrupción del Temporizador (Timer) Mascarable y la Interrupción "Software Trap" No Mascarable. Todas las interrupciones causan saltos a direcciones específicas en el programa de memoria, el código de programa hacia éstas direcciones determina la prioridad relativa de las interrupciones mascarables.

B.1 DIAGRAMA A BLOQUES

El diagrama a bloques de la arquitectura de la Familia Básica de COP8 se muestra a continuación. Todos los dispositivos de la Familia Básica contienen los elementos mostrados en el diagrama. Estos elementos incluyen: la Unidad Lógica Aritmética (ALU), Memoria de Datos, Memoria de Programa, Temporizador 1 (Timer 1), el Registro de Corrimiento "MICROWIRE/PLUS", puertos I/O e interrupciones lógicas. Los bloques no comunes en todos los miembros de la Familia Básica no se muestran en el diagrama A.1.



TSP-COP820-01

Diagrama B.1 Bloques del Microcontrolador de la Familia Básica COP8 de National Semiconductor.

B.3 ORGANIZACION DE LA MEMORIA

Los microcontroladores de la Familia Básica COP8 están basados en una arquitectura Harvard Modificada. Este tipo de arquitectura separa la Memoria de Programa de la Memoria de Datos. Cada tipo de memoria tiene su propio espacio de direcciones, canal de direcciones y canal de datos. La siguiente sección describe la estructura de la memoria del microcontrolador de la Familia Básica COP8.

B.3.1 Memoria de programa

La memoria de programa de la Familia Básica COP8 es un bloque de memoria no volátil ROM o memoria EPROM, que mantiene las instrucciones del programa o datos. El rango de direcciones de la memoria de programa es de 32 KBytes. Un Contador de Programa de 16 bits (PC) es utilizado para direccionar la memoria de programa, que se subdivide en segmentos de 4 KBytes, con respecto a ciertas instrucciones.

La división de segmentos de 4 KByte dentro de la memoria de programa, está relacionada con las instrucciones de 2 Bytes: Salto Absoluto (JMP) y Salto a Subrutina (JSR); estas instrucciones causan que los 12 bits más bajos del PC sean reemplazados por el valor que se especificó en la instrucción, mientras los 3 bits superiores quedan inalterados. Así, estas instrucciones se ramifican sólo dentro del actual segmento de direcciones de 4 KBytes de la memoria de programa

Las instrucciones indirectas, Salto Indirecto (JID) y Carga Indirecta del Acumulador (LAID), funcionan solo con bloques de la memoria de programa de 256 Bytes (0FF Hex). Esta restricción existe porque solo los 8 bits bajos del PC (PCL) son reemplazados durante la búsqueda de las tablas en la memoria de programa. Los 8 bits superiores del PC (PC Up) permanecen sin cambios. Reemplazando solo el PCL se minimiza el tiempo de ejecución de esta instrucción. Los programadores deben garantizar que las instrucciones LAID y JID y las tablas asociadas no crucen el límite de 256 Bytes de la memoria de programa.

El Salto Relativo (Jump Relative, JP) es una instrucción muy económica, es completamente independiente de los bloques de la memoria de programa y los límites del segmento de memoria. Esta instrucción JP de un Byte permite saltar hacia adelante desde 2 a 32 localidades y desde 0 a 31 localidades hacia atrás (con el cero se representa un ciclo cerrado infinito), relativos al contenido actual del contador de programa. Un salto hacia adelante de uno no está permitido, para este caso puede implementarse con la instrucción NOP.

B.3.2 Memoria de Datos

La Memoria de Datos de la Familia Básica COP8 consiste de algunos bloques de RAM y/o memoria EEPROM. Los dispositivos que incluyen más de 128 Bytes de RAM, utilizan un registro de extensión del segmento de datos, para incrementar el direccionamiento de datos más allá de los primeros 128 Bytes. Esto es necesario porque el registro de direccionamiento de memoria (MAR) es utilizado para acceder a todas las localidades de memoria en solo 8 bits

El segmento base de memoria de datos de la Familia Básica COP8 puede verse como dos secciones: el rango de direcciones bajas de 0000 a 006F Hex y el rango de direcciones altas de 0080 a 00FF Hex. El segmento bajo de memoria, contiene la pila (SP) del programa y la memoria de datos de propósito general. El rango de direcciones altas contiene el registro de datos y los registros de los puertos I/O mapeados en memoria, registros de control, temporizador (Timer) y el registro de captura asociado, el registro de corrimiento "MICROWIRE/PLUS", etc.

La memoria de datos es directamente direccionable por instrucciones o indirectamente por los registros apuntadores B, X y SP. El conjunto de instrucciones de la Familia Básica COP8 permite fijar cualquier bit en la memoria de datos, para fijar, inicializar o probar. Todos los puertos I/O, registros, apuntadores y contadores en la Familia COP8 (excepto A y PC) son mapeados en la memoria de datos. Por consiguiente, todos los bits de I/O y registros pueden fijarse individualmente, inicializar y probar.

Dieciséis Bytes de RAM son mapeados como "registros" de propósito general, en las direcciones 00F0 a 00FF Hex ; ciertas instrucciones trabajan solo con estos registros; mientras otras instrucciones son más eficientes cuando se utilizan con estos registros en lugar de otro tipo de memoria. Los tres registros apuntadores X, SP y B son mapeados en el espacio de la memoria de registros en las direcciones de 00FC a 00FE Hex, respectivamente. En los dispositivos de la Familia Básica COP8 con más de 128 Bytes de RAM, el registro de extensión del segmento de datos es mapeado en la localidad 00FF Hex.

Las primeras dieciséis localidades de la memoria de almacenamiento de datos (0000 a 000F Hex) tienen un significado especial para la instrucción de carga B con el dato inmediato. Esta instrucción es extremadamente eficiente para cargar el apuntador B con la dirección en este rango, porque es de un Byte sencillo, instrucción de un ciclo sencillo. Cargando B con las direcciones y/o valores mayores que 000F Hex, requiere de dos Bytes, instrucción de tres ciclos.

Toda la RAM, puertos de I/O, contadores (counter) y registros (excepto A y PC) son mapeados en el espacio de direcciones de la memoria de datos como se muestra en la tabla B.1.

Address	Contents
00-6F	On-chip RAM Address Space
70-BF	On-chip Data Memory Address Space
C0-CF	I/O and Register Address Space
D0	Port L Data Register
D1	Port L Configuration Register
D2	Port L Input Pins (read only)
D3	Reserved for Port L
D4	Port G Data Register
D5	Port G Configuration Register
D6	Port G Input Pins (read only)
D7	Port I Input Pins (read only)
D8	Port C Data Register
D9	Port C Configuration Register
DA	Port C Input Pins (read only)
DB	Reserved for Port C
DC	Port D Data Register
DE	Reserved for Port D
DF	Reserved for Port D
E0-E8	On-chip Functions and Registers
E9	MICROWIRE shift register
EA	Timer 1 Lower Byte
EB	Timer 1 Upper Byte
EC	Timer 1 Autoload Register Lower Byte
ED	Timer 1 Autoload Register Upper Byte
EE	CNTRL Control Register
EF	PSW Register
F0 to FF	On-chip RAM mapped as Registers
FC	X Register
FD	SP Register
FE	B Register
FF	S Register or General Purpose Register

Tabla B.1 Mapa de Memoria de Datos

B.4 MAPEO DE LA MEMORIA DE REGISTROS DE I/O

Los dispositivos de la Familia Básica COP8 tienen tres tipos de puertos: entrada/salida configurables, salidas dedicadas y entradas dedicadas. Cada puerto de entrada/salida puede especificarse en el mapa de memoria de registros de I/O y con direcciones asociadas con él, dependiendo del tipo de puerto. La siguiente sección describe la estructura de registros para cada tipo de puerto I/O.

Todos los registros y terminales, son mapeados en el espacio de direcciones, en el almacenamiento de la memoria de datos. Por consiguiente, las instrucciones que operan en la memoria de datos, también operan en los registros de los puertos y terminales. Estas instrucciones se usan para fijar, inicializar y probar los bits individualmente. Las

direcciones de los registros, para especificar los puertos de I/O se encuentran listados en el mapa de memoria en la tabla B.1.

B.4.1 Puertos Entrada/Salida Configurables

Cada puerto de entrada/salida configurable tiene tres registros asociados: en la tabla B.2 se tiene un registro de configuración del puerto y un registro de datos. Estos dos registros mapeados en memoria dejan que las terminales del puerto puedan ser individualmente configurables como entrada o salida y pueden ser cambiados individualmente de uno a otro (entrada a salida o viceversa) por software. El registro de configuración es utilizado para configurar las terminales como entrada o salida. Una terminal puede ser configurada como entrada escribiendo un "0" o como salida escribiendo un "1" al bit asociado del registro de configuración. Si una terminal es configurada como una salida, el bit asociado del registro de datos, representa el estado de la terminal (1 = alto, 0 = bajo). Si la terminal es configurada como entrada, el bit asociado del registro de datos determina si la terminal es un "pull-up" o entrada de alta impedancia. La configuración del puerto y los registros de datos son de lectura/escritura.

Una tercera dirección de la memoria de datos es asignada a cada puerto I/O. Leyendo esta dirección de memoria regresa el valor de la terminal del puerto indiferentemente de como se configuraron las terminales.

Configuration Bit	Data Bit	Port Pin Setup
0	0	Hi-Z input (TRI-STATE output)
0	1	Input with weak pull-up
1	0	Push-pull zero output
1	1	Push-pull one output

Tabla B.2 Configuración de puertos de I/O

B.4.2 Salidas Dedicadas

Los puertos que son salidas dedicadas tienen un registro asociado al puerto. Este registro mapeado en memoria es usado para fijar las terminales del puerto a un estado lógico alto o bajo. Una terminal del puerto puede ser individualmente fijada o inicializada escribiendo unos o ceros en el bit asociado del registro de datos. El registro de datos del puerto puede ser de lectura o escritura.

B.4.3 Entradas Dedicadas

Los puertos que son entradas dedicadas tienen un registro asociado. Sin embargo, una dirección de la memoria de datos es asignada a las terminales del puerto, para leer la entrada del puerto. Las direcciones de las terminales del puerto son localidades de memoria de solo lectura.

B.5 EL NÚCLEO DE LOS REGISTROS

Todos los microcontroladores de la Familia Básica COP8 comparten un bloque común de lógica, referido como el núcleo de microcontrolador de la Familia Básica COP8. Este núcleo incluye la Unidad Central de Proceso (CPU), el bloque del Temporizador 1 (Timer 1) y el bloque del "MICROWIRE/PLUS". Los registros contenidos dentro de estos bloques son los registros del núcleo. Los registros del CPU incluyen: un Contador de Programa (PC) de 16 bits, un Acumulador (A) de 8 bits, una Palabra de Estado del Procesador (PSW), un Registro de Control del Núcleo (CNTRL) y dieciséis registros de 8 bits de la memoria de programa. Todos los registros del núcleo son mapeados en el espacio de direcciones de la memoria de datos, excepto por el contador de programa (PC) y el acumulador (A).

B.5.1 Acumulador

Todos los dispositivos de la Familia COP8 tienen un acumulador de 8 bits. El acumulador es utilizado en todas las operaciones lógicas y aritméticas, como son ADD y XOR. Además, este es utilizado con las instrucciones de intercambio JID y LAID. Las instrucciones lógicas y aritméticas utilizan al acumulador, como registro de operandos y de resultados. Si se requiere un segundo registro de operando, esta el registro de instrucciones (IR) que contiene un dato inmediato o un registro en la memoria de datos.

B.5.2 Contador de programa

El CPU contiene un contador de programa de 16 bits utilizado en el direccionamiento de memoria de programa. Se inicializa a cero el PC y se incrementa una vez por cada Byte de una instrucción de código. Saltos, saltos a subrutinas, interrupciones y la instrucción JID causan que algunos o todos los bits de PC puedan ser remplazados. Las instrucciones de transferencia de control que reemplazan sólo algunos de los bits del PC tienen un rango de salto limitado.

El núcleo de la Familia Básica contiene dos registros de control de 8 bits (PSW y CNTRL). Los siguientes párrafos y tablas (B.3 y B.4) muestran el contenido de los bits en cada registro.

B.5.3 Registro PSW (Dirección 00EF Hex)

El registro Palabra del Estado del Procesador (PSW) contiene ocho diferentes bits de banderas (Ver tabla B.3). Los bits del registro son asignados como sigue:

GIE	Global interrupt enable
ENI	External interrupt enable
BUSY	MICROWIRE busy shifting flag
IPND	External interrupt pending
ENI1	Timer T1 interrupt enable

TPND Timer T1 interrupt pending
 C Carry Flip/Flop
 HC Half-Carry Flip/Flop

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HC	C	TPND	ENTI	IPND	BUSY	ENI	GIE

Tabla B.3 Registro PSW

B.5.4 Registro de Control CNTRL (Dirección 00EE Hex)

El registro de control "MICROWIRE" (CNTRL) contiene: "MICROWIRE/PLUS", selección del flanco de la Interrupción Externa (External Interrupt Edge) y las banderas de control del Temporizador (Timer Control flags), ver tabla B.4. Las banderas del "MICROWIRE/PLUS" incluye SL0 y SL1, que selecciona el factor de la división del reloj del MICROWIRE/PLUS y "MSEL" que selecciona el puerto G, nombradas G5 y G4 como SK y SO, respectivamente señales del "MICROWIRE/PLUS". La bandera de control de la Interrupción Externa (External Interrupt Edge) selecciona la polarización de la señal de entrada. La bandera de control del Temporizador (Timer) incluye el "TRUN", que es utilizado para iniciar o detener el "timer/counter" y tres señales de control del modo del temporizador (Timer Mode Control).

Los bits del registro de control "Timer" y el "MICROWIRE" son:

SL1 & SLO Select the MICROWIRE clock divide-by (00=2,01=4,1x=8)
 IEDG External interrupt edge polarity (0 = rising edge, 1 = falling edge)
 MSEL Selects G5 and G4 as MICROWIRE signals SK and SO, respectively
 TRUN Used to start and stop the timer/counter (1 = run, 0 = stop)
 TC1 Timer T1 Mode Control Bit
 TC2 Timer T1 Mode Control Bit
 TC3 Timer T1 Mode Control Bit

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1	TC2	TC3	TRUN	MSEL	IEDG	SL1	SLo

Tabla B.4 Registro de Control CNTRL

B.5.5 Registro de Datos

La Familia Básica COP8 contiene dieciséis localidades de registros de datos de 8 bits en la memoria de datos de la dirección 00F0 a la 00FF Hex. Cuatro de estos registros, 00FC a 00FF Hex, tienen funciones especiales. Las localidades de 00FC y 00FE Hex contienen los apuntadores de la memoria de datos de 8 bits X y B, respectivamente. La localidad 00FD contiene el apuntador de pila (SP) de 8 bits para la memoria de datos. La localidad 00FF esta reservada para la extensión del registro del segmento de datos, que es usada en algunos dispositivos de la Familia Básica COP8 para extender la memoria de datos más allá de 128 Bytes, este registro esta disponible para uso general en dispositivos

que contienen 128 Bytes o menos. Los restantes doce registros, de 00F0 a 00FB Hex, siempre están disponibles para uso de propósito general.

Ciertas instrucciones de la Familia Básica COP8 diferencian registros de datos de otras localidades de memoria de datos, tal como la instrucción DRSZ (Decrementa Registro y Salta si es Cero), que utiliza un byte por ciclo de reloj cuando es ejecutada utilizando el registro B. DRSZ resta "1" al registro de datos y salta a la siguiente instrucción, si el resultado del decremento es cero. Esta instrucción es sumamente útil en la construcción de ciclos, y hace que el registro de datos sea ideal para contadores

B.5.6 Apuntador de Pila

El Apuntador de Pila (SP) se encuentra mapeado en memoria en la localidad 00FD Hex de la memoria de datos. El apuntador de pila debe ser inicializado antes de que ocurra cualquier llamada a subrutina o una interrupción. Normalmente, el apuntador de pila es inicializado en la cima del segmento base de la memoria de datos. Para los dispositivos de 64 Bytes de RAM, ésta es la localidad memoria 002F Hex. Para los dispositivos con 128 Bytes de RAM, ésta es la localidad de memoria 006F Hex.

Cargar direcciones en la pila causa que se decremente hacia la dirección cero. Extraer direcciones de la pila causa, que esta se incremente hacia arriba (por ejemplo de "0" hacia 006F Hex). Si el apuntador de pila es inicializado hasta la parte superior del segmento base de memoria, sobreextraer de la pila causa un error de interrupción "Software Trap".

Si el límite inferior de la pila es la dirección 0000 Hex: Sobrecargar direcciones a la pila causa que la pila esté alrededor de la dirección 00FF y 00FE Hex (las llamadas a subrutina e interrupciones causan una carga de doble-Byte). Esto debe ser evitado porque interfiere con el apuntador B, que está mapeado en memoria en la localidad 00FE Hex.

El usuario puede inicializar el apuntador de pila en cualquier parte de la memoria del segmento base, la pila crece hacia la dirección cero, pero esta ya no tiene la interrupción "Software Trap" para protección de sobreextraer. Inicializar el apuntador de la pila en la parte superior del segmento base del registro de datos direccionadas en 00F0 a 00FB Hex, es potencialmente muy arriesgado. La memoria disponible de la pila es estrictamente limitada y si la pila se carga más allá de la localidad de memoria 00F0, ocurre interferencia con los registros de control PSW y CNTRL, que están mapeados en memoria en las localidades de memoria 00EF y 00EE Hex, respectivamente.

B.5.7 Apuntadores de Memoria de Datos (Registros Índice)

La Familia Básica COP8 contiene dos registros, X y B, que pueden ser usados como apuntadores. Estos registros permiten el direccionamiento indirecto de todas las localidades mapeadas en el espacio de direcciones de memoria de datos. En suma, estos registros pueden ser automáticamente incrementados o decrementados por ciertas instrucciones que usan el direccionamiento de registro indirecto. Las características de autodecremento y autoincremento permiten fácilmente al usuario el paso por localidades de memoria de datos.

B.5.8 Registro "MICROWIRE/PLUS"

El "MICROWIRE/PLUS", es un sistema de comunicaciones serie de tres líneas que contiene un registro de corrimiento serie de 8 bits (SIOR) mapeado en memoria. La entrada de datos serie y las señales de salida al registro SIOR son proporcionadas por SI y SO, respectivamente. El registro de corrimiento es cronometrado por la señal SK. Los datos son recorridos a través del SIOR del orden bajo al alto y al ocurrir el flanco de bajada de la señal de reloj SK.

B.5.9 Registros Temporizadores (Timer)

El núcleo de la Familia Básica COP8 contiene un bloque temporizador. El bloque del temporizador consiste de un temporizador/contador de 16 bits con un registro asociado autocarga/captura de 16 bits. Los Bytes altos y bajos direccionados para los registros: temporizador y autocarga/captura, se encuentran mapeados en memoria.

B.6 FUNCIONAMIENTO DEL CPU

Esta sección describe el funcionamiento de la Unidad Central de Procesos (CPU) del COP8 que se muestra en el diagrama B.2. A continuación se da una breve descripción de la lógica de control y la Unidad Lógica Aritmética.

B.6.1 Lógica de Control

La lógica de control (Ver diagrama B.2) del CPU controla virtualmente todas las operaciones dentro del dispositivo. Este incluye el contador de memoria, registro de direcciones de memoria, el registro de palabra de estado del procesador, y registros para almacenar información, también incluye la lógica para el direccionamiento "Fetch" de memoria, instrucciones de decodificación y ejecución, y manejo de interrupción/error. El recibe entradas del ALU y periféricos internos, incluyendo el temporizador (s) y la interfaz "MICROWIRE/PLUS" y señales generales de control para este y otras partes del dispositivo.

B.6.2 Unidad Lógica Aritmética (ALU)

La Unidad Lógica Aritmética (ALU) ejecuta las operaciones lógicas y aritméticas. Las entradas al ALU (Ver diagrama B.2) son proporcionadas por el acumulador, algunos sistemas de lógica-alambrada de datos constantes, los bits de acarreo y medio acarreo y el Registro de Memoria de Datos (MDR). Las entradas del ALU dadas para una instrucción están especificadas por el código de operación. El acumulador funciona como ambos fuente y destino para el ALU y es utilizada en todas las instrucciones lógicas y aritméticas. El siempre contiene el resultado de la última ejecución lógica, aritmética o la instrucción del acumulador carga/intercambio. La lógica alambrada de datos constantes, que incluye

0000, 0001 y 00FF Hex, son utilizadas en instrucciones como CLR A, INC A y DEC A. Estas instrucciones tienen un modo de direccionamiento implícito. Los bits de acarreo (C) y medio acarreo (HC) son utilizados en instrucciones como ADC y SUBC. Todas las instrucciones aritméticas y lógicas tienen dos operandos, utilizando el MDR (Memory Data Register) como una entrada al ALU. El MDR puede ser cargado con operandos desde la memoria de datos o registro (dato inmediato especificado en una instrucción). Desde el MDR, las instrucciones lógicas y aritméticas no se pueden ejecutar directamente en dos operandos de la memoria de datos y/o programa. Tal funcionamiento requiere un operando de memoria para ser cargado en el acumulador antes de ser ejecutada.

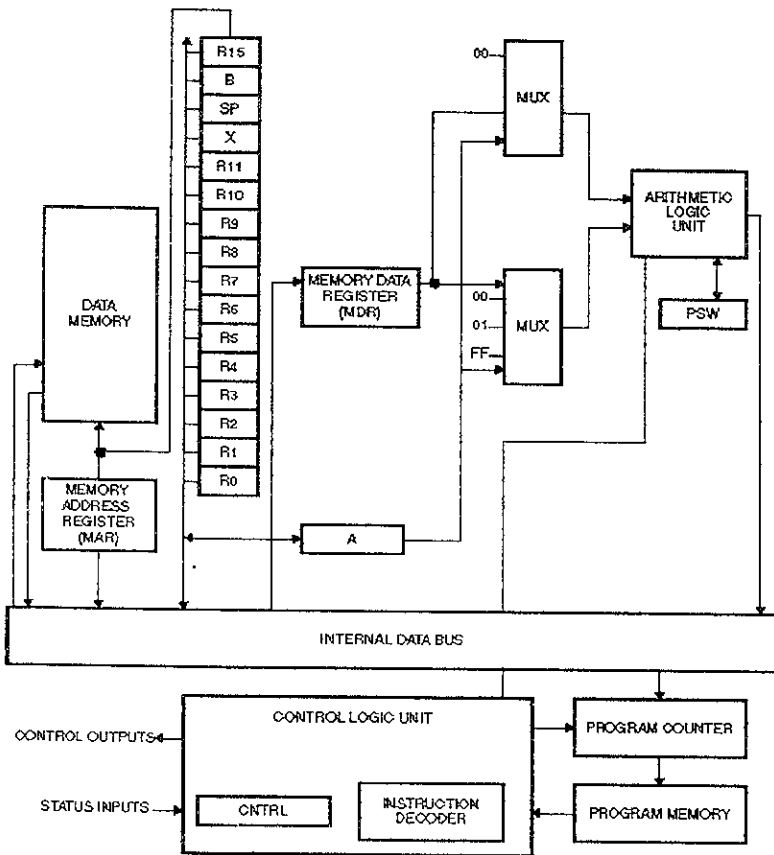


Diagrama B.2 Interfaz del CPU del COP8.

B.7 "RESET"

El microcontrolador de la Familia Básica COP8 entra al estado de "RESET" inmediatamente que detecta un cero lógico en la terminal de "RESET". Cuando la terminal de "reset" es puesta en alto, el dispositivo ejecuta el código en dos ciclos de instrucción. La terminal de "RESET" debe mantenerse en bajo con un mínimo de un ciclo de instrucción para garantizar un "RESET" válido. Durante el encendido, el "hardware" del usuario debe de garantizar que la terminal de "RESET" se mantenga en bajo hasta que el microcontrolador de la Familia Básica COP8 se encuentre dentro de las especificaciones de voltaje Vcc. Además, el usuario debe asegurar que el oscilador se estabilice a tiempo. Se recomienda un circuito R/C en la terminal de "RESET" con un retardo 5 veces mayor que el rizo de la fuente de poder.

Todos los microcontroladores de la Familia Básica COP8 contienen la lógica para inicializar sus circuitos internos durante el estado de "RESET".

- El contador de programa es cargado con 0000 Hex.
- Todos los bits de los registros PSW y CNTRL son inicializados. Este desactiva todas las interrupciones, detiene el temporizador 1 (Timer 1) y deshabilita el MICROWIRE/PLUS.

El acumulador y toda la memoria de datos y registros de datos, incluyendo los apuntadores B, X y SP, no son inicializados en el "RESET".

B.8 OPCIONES DE RELOJ

La mayor parte de la Familia Básica COP8 soporta tres opciones de reloj; un oscilador de cristal, un circuito CR y un oscilador externo. Dependiendo del tipo de dispositivo, la opción del reloj es seleccionada por medio de la máscara o programada por el usuario en el dispositivo.

La selección de la opción de un reloj en específico afecta la frecuencia de operación, la exactitud del reloj y el consumo de potencia de un dispositivo en particular

B.8.1 Oscilador de Cristal

La terminal dedicada CKI (entrada de reloj) y G7 (CKO) en los dispositivos de la Familia Básica COP8 pueden ser conectados para ser controlados por un Oscilador de Cristal. La terminal G7 usada como CKO, no esta disponible para uso de proposito general.

B.8.2 Circuito Oscilador RC

La terminal dedicada CKI, puede ser utilizada para contruir un oscilador RC. Con esta opción, G7 esta disponible como una estrada de proposito general. En el COP840CJ, tiene un capacitor interno.

B.8.3 Circuito Oscilador RC solo para el COP840CJ

La opción R/C del COP840CJ tiene un capacitor interno, donde el usuario sólo tiene que proporcionar la resistencia. Esto ayuda a reducir el ruido EMI y el costo al cliente.

El COP820CJ no tiene un capacitor interno, por lo tanto, el usuario tiene que proporcionar el capacitor y la resistencia.

La tabla B.5 muestra la tolerancia en la frecuencia de oscilación en el COP840CJ como función del valor de la componente (R). El capacitor es interno.

La tabla B.5 muestra los datos preliminares solamente para el COP840CJ.

R (kOhm)	CKI Freq. (MHz)	Instr. Cycle (msec)	Conditions
5.6	3.3 ±10%	3.0 ±10%	Vcc = 5 Volts
15	1.3 ±10%	7.7 ±10%	Vcc = 5 Volts
27	0.75 ±10%	13.3 ±10%	Vcc = 5 Volts

Tabla B.5 Configuración del Circuito Oscilador R/C

B.8.4 Oscilador Externo

La terminal dedicada CKI puede ser manejada para una señal de reloj externa, que debe de cumplir con las especificaciones de ciclo, tiempos de levantamiento/caída y niveles de entrada. Con esta opción, G7 esta disponible como entrada de propósito general.

B.9 MODOS DE DIRECCIONAMIENTO

El conjunto de instrucciones de la Familia Básica COP8 ofrece una variedad de métodos para especificar el direccionamiento de la memoria. Cada método es llamado "Modo de Direccionamiento". Estos métodos son clasificados en dos categorías:

- Modos de direccionamiento de "Operando"
- Modos de direccionamiento de "Transferencia de Control".

El modo de direccionamiento de operando tiene varios métodos de especificar una dirección para el acceso a datos (lectura o escritura). El modo de direccionamiento de transferencia de control es utilizado en conjunto con la instrucción "Jump" para control de la secuencia de ejecución del programa ("software").

B.9.1 Modo de Direccionamiento de Operando

El operando de una instrucción indica que localidad de memoria puede ser afectada por la instrucción. Algunos diferentes modos de direccionamiento de operando estan disponibles y permiten que localidades de memoria sean especificadas en una variedad de caminos. Una instrucción puede ser especificada como un direccionamiento directo o indirectamente para especificar un apuntador de registro, el contenido del

registro (en algunos casos dos registros) apuntan a una localidad de memoria deseada. En el modo "inmediato", el Byte de datos puede ser utilizado en la misma instrucción. Cada modo de direccionamiento tiene sus ventajas y desventajas, con respecto a su flexibilidad, velocidad de ejecución y tamaño reducido del programa. No todos los modos están disponibles en todas las instrucciones. La instrucción de carga ("LD") ofrece el número más grande de modos de direccionamiento.

Los modos de direccionamiento disponibles son:

- Directo
- Indirecto, registro B o X
- Indirecto con incremento/decremento
- Inmediato
- Inmediato corto
- Indirecto de la memoria de programa

Los modos de direccionamiento son descritos a continuación:

- **Directo**

El direccionamiento de la memoria es directamente especificado como un Byte en la instrucción. En lenguaje ensamblador, el direccionamiento directo es escrito como un valor numérico (o una etiqueta que ha sido definida en cualquier parte del programa con un valor numérico).

Ejemplo: "Load Accumulator Memory Direct"
LD A,05

Memoria Reg/Datos	Antes de ejecución	Después de ejecución
Acumulador	XX Hex	A6 Hex
Localidad de memoria 0005 Hex	A6 Hex	A6 Hex

- **Indirecto, Registro B o X.**

El direccionamiento de memoria es especificado por el registro B o X (registros apuntadores). En lenguaje ensamblador, la notación [B] o [X] especifican que el registro sirve como apuntador.

Ejemplo: "Exchange Memory with Accumulator, B Indirect"
X A, [B]

Memoria Reg/Datos	Antes de ejecución	Después de ejecución
Acumulador	XX Hex	87 Hex
Localidad de memoria 0005 Hex	87 Hex	XX Hex
Apuntador B	05 Hex	05 Hex

- **Indirecto, registro B o X con incremento/decremento (después de la ejecución).**

Este relevante modo de direccionamiento de memoria es especificado por el contenido del registro B o X (registros apuntadores). El registro apuntador es automáticamente incrementado o decrementado después de la ejecución, permite una fácil manipulación de bloques de memoria con ciclos de programas. En lenguaje ensamblador, la notación [B+], [B-], [X+], o [X-] especifica que registros se utilizan como apuntadores y si el apuntador puede ser incrementado o decrementado.

Ejemplo: "Exchange Memory with Accumulator, B Indirect with Post-Increment"
X A,[B+]

Memoria Reg/Datos	Antes de ejecución	Después de ejecución
Acumulador	03 Hex	62 Hex
Localidad de memoria 0005 Hex	62 Hex	03 Hex
Apuntador B	05 Hex	06 Hex

- **Inmediato**

Los datos para la operación continúan en el código de instrucción en la memoria de programa. En lenguaje ensamblador, el signo (#) de carácter de número indica una operación inmediata.

Ejemplo: "Load Accumulator Immediate"
LD A,#05

Memoria de Reg/Datos	Antes de Ejecución	Después de Ejecución
Acumulador	XX Hex	05 Hex

- **Inmediato corto**

Este es un caso especial de una instrucción inmediata. En la instrucción "Load B immediate", el valor inmediato de 4 bits es cargado en la instrucción, en la parte baja del registro B. La parte superior del registro B es inicializada a 0000 binario

Ejemplo: "Load B Register immediate Short"
LD B,#7

Memoria de Reg/Datos	Antes de Ejecución	Después de Ejecución
Apuntador B	XX Hex	07 Hex

- **Indirecto de la Memoria de Programa**

Este es un caso especial de un direccionamiento indirecto que permite acceso a tablas de datos almacenadas en la memoria de programa. En la instrucción "Load Accumulator Indirect" (LAID), los Bytes superiores y bajos del contador de programa (PCU y PCL) son utilizados temporalmente como un apuntador a la memoria de programa. Para propósito de acceso a la memoria de programa, el contenido del acumulador y el PCL son intercambiados. Los datos apuntados por el contador de programa son cargados en el acumulador, y simultáneamente, el contenido original del PCL es almacenado, por lo que el programa puede continuar su ejecución normal.

Ejemplo: "Load Accumulator Indirect"
LAID

Memoria de Reg/Datos	Antes de Ejecución	Después de Ejecución
PCU	04 Hex	04 Hex
PCL	35 Hex	36 Hex
Acumulador	1F Hex	25 Hex
Localidad de memoria 041F Hex	25 Hex	25 Hex

B.9.2 Modos de transferencia de control

Las instrucciones del programa son usualmente ejecutadas en orden secuencial. Sin embargo, las instrucciones de salto pueden cambiar la secuencia de ejecución normal. Algunos modos de direccionamiento están disponibles para especificar saltos de direcciones.

Un cambio en el flujo del programa requiere un no-incremento en el contenido del contador de programa. El contador de programa consiste de dos Bytes, designados como Byte superior (PCU) y Bytes bajo (PCL). El bit más significativo no es utilizado, se dejan 15 bits para direccionar la memoria de programa.

Diferentes modos de direccionamiento son utilizados para especificar una nueva dirección por el contador de programa. Los cambios de modos de direccionamiento dependen primeramente de la distancia del salto. Los saltos más lejanos algunas veces requieren más Bytes de instrucción en orden para especificar completamente el nuevo contenido del contador de programa

Los Modos de Direccionamiento de Transferencia de Control son:

- Salto Relativo
- Salto Absoluto
- Salto Absoluto Largo
- Salto Indirecto

Los modos de direccionamiento de Transferencia de Control son descritos a continuación. Cada descripción incluye un ejemplo de una instrucción de salto con un particular modo de direccionamiento.

- **Salto Relativo**

En esta instrucción de 1 Byte, seis de los bits del código de la instrucción especifican la distancia del salto de la localidad de la memoria de programa actual. La distancia del salto está en el rango de -31 a +32. La instrucción JP +1 no está permitida. El programador puede utilizar una instrucción NOP.

Ejemplo: "Jump Relative"
JP 0A

Registro	Antes de ejecutar	Después de ejecutar
PCU	02 Hex	02 Hex
PCL	05 Hex	0F Hex

- **Salto Absoluto**

En esta instrucción de 2 Bytes, 12 bits del código de la instrucción especifican el nuevo contenido del contador de programa. Los tres bits superiores del contador de programa permanecen sin cambio, restringiendo a la nueva dirección del contador de programa dentro de los mismos 4 KBytes de la instrucción actual. (Esta restricción es válida solo para los dispositivos que utilizan más de 4 KBytes de espacio en la memoria de programa).

Ejemplo: "Jump Absolute"
JMP 0125

Registro	Antes de ejecutar	Después de ejecutar
PCU	0C Hex	01 Hex
PCL	77 Hex	25 Hex

- **Salto Absoluto Largo**

En esta instrucción de 3 Bytes, 15 bits del código de la instrucción especifican el nuevo contenido del contador de programa.

Ejemplo: "Jump Absolute Long"
JMP 03625

Registro	Antes de ejecutar	Después de ejecutar
PCU	42 Hex	36 Hex
PCL	36 Hex	25 Hex

- **Salto indirecto**

En esta instrucción de un Byte, la parte baja de la dirección de salto es obtenida de una tabla almacenada en la memoria de programa, con el Acumulador funcionando como Byte de bajo orden de un apuntador en la memoria de programa. Para propósito de acceso a la memoria de programa, el contenido del acumulador es escrito al PCL (temporalmente), después los datos apuntados por el contador de programa (PCH/PCL) son cargados en el PCL, mientras el PCH queda inalterado.

Ejemplo: "Jump Indirect"
JID

Registro	Antes de ejecutar	Después de ejecutar
PCU	01 Hex	01 Hex
PCL	C4 Hex	32 Hex
Acumulador	26 Hex	26 Hex
Localidad de memoria 0126 Hex	32 Hex	32 Hex

B.10 MODO DE "HALT"

Los dispositivos soportan un modo de operación de ahorro de energía: "Halt". El microcontrolador es puesto en modo "Halt" configurando el bit 7 del registro de datos del puerto G, otro método es deteniendo el reloj en el caso de utilizar una señal externa de reloj. En el Modo "Halt" todos los procesos activos internos incluyendo el reloj oscilador son detenidos. La arquitectura antiestática mantiene un estado de control y toda la información hasta que continúe el programa. En el Modo "Halt" los requerimientos de potencia son mínimos. El voltaje aplicado Vcc puede bajar a Vr (voltaje mínimo para mantener la información en RAM) sin alterar el estado de la máquina.

Cuando se utiliza el método del Puerto G, existen dos caminos para salir del Modo "Halt": por medio del "RESET" o por la terminal G7. Un cero en la terminal de "RESET" reinicia el microcontrolador y comienza la ejecución del programa en la dirección 0000H. En los modos de: oscilador externo y circuito oscilador RC, una transición de bajo a alto en la terminal G7 debe causar que el microcontrolador salga del Modo "Halt". Excepto por el bit de Datos G7, que permite reiniciar, todas las localidades de memoria retienen los valores que tenían prioridad a la ejecución de la instrucción "Halt". Se requiere que después de una instrucción "Halt" continúe una instrucción "NOP", para sincronizar el reloj.

B.11 INTERRUPCIONES

Los dispositivos tienen una sofisticada estructura de interrupciones que permite fácilmente conectarse con el mundo real. Hay tres posibles fuentes de Interrupción, que se muestran a continuación:

- Una interrupción mascarable en una entrada externa G0 (con flanco de sensibilidad positivo o negativo controlado por "software").
- Una interrupción mascarable en el temporizador de sobreflujo o temporizador de captura.
- Una interrupción no-mascarable "software/error" en código cero (opcode zero).

B.11.1 Control de Interrupción

El bit GIE (Global Interrupt Enable) habilita la función de interrupción. esta es utilizada en conjunto con ENI y ENTI para seleccionar una o ambas fuentes de interrupción. Este bit (GIE) es inicializado cuando la interrupción es reconocida (Ver diagrama B.3).

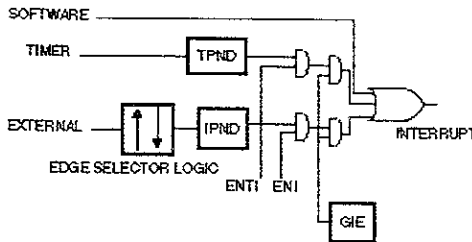


Diagrama B.3 Control de Interrupción.

Los bits ENI y ENTI seleccionan las interrupciones externa y temporizador respectivamente. Así, el usuario puede seleccionar una o ambas fuentes para interrumpir al microcontrolador cuando el GIE este habilitado.

IEDG selecciona el flanco de la interrupción externa (0=flanco de subida,1=flanco de bajada). El usuario puede tener una interrupción con los dos flancos de subida o de bajada para mantener el estado del bit IEDG después de cada interrupción.

IPND y TPND son señales para cuando una interrupción esta pendiente. Después que una interrupción es reconocida, el usuario puede probar estos dos bits para determinar si una interrupción esta pendiente. Esto permite que una interrupción tenga prioridad bajo "software". Las banderas de "pendiente" pueden ser limpiadas por el usuario. Se puede probar una interrupción configurando en alto el bit GIE dentro de una subrutina de interrupción.

La interrupción por "software" no inicializa el bit GIE, esto significa que el controlador puede ser interrumpido por otra fuente de interrupción mientras se sirve una interrupción por "software".

B.11.2 Procesamiento de una interrupción

La interrupción, una vez reconocida, almacena el contador de programa (PC) en la pila y el apuntador de pila (SP) es decrementado dos veces. El bit GIE (Global Interrupt Enable) es inicializado para deshabilitar otras interrupciones. El microcontrolador apunta a la dirección 00FFH y termina su ejecución de esa dirección. Este proceso toma 7 ciclos para completarse. Para que termine una subrutina de interrupción, cualquiera de las siguientes tres instrucciones regresan el proceso al programa principal: RET, RETSK o RETI. Cualquiera de las instrucciones puede sacar la pila del contador de programa (PC). El apuntador de pila es incrementado dos veces. La instrucción RETI además configura el GIE para rehabilitar las demás interrupciones.

Cualquiera de las tres interrupciones pueden regresar de una subrutina de interrupción por "hardware". La instrucción RETSK puede ser utilizada para regresar de una subrutina de interrupción por "software" para evitar entrar a un ciclo infinito.

B.11.3 Detección de Condiciones Ilegales

El microcontrolador incorpora "hardware" que permite detectar una condición ilegal, que puede ocurrir por errores de código, ruido y cuando cae el voltaje. Específicamente, detecta casos de fuera de ejecución, de áreas indefinidas de EPROM y situaciones de desbalanceo de la pila.

Al leer una localidad de área indefinida EPROM regresa un 00 Hex como contenido. El código de operación para una interrupción por "software" también es "00". Así, un acceso a un programa indefinido EPROM causa una interrupción por "software".

Leer una localidad de área indefinida RAM regresa un FF Hex. La subrutina de la pila en el dispositivo crece hacia abajo para cada llamada a subrutina. Para inicializar el apuntador de pila hasta la cima de la RAM; el primer retorno de instrucción desbalanceada, puede causar que la pila apunte a una dirección indefinida de RAM. Como resultado el programa puede intentar ejecutar un FFFF Hex, que es una localidad EPROM indefinida y puede disparar una interrupción por "software".

B.12 TIMER/COUNTER

El dispositivo tiene un temporizador de 16 bits con un registro asociado de 16 bits que habilita ejecutar funciones del temporizador. El temporizador T1 y el registro R1 son organizados cada uno en registros de lectura/escritura de 8 bits. Los bits de control del registro CNTRL permiten al temporizador comenzar y detenerse bajo el control del "software". La pareja registro-temporizador en uno de los tres posibles modos. La tabla B.6 muestra varias formas de operación del temporizador y los requisitos de configuración.

CNTRL Bits 7 6 5	Modo de Operación	Interrupción T	Inicio del Temporizador
0 0 0	External Counter W/Auto-Load Reg.	Timer Underflow	TIO Post.Edge
0 0 1	External Counter W/Auto-Load Reg.	Timer Underflow	TIO Neg. Edge
0 1 0	Not Allowed	Not Allowed	Not Allowed
0 1 1	Not Allowed	Not Allowed	Not Allowed
1 0 0	Timer W/Auto-Load Reg.	Timer Underflow	tc
1 0 1	Timer W/Auto-Load Reg./Toggle TIO Out	Timer Underflow	tc
1 1 0	Timer W/Capture Register	TIO Post. Edge	tc
1 1 1	Timer W/ Capture Register	TIO Neg. Edge	tc

Tabla B.6 Modos de Operación del Temporizador

B.12.1 Modo 1.- Temporizador con Registro de Auto Carga

En este modo de operación el temporizador T1 cuenta hacia abajo a una razón de un ciclo de instrucción. En sobreflujo el valor en el registro R1 hace que se recargue automáticamente el temporizador, continuando con el conteo hacia abajo. El temporizador de sobreflujo puede programarse para interrumpir al microcontrolador. Un bit en el registro de control CNTRL habilita la terminal TIO (G3) en cada interrupción del temporizador de sobreflujo. Esto permite la generación de señales de onda cuadrada a la salida o Modulación de Ancho de Pulsos (PWM) bajo el control de "software" (Ver diagrama B.4)

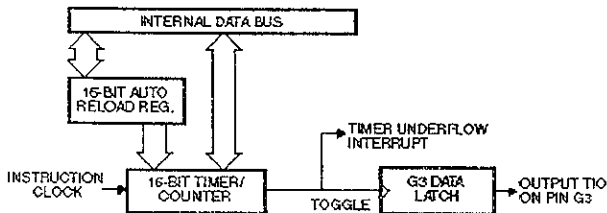


Diagrama B.4 Temporizador en Modo PWM.

B.12.2 Modo 2.- Contador de Eventos Externo

En este modo el temporizador T1 se hace un contador de eventos externo de 16 bits. El contador cuenta hacia abajo en un flanco, en la terminal TIO. Los bits de control en el registro CTNRL programan al contador para decrementarse en cada flanco de bajada o flanco de subida. En un sobreflujo el contenido del registro R1 es automáticamente copiado al contador. El sobreflujo puede ser programado para generar una interrupción (Ver diagrama B.5)

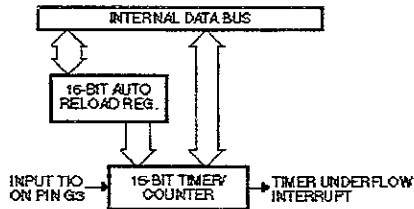


Diagrama B.5 Contador de Eventos Externo.

B.12.3 Modo 3.- Temporizador con Registro de Entrada de Captura

El temporizador T1 puede ser utilizado para mediciones precisas de frecuencias externas o eventos en este modo de operación. El temporizador T1 cuenta hacia abajo a razón de un ciclo de instrucción. En el momento en que ocurre el flanco especificado en la terminal TIO el contenido de la terminal T1 es copiado al registro R1. Los bits en el registro de control CNTRL permiten especificar el flanco como: flanco positivo o flanco negativo. En este modo el usuario puede elegir una interrupción en un flanco específico (Ver Diagrama B.6).

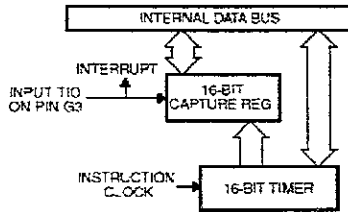


Diagrama B.6 Temporizador con Registro de Entrada de Captura.

APÉNDICE C

CONJUNTO DE INSTRUCCIONES DE LA FAMILIA COP8

El presente capítulo está pensado en las personas que quieren aprender el uso de las instrucciones para programar el microcontrolador de la Familia Básica COP8, dado que son muy pocas instrucciones se puede mencionar cada una de ellas y entender mejor su funcionamiento.

A continuación se define el conjunto de instrucciones de los miembros de la Familia Básica COP8. Las instrucciones de operaciones aritméticas, de comparación y transferencia de datos con los tres modos de direccionamiento (direccionamiento registro indirecto con el apuntador B, direccionamiento directo de memoria y direccionamiento inmediato) pueden incrementar el total de las instrucciones. En la tabla 3.0 al final del capítulo se encuentra el resumen del conjunto de instrucciones del COP8.

C.1 DESCRIPCIÓN DEL CONJUNTO DE INSTRUCCIONES DE LA FAMILIA BÁSICA COP8

C.1.1 ADC— Add with Carry

Sintaxis:

- a) ADC A,[B]
- b) ADC A,#
- c) ADC A,MD

Descripción: Suma con Acarreo el contenido de :

- a) La localidad de la memoria de datos referida por el apuntador B, con el acumulador.
- b) El valor inmediato encontrado en el segundo Byte de la instrucción con el acumulador.
- c) La localidad de la memoria de datos referida por el segundo Byte de la instrucción es sumada al contenido del acumulador, y el resultado simultáneamente se incrementa si la bandera de acarreo a sido habilitada anteriormente.

El resultado es colocado en el acumulador, y la bandera de acarreo se fija o restablece (reset), dependiendo de la presencia o ausencia del acarreo de los resultados. Similarmente, la bandera de medio acarreo se fija o restablece, dependiendo de la presencia o ausencia de un acarreo del "nibble" de bajo orden.

Operación:

$$A \leftarrow A + \text{VALUE} + C$$
$$C \leftarrow \text{CARRY}; \text{HC} \leftarrow \text{HALF CARRY}$$

C.1.2 ADD — Add

Sintaxis:

- a) ADD A,[B]
- b) ADD A,MD
- c) ADD A,#

Descripción: Suma el acumulador A con la localidad de memoria de datos referenciada por:

- a) El apuntador B
- b) La dirección en el segundo Byte de la instrucción

c) El valor inmediato que aparece en el segundo Byte de la instrucción es sumado al contenido del acumulador, y el resultado es colocado en el acumulador. Las banderas de acarreo y medio acarreo no cambian.

Operación:

$A \leftarrow A + \text{VALUE}$

C.1.3 AND — And

Sintaxis:

- a) AND A,[B]
- b) AND A,#
- c) AND A,MD

Descripción:

Realiza una operación AND entre los correspondientes bits del acumulador y :

- a) El contenido de la localidad de memoria de datos referida por el apuntador
- b) B.
- b) El valor inmediato aparece en el segundo Byte de la instrucción.
- c) El contenido de la localidad de memoria de datos referida por la dirección en el segundo Byte de la instrucción.

El resultado es guardado en el acumulador.

Operación:

$A \leftarrow A \text{ AND VALUE}$

C.1.4 CLR — Clear Accumulator

Sintaxis: CLR A

Descripción:

El acumulador es limpiado con ceros.

Operación:

$A \leftarrow 0$

C.1.5 DCOR — Decimal Correct

Sintaxis: DCOR A

Descripción:

Esta instrucción, cuando es usada después de un DC (suma con acarreo) o una instrucción SUBC (resta con acarreo) puede hacer una corrección decimal, para corregir el resultado de una suma binaria o resta. Note que la instrucción ADC puede ser precedida con una instrucción AAD A, #066 (suma 66 en hexadecimal) para corrección de suma decimal. Esta instrucción asume que dos operandos son en formato BCD (Código Binario Decimal) y da como resultado el mismo formato BCD. Las banderas de acarreo y medio acarreo permanecen sin cambio.

Operación:

$A (\text{BCD FORMAT}) \leftarrow A (\text{BINARY FORMAT})$

C.1.6 DEC — Decrement Accumulator

Sintaxis: DECA

Descripción:

Esta instrucción decrementa el contenido de el acumulador y deja el resultado en el acumulador. Las banderas de acarreo y medio acarreo permanecen sin cambio.

Operación:

A ← A - 1

C.1.7 DRSZ REG# — Decrement Register and Skip if Result is Zero

Sintaxis: DRSZ REG#

Descripción:

Esta instrucción decrementa el contenido del registro de la memoria seleccionada (selección por #, donde # = 0 a F) y regresa el resultado al mismo registro. Si el resultado es cero, salta la siguiente instrucción.

Esta instrucción es de gran uso cuando se desea repetir una secuencia de instrucciones un número de veces. El número de veces que se desea que la secuencia de instrucciones que pueden ser ejecutadas en el registro y una instrucción DRSZ que con este registro es codificado al final sigue la secuencia enseguida de la instrucción JP (Salto Relativo) los saltos regresan al inicio de la secuencia de la instrucción. La instrucción JP cada vez, alrededor de una instrucción de salto regresa hasta que el registro contador es decrementado a cero, al mismo tiempo la instrucción JP salta hacia fuera del ciclo del programa.

Operación:

```
REG ← REG - 1
IF (REG - 1) = 0,
THEN SKIP NEXT INSTRUCTION
```

C.1.8 IFBIT — Test Memory Bit

Sintaxis:

- a) IFBIT #,[B]
- b) IFBIT #,MD

Descripción: Al seleccionar un bit (# = 0 a 7, siendo 7 el orden más alto) de una localidad de memoria de programa referenciada por el :

- a) El apuntador B es probado (tested).
 - b) La dirección en el segundo Byte de la instrucción es probada.
- Si el bit seleccionado es alto (= 1) , entonces la siguiente instrucción es ejecutada

De otro modo, la siguiente instrucción es saltada

Operación:

```
IF BIT (#) SELECTED FROM MEMORY
IS EQUAL TO 0,
THEN SKIP NEXT INSTRUCTION
```

C.1.9 IFBNE # — If B Pointer Not Equal

Sintaxis IFBNE #

Descripción:

Si el nibble de bajo orden del apuntador B no es igual a # (donde # = 0 a F) la siguiente instrucción es ejecutada. De otra forma, la siguiente instrucción es saltada. Esta instrucción es útil cuando el apuntador B forma parte de una secuencia de instrucción en un ciclo cerrado. La instrucción IFBNE es codificada hasta al final de la secuencia, seguida por una instrucción JP (Salto Relativo, "Jump Relative") que salta al inicio de una secuencia de la instrucción.

El # codificado con el IFBNE representa la siguiente instrucción más allá del campo de los datos

La instrucción del apuntador B con pos-incremento o decremento del apuntador puede ser usado a través del campo de datos en cualquier dirección

La secuencia de la instrucción hace saltos y se repite hasta que el nibble de bajo orden del apuntador B es igual al # (representa a la siguiente instrucción hasta el campo de datos) a la vez la instrucción JP salta del programa fuera del ciclo.

Operación:

```
IF B POINTER LOW-ORDER NIBBLE EQUALS #,  
THEN SKIP NEXT INSTRUCTION
```

C.1.10 IFC — Test if Carry

Sintaxis: IFC

Descripción:

La siguiente instrucción es ejecutada si la bandera de acarreo es alta. De otra forma, la siguiente instrucción es saltada. La bandera de acarreo queda sin cambio.

Operación:

```
IF NO CARRY (C = 0),  
THEN SKIP NEXT INSTRUCTION
```

C.1.11 IFEQ — Test if Equal

Sintaxis:

- a) IFEQ A,[B]
- b) IFEQ A,#
- c) IFEQ A,MD

Descripción:

- a) El contenido de la localidad de la memoria de datos referida por el apuntador B es comparada con el contenido del acumulador .
- b) El valor inmediato encontrado en el segundo Byte de la instrucción es comparado con el contenido del acumulador .
- c) El contenido de la localidad de la memoria de datos referenciada por la dirección en el segundo Byte de la instrucción es comparada con el contenido del acumulador .

El resultado de comparación de igualdad exitosa es la ejecución de la siguiente instrucción.

De otra forma, la siguiente instrucción es saltada.

Operación:

```
IF A ≠ VALUE  
THEN SKIP NEXT INSTRUCTION
```

C.1.12 IFGT — Test if Greater Than

Sintaxis:

- a) IFGT A,[B]
- b) IFGT A,#
- c) IFGT A,MD

Descripción:

El contenido del acumulador es probado para ser mayor que:

- a) El contenido de la localidad de la memoria de datos referenciada por el apuntador B.
- b) El valor inmediato aparece en el segundo Byte de la instrucción.
- c) El contenido de la localidad de memoria de datos referenciada por la dirección en el segundo Byte de la instrucción.

El resultado probado de un "mayor que" exitoso es la ejecución de la siguiente instrucción.

De otra forma, la siguiente instrucción es saltada.

Operación:

```
IF A ≤ VALUE  
THEN SKIP NEXT INSTRUCTION
```

C.1.13 IFNC — Test if No Carry

Sintaxis: IFNC

Descripción:

La siguiente instrucción es ejecutada si la bandera de acarreo es restablecida (reset). De otra forma, la siguiente instrucción es saltada.

La bandera de acarreo permanece sin cambio.

Operación:

```
IF CARRY (C=1),  
THEN SKIP NEXT INSTRUCTION
```

C.1.14 INC — Increment Accumulator

Sintaxis: INC A

Descripción:

Esta instrucción incrementa el contenido del acumulador y el resultado regresa al acumulador.

Las banderas de acarreo y medio acarreo permanecen sin cambios

Operación:

```
A ← A + 1
```

C.1.15 INTR — Interrupt (Software Trap)

Sintaxis: INTR

Descripción:

La instrucción de "error por software" (software trap) código cero, primero almacena el regreso de la dirección en la pila de la memoria de datos y entonces el programa salta a la localidad de memoria 00FF.

Esta localidad de memoria es el punto común de cambio (switching) para toda la familia "COP8 Basic Family interrupts", ambos software y hardware. El programa inicia en la localidad de memoria 00FF, ordena la prioridad de varias interrupciones y de los vectores para el correcto servicio de rutina.

Para guardar en orden el regreso de la dirección, el contenido del PCL (los 8 bits bajos del PC) son transferidos a la localidad de la memoria de datos referenciada por SP (apuntador de Pila, "Stack Pointer"). Entonces el SP es decrementado, seguido por el PCU (los 7 bits superiores del PC) siendo transferidos a la nueva localidad de memoria de datos referenciado por el SP. Entonces el SP es de nuevo decrementado para configurar por "software stack" la siguiente interrupción o subrutina. El retorno de la dirección ahora a sido guardada en el "software stack" en la memoria de datos RAM.

La instrucción INTR no quiere decir que sea programable explícitamente, pero más bien se invoca automáticamente cuando a ocurrido una condición de error

La lectura indefinida (no existe) de la memoria de programa ROM produce puros ceros, que en el regreso invoca la instrucción INTR

Un similar "software trap" puede ser configurado si la subrutina "Stack Pointer" (SP) es inicializada en la cima de la localidad de la memoria de datos de espacio RAM del usuario. Entonces si en el "software stack" alguna vez, se sacan más datos de la pila (más subrutinas o interrupciones regresan estas llamadas) regresa toda RAM indefinida (no

existe). Esto puede causar que el programa regrese a la dirección FFFF Hex, que el regreso puede leer solamente ceros y de nuevo invocar una instrucción INTR "software trap".

Dos precauciones más pueden ser observadas, cuando se trata de una interrupción por software y es asociada con una rutina de servicio de interrupción. Primero, diferentes interrupciones por hardware, el "software interrupt" no inicializa ("reset") la bandera GIE (Habilita Interrupción Global, "Global Interrupt Enable"). Consecuentemente, la Familia Básica de Microcontroladores COP8 puede ser interrumpida por otras fuentes de interrupción mientras se atiende la interrupción por software ("software interrupt"). Segundo, una instrucción RETSK (regreso y salta) puede ser usada cuando retorne de una rutina de servicio de interrupción por software, en lugar del retorno normal de la instrucción de interrupción (RETI). La instrucción RETI retorno sencillo a la misma instrucción "INTR software trap", da por resultado un ciclo de programa redundante infinito.

Operación:

[SP] ← PCL

[SP - 1] ← PCU

[SP - 2] : SET UP FOR NEXT STACK REFERENCE

PC ← 0FF

C.1.16 JID — Jump Indirect

Sintaxis: JID

Descripción:

La instrucción JID usa el contenido del acumulador para poner indirectamente en la tabla de vectores la dirección de programa. El contenido del acumulador es transferido al PCL (8 bits bajos del PC), después que el acceso de datos de la localidad de memoria de programa direccionada por el PC es transferida al PCL. El programa entonces salta a la localidad de la memoria del programa accedió por el PC. Esto puede ser observado en el PCU (7 bits altos del PC) que nunca cambia durante la instrucción JID, por lo que el salto indirecto, puede saltar a una localidad en el programa actual en una pagina de memoria de 256 direcciones. Sin embargo, si la instrucción es localizada en la última dirección de la pagina, el contador PC ya se haya incrementado encima del límite de la página, y ambos accesos a la memoria de programa (tabla de vectores y nuevas instrucciones), pueden ser decodificados desde la pagina siguiente de 256 Bytes.

Operación:

PCL ← A

PCL ← ROM (PCU,A)

C.1.17 JMP — Jump Absolute

Sintaxis: JMP ADDR

Descripción:

Esta instrucción salta a la dirección de memoria programada. El valor encontrado en el nibble bajo (4 bits) del primer Byte de la instrucción es transferida al nibble bajo del PCU (7 bits superiores del PC), y entonces el valor encontrado en el segundo Byte de la instrucción es transferido al PCL (8 bits bajos del PC). El programa entonces salta a la localidad de memoria de programa que accedió por PC.

Se debe notar que los 3 bits superiores del PC (12-14) no cambian, así que la instrucción Salto Absoluto, debe saltar a una localidad de dirección en el segmento del programa de memoria actual de 4-KByte. Sin embargo, si la instrucción es programada en la última dirección del segmento de memoria, el contador PC puede estar listo para sobreincrementar el límite del segmento de memoria, por consiguiente, el salto es a una localidad de memoria en los siguientes 4-KByte del segmento de memoria.

Operación:

PC11-8 ← HIADDR (LOW NIBBLE OF FIRST BYTE OF INSTRUCTION)
PC7-0 ← LOADDR (SECOND BYTE OF INSTRUCTION)

C.1.18 JMPL — Jump Absolute Long

Sintaxis: JMPL ADDR

Descripción:

La instrucción JMPL puede saltar a cualquier parte, en el espacio de la memoria de programa de 32-KByte. Los valores encontrados en el segundo y tercer Byte de la instrucción son transferidos al PCU (7 bits superiores del PC) y PCL (8 bits bajos del PC) respectivamente.

Operación:

PC14-8 ← HIADDR (SECOND BYTE OF INSTRUCTION)
PC7-0 ← LOADDR (THIRD BYTE OF INSTRUCTION)

C.1.19 JP — Jump Relative

Sintaxis: JP DISP

Descripción:

El valor relativo de desplazamiento encontrado en la instrucción (todos de 8 bits), es direccionado por el Contador de Programa (PC). El incremento normal del PC es asimismo ejecutado.

El valor del desplazamiento deja un retorno de 0 a 31 lugares (con el cero se representa un ciclo cerrado infinito) y un retorno de 2 a 32 lugares.

Un retorno de 1 no es permitido, por conflicto de "opcode zero" con "INTR software trap".

Operación:

$PC \leftarrow PC + DISP + 1$ (DISP 0)

C.1.20 JSR — Jump Subroutine

Sintaxis: JSR ADDR

Descripción:

Esta instrucción pone el regreso de dirección en el "software stack" en la memoria de datos y entonces salta a la dirección de la subrutina. El contenido del PCL (8 bits bajos del PC) son transferidos a la localidad de memoria de datos referida por SP (Apuntador de Pila).

El SP es entonces decrementado, seguido por el contenido del PCU (los 7 bits superiores del PC) comienza a transferir la nueva localidad de la memoria de datos referenciada por SP.

La dirección de retorno ahora es guardada en el "software stack" en la memoria de datos RAM. Luego el SP es nuevamente decrementado para configurar el "software stack" referenciado por la subrutina.

Luego, el valor encontrado en el nibble bajo (4 bits del primer Byte de la instrucción) es transferido al nibble bajo del PCU, y el valor encontrado en el segundo Byte es transferido al PCL. El programa cuando salta a la localidad de memoria accesado por el PC. Es posible notar que los 3 bits superiores del PC (12-14) no cambian, por lo que la subrutina puede localizar en el segmento actual de la memoria de programa de 4 KByte, sin embargo, si una instrucción es programada en la última dirección del segmento de memoria, el contador PC puede estar listo para sobreincrementar el límite del segmento de memoria, por consiguiente, la subrutina puede localizarse en el siguiente segmento de memoria.

Operación:

```
[SP] ← PCL
[SP - 1] ← PCU
[SP - 2]: SET UP FOR NEXT STACK REFERENCE
PC11-8 ← HIADDR (LOW NIBBLE OF FIRST BYTE OF INSTRUCTION)
PC7-0 ← LOADDR (SECOND BYTE OF INSTRUCTION)
```

C.1.21 JSRL — Jump Subroutine Long

Sintaxis: JSRL ADDR

Descripción:

La instrucción JSRL deja que la subrutina pueda ser localizada en cualquier parte en el espacio de 32-KByte de la memoria de programa. La instrucción pone el retorno de la dirección en el "software stack" en la memoria de datos y entonces salta a la dirección de la subrutina.

El contenido del PCL (8 bits bajos del PC) son transferidos a la localidad de la memoria de datos referenciada por SP (Apuntador de la Pila). SP entonces es decrementado, seguido por la nueva localidad la memoria de datos referenciada por el SP. La dirección de retorno es ahora guardada en el "software stack" en la memoria de datos RAM. Entonces el SP es nuevamente decrementado para configurar el "software stack" referenciada por la siguiente subrutina.

Luego, el valor encontrado en el segundo y tercer Byte de la instrucción es transferida al PCU y PCL respectivamente. El programa entonces salta a la localidad de memoria de programa accesada por el PC.

Operación:

```
[SP] ← PCL
[SP - 1] ← PCU
[SP - 2]: SET UP FOR NEXT STACK REFERENCE
PC14-8 ← HIADDR (SECOND BYTE OF INSTRUCTION)
PC7-0 ← LOADDR (THIRD BYTE OF INSTRUCTION)
```

C.1.22 LAID — Load Accumulator Indirect

Address Mode: INDIRECT

Descripción:

La instrucción LAID usa el contenido del acumulador para apuntar a una tabla de datos fija almacenada en la memoria de programa. La tabla de datos usualmente representa una matriz traducida, tal como la entrada de un teclado o la salida de una pantalla ("display").

El contenido del acumulador se intercambiado con el contenido del PCL (8 bits bajos del PC). Los datos introducidos desde la localidad de memoria de programa dirigidos por el PC son entonces transferidos al acumulador.

Simultáneamente, el contenido original del PCL es transferido de regreso al PCL desde el acumulador. Se puede observar que el PCU (7 bits superiores del PC) no cambian durante la instrucción LAID, de manera que el acumulador de carga indirecta junto con la tabla de datos fija asociada puedan ser ambos localizados en la página de memoria actual de 256 Bytes. Sin embargo, si la instrucción LAID es localizada en la última dirección de la pagina, el contador PC debe estar listo para incrementar encima de la página resultante en la operación está decodificada desde la siguiente página. En consecuencia, en esta instancia, la tabla de datos fija puede estar residente en la siguiente pagina de 256 Bytes en la memoria de programa.

Operación:

$A \leftarrow \text{ROM (PCU, A)}$

C.1.23 LD — Load Accumulator

Sintaxis:

- a) LD A,[B]
- b) LD A,[B+]
- c) LD A,[B-]
- d) LD A,#
- e) LD A,MD
- f) LD A,[X]
- g) LD A,[X+]
- h) LD A,[X-]

Descripción:

- a) El contenido de la localidad de la memoria de datos referenciada por el apuntador B es cargado en el acumulador.
- b) El contenido de la localidad de la memoria de datos referenciada por el apuntador B es cargado en el acumulador, y entonces el apuntador B es pos-incrementado.
- c) El contenido de la localidad de la memoria de datos referenciada por el apuntador B es cargado en el acumulador, y entonces el apuntador B es pos-decrementado.
- d) El valor inmediato encontrado en el segundo Byte de la instrucción es cargado en el acumulador
- e) El contenido de la localidad de memoria de datos referenciada por la dirección en el segundo Byte de la instrucción es cargada en el acumulador.
- f) El contenido de la localidad de memoria de datos referenciada por el apuntador X es cargado en el acumulador.
- g) El contenido de la localidad de memoria de datos referenciada por el apuntador X es cargado al acumulador, y entonces el apuntador X es pos-incrementado.
- h) El contenido de la localidad de memoria de datos referenciada por el apuntador X es cargado en el acumulador, y entonces el apuntador X es pos-decrementado.

Operación :

$A \leftarrow \text{VALUE}$

C.1.24 LD — Load B Pointer

Sintaxis. LD B,# (# < 16)

Descripción:

El complemento de uno de los valores encontrados en el nibble bajo (4 bits) de la instrucción es transferido a la posición del nibble bajo del apuntador de registro B, con la posición del nibble superior comienza a limpiar con ceros.

Operación:

$B3-0 \leftarrow (15 - \#) \text{ (1's complement of \#)}$

$B7-4 \leftarrow 0$

C.1.25 LD — Load Memory

Sintaxis:

- a) LD [B],#
- b) LD [B+],#
- c) LD [B-],#

d) LD MD,#

Descripción:

- a) El valor inmediato encontrado en el segundo Byte de la instrucción es cargado en la localidad de la memoria de datos referenciada por el apuntador B.
- b) El valor inmediato encontrado en el segundo Byte de la instrucción es cargado en la localidad de memoria de datos referenciada por el apuntador B, y entonces el apuntador B es pos-incrementado.
- c) El valor inmediato encontrado en el segundo Byte de la instrucciones cargada en la localidad de memoria de datos referenciado por el apuntados B, y entonces el apuntador B es pos-decrementado.
- d) El valor encontrado en el tercer Byte de la instrucción es cargado en la localidad de memoria de datos referenciada por la dirección en el segundo Byte de la instrucción.

Operación:

- a) $[B] \leftarrow \#$
- b) $[B] \leftarrow \#; B \leftarrow B + 1$
- c) $[B] \leftarrow \#; B \leftarrow B - 1$
- d) $MD \leftarrow \#$

C.1.26 LD — Load Register

Sintaxis: LD REG,#

Descripción:

El valor inmediato encontrado en el segundo Byte de la instrucción es cargado en el registro de la memoria de datos referenciada por nibble bajo del primer Byte de la instrucción.

Operación:

$REG \leftarrow \#$

C.1.27 NOP — No Operation

Sintaxis: NOP

Descripción:

Ninguna operación es ejecutada por esta instrucción, por lo que la siguiente operación resulta ser un retardo de un ciclo de instrucción.

Operation:

NO OPERATION

C.1.28 OR — Or

Sintaxis:

- a) OR A,[B]
- b) OR A,#
- c) OR A,MD

Descripción:

Una operación OR es ejecutada en bits correspondientes del acumulador con:

- a) El contenido de la localidad de la memoria de datos referenciada por el apuntador B.
- b) El valor inmediato encontrado en el segundo Byte de la instrucción.
- c) El contenido de la localidad de la memoria de datos referenciada por la dirección en el segundo Byte de la instrucción.

El resultado es retornado al acumulador.

Operación:

$A \leftarrow A \text{ OR VALUE}$

C.1.29 RBIT — Reset Memory Bit

Sintaxis:

- a) RBIT #,[B]
- b) RBIT #,MD

Descripción:

El bit seleccionado (# = 0 a 7, con orden superior comenzando en 7) de la localidad de memoria de datos referenciada por :

- a) El apuntador B es inicializado a cero ("reset").
- b) La dirección en el segundo Byte de la instrucción es inicializada a cero.

Operación:

$[\text{Address:\#}] \leftarrow 0$

C.1.30 RC — Reset Carry

Sintaxis: RC

Descripción:

Las banderas de acarreo y medio acarreo, ambas son inicializadas a cero ("reset").

Operación:

$C \leftarrow 0$
 $HC \leftarrow 0$

C.1.31 RET — Return from Subroutine

Sintaxis: RET

Descripción:

El apuntador de pila (SP) primero es incrementado. El contenido de la localidad de la memoria de datos referenciado por SP son transferidos al PCU (7 bits superiores del PC), después que el SP es de nuevo incrementado. Continúa, el contenido de la localidad de memoria de datos referenciada por SP es transferida al PCL (8 bits bajos del PC) El retomo de la dirección ahora a sido recuperada del "software stack" en la memoria de datos RAM. El programa ahora salta a la localidad de la memoria de programa accesada por el PC.

Operación:

$PCU \leftarrow [SP + 1]$
 $PCL \leftarrow [SP + 2]$
 $[SP + 2] \cdot \text{SET UP FOR NEXT STACK REFERENCE}$

C.1.32 RETI — Return from Interrupt

Sintaxis: RETI

Descripción:

El apuntador de pila (SP) primero es incrementado. El contenido de la localidad de memoria de datos referenciada por el apuntador SP es transferido al PCU (los 7 bits superiores del PC), y el SP es nuevamente incrementado. Luego, el contenido de la localidad de la memoria de datos referenciada por el SP es transferido al PCL (los 8 bits bajos del PC). La dirección de retomo ahora es recuperada del "software stack" en la memoria de datos RAM. El programa ahora salta a la localidad de la memoria de programa accesada por el PC. La bandera global de habilitación de interrupción (GIE) es configurada a uno.

APÉNDICE D

**SISTEMAS DE DESARROLLO PARA LA FAMILIA BASICA DE
MICROCONTROLADORES COP8.**

D.1 Introducción

El Sistema de Desarrollo para la familia de microcontroladores COP8 cuenta con: un ensamblador ASMCOP, un ligador LNCOP, se puede combinar ASMCOP con librerías del COP8 utilizando el LIBCOP, las siguientes utilerías son proporcionadas:

- DUMPCOFF es un programa utilizado para mostrar el contenido de un archivo COFF (generado por LNCOP) en forma legible.
- PROCOP es utilizado para convertir un archivo COFF en un "Intel-hex" con el proposito de de crear PROMs.
- HEXLM y LMHEX convierten un archivo "Intel-hex" a formato NSC LM o un formato LM a "Intel-hex".

Y una tarjeta de desarrollo, llamada EPU (Emulating Programing Unit), con la que se pueden grabar los dispositivos y simular los programas.

A continuación se muestra cada uno de los pasos que se realizan con los respectivos comandos para fines de simulación de un programa fuente (Ver figura D.1).

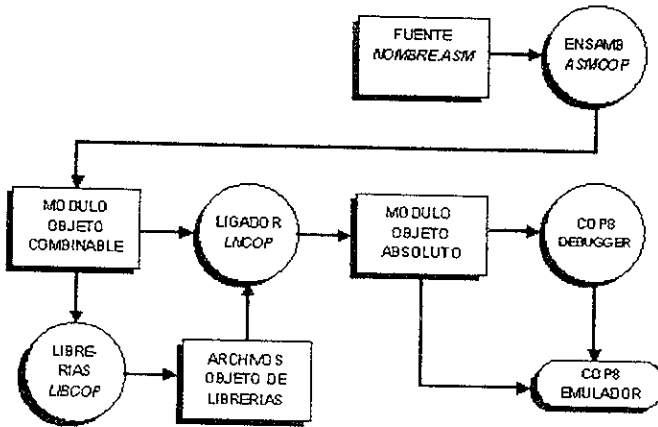


Figura D.1 Eventos Realizados por el Software COP8. (ASMCOP, LNCOP, LIBCOP)

D.2 DESCRIPCIÓN DEL SOFTWARE DE DESARROLLO

El software de desarrollo para la familia básica del microcontrolador COP8 está compuesto por: un ensamblador ASMCOP, un ligador LNCOP y un visualizador de datos DUMPCOFF.

D.2.1 ENSAMBLADOR ASMCOP

En esta sección se describirá la función, características y directivas del ensamblador ASMCOP.

La función del ensamblador ASMCOP es convertir un archivo fuente, que contiene instrucciones en dicho lenguaje ensamblador, a un archivo objeto combinable el cual contiene las instrucciones en código máquina (binario). Este archivo objeto combinable es ligado con el comando LNCOP para generar por último un archivo objeto absoluto el cual puede ser cargado en el EPU (Emulator and Programing Unit) y así simularlo y poder corregir errores de programación. El archivo objeto combinable puede guardarse en una librería para después ser usado en otros programas con el comando LIBCOP.

El ensamblador tiene opciones para generar archivos en los que se presenta cierto tipo de información útil para el programador, por ejemplo: un listado en donde se presenta el programa completo, el tamaño de cada una de las instrucciones en Bytes, mensajes de error y los códigos de operación de cada instrucción, todos ellos útiles para corregir errores y verificar el programa fuente.

También cuenta con directivas, con las cuales se controla las operaciones del ensamblador, y una lista de errores con sus correspondientes posibles causas que lo originaron, lo cual facilita la corrección de errores.

Elementos del ensamblador

Los caracteres que se pueden usar en las declaraciones son los siguientes:

Letras	A a la Z (a - z)
Números	0 al 9
Otros	! \$ % ' () * + , - . / ; : < = > & # ? _ b (Donde "b" indica un espacio.)

Símbolos y etiquetas son usados para asignar un nombre conveniente a direcciones de memoria y valores. Las reglas para construir símbolos y etiquetas son los mismos y sólo se distingue un símbolo de una etiqueta cuando se usan en un programa.

A continuación se mencionan algunas reglas para la elaboración de símbolos y etiquetas:

1. El primer caracter puede ser una letra, un signo de interrogación (?), una guión bajo (_), un signo de dólar (\$) y un punto (.

2. Todos los demás caracteres pueden ser alfanuméricos, signos de dólar, signos de interrogación (?) o líneas (_).

Ejemplos:

```
REGloop
?1
_S
```

3. Símbolos que comienzan con el signo de dólar son designados como locales y definidos solo en una región local (Ver directiva `.LOCAL`).

Expresiones con operadores

Una expresión esta compuesta por uno o más términos y operadores. Y estos pueden ser cualquiera de los siguientes:

- Números en decimal, hexadecimal, octal y binario
- Cadena de constantes
- Símbolos y etiquetas
- Símbolos para direccionar localidades (Etiqueta : `.= LOC`)

Ejemplos:

```
X'5B
HIGH(X'13CF) ;selecciona la parte alta X'13
36 + SUB
```

Cada término es descrito por cuatro atributos: valor, tipo relocalizable, tipo de memoria (RAM, ROM,) y tamaño (BYTE Y WORD). Un tipo relocalizable puede ser absoluto o relocalizable. Un término absoluto es aquel que es conocido al momento de ensamblar Y un término relocalizable es definido como etiqueta dentro de una sección relocalizable (Ver directiva `SECT`).

Una expresión tiene el mismo atributo que el término resultante que se obtiene entre los términos que componen la expresión, y sólo determinadas combinaciones de términos son válidas. Tales combinaciones se muestran a continuación ("aterm" es absoluto, "rterm" es relocalizable, "op" represente cualquier operador, "cop" representa operadores condicionales):

```
rterm = rterm + aterm
rterm = rterm - aterm
rterm = aterm + rterm
aterm = aterm op aterm
aterm = rterm - rterm
aterm = rterm cop rterm
```

En los últimos dos casos, los términos relocalizables deben estar dentro de una misma sección (Ver directiva `SECT`).

El tipo de memoria y tamaño de una expresión se deriva de una manera similar. Para una expresión de un tipo de memoria, tamaño y un valor numérico, se aplican la siguientes reglas

```
type = type + number
```

type = type - number

type = number + type

Cualquier otra combinación es nula.

La magnitud de cada expresión debe ser compatible con la memoria designada para cada expresión. Por ejemplo, si una expresión va a ser almacenada en 1 Byte de memoria, entonces el valor de la expresión no debe exceder el valor X'FF.

Operadores

Los operadores en una expresión son:

- Operadores Aritméticos (+, -, *, /, MOD, SHL, ROL, y ROR).
- Operadores Lógicos (NOT, AND, OR, y XOR).
- Operadores de Relación (EQ, EN, GT, LT, GE, y LE).
- Operadores de Extracción de alto y bajo Byte (HIGH y LOW).
- Operador "untype" (&).

Términos constantes

El ensamblador reconoce doce tipos de términos constantes. Las notaciones permitidas se muestran a continuación:

- En decimal: 4, 356, -10, D'5.
- En hexadecimal: X'32E, H'32E, 0x32E, 032E, 032EH.
- En octal: O'40, Q'40.
- En binario: B'1100, B'0110111.
- Cadena: 'Z', '\$', '23', ", '""', ""'".
- Etiquetas: INICIO:
- Símbolos: _1, \$1, ?1
- Contador de localidad JP.
- Medio Bajo y Medio Superior:
HIGH X'172F es reemplazado por X'17
LOW X'172F es reemplazado por X'2F

- Tamaño: BYT = 7:BYTE
Se puede anular el tamaño asignado a un símbolo con el operador "&", (&BYT).

- "B_SECT" y "E_SECT" :

Los operadores B_SECT y E_SECT se usan para obtener la dirección de inicio y la dirección final de una sección relocalizable (Ver directiva .SECT). La declaración es la siguiente:

B_SECT (nombre de sección)
E_SECT (nombre de sección)

Ejemplo:

```
LD A, #E_SECT(ONE) - B_SECT (ONE)
```

- Números: carga ACC con tamaño de la sección ONE.
Con signo: -32768 (X'8000) a +32767 (X'7FFF)
Sin signo: 0 a 65535 .
- Expresión: 5, X'3C, 'Q', LOW(SUB), 36+SUB, X'3F010-10 .

La siguiente lista (Tabla D.1) muestra los operadores con su función correspondiente.

Operador	Op. Opcional	Función
+		Suma
-		Resta
*		Multiplicación
/		División
MOD		Módulo
SHL		Corrimiento Izq. (1 bit)
SHR		Corrimiento Der. (1 bit)
ROL		Rotación Izq.
ROR		Rotación Der.
NOT	%	Negación
AND	&	AND Lógico
OR	!	OR Lógico
XOR		XOR Lógico
LT	<	"Menor Que"
EQ	=	"Igual Que"
GT	>	"Mayor Que"
LE	<=	"Menor o Igual Que"
GE	>=	"Mayor o Igual Que"
NE	<>	"No Igual Que"
LOW	L	8 bits Menos Significativos
HIGH	H	8 bits Más Significativos
B_SECT		Inicio de Sección
E_SECT		Fin de Sección

Tabla D.1. Operadores Aritméticos, Lógicos y de Relación

Proceso de ensamblado

El ensamblador realiza sus funciones al leer las declaraciones en el lenguaje ensamblador, generando así el código de máquina

ASMCOP ejecuta dos pasos a través de las declaraciones en el lenguaje ensamblador

Paso 1

En este primer paso el ensamblador no genera el archivo objeto combinable, solo asigna valores a símbolos y etiquetas usando un contador de localidades interno. Cada sección de programa tiene su propio contador de localidades, el cual indica donde se encuentran las instrucciones y datos en relación con el inicio de la sección de programa. Cada vez que el ensamblador encuentra una instrucción, el contador de localidades se incrementa de acuerdo al tamaño en Bytes de la instrucción, y cuando encuentra una

etiqueta, se le asigna a esta el valor actual del contador de localidades. Cuando se trata de un símbolo en el lenguaje ensamblador, éste es guardado con su valor correspondiente.

Paso 2

Ahora el ensamblador genera el archivo objeto combinable. Se usan las tablas de etiquetas y símbolos para generar los valores del código de máquina de cada instrucción. También hace uso del contador de localidades para determinar el tamaño de la sección de memoria para cada instrucción. Finalmente el ligador hace el trabajo final de asignar el lugar en la memoria para cada sección de programa y generar el archivo objeto absoluto.

Asignación de las declaraciones

Sintaxis:

[Etiqueta:] símbolo = expresión: BYTE,[WORD],[ninguno] [:comentario]

Ejemplos:

```

.=X'100           ;posesiona el contador de localidades en la
                  ;dirección X'100
COL=.            ;almacena el valor del contador de localidades en "COL"
    
```

Mensajes de error y de atención ("warning")

Cuando ocurre un mensaje de error o de atención, el símbolo "^" indica donde se encuentra el error, esto se puede ver en el monitor después de haber invocado al ensamblador, también se muestra la clave del error y la causa que lo generó (Ver tabla D.2).

# Mensaje	Mensaje y Causas
1	"Cod-op" Inválido o Inexistente Número señalado después de etiqueta Caracter después de etiqueta no es comentario "cod-op" tiene una terminación errónea
2	"Cod-op" No Definido cod-op señalado no es valido
3	Error de Símbolo primer caracter en línea no es permitido símbolo local inválido
5	Símbolo o Etiqueta Duplicado Símbolo ya definido Símbolo definido con .SET ya esta definido Parámetro duplicado
7	Símbolo No Definido
9	Error de Sintaxis Operador erróneo Combinación ilegal de operadores
11	Término Numérico Inválido

Tabla D.2 Ejemplos de Errores al Ensamblar

Directivas

Las directivas forman parte del programa fuente y controlan el proceso de ensamblado o manipulan datos. Todas ellas comienzan con un punto, pueden llevar etiquetas y comentarios.

A continuación se muestran las directivas y sus respectivas funciones .

1) **.addr**

Sintaxis: [etiqueta:] **.ADDR** expresión [,expresión]...[;comentarios]

Descripción:

Esta directiva almacena datos de 8 bits, por cada una de las expresiones especificadas, en forma sucesiva en localidades de memoria. Las expresiones son etiquetas que se usan como apuntador de dirección para la instrucción JID (Salto Indirecto), la cual transfiere el control del programa al contenido de la dirección generada por la directiva **.ADDR** .

La instrucción JID, la directiva **.ADDR** y las expresiones deben estar en una misma página. De otra manera se generara un mensaje de error.

Ejemplo:

```
                .SECT EJEM, ROM, INPAGE
SALT: ADD A, #LONG
                JID
                ADDR TABL1, TABL2
TABL1.          , códigos de TABL1
...
TABL2          ; códigos de TABL2
..
                .ENDSECT
```

2) **.byte, .db**

Sintaxis: [etiqueta:] **BYTE** expresión [,expresión]...[;comentarios]
[etiqueta:] **DB** expresión [,expresión]...[;comentarios]

Descripción: Estas dos directivas generan datos consecutivos de 8 bits para cada expresión dada. Si la directiva tiene una etiqueta, entonces el valor de la primera expresión se le asigna a la etiqueta. Las directivas **.BYTE** y **DB** son válidas solo en secciones tipo ROM. Valores en hexadecimal de caracteres ASCII pueden ser almacenados en memoria usando **BYTE** (y **.DB**) y una expresión, especificando una cadena de caracteres o su equivalente en hexadecimal.

Ejemplos:

```
1 BYTE    X'FF' ; almacena el número X'FF en un Byte de
             memoria
2 BYTE    'NO'
3 DB      X'4E,X'4F
```

Los ejemplos 2 y 3 almacenan la cadena en ASCII (NO) en Bytes consecutivos de memoria


```

...
...
END INICIO      ;fin de programa

```

7) .fb, .fw

Sintaxis: [etiqueta.] .FB tamaño, valor
[etiqueta:] .FW tamaño, valor

Descripción: Estas directivas cargan un bloque de memoria el cual tiene un tamaño en Bytes o en palabras (Word). La etiqueta valor especifica el valor al cual será cargado cada Byte o palabra; este valor puede ser constante o variable. La etiqueta se referirá entonces a la dirección del primer Byte del dato.

Para .FB, el tipo Byte se le asigna a la etiqueta y para .FW, el tipo "Word" se asigna a la etiqueta.

.FB y .FW son válidas en el tipo de sección ROM.

Ejemplos: INI: .FB X'100,0 ; pone X'100 Bytes en cero
 MED: .FW 20,X'FFF ; 20 palabras cargadas con X'FFF

8) .includ

Sintaxis: [etiqueta:] .INCLD archivo [;comentarios]

Descripción: Esta directiva incluye declaraciones en lenguaje ensamblador que se encuentran en un archivo como parte del programa fuente. La extensión, por omisión, es determinada por el archivo fuente en la línea de invocación.

Ejemplo: INCLD SUMBIN ; incluye el archivo SUMBIN.ASM

9) .local

Sintaxis: [etiqueta:] .LOCAL [;comentarios]

Descripción: Establece una nueva sección de programa para etiquetas locales (etiquetas que inician con el signo de dólar [\$]). Todas las etiquetas locales que se encuentran divididas por la directiva .LOCAL, tienen sus propios valores, aunque sean idénticas, dentro de su respectiva sección en el programa

Ejemplo: \$Y. .BYTE 1 ,primera etiqueta \$Y
 LOCAL ;establece una nueva sección para símbolos locales
 \$Y BYTE 1 ,segunda etiqueta \$Y, no hay confusión puesto que se
 ,encuentran en diferentes bloques.

10) .org

Sintaxis: [etiqueta] ORG expresión [;comentarios]

Descripción: La directiva `.ORG` es usada para mover el contador de programa a un nuevo valor. Opcionalmente, el nuevo valor del contador del programa es cargado en la etiqueta.

`.ORG` expresión es equivalente a `=` expresión.

Ejemplos: `.ORG 0100` ; contador de programa = 0100

11) `.sect`, `.endsect`

Sintaxis:

```
[etiqueta:] .SECT nombre, [memoria [,REL][,INPAGE]][:comentarios]
[etiqueta:] .SECT nombre [memoria,ASB=dirección[,INPAGE]][:comentarios]
[etiqueta:] .ENDSECT [:comentarios]
```

Donde:

nombre	Especifica el nombre de la sección.
memoria	ROM memoria de sólo lectura y almacenamiento de códigos y datos.
RAM	memoria de lectura/escritura y almacenamiento de datos
BASE	memoria RAM, rango de 0 a f.
REG	memoria RAM, rango de 0f0 a 0ff.
REL	REL es el tipo de sección relocizable (por omisión). Cuando se tienen varios módulos en un programa, el tipo de sección REL se encarga de reacomodarlos de tal forma que no se traslapen los módulos.
ABS	ABS es el tipo de sección absoluta. La dirección de inicio de la sección es especificada, ejem.,ABS=0100.
INPAGE	INPAGE es usada para garantizar que la sección entera caiga dentro de una página (división de direcciones de 0x100).

Descripción: La directiva `.SECT` define una sección de programa. Esta directiva especifica el nombre de la sección y atributos. `.SECT` debe aparecer al inicio del módulo antes del uso de datos e instrucciones.

`.ENDSECT` es usada opcionalmente para finalizar una sección de programa e iniciar otra, ya que, una nueva sección finaliza la anterior.

Ejemplo:

```
.SECT PSECT, ROM
.DB 3
.SECT NSECT, RAM
.DSB 2
.ENDSECT ; fin de la sección NSECT
.DW 1 ; es parte de la sección PSECT
```

13) `.set`

Sintaxis: [etiqueta:] `.SET` símbolo, expresión [:tipo][:comentarios]

Descripción: Esta directiva se usa para asignar valores a símbolos. En contraste con la declaración de variables tradicional, la directiva `.SET` puede asignar nuevos valores a los mismos símbolos en cualquier sitio dentro la sección de programa. Opcionalmente se puede especificar un tamaño Byte o "Word" y asignar este tipo al símbolo.

Ejemplos .SET E1,2 ; E1=2
 SET G1,E1+4 ; G1=E1+4
 SET H1,E1:BYTE ; H1= al valor 2 con tamaño Byte

14) .title

Sintaxis: [etiqueta:] .TITLE [símbolo][,'cadena'][:comentarios]

Descripción: Esta directiva coloca el símbolo en la primera línea de cada página del listado de salida y la cadena se imprime en la segunda línea, a partir de la segunda página del listado. Esta directiva debe aparecer en la primera línea del programa fuente.

Ejemplo: .TITLE ABCSUM, 'SUMA'

 ABCSUM aparece como encabezado en la primera línea.
 SUMA aparece como segundo encabezado en la segunda página.

15) .word, .dw

Sintaxis: [etiqueta:] .WORD expresión[,expresión. .][:comentarios]
 [etiqueta:] .DW expresión[,expresión...][:comentarios]

Descripción: Estas dos directivas generan palabras consecutivas para almacenar datos de 16 bits. Si las directivas tienen una etiqueta, ésta se referirá sólo a la dirección de la primera palabra y se le asignará el tipo word a la etiqueta. Estas dos directivas sólo se pueden usar en sección tipo ROM.

Ejemplos

.WORD 02 ,almacena el número 02A en una palabra de memoria
 A1. .DW 0FFF,'CD' ,almacena dos valores de 16 bits en forma consecutiva

Control de opciones del ensamblador

Las opciones de control del ensamblador, que se usan sobre el programa fuente, se seleccionan desde la línea de invocación.

La forma de invocar al ensamblador y las opciones es la siguiente

ASMCOP (información de ayuda)
ASMCOP asmfuente [,asmfuente [, .]][opción]

Donde.

asmfuente Es el nombre del archivo(s) a ensamblar. Si son varios los archivos que se van a ensamblar se separan con comas. La extensión del archivo asmfuente es por omisión: .asm

opción: Puede ser una serie de opciones que tiene el ensamblador. Si no se especifican opciones, entonces el ensamblador, por omisión hace uso de unas ya determinadas (COMMENTlíneas, CONStants, Headings, etc).

Ejemplo:

```
ASMCOP \EJEM\PRIMER /L /I
```

Esta invocación ensambla el archivo PRIMER.ASM que se encuentra en el directorio EJEM y crea un listado, PRIMER.LIS, en el directorio actual. Además, se incluye las líneas del archivo especificado por .INCLD en el archivo PRIMER.ASM.

Opciones de Control

1) Crossref

Sintaxis: [NO]Crossref

Descripción: CROSSREF proporciona una tabla de referencia la cual es incluida en el listado de salida. Esta tabla contiene los símbolos; su valor de longitud, número de la línea donde se encuentran y el tipo de sección donde se ubican.

2) llines

Sintaxis: [NO]llines

Descripción: llines incluye al listado de salida las líneas del archivo seleccionado por .INCLD.

3) Include

Sintaxis: Include=directorio

Descripción: Cuando se usa la directiva .INCLD se debe especificar la ruta del archivo que se incluirá en el archivo fuente por medio de Include.

Ejemplo: I=C:\INCLUDES ; busca el archivo especificado en .INCLD a partir de este directorio.

4) Listfile

Sintaxis: [NO]Listfile [=nuevonombre]

Descripción: LISTFILE escribe un listado de todas las líneas del archivo fuente especificado. Por omisión la extensión es .lis y el nombre es el mismo del archivo fuente; opcionalmente se puede cambiar el nombre del listado.

5) **Verify**

Sintaxis [NO]Verify

Descripción. Genera un listado durante cada uno de los pasos del ensamblador. VERIFY es usada principalmente para examinar el efecto de optimización o para determinar en que fase ocurre un error

D.2.2 **LIGADOR LNCOP**

El ligador LNCOP lee el módulo objeto combinable que se genera por el ASMCOP para producir por último el módulo objeto absoluto. Al usar el LNCOP se genera un archivo con extensión .map, el cual muestra como se distribuyen los datos en la memoria.

Proceso de Ligado

El ligador coloca cada sección de programa en la memoria basandose en los atributos de la sección y del espacio disponible de memoria. No hay manera de controlar el orden de instalación de cada sección. Sin embargo, las secciones son procesadas en el orden en que se le den a conocer al ligador.

Las secciones de programa son instaladas en memoria de la siguiente manera:

1. Las secciones absolutas (ABS) son colocadas en memoria a partir de la dirección de inicio especificada. La segmentos de memoria donde se colocará esta sección debe darse de alta por medio de la directiva .CHIP (ASMCOP).
2. Cada sección de programa es colocada en la memoria dentro de un rango específico
3. Las demás secciones (REL y INPAGE) son acomodadas de la siguiente manera: Cada sección es examinada para determinar la longitud de esta y buscar el espacio de memoria suficiente para colocarla. El primer espacio de memoria lo suficientemente grande para contener a esta sección es tomado. Si no hay memoria suficiente para colocar a una sección se produce un mensaje de error

Invocación y Opciones de Control

La invocación del ligador en el sistema MS-DOS es la siguiente.

LNCOP (información de ayuda)
LNCOP objcomb [opción]

Donde

objcomb Es el nombre del archivo objeto combinable a ligar

opción: Son opciones que se pueden especificar para manipular al ligador.

Ejemplos: LNCOP \EJEM\MWPLUS /F=HEX

Este ejemplo liga el archivo mwplus.obj que se encuentra en el directorio EJEM. Se genera un archivo de salida mwplus.hex, con formato hexadecimal y un archivo mwplus.map el cual contiene la longitud del programa, ubicación de los símbolos y rangos de la memoria.

Opciones de control

Format

Sintaxis: Format = tipo [=opción]

Descripción: Este comando especifica el formato de salida para el módulo objeto. El tipo y la opción pueden ser uno de los siguientes:

lm	Para National Semiconductor, formato load módulo.
hex	Para Intel, formato hex. Por omisión, los Bytes no usados son cargados con ceros.
coff	Para formato COFF (Comun Object Format File).

D.2.3 VISUALIZADOR DE DATOS, DUMPCOFF

DUMPCOFF es usado para desplegar datos del archivo de salida, con formato COFF, generado por el ligador LNCOP.

Invocación y Opciones de Control

La invocación es la siguiente:

DUMPCOFF (información de ayuda)
DUMPCOFF archcof [opción]

Donde:

archcof	Es el archivo de salida del ligador con formato COFF. La extensión por omisión es .cof.
opción	Es una o más de las siguientes opciones: /b /e /h /l /s /t

Opciones de Control

- /b** Esta opción despliega información de las secciones, módulos y rangos usados en el programa.
- /e** Con esta opción se habilitan todas las demás opciones menos la opción /h
- /h** Con esta opción se despliega información normalmente "oculta" que es útil para el programador.
- /l** La opción /l despliega el número líneas de entrada.
- /s** Despliega información simbólica contenida en el archivo de salida
- /t** Con esta opción se despliegan los datos de Byte contenidos en una sección de texto COFF, en Hex y ASCII.

D.3 DESCRIPCIÓN DEL HARDWARE PARA DESARROLLO

El Hardware para desarrollo esta compuesto por una tarjeta iceMASTER-EPU-COP8, para emular y programar, y una computadora con ciertas características que se describen más adelante

Tarjeta iceMASTER-EPU-COP8

El EPU (Evaluation and Programming Unit) es una herramienta de evaluación diseñada para emular a la familia básica de microcontroladores y programar a los dispositivos COP8780CJ (UV Borrable) y COP8780CJ(OTP).

Funcionamiento del simulador

El circuito interno simulador combina características de un simulador y un circuito emulador. El emulador que se encuentra en la tarjeta realiza la función de un microcontrolador, ya que este ejecuta los códigos usando un microcontrolador COP880C real. Sin embargo el simulador no corre en tiempo real, ya que el microcontrolador es controlado por software ciclo por ciclo.

Grabación del microcontrolador

Una vez que el programador finalizó el proceso de verificación del programa, puede proceder a grabar el microcontrolador. En la pantalla de simulación tenemos un menú de opciones, para grabar un microcontrolador debemos elegir la opción Misc y seleccionar la segunda opción PROM Programmer. En la pantalla aparecerán otras opciones, por ejemplo el microcontrolador que se grabará y la longitud del segmento de memoria a grabar

Descripción de la tarjeta

La tarjeta EPU cuenta con 3 LEDS indicadores, cada uno de ellos con diferentes funciones:

POWER LED :Este LED (rojo) indica que se esta suministrando corriente a la tarjeta EPU.

PASS LED : LED PASS (verde) indica el estado final de la operación de programación.
Si el LED se enciende, entonces la operación de programación fue un éxito.
Si el LED esta apagado, quiere decir que aún no termina la operación de programación.

STATUS LED : LED STATUS (rojo) es usado para indicar el estado de operación de programación. Si el LED destella, indica que la operación de programación sigue en proceso. Y si se mantiene encendido quiere decir que hubo un error en la operación de programación.

Componentes de Sistema EPU

- 1) DB-9 RS-232
- 2) Fuente de Poder (+16 a +20V DC, 300 mA, voltaje rizo máx. 50 mVp-p)
- 3) Cable de Interfase de 40-Pin EPU-Proto board

Requerimientos del Sistema de Computo

El EPU es controlado por una IBM PC bajo el ambiente PC-DOS/MS-DOS.
La computadora debe contar, como minimo, con las siguientes características:

- 1) IBM PC/XT/AT, PS/286,386,486, Pentium, ISA, EISA.
- 2) 640K Bytes de RAM
- 3) un floppy drive de 3.5", 1.44 MB
- 4) un disco duro
- 5) un puerto serial
- 6) DOS 3.1 ó posteriores.

Ejemplo de uso de directivas básicas en un programa sencillo:

```

.TITLE  Prueba, 'TABLA'          ; Con la directiva .TITLE se puede poner un
                                ; encabezado en cada una de las páginas del
                                ; listado de salida
.INCLD  COP880.INC              ; INCLD incluye las declaraciones hechas en el
                                ; archivo COP880.INC a este programa
.SECT   INICIO, ROM, ABS = 0    ; .SECT indica el tipo de memoria en el que se
                                ; almacenaran los siguientes códigos de operación,
                                ; así como, la dirección inicial del programa
                                ; selecciona por ABS=(dirección de inicio)

PROG:                                ; Etiqueta de inicio de programa.
    LD      B, #00
LIMPIA: LD      [B+], #00
        IFBNE #00
        JP      LIMPIA

.END  PROG:                        ; .END indica el final del código de operación del
                                ; programa. La etiqueta de restablecer PROG indica
                                ; el módulo actual a ejecutar

```


APÉNDICE E

COMPARACIÓN DEL MICROCONTROLADOR COP8 CON OTROS MICROCONTROLADORES

E.1 COMPARACIÓN DEL MICROCONTROLADOR COP8 CON OTROS MICROCONTROLADORES

A continuación se compara el conjunto de instrucciones, la arquitectura, modos de direccionamiento y características sobresalientes de los siguientes microcontroladores:

- COP800 Fabricado por National Semiconductor (Familia básica del COP8).
- M68HC05 Fabricado por Motorola.
- 80C51 Fabricado por Intel.
- PIC16C5X Fabricado por Microchip.

COP800 :

El COP8 tiene una arquitectura Harvard modificada (tiene separada la memoria de datos de la memoria de programa) con mapeo de memoria de entrada/salida. Los registros del CPU incluyen un acumulador (A) de 8 bits, un contador de programa de 16 bits (PC), dos apuntadores de datos de 8 bits (B y X), un apuntador de pila (SP), un estado de palabra del procesador (PSW), y un registro de control (CNTRL). La memoria de datos incluye un banco de 16 registros (incluyendo los tres apuntadores) que tienen atributos especiales. La sección genérica de estradas/salidas del COP800 incluye: dos puertos I/O de 8 bits, cada uno asociado a un registro de configuración de 8 bits y un registro de datos de 8 bits (puertos L y G), también contiene un puerto de salida dedicada (puerto D) de 8 bits con un registro de datos asociado de 8 bits, un registro de entrada dedicada (puerto I) de 8 bits, un puerto de propósito especial (puerto C) de 4 bits. Los puertos I/O son configurables por "software".

M68HC05 :

El M68HC05 tiene una arquitectura Von Neumann con mapeo de memoria de entrada /salida. Los registros del CPU incluyen: un acumulador de 8 bits (A), un contador de programa de 16 bits (PC), un registro índice (X), un apuntador de pila (SP), registro de código de condición (CCR). La sección del temporizador incluye contador de funcionamiento libre, dos registros de contador de 16 bits para leer el contador, un registro de entrada de captura de 16 bits, una salida del contador de comparación de registros, un registro de control del temporizador de 8 bits y un registro de estados del temporizador de 8 bits. La sección de I/O incluye tres puertos bidireccionales de 8 bits cada uno asociado a un registro de datos de 8 bits y un registro de direcciones. La sección de I/O también incluye un puerto de entrada de 7 bits asociado a un registro de entrada

80C51 :

El 80C51 tiene una arquitectura Harvard modificada con mapeo de memoria de entrada y salida. Los registros del CPU incluyen un acumulador (A) de 8 bits, un registro auxiliar de 8 bits (B), para multiplicar y dividir, un contador de programa de 16 bits (PC), una palabra de estado de programa de 8 bits (PSW), un apuntador de pila (SP) de 8 bits y un apuntador de datos de 16 bits (DPTR). El banco de 8 registros de trabajo especial "R0" a "R7". Los registros "R0" y "R1" pueden ser utilizados como apuntadores de

direccionamiento indirecto a la memoria de datos. La sección del temporizador incluye dos temporizadores de 16 bits, un registro del modo del temporizador (TMOD) y un registro de control del temporizador de 8 bits (TCON). La sección de entrada/salida I/O incluye cuatro puertos I/O de 8 bits, cada uno asociado con un registro de 8 bits.

PIC16C5X :

El PIC16C5X tiene una arquitectura Harvard modificada con mapeo de memoria de entrada /salida. El PIC16C5X, también tiene un tipo de arquitectura RISC (Conjunto de Instrucciones Reducidas de Computadora) con 33 instrucciones básicas. Actualmente éstas 33 instrucciones que pueden ser expandidas a 47 para propósito de comparación con otros microcontroladores. Esto es debido a que 14 de las 33 instrucciones tienen como propósito programar bits, que seleccionan uno de dos destinos para el resultado de la instrucción. Consecuentemente , cada una de estas 14 instrucciones pueden tomarse como una instrucción doble.

Los registros del CPU incluyen un registro de trabajo de 8 bits (W) que se utiliza como un pseudo acumulador en tanto mantiene el segundo operando, recibe la literal en la instrucción de tipo inmediato, y también puede ser seleccionado como el registro de destino. Sin embargo, un banco de 31 archivos de registros se utilizan como el acumulador primario en el que representan el primer operando y que pueden ser seleccionados por programa como registro destino. Los primeros 8 archivos de registro incluyen un registro temporizador real reloj/contador (RTCC) mapeado como F1, un contador de programa de 9 bits (PC) mapeado como F2, un registro de estado de palabra de 8 bits (SWR) mapeado como F3 y los registros del puerto I/O de 8 bits mapeados como F5 a F7. El registro de selección de archivo de 8 bits (FSR) es mapeado como F4, cuyos 5 bits de orden bajo eligen uno de los 31 registros de archivos en el modo de direccionamiento indirecto. Invocando al archivo F0 en cualquier instrucción orientada al modo de direccionamiento indirecto y sera utilizado el FSR. Se debe notar que el registro de archivo F0 no es un registro implementado físicamente. El CPU también contiene dos niveles de lógica-alambrada "push/pop" pila para ligar subrutinas. El PIC16C5 también tiene un "temporizador vigilante" así como un registro reloj/contador de tiempo real (RTCC), que no tienen ninguna interrupción física, la ausencia de estas implementaciones son una carga para el CPU; por ejemplo, el registro contador debe ser muestreado por programa para detectar cualquier condición de sobreflujo.

El número de instrucciones en un programa debe ser ajustado en forma diferente en el PIC16C5X, debido a que una palabra de instrucción es de 12 bits de longitud; por lo tanto, veinte instrucciones de una palabra de 12 bits pueden contener el equivalente a 30 instrucciones de 8 bits del COP800. El tamaño mayor de la palabra de la instrucción es una herramienta en la implementación de la arquitectura RISC, pero, a mayor tamaño de la instrucción se traduce en más espacio en ROM.

E.2 MODOS DE DIRECCIONAMIENTO

COP800.

- 1.- Directo
- 2.- Registro Indirecto
- 3.- Registro Indirecto con Post Incremento/Decremento
- 4.- Inmediato
- 5.- Inmediato Corto
- 6.- Indirecto desde el programa de memoria
- 7.- Salto Relativo
- 8.- Salto Absoluto
- 9 - Salto Absoluto largo
- 10.- Salto Indirecto

68HC05:

- 1.- Inherente
- 2.- Inmediato
- 3.- Extendido
- 4.- Directo
- 5.- Índice sin corrimiento
- 6.- Índice con desplazamiento de 8 bits
- 7 - Índice con desplazamiento de 16 bits
- 8.- relativo

80C51:

- 1.- Registro
- 2 - Directo
- 3.- Indirecto
- 4.- Inmediato
- 5.- Relativo
- 6.- Absoluto
- 7.- Largo
- 8.- Índice

PIC16C5X.

- 1 - Datos Directos
- 2.- Datos Indirectos
- 3.- Inmediato
- 4 - Programa Directo
- 5.- Programa Indirecto
- 6.- Relativo

E.3 TIPOS DE INSTRUCCIÓN

- COP800: Total de instrucciones básicas: 40
Total de instrucciones incluyendo modos de direccionamiento: 87
- 1.- Aritmética
 - 2.- Carga e Intercambio
 - 3.- Lógica
 - 4.- Manipulación de bit
 - 5.- Condicional
 - 6.- Transferencia de Control
- 68HC05: Total de instrucciones básicas: 62
Total de instrucciones incluyendo modos de direccionamiento: 210
- 1.- Registro/Memoria
 - 2.- Lectura/Modificación/Escritura
 - 3.- Ramificaciones
 - 4.- Control
- 80C51: Total de instrucciones básicas: 51
Total de instrucciones incluyendo modos de direccionamiento: 111
- 1.- Aritmética
 - 2.- Lógica
 - 3.- Transferencia de Datos
 - 4.- Variable Booleana
 - 5.- Programa de Ramificaciones
- PIC16C5X: Total de instrucciones básicas: 33
Total de instrucciones incluyendo modos de direccionamiento: 47
- 1.- Registro de Archivo Orientado a Byte
 - 2.- Registro de Archivo Orientado a Bit
 - 3.- Literal y Control

E.4 CARACTERÍSTICAS DEL CONJUNTO DE DIRECCIONES

CARACTERÍSTICAS DEL CONJUNTO DE DIRECCIONES DEL COP800 (De la Familia básica de COP8)

1. Mejoras en el Byte sencillo de código de instrucción para minimizar el tamaño del programa.
2. Una instrucción por ciclo para minimizar el tiempo de ejecución del programa.
3. Muchas instrucciones de Bytes-sencillo de función múltiple como es DRSZ.
4. Tres apuntadores mapeados en memoria: dos apuntadores de datos (B y X) por direccionamiento de registro indirecto y un apuntador de pila (SP) para memoria de programa
5. Dieciséis registros mapeados en memoria que permiten una implementación óptima de ciertas instrucciones.
6. Habilidad para configurar, inicializar y probar cualquier bit en el espacio de direcciones de la memoria de datos, incluyendo los puertos I/O mapeados en memoria y los registros asociados.
7. Las instrucciones de registro indirecto: Carga e Intercambio con post incremento/decremento de un apuntador de registro (B y X). Estas permiten gran eficiencia (en tiempo de ejecución y código de programa) en carga y procesamiento de campos en memoria de datos.
8. Instrucciones únicas para optimizar el tamaño del programa y eficiencia.
Algunas de estas instrucciones son :

DRSZ	IFBNE	DCOR
RETSK	RRC	LAID
9. Cuarenta instrucciones básicas.
10. Diez modos de direccionamiento.
11. Fácil de programar.

CARACTERÍSTICAS DEL CONJUNTO DE DIRECCIONES DEL M68HC05

1. Estructura flexible de ramificaciones con 21 diferentes instrucciones de salto
2. Doce diferentes instrucciones lectura/modificación/escritura
3. Cinco instrucciones de manipulación de bits (Limpiar, Configurar, Salta Sí Limpia Bit, Salta Sí Configura Bit, Prueba de Bit con Acumulador.
4. Direccionamiento Índice con opciones de no desplazamiento, desplazamiento de 8 bits o 16 bits.
5. Tabla de datos localizados en la página "0", en el espacio de direcciones (0000-00FFH) toma ventajas del modo de direccionamiento directo para un código óptimo.
6. Instrucción de multiplicación (sin signo 8X8).
7. Ocho modos de direccionamiento.

CARACTERÍSTICAS DEL CONJUNTO DE DIRECCIONES DEL 80C51

1. Gran velocidad, modos de direccionamiento compactos para acceso a memoria de datos para facilitar las operaciones en pequeñas estructuras de datos
2. Ofrece soporte para variables de un bit, permite la manipulación directa del bit en control y sistemas de lógica que requieran procesamiento Booleano.
3. Ocho registros de trabajo (R0 - R7) seleccionados por tres bits permiten una combinación de código de función y direccionamiento de registros en un solo Byte sencillo de instrucción.
4. Los registros R0 y R1 sirven de direccionamiento indirecto apuntado a la memoria de datos.
5. Cuatro bancos de registros de trabajo con sólo un banco activo a la vez, permiten un rápido y efectivo "cambio de contexto".
6. Varias instrucciones especifican registros, referentes al acumulador (A), el acumulador (A) y el registro auxiliar (B) como registro par (AB), la bandera de acarreo (C). El apuntador de datos (DPTR) y el contador de programa dan gran eficiencia.
7. Direccionamiento Índice utiliza un registro base (el contador de programa (PC) o el apuntador de datos (DPTR)) y un corrimiento en el acumulador dan gran flexibilidad.
8. Instrucciones para multiplicar y dividir (utilizando los registros A y B).
9. Cincuenta y un instrucciones.
10. Ocho modos de direccionamiento.

CARACTERÍSTICAS DEL CONJUNTO DE DIRECCIONES DEL PIC16C5X

1. Todas la instrucciones son de 12 bits.
2. Todas las instrucciones son de un ciclo sencillo excepto las instrucciones del tipo salto (GOTO y CALL) y las instrucciones de prueba de bit (DECFSZ, INCFSZ, BTFSC y BTFSS) que son de dos ciclos.
3. Los 32 registros de archivos pueden ser direccionados directa o indirectamente y sirven como acumulador para proporcionar el primer operando.
4. El registro de trabajo (W) se utiliza como pseudo acumulador, proporcionando el segundo operando. El registro de trabajo (W) también se usa como destino de una literal desde memoria de programa en MOVLW (Mueve Literal a W) y la instrucción RETLW (Regresa Literal a W).
5. Muchas instrucciones incluyen un bit destino que selecciona un registro de archivo o el acumulador como destino del resultado.
6. Cuatro instrucciones de manipulación de Bit (Set, Clear, Test and Skip if set, Test and Skip if Clear).
7. Registro de Estado de Palabra (SWR) mapeado en memoria como el registro archivo F3 permite probar el estado de bits (acarreo, dígito de acarreo, cero, potencia abajo y tiempo fuera).
8. Un contador de programa (PC) mapeado en memoria como el registro F2 permite ser utilizado como registro de desplazamiento para direccionamiento indirecto de la memoria de programa.
9. En el Modo de Direccionamiento Indirecto, el apuntador de datos FSR (Registro de Selección de Archivo) es mapeado en memoria como registro de archivo F4. Direccionar F0 produce que FSR pueda ser utilizado para seleccionar el registro de archivo.

10. La literal en la instrucción RETLW (Regreso de Literal a W) combinada con el registro de archivo contador de programa permite que las tablas de datos puedan ser accedidas desde la memoria de programa.
11. Dos niveles de "hardware" "push/pop" apilados de 12 bits para enlazar subrutinas utilizando las instrucciones CALL y RETLW.
12. Treinta y tres instrucciones Básica.
13. Seis modos de direccionamiento.
14. Fácil de programar.

E.5 COMPARACIÓN DE CARACTERÍSTICAS SOBRESALIENTES

Modos de Direccionamiento

Los tres tipos de direccionamiento Índice (sin desplazamiento, desplazamiento de un Byte sencillo y desplazamiento de Byte doble) en el MC68HC05 da flexibilidad en pasar de dos o tres campos de datos (dos operandos, resultado) simultáneamente. Los otros tres microcontroladores alcanzan el mismo resultado con el direccionamiento indirecto y apuntadores. El COP800 y el 80C51 tienen dos apuntadores para el direccionamiento indirecto de datos (en el COP800 son los apuntadores B y X , en el 80C51 son R0 y R1). Solamente el PIC16C5X tiene un apuntador indirecto de datos F4, que es llamado para usar F0. Por lo tanto, procesar dos operandos de Bytes múltiples con el PIC16C5X requiere cargar alternativamente de la dirección del operando en el apuntador F4 como el Byte múltiple del campo de datos a ser procesados.

Los cuatro microcontroladores tienen alguna forma de direccionamiento indirecto para la memoria de programa, que son útiles en hacer saltos a tablas. El COP800 utiliza la instrucción JID (salto indirecto) mientras el PIC16C5x puede direccionar el contador de programa como el registro de archivo F2 y sumar un desplazamiento a él. El M68HC05 y el 80C51 ambos utilizan modos de direccionamiento de índice para lograr direccionar indirectamente la memoria de programa. Pero, saltar a tablas y ver tablas es más fácil crearlas con el modo de direccionamiento de índice. El COP800 utiliza la instrucción LAID (carga el acumulador indirectamente) para implementar tablas, con el PIC16C5X se utiliza la instrucción RETLW (regresa literal a W) como entrada de tablas precedidas a una cabecera que produce una tabla de desplazamiento de direcciones

E.6 CARGA DE MEMORIA INMEDIATA

Sólo el COP800 y el 80C51 tienen instrucciones que pueden cargar la memoria con un valor inmediato. La localidad de memoria puede ser cargada para ser seleccionada como direccionamiento directo o indirecto por ambos microcontroladores

E.7 PUERTOS DE ENTRADA/SALIDA

Los cuatro microcontroladores tienen múltiples puertos I/O, algunos de ellos (COP800, M68HC05) también tienen puertos dedicados de entrada y/o puertos dedicados de salida. El COP800 y el M68HC05 tienen registros de configuración de cada puerto I/O, con cada bit de configuración se determina que bit del puerto está seleccionado como bit de entrada o de salida. El PIC16C5X contiene instrucciones de tres estados (TRIS) cuyo operando determina si o no está asociado a un bit de puerto puesto en alta impedancia o se utiliza como entrada o salida. Por otra parte los bits del puerto asociado del 80C51 están seleccionados como bits de salida. El COP800 tiene tres direcciones asociadas a cada puerto I/O; las primeras dos direccionan los registros de datos y configuración, mientras la tercer dirección lee las terminales (pins) del puerto.

E.8 POST INCREMENTO O DECREMENTO DE LOS APUNTADORES DE DATOS

Sólo el COP800 tiene instrucciones que pueden post incrementar, post decrementar o mantener el apuntador de datos sin cambio. Esta característica aplica a las instrucciones de carga e intercambio y pueden ser utilizadas con cualquiera de los dos apuntadores de datos B y X.

E.9 CORRECCION DECIMAL BCD

Sólo el COP800 y el 80C51 proporcionan instrucciones para agilizar el proceso BCD. La instrucción DCOR (Corrección Decimal) del COP800 es utilizado después de una instrucción de suma o resta para obtener el resultado correcto en BCD. Note que el 66 Hex puede ser sumado al primer operando para comenzar el proceso de suma. La instrucción DA (Ajuste Decimal) del 80C51 es utilizado después de una instrucción de suma para obtener el resultado correcto. La instrucción de Ajuste Decimal no trabaja después de una resta, en consecuencia, la resta en BCD en el 80C51 puede ser implementada sumando el complemento del segundo operando y entonces utilizar la instrucción de ajuste decimal.

E.10 SELECCION DEL MICROCONTROLADOR

Para el Sistema de Alarma y Seguridad se utilizó el microcontrolador COP800 se selecciono por que es relativamente sencillo de programar, es rápido en comparación con el M68HC05, el tamaño del código es compacto, por lo tanto, se puede tener un programa complejo en una pequeña cantidad de memoria ROM, la manipulación de los puertos es de gran ayuda, ya que por medio de el se puede manejar y/o probar el estado de los sensores, teclado y pantalla, otra característica importante es que el microcontrolador COP800 tiene una arquitectura Harvard modificada por lo tanto tiene pocas instrucciones lo que permite memorizar y manejar rápidamente al microcontrolador. Además de que el Laboratorio de Electrónica cuenta con el sistema de desarrollo e información para realizar proyectos usando el microcontrolador COP8.

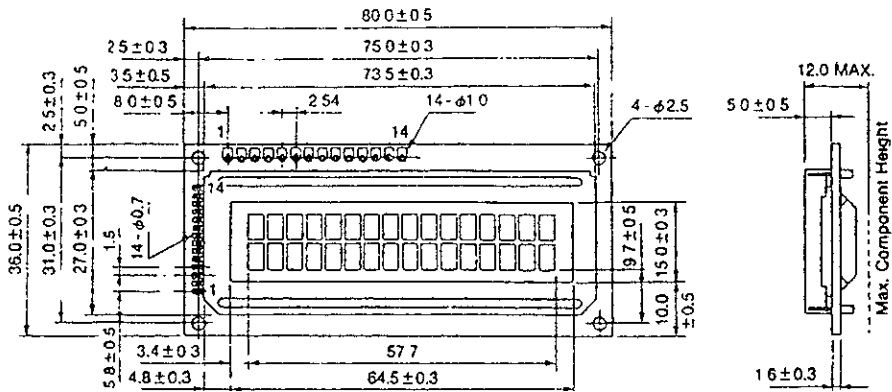
El COP8 al ser un microcontrolador RISC tiene la memoria ROM y memoria RAM separadas por lo tanto la forma de programar es diferente; en el momento de programar y cargar el programa (programa y tablas) en el microcontrolador se utiliza memoria ROM y la memoria de datos RAM es directamente direccionable por instrucciones o indirectamente por los registros apuntadores B, X y SP.

ESPECIFICACIONES

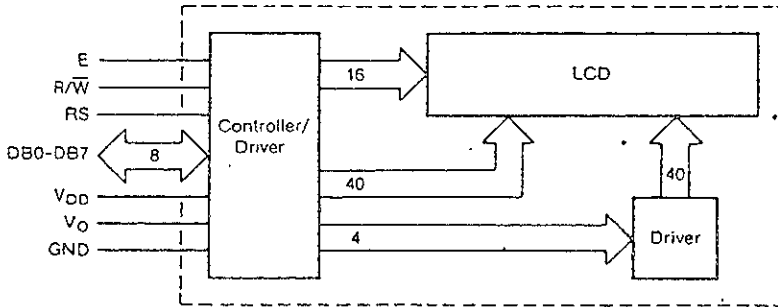
Las especificaciones para el sensor de movimiento pasivo son las siguientes (datos del fabricante).

Alcance:	15m x 21m, 100°
Zonas de detección:	22 zonas (9 de largo alcance, 8 intermedias y 5 de corto alcance).
Contador de impulsos: (Sensibilidad)	1, 2 o 3 eventos seleccionables.
Velocidad de paso detectable:	0.15 - 2.3m/seg.
Altura de montaje:	nominal, 2.5m
Indicador:	LED rojo con selector de activación/desactivación. Se ve a través de la lente.
Relé de alarma:	Capacidad nominal de los contactos de 0.5 A a 30 Vcc, resistencia protectora de 10 ohmios.
Tensión de entrada:	12 Vcc.
Consumo de corriente	15 mA
Temperatura de funcionamiento	-10°C a 50°C
Dimensiones:	107mm x 64mm x 50mm

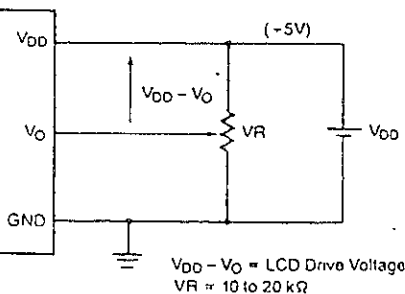
Dimensional Outline (in millimeters)



Block Diagram



Power Supply



Recommended Power Supply for LCD Drive (V_O)

LCD Panel is driven by the voltage $V_{DD} - V_O$ so adjustable V_O is required for contrast control and temperature compensation

Temperature	V_O
0°C	0.0V
+25°C	5V
+50°C	10V

MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
V _{CC}	DC Supply Voltage (Referenced to GND)	-0.5 to +7.0	V
V _{in}	DC Input Voltage (Referenced to GND)	-1.5 to V _{CC} + 1.5	V
V _{out}	DC Output Voltage (Referenced to GND)	-0.5 to V _{CC} + 0.5	V
I _{in}	DC Input Current, per Pin	±20	mA
I _{out}	DC Output Current, per Pin	±25	mA
I _{CC}	DC Supply Current, V _{CC} and GND Pins	±50	mA
P _D	Power Dissipation in Still Air Plastic DIP† SOIC Package‡	750 500	mW
T _{stg}	Storage Temperature	-65 to +150	°C
T _L	Lead Temperature, 1 mm from Case for 10 Seconds (Plastic DIP or SOIC Package)	260	°C

*Maximum Ratings are those values beyond which damage to the device may occur

Functional operation should be restricted to the Recommended Operating Conditions.

†Derating — Plastic DIP -10 mW/°C from 65° to 125°C

SOIC Package: -7 mW/°C from 65° to 125°C

For high frequency or heavy load considerations, see Chapter 2.

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range GND ≤ (V_{in} or V_{out}) ≤ V_{CC}. Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}). Unused outputs must be left open.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V _{CC}	DC Supply Voltage (Referenced to GND)	4.5	5.5	V
V _{in} , V _{out}	DC Input Voltage, Output Voltage (Referenced to GND)	0	V _{CC}	V
T _A	Operating Temperature, All Package Types	-55	+125	°C
t _r , t _f	Input Rise and Fall Time (Figure 1)	0	500	ns

DC ELECTRICAL CHARACTERISTICS (Voltages Referenced to GND)

Symbol	Parameter	Test Conditions	V _{CC} V	Guaranteed Limit			Unit
				-55 to 25°C	≤ 85°C	≤ 125°C	
V _{IH}	Minimum High-Level Input Voltage	V _{out} = 0.1 V or V _{CC} - 0.1 V I _{out} ≤ 20 μA	4.5	2.0	2.0	2.0	V
			5.5	2.0	2.0	2.0	
V _{IL}	Maximum Low-Level Input Voltage	V _{out} = 0.1 V or V _{CC} - 0.1 V I _{out} ≤ 20 μA	4.5	0.8	0.8	0.8	V
			5.5	0.8	0.8	0.8	
V _{OH}	Minimum High-Level Output Voltage	V _{in} = V _{IH} or V _{IL} I _{out} ≤ 20 μA	4.5	4.4	4.4	4.4	V
			5.5	5.4	5.4	5.4	
			4.5	3.98	3.84	3.7	
V _{OL}	Maximum Low-Level Output Voltage	V _{in} = V _{IH} or V _{IL} I _{out} ≤ 20 μA	4.5	0.1	0.1	0.1	V
			5.5	0.1	0.1	0.1	
			4.5	0.26	0.33	0.4	
I _{in}	Maximum Input Leakage Current	V _{in} = V _{CC} or GND	5.5	±0.1	±1.0	±1.0	μA
I _{OZ}	Maximum Three-State Leakage Current	Output in High-Impedance State V _{in} = V _{IL} or V _{IH} V _{out} = V _{CC} or GND	5.5	±0.5	±5.0	±10	μA
I _{CC}	Maximum Quiescent Supply Current (per Package)	V _{in} = V _{CC} or GND I _{out} = 0 μA	5.5	40	40	160	μA
ΔI _{CC}	Additional Quiescent Supply Current	V _{in} = 2.4 V, Any One Input V _{in} = V _{CC} or GND, Other Inputs I _{out} = 0 μA	5.5	≥ -55°C 2.9	25°C to 125°C 2.4		mA

NOTE: Information on typical parametric values can be found in Chapter 2.

ELECTRICAL CHARACTERISTICS ($V_{CC} = 5.0 \text{ V} \pm 10\%$, $C_L = 50 \text{ pF}$, Input $t_r = t_f = 6.0 \text{ ns}$)

Symbol	Parameter	Guaranteed Limit			Unit
		-55 to 25°C	≤ 85°C	≤ 125°C	
t _{PLH} , t _{PLL}	Maximum Propagation Delay, Input D to Output Q (Figures 1 and 5)	30	38	45	ns
t _{PLH} , t _{PLL}	Maximum Propagation Delay, Latch Enable to Q (Figures 2 and 5)	30	38	45	ns
t _{ZLH} , t _{ZLZ}	Maximum Propagation Delay, Output Enable to Q (Figures 3 and 6)	28	35	42	ns
t _{ZLH} , t _{ZLZ}	Maximum Propagation Delay, Output Enable to Q (Figures 3 and 6)	28	35	42	ns
t _{PLH} , t _{PLL}	Maximum Output Transition Time, any Output (Figures 1 and 5)	12	15	18	ns
C _i	Maximum Input Capacitance	10	10	10	pF
C _o	Maximum Three-State Output Capacitance (Output in High-Impedance State)	15	15	15	pF

For propagation delays with loads other than 50 pF, and information on typical parametric values, see Chapter 2.

Power Dissipation Capacitance (Per Enabled Output)*	Typical @ 25°C, $V_{CC} = 5.0 \text{ V}$	pF
	48	

*To determine the no-load dynamic power consumption $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$. For load considerations, see Chapter 2.

REQUIREMENTS ($V_{CC} = 5.0 \text{ V} \pm 10\%$, $C_L = 50 \text{ pF}$, Input $t_r = t_f = 6.0 \text{ ns}$)

Parameter	Fig.	Guaranteed Limit						Unit
		-55 to 25°C		≤ 85°C		≤ 125°C		
		Min	Max	Min	Max	Min	Max	
Minimum Setup Time, Input D to Latch Enable	4	10		13		15		ns
Minimum Hold Time, Latch Enable to Input D	4	50		50		50		ns
Minimum Pulse Width, Latch Enable	2	15		19		22		ns
Maximum Input Rise and Fall Times	1		500		500		500	ns

3

SWITCHING WAVEFORMS

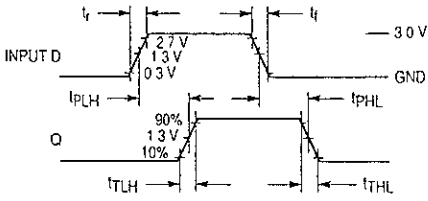


Figure 1.

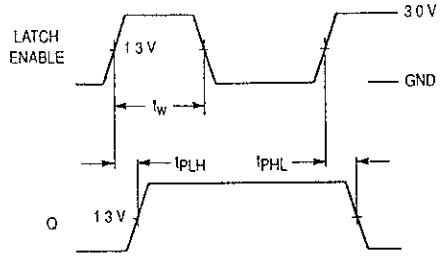


Figure 2.

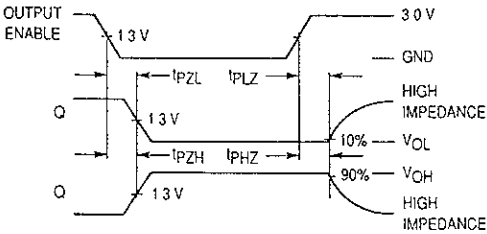


Figure 3.

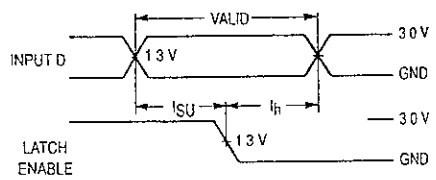
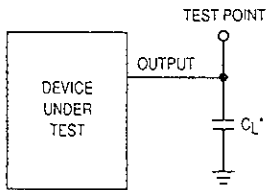
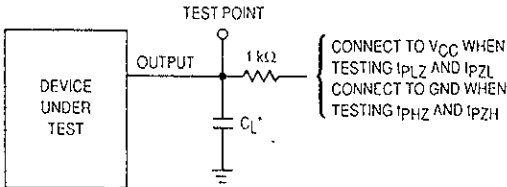


Figure 4.



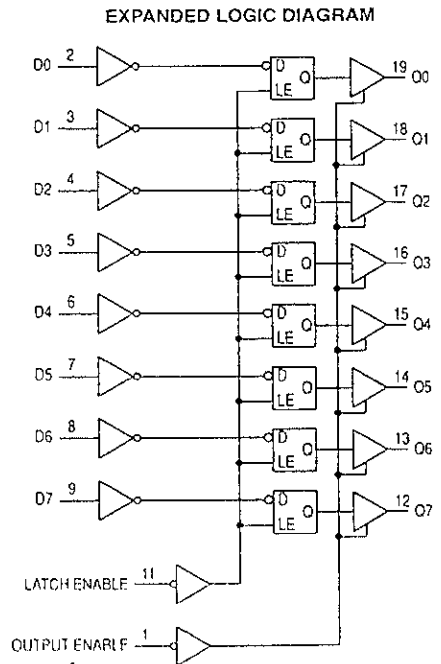
* Includes all probe and jig capacitance

Figure 5 Test Circuit



* Includes all probe and jig capacitance

Figure 6 Test Circuit





LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available in several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the limit sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

Features

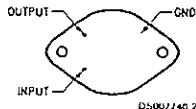
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

Connection Diagrams

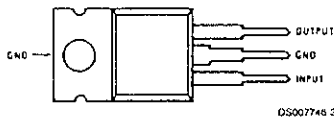
**Metal Can Package
TO-3 (K)
Aluminum**



Bottom View

Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

**Plastic Package
TO-220 (T)**

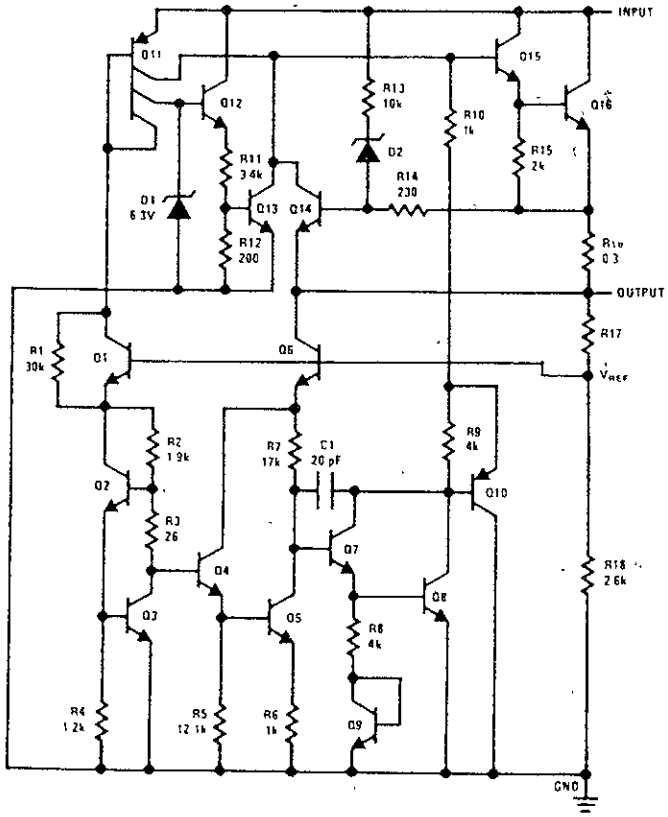


Top View

Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B



Schematic



DS007746-1

MOTOROLA

SEMICONDUCTOR TECHNICAL DATA



6-Pin DIP Random-Phase Optoisolators Triac Driver Output (250 Volts Peak)

The MOC3010 Series consists of gallium arsenide infrared emitting diodes, optically coupled to silicon bilateral switch and are designed for applications requiring isolated triac triggering, low-current isolated ac switching, high electrical isolation (to 7500 Vac peak), high detector standoff voltage, small size, and low cost.

To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.

Recommended for 115 Vac(rms) Applications:

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115 Vac Peripherals
- Motor Controls
- Static ac Power Switch
- Solid State Relays
- Incandescent Lamp Dimmers

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

INFRARED EMITTING DIODE

Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Negligible Power in Transistor Derate above 25°C	P_D	100	mW
		1.33	mW/ $^\circ\text{C}$

OUTPUT DRIVER

Off-State Output Terminal Voltage	V_{DRM}	250	Volts
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	I_{TSM}	1	A
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	300	mW
		4	mW/ $^\circ\text{C}$

TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 Second Duration)	V_{ISO}	7500	Vac(pk)
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	330	mW
		4.4	mW/ $^\circ\text{C}$
Junction Temperature Range	T_J	-40 to +100	$^\circ\text{C}$
Ambient Operating Temperature Range ⁽²⁾	T_A	-40 to +85	$^\circ\text{C}$
Storage Temperature Range ⁽²⁾	T_{stg}	-40 to +150	$^\circ\text{C}$
Soldering Temperature (10 s)	T_L	260	$^\circ\text{C}$

1. Isolation surge voltage, V_{ISO} , is an internal device dielectric breakdown rating.

For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.

MOC3010

[IFT = 15 mA Max]

MOC3011

[IFT = 10 mA Max]

MOC3012*

[IFT = 5 mA Max]

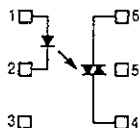
*Motorola Preferred Device

STYLE 6 PLASTIC



STANDARD THRU HOLE
CASE 730A-04

COUPLER SCHEMATIC



- 1 ANODE
- 2 CATHODE
- 3 NC
- 4 MAIN TERMINAL
5. SUBSTRATE
DO NOT CONNECT
- 6 MAIN TERMINAL



MOC3010 MOC3011 MOC3012

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
INPUT LED					
Reverse Leakage Current ($V_R = 3\text{ V}$)	I_R	—	0.05	100	μA
Forward Voltage ($I_F = 10\text{ mA}$)	V_F	—	1.15	1.5	Volts
OUTPUT DETECTOR ($I_F = 0$ unless otherwise noted)					
Peak Blocking Current, Either Direction (Rated $V_{DRM}^{(1)}$)	I_{DRM}	—	10	100	nA
Peak On-State Voltage, Either Direction ($I_{TM} = 100\text{ mA Peak}$)	V_{TM}	—	1.8	3	Volts
Critical Rate of Rise of Off-State Voltage (Figure 7, Note 2)	dv/dt	—	10	—	$\text{V}/\mu\text{s}$
COUPLED					
LED Trigger Current, Current Required to Latch Output (Main Terminal Voltage = 3 V ⁽³⁾)	I_{FT}	—	8	15	mA
	MOC3010	—	8	15	
	MOC3011	—	5	10	
	MOC3012	—	3	5	
Holding Current, Either Direction	I_H	—	100	—	μA

1. Test voltage must be applied within dv/dt rating.
2. This is static dv/dt . See Figure 7 for test circuit. Commutating dv/dt is a function of the load-driving thyristor(s) only.
3. All devices are guaranteed to trigger at an I_F value less than or equal to max I_{FT} . Therefore, recommended operating I_F lies between max I_{FT} (15 mA for MOC3010, 10 mA for MOC3011, 5 mA for MOC3012) and absolute max I_F (60 mA).

TYPICAL ELECTRICAL CHARACTERISTICS $T_A = 25^\circ\text{C}$

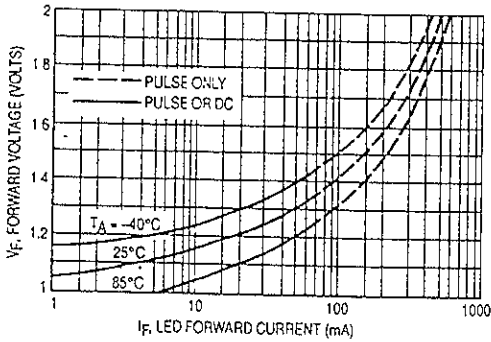


Figure 1. LED Forward Voltage versus Forward Current

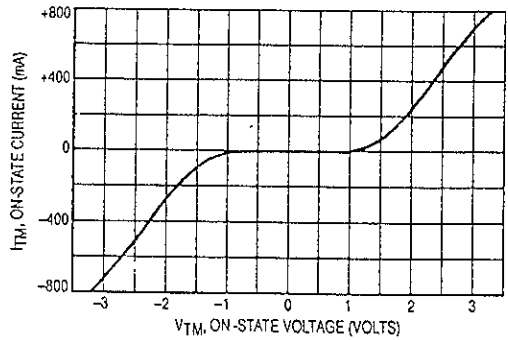


Figure 2. On-State Characteristics

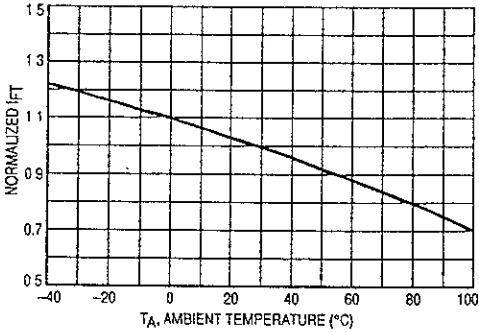


Figure 3. Trigger Current versus Temperature

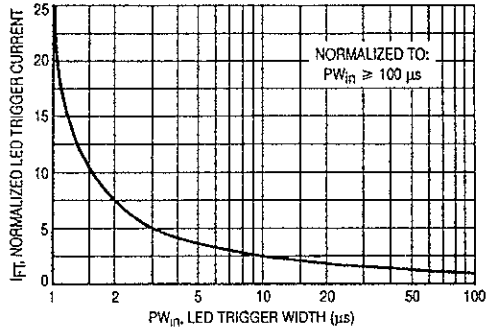


Figure 4. LED Current Required to Trigger versus LED Pulse Width

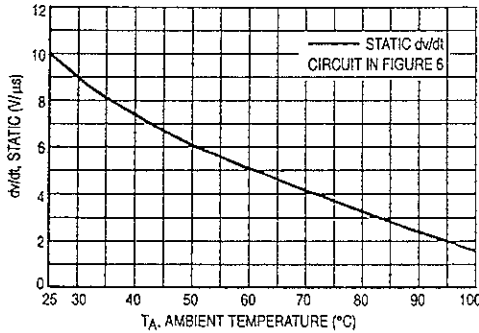
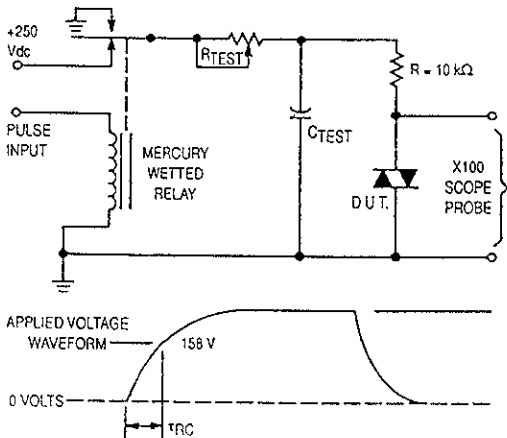


Figure 5. dv/dt versus Temperature



1. The mercury wetted relay provides a high speed repeated pulse to the D.U.T.
2. 100x scope probes are used, to allow high speeds and voltages.
3. The worst-case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable R_{TEST} allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering. τ_{RC} is measured at this point and recorded.

$$dv/dt = \frac{0.63 V_{max}}{\tau_{RC}} = \frac{158}{\tau_{RC}}$$

Figure 6. Static dv/dt Test Circuit

MOC3010 MOC3011 MOC3012

TYPICAL APPLICATION CIRCUITS

NOTE: This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only. Additional information on the use of the MOC3010/3011/3012 is available in Application Note AN-780A.

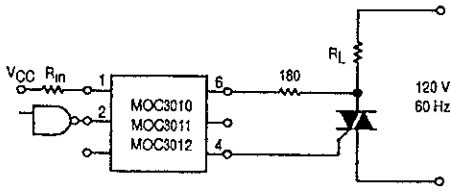


Figure 7. Resistive Load

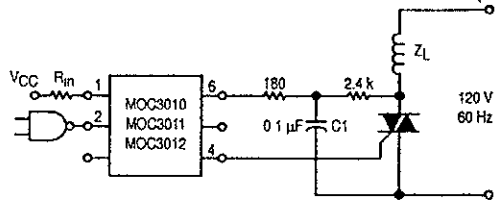


Figure 8. Inductive Load with Sensitive Gate Triac ($I_{GT} \leq 15 \text{ mA}$)

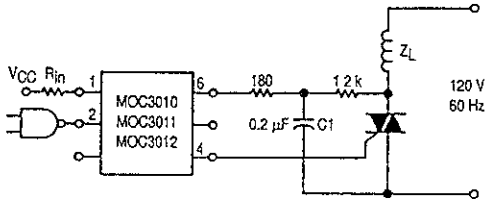


Figure 9. Inductive Load with Non-Sensitive Gate Triac ($15 \text{ mA} < I_{GT} < 50 \text{ mA}$)

APÉNDICE G

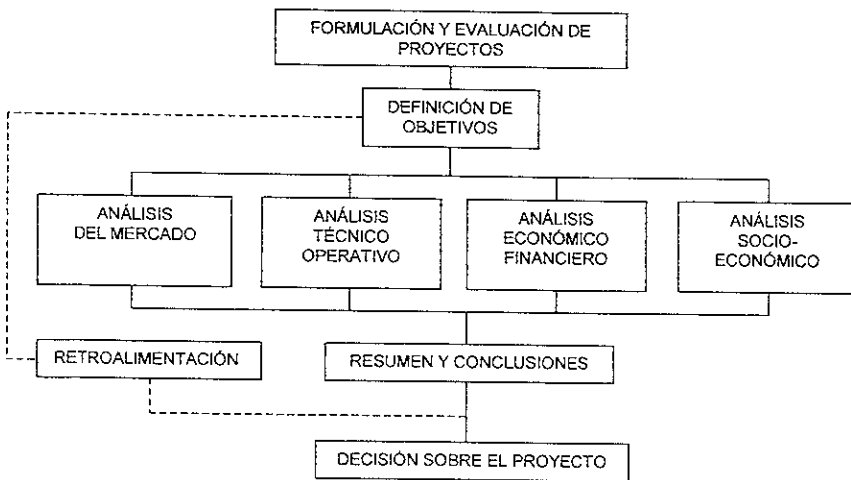
Fundamentos para la evaluación de proyectos

Fundamentos para la evaluación de proyectos

Un proyecto es la búsqueda de una solución inteligente al planeamiento de un problema tendente a resolver, entre muchas, una necesidad humana.

Un "proyecto de inversión" se puede describir como un plan que, si se le asigna determinado monto de capital y se le proporcionan insumos de varios tipos, podrá producir un bien o un servicio, útil al ser humano o a la sociedad en general. La evaluación de un proyecto de inversión, cualquiera que éste sea, tiene como objetivo conocer su rentabilidad económica y social.

La estructura general de la metodología de la evaluación de proyectos puede ser representada por el siguiente diagrama:



Introducción: breve reseña histórica, factores que influyen en el consumo del producto, etc.

Marco de Referencia: verificar si hay un mercado potencial, Demostrar que tecnológicamente es posible producirlo y demostrar que es económicamente rentable.

Estudio del mercado: investigación formal del estudio, cuantificación de la oferta y la demanda, el análisis de los precios y el estudio de la comercialización

Estudio técnico: dividido en cuatro partes, determinación del tamaño óptimo de la planta, determinación de la localización óptima de la planta, ingeniería del proyecto y análisis administrativo.

Estudio económico: ordenar y sistematizar la información de carácter monetario que proporcionan las etapas anteriores y elaborar los cuadros analíticos que sirven de base para la evaluación económica (tasa de rendimiento mínima aceptable y el cálculo de los flujos netos de efectivo).

Evaluación económica: métodos de evaluación que toman en cuenta el valor del dinero a través del tiempo, como son la tasa interna de rendimiento y el valor presente neto

Análisis y administración de riesgos: este criterio puede aplicarse en economías inestables. El resultado de una evaluación económica no permite prever el riesgo de una posible bancarrota al corto o mediano plazo.

BIBLIOGRAFÍA

- 1 - Hans-Werner Bastian, *Sistemas de Alarma*
Ediciones ceac, 1994
(TH9739 B3718)
- 2.- Editec / Rede, *Circuitos Prácticos de Protección y Seguridad*
Ed. Barcelona, 1989
(TH9739-C57)
- 3.- Mavery Swearer, *Alarma y Protección*
Ed. PARANINFO, 1979
(TH9739 S8418)
- 4.- NATIONAL SEMICONDUCTOR, *COP8 Basic Family User's Manual*
Mayo 1995
- 5.- NATIONAL SEMICONDUCTOR, *COP8 Assembler/Linker/Librarian User's Manual*
Mayo 1995
- 6.- NATIONAL SEMICONDUCTOR, *COP8780C/COP8781C/COP8782C Single-Chip EPROM/ OTP Microcontrollers*
Mayo 1995
- 7 - NATIONAL SEMICONDUCTOR, *COP8 Development Tools*
- 8.- MOTOROLA, *Small-Signal Transistor, Fets and Diodes*
Ed. 1994
- 9.- ECG Semiconductors, *Master Replacement Guide*
Ed. 1996