



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Dpto. Ingeniería Eléctrica

“SISTEMA DE SEGURIDAD PARA LA SUPERVISIÓN DE VIVIENDAS MEDIANTE MICROCONTROLADOR PIC Y RED SMS”

PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA INDUSTRIAL: ELECTRÓNICA
INDUSTRIAL

AUTOR: **Rubén Morón Uche**

DIRECTOR DE PROYECTO: **Pedro Abad Martín**

CONVOCATORIA: **Septiembre 2013**

RESUMEN

En el presente documento se justifica el trabajo que se ha ido efectuando durante la realización del proyecto titulado SISTEMA DE SEGURIDAD PARA LA SUPERVISION DE VIVIENDAS MEDIANTE MICROCONTROLADOR PIC Y RED SMS.

Este prototipo de control domótico, que brinda un sistema de seguridad y un simulador de presencia mediante un sistema de comunicación vía SMS, está basado en los pilares básicos de la domótica como son: seguridad, comunicaciones, confort, bienestar y control energético. Todo esto hace que este proyecto sea de gran interés para muchos constructores para mejorar y elevar de nivel sus obras.

El primer capítulo del proyecto da una breve descripción de lo que significa domótica, dando características generales tanto del simulador de presencia como del panel de control a ser diseñado e implementado.

El segundo capítulo muestra los diferentes tipos de sistemas de comunicación que han existido e indica el módem que se escogería para la realización de este proyecto, dando sus características.

En el tercer capítulo se muestra todo el diseño del hardware del sistema, indicando paso a paso los estudios realizados y la selección de elementos para la elaboración del mismo. El software del programa principal se encuentra descrito de una manera concisa y representado a través de diagramas de flujo y sus partes de código correspondientes.

El cuarto capítulo indica las pruebas y resultados realizados al proyecto, con sus respectivas observaciones.

En el quinto capítulo se encuentran las conclusiones y recomendaciones se que han podido extraer sobre la realización de este proyecto y algunos planteamientos para posibles líneas futuras.

En el sexto capítulo se detallan las siglas, abreviaciones y términos utilizados a lo largo de todo el documento.

En el séptimo capítulo se detallan los trabajos de normalización de la domótica que se están realizando tanto en organismos europeos, internacionales y nacionales.

Además, se encuentran los anexos, que incluyen el código completo en lenguaje C del programa, un manual de usuario del prototipo y los datasheet de los componentes del sistema y de los sensores elegidos que por problemas económicos no se han podido comprar.

ÍNDICE

CAPÍTULO I. INTRODUCCIÓN.....	6
1.1 SISTEMAS DOMÓTICOS.....	6
1.1.1 Características de los sistemas domóticos.....	6
1.1.2 Tipos de sistemas de control.....	8
1.1.3 Descripción de un sistema domótico.....	10
1.1.4 Valoración de sistemas domóticos.....	15
1.2 DESCRIPCIÓN DEL PROYECTO.....	17
1.2.1 Características de los sistemas de alarma.....	17
1.2.2 Características de los simuladores de presencia.....	17
1.3 CARACTERÍSTICAS GENERALES DEL SISTEMA.....	17
 CAPÍTULO II. SISTEMAS DE COMUNICACIÓN.....	 19
2.1 SISTEMAS MOVILES CELULARES.....	19
2.1.1 Sistemas GSM.....	21
2.2 MODEMS DE DATOS.....	24
2.2.1 Funcionamiento.....	24
2.2.2 Estándares de modulación.....	24
2.2.3 Tipos de modulación.....	25
2.2.4 Comandos Hayes.....	26
2.3 MODEM GSM Y PROGRAMACIÓN DEL MICROCONTROLADOR.....	28
2.3.1 Módem GSM.....	28
2.3.2 Programación del microcontrolador.....	29

CAPÍTULO III. DISEÑO DEL HARDWARE DEL SISTEMA.....	33
3.1 DESARROLLO DE ESPECIFICACIONES.....	33
3.1.1 Medios de transmisión.....	33
3.1.2 Elementos.....	34
3.2 ESTUDIO DE LOS PLANOS DE LA VIVIENDA.....	49
3.2.1 Determinación del número de circuitos de iluminación.....	50
3.2.2 Determinación de la ubicación de sensores de movimiento y contactos magnéticos.....	51
3.2.3 Estudio de planos eléctricos y otras instalaciones de interés a instalar en la vivienda.....	53
3.3. DISEÑO DEL HARDWARE DE CONTROL.....	54
3.3.1 Herramientas de desarrollo.....	54
3.3.2 Requerimientos de desarrollo.....	56
3.4 PROGRAMACIÓN DEL CONTROLADOR.....	56
3.4.1 Lógica de programación.....	59
3.4.2 Hardware de programación.....	77
CAPITULO IV. PRUEBAS AL SISTEMA Y RESULTADOS.....	79
4.1 PRUEBAS AL SISTEMA.....	79
4.2 RESULTADOS.....	79
CAPITULO V. CONCLUSIONES, RECOMENDACIONES Y FUTUROS DESARROLLOS.....	85
5.1 CONCLUSIONES.....	85
5.2 RECOMENDACIONES.....	85
5.3 FUTUROS DESARROLLOS.....	86

CAPITULO VI. DEFINICIONES Y ABREVIATURAS.....	87
6.1 DEFINICIONES.....	87
6.2 ABREVIATURAS.....	100
 CAPITULO VII. NORMATIVA.....	 101
7.1 CENELEC.....	101
7.1.1 Comité técnico 205.....	101
7.1.2 Normas publicadas.....	101
7.2 CEN.....	102
7.2.1 Comité técnico 247.....	102
7.2.2 Normas publicadas.....	102
7.3 ISO/IEC.....	102
7.3.1 Subcomité 25.....	102
7.3.2 Normas publicadas.....	102
7.4 AENOR.....	103
7.4.1 Subcomité 205.....	103
7.4.2 Norma EA 0026.....	103
 CAPITULO VIII. BIBLIOGRAFÍA.....	 104
8.1 Páginas web.....	104
8.2 Libros.....	105
 CAPITULO IX. ANEXOS.....	 106
9.1 Código del programa.....	106
9.2 Manual de usuario del sistema de alarma.....	117
9.3 Datasheets y catálogos.....	124

CAPÍTULO I

INTRODUCCIÓN

1.1 SISTEMAS DOMÓTICOS

1.1.1 Características de los sistemas domóticos

La domótica se conoce y se aplica desde los años 80 y su desarrollo más importante ha tenido lugar en países como Estados Unidos y Japón, aunque de manera diferente. En los últimos años a esta corriente se han unido países europeos y especialmente los nórdicos, impulsados por las nuevas tecnologías en comunicaciones. En países en vías de desarrollo, hasta ahora, la situación ha sido bastante diferente y, aunque también se conoce desde hace unos años, nunca ha acabado de alzar el vuelo.

El término domótica proviene de la unión de las palabras domus (que significa casa en latín) y robótica (de robota, que significa esclavo, sirviente en checo). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto.

Una de las funciones de la Domótica es la de crear un entorno más amigable, según sean las capacidades del individuo. Además de crear diversos sistemas que mejoran la calidad de vida, da la posibilidad de su monitorización y control a distancia, gracias al bus domótico. De esta manera, se habla de un control local que permite estar al tanto del funcionamiento de la gran parte de los servicios que se tienen al alcance de la mano (como pueden ser: la temperatura del termostato, en que ciclo se encuentra la lavadora, el estado del horno microondas entre otros) y a su vez modificarlos (cambiar la temperatura, acelerar el proceso de lavado). Pero por otro lado también existe un control remoto, mediante el cual se puede obtener información, y modificarla si es requerido, mediante otras interfaces, y controlar los dispositivos también desde interfaces diferentes.

Para que un sistema pueda ser considerado inteligente ha de incorporar sistemas basados en las Nuevas Tecnologías de la Información (NTI). El uso de las NTI en la vivienda genera nuevas aplicaciones y tendencias basadas en la capacidad de proceso de información y en la integración y comunicación

entre los equipos e instalaciones. Así concebida, una vivienda inteligente puede ofrecer una amplia gama de aplicaciones en áreas tales como:

- Monitorización de salud
- Seguridad
- Gestión de la energía
- Automatización de tareas domésticas
- Formación, cultura y entretenimiento
- Teletrabajo
- Operación y mantenimiento de las instalaciones

Estas aplicaciones, junto con las consecuencias inmediatas emanadas de su uso, son las siguientes:

- **Control remoto desde dentro de la vivienda:** A través de un esquema de comunicación con los distintos equipos (mando a distancia, bus de comunicación, etc.). Reduce la necesidad de moverse dentro de la vivienda, este hecho puede ser particularmente importante en el caso de personas de la tercera edad o minusválidos.
- **Control remoto desde fuera de la vivienda:** Presupone un cambio en los horarios en los que se realizan las tareas domésticas (por ejemplo: la posibilidad de que el usuario pueda activar la cocina desde el exterior de su vivienda, implica que previamente ha de preparar los alimentos) y como consecuencia permite al usuario un mejor aprovechamiento de su tiempo.
- **Programabilidad:** El hecho de que los sistemas de la vivienda se pueden programar ya sea para que realicen ciertas funciones con sólo tocar un botón o que las lleven a cabo en función de otras condiciones del entorno (hora, temperatura interior o exterior) produce un aumento del confort y un ahorro de tiempo.
- **Acceso a servicios externos:** Servicios de información, telecompra, telebanco. Para ciertos colectivos estos servicios pueden ser de gran utilidad ya que producen un ahorro de tiempo.

La esencia que debe caracterizar a todos los sistemas domóticos, es la INTEGRACIÓN. Asimismo, todas las mejoras, ventajas y posibilidades se pueden agrupar en torno a cuatro grupos o pilares básicos, que son los siguientes:

1. Seguridad o protección patrimonial
2. Control energético
3. Confort, bienestar y calidad de vida
4. Unificación de las comunicaciones

En el pilar de la protección patrimonial se encuentran:

- Simulación de presencia.
- Detección de conatos de incendio, fugas de gas, escapes de agua.
- Alerta médica. Teleasistencia.
- Cerramiento de persianas puntual y seguro.
- Acceso a cámaras IP.

En el pilar de la eficiencia y ahorro energético se encuentran:

- Climatización: programación y zonificación.
- Gestión eléctrica.
- Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado. Reduce la potencia contratada.
- Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
- Uso de energías renovables.

En el pilar del nivel de confort, bienestar y calidad de vida se encuentran:

- Iluminación.
- Apagado general de todas las luces de la vivienda.
- Automatización del apagado/ encendido en cada punto de luz.
- Regulación de la iluminación según el nivel de luminosidad ambiente.
- Automatización de todos los distintos sistemas, instalaciones y equipos dotándolos de control eficiente y de fácil manejo.
- Integración del portero al teléfono, o del videoportero al televisor.
- Control vía Internet.
- Gestión multimedia y del ocio electrónico.
- Generación de macros y programas de forma sencilla por parte del usuario.

En el pilar de la unificación de las comunicaciones se encuentran:

- Ubicuidad en el control tanto externo como interno, control remoto desde Internet, PC, mandos inalámbricos.
- Transmisión de alarmas.
- Intercomunicaciones.

1.1.2 Tipos de sistemas de control

1.1.2.1 Sistema de lazo abierto

Son los sistemas en los cuales la salida no afecta la acción de control. En un sistema en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada.

En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Por tanto a cada entrada de referencia le corresponde una condición operativa fija; como resultado, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se utiliza si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas. Es evidente que estos sistemas no son de control realimentado.

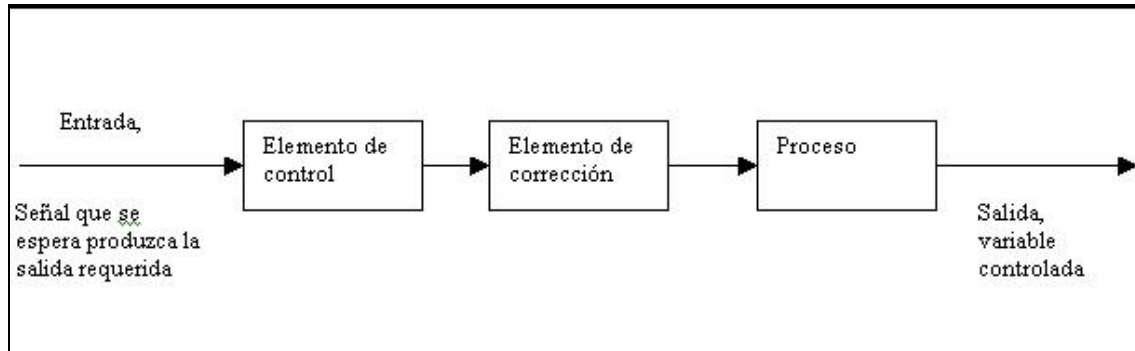


Ilustración 1. Sistema lazo abierto

1.1.2.2 Sistema de lazo cerrado

Los sistemas de control realimentados se denominan también sistemas de control de lazo cerrado. En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente.

En un sistema de control en lazo cerrado, se alimenta al controlador mediante la señal de error de actuación, que es la diferencia entre la señal de entrada y la salida de realimentación (que puede ser la señal de salida misma o una función de la señal de salida y sus derivadas o/y integrales) a fin de reducir el error y llevar la salida del sistema a un valor conveniente. El término control en lazo cerrado siempre implica el uso de una acción de control realimentando para reducir el error del sistema.

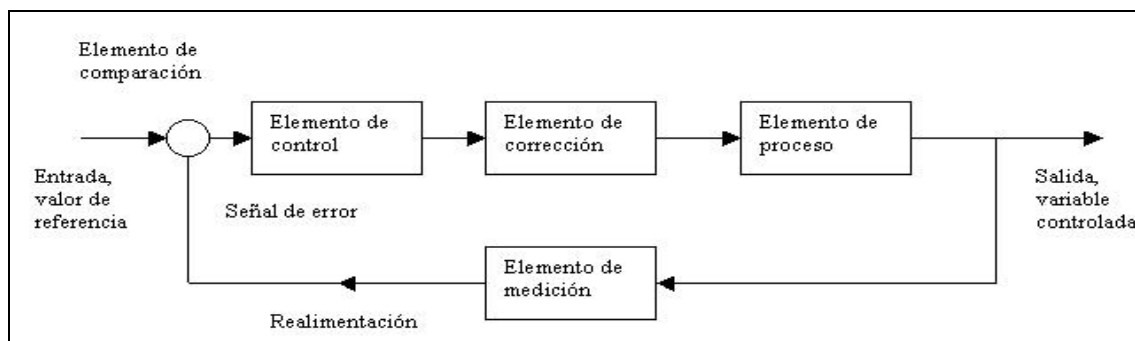


Ilustración 2. Sistema lazo cerrado

Al conocer las características de los distintos tipos de sistemas de control se elige el Sistema de Control de Lazo Abierto, puesto que éste se adapta a la forma de controlar el simulador de presencia, en donde la salida no tiene efecto sobre la acción de control, es decir, la salida no se mide ni se retroalimenta para compararla con la entrada.

En el simulador de presencia, se envía la orden desde el panel del sistema (entrada), y los distintos periféricos la ejecutan (salidas), entonces, al no comparar la entrada con la salida para efectuar la acción, por consiguiente el sistema de control que ha de utilizarse es el de lazo abierto.

1.1.3 Descripción de un sistema domótico

De los sistemas domóticos que se ofrecen en el mercado se pueden apreciar configuraciones genéricas y comunes que permiten su clasificación atendiendo a determinados criterios.

1.1.3.1 Tipo de arquitectura

La arquitectura de un sistema domótico, como la de cualquier sistema de control, especifica el modo en que los diferentes elementos de control del sistema se van a ubicar. Existen dos arquitecturas básicas: la arquitectura centralizada y la distribuida:

Arquitectura centralizada: Es aquella en la que los elementos a controlar y supervisar (sensores, luces, válvulas, etc.) han de cablearse hasta el sistema de control de la vivienda (PC o similar).

El sistema de control es el 'corazón' de la vivienda, en cuya falta todo deja de funcionar, y su instalación no es compatible con la instalación eléctrica convencional en cuanto que en la fase de construcción hay que elegir esta topología de cableado, no es posible su ampliación.

Arquitectura distribuida: Es aquella en la que el elemento de control se sitúa próximo al elemento a controlar. Este tipo de arquitectura es más abierta a cambios una vez realizada la instalación ya que la base de su funcionamiento es: elemento a controlar-controlador.

Una red domótica de arquitectura distribuida está compuesta por una serie de nodos que se conectan unos con otros a través del bus de comunicaciones, el cual lleva dos hilos para datos y dos para la alimentación.

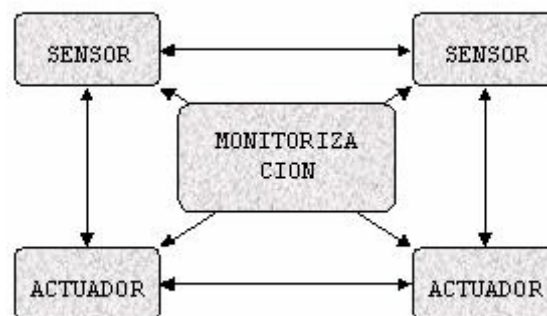
Hay sistemas que son de arquitectura distribuida en cuanto a la capacidad de proceso, pero no lo son en cuanto a la ubicación física de los diferentes elementos de control y viceversa, sistemas que son de arquitectura distribuida en cuanto a su capacidad para ubicar elementos de control físicamente distribuidos, pero no en cuanto a los procesos de control, que son

ejecutados en uno o varios procesadores físicamente centralizados. En los sistemas de arquitectura distribuida se deben de tener en cuenta para poder realizar comparaciones objetivas los siguientes criterios:

- Medio de transmisión de las comunicaciones (bus).
- Velocidad de comunicaciones.
- Topología de la red.
- Protocolo de comunicaciones.



Arquitectura Centralizada.



Arquitectura Distribuida.

Ilustración 3. Arquitectura distribuida y centralizada

COMPARATIVA ENTRE LOS SISTEMAS CENTRALIZADOS Y DISTRIBUIDOS

Sistemas Centralizados

Ventajas:

- Equipos más económicos. Este aspecto que a priori puede ser incierto, hay que complementarlo con los costes derivados de una instalación más complicada, y, otro factor muy importante son los costes ocultos de la instalación de grandes cuadros equipados con relés de potencia o telerruptores, ya que estos equipos no son capaces de conmutar cargas eléctricas.

Inconvenientes:

- Gran cantidad de cableado.

- Centralización de funciones.
- Complicados Interface Hombre-Máquina.
- No se corresponde con la filosofía de sistemas domóticos.
- Muchos sistemas son Autómatas programables.
- Otros carecen de bus de comunicaciones.
- Inexistencia de comunicación entre equipos.

Sistemas Distribuidos

Ventajas:

- Sistemas robustos al fallo.
- Fácil diseño de instalaciones.
- Gran facilidad de uso.
- Cumple con todos los requisitos que un sistema domótico debe cumplir.

Inconvenientes:

- Dependiendo del sistema empleado puede tener una mala relación punto controlado – precio.

1.1.3.2 Medios de Transmisión

Como medio de transmisión se entiende el soporte físico sobre el cual son transportados los datos de comunicaciones, básicamente son:

- Corrientes portadoras.
- Cable (par trenzado)
- Radiofrecuencia.
- Fibra óptica.

1.1.3.3 Topología

Para los sistemas de cable, existe un concepto a tener en cuenta que es la topología de la red de comunicaciones. La topología de la red se define como la distribución física de los elementos de control respecto al medio de comunicación (cable), estos pueden ser clasificados en bus, anillo, topología libre.

1.1.3.4 Velocidad

En todo sistema domótico con arquitectura distribuida, los diferentes elementos de control deben intercambiar información unos con otros a través de un soporte físico (par trenzado, línea de potencia o red eléctrica, radio, infrarrojos, etc.). La velocidad a la cual se intercambian información los diferentes elementos de control de la red se denomina velocidad de transmisión.

1.1.3.5 Protocolo

Una vez establecido el soporte físico y la velocidad de comunicaciones, un sistema domótico se caracteriza por el protocolo de comunicaciones que

utiliza, que no es otra cosa que el 'idioma' o formato de los mensajes que los diferentes elementos de control del sistema deben utilizar para entenderse unos con otros y que puedan intercambiar su información de una manera coherente.

Dentro de los protocolos existentes, se puede realizar una primera clasificación atendiendo a su estandarización:

Protocolos estándar: Los protocolos estándar son los que de alguna manera son utilizados ampliamente por diferentes empresas y estas fabrican productos que son compatibles entre sí.

Protocolos propietarios: Son aquellos que desarrollados por una empresa, solo ella fabrica productos que son capaces de comunicarse entre sí.

1.1.3.6 Bandas de frecuencia

Este modo de transmisión ha sido, poco utilizado para comunicaciones domésticas si bien las compañías eléctricas utilizan esta tecnología para controlar su red desde hace muchos años.

Las compañías eléctricas utilizan señales a baja frecuencia lo que les permite no estar bloqueadas por la impedancia de los transformadores.

Básicamente se trata de sobreponer sobre la senoide a 50Hz de la red una señal a una frecuencia mucho más alta que contendrá las mismas informaciones que pueda tener una señal radio.

La norma europea básica para los sistemas CPL es la EN50065-0, Transmisión de señales sobre las redes eléctricas de baja tensión en la banda de frecuencias de 3 kHz a 148,5 kHz. Parte 1: Reglas generales, bandas de frecuencia y perturbaciones electromagnéticas.

Esta norma establece unas bandas de frecuencia para limitar la influencia mutua entre los equipos de transmisión de señales en las instalaciones eléctricas. También limita las interferencias generadas por los equipos de transmisión de señales a otros equipos electrónicos sensibles. Se aplica a aquellos equipos cuyas señales utilicen frecuencias entre 3kHz y 148,5 kHz, en redes de baja tensión, tanto públicas como de consumidores.

Se establecen una serie de bandas de frecuencia en función del uso:

- **Banda de 3 a 9kHz**: limitado a los distribuidores de electricidad (Compañías Eléctricas).
- **Banda de 9 a 95kHz**: limitado a los distribuidores de electricidad y a quienes hayan dado licencia para ello.

- **Banda de 95kHz a 125kHz**: no necesita protocolo de acceso.
- **Banda de 140kHz a 148,5kHz**: no necesita protocolo de acceso.
- Dentro de la banda de 125kHz a 140kHz, se define que para interoperar varios sistemas, éstos deben ser capaces de conocer la ocupación de la banda y no emitir señal alguna si está ocupada. Se limita la duración de las transmisiones y se define el nivel de potencia mínimo para considerar que hay una señal. La frecuencia para indicar que una transmisión está en curso es la de 132,5kHz.

1.1.3.7 Descripción del tipo de nodos

Una red domótica de arquitectura distribuida está compuesta por una serie de nodos que se conectan unos con otro a través del bus de comunicaciones, el cual lleva dos hilos para datos y dos para la alimentación. Así tenemos:

Nodos de control estándar: Son los encargados de controlar los parámetros de cada estancia. Cada uno soporta dos circuitos independientes de conmutación y dos entradas extra para sensores. La funcionalidad del nodo depende del programa (firmware) que se cargue en el nodo. ISDE suministra un conjunto de programas con las diferentes funcionalidades que cubren la mayoría de las necesidades de control de las estancias de una vivienda.

Nodos de supervisión: Son nodos dedicados a realizar la interface con el usuario. Cada función que el usuario necesita para supervisar y controlar el sistema está implementada en el correspondiente nodo. De esta manera, el usuario puede elegir para su vivienda las funciones que considere necesarias.

- Nodo de alarmas técnicas. (Agua, Gas, Humo y Fuego)
- Nodo de vigilancia de intrusión. (Simulación de presencia, vigilancia)
- Nodo de sirena interior. (Prueba de avisador acústico externo y rearme de alarmas)
- Nodo de luces exteriores. (Activación manual y automática con el sensor de luz)

Nodo telefónico: Realiza la interface entre la red domótica y la red telefónica, tanto la interior de la vivienda como la exterior. A través de este nodo se pueden controlar todas las funciones de la vivienda con el propio teclado del teléfono y confirma la ejecución de las funciones realizadas mediante voz natural.

Nodo de portero: Realiza la interface entre el portero electrónico y el teléfono interior de la vivienda, de tal manera que al realizar una llamada en el portero, el usuario puede atender la llamada y abrir la puerta desde el propio teléfono de la vivienda.

Nodo de televisión: Realiza de interface entre la red domótica y la televisión de la vivienda. Este nodo presenta en la pantalla de televisión la situación de los elementos de supervisión y el usuario puede controlar su vivienda con el mando a distancia.

Nodos exteriores: dentro de este tipo de nodos se agrupas aquellos que siendo de uso dedicado se instalan en el exterior de la vivienda. Dentro de ellos podemos destacar el nodo de sirena exterior y el nodo medidor de luz exterior.

Nodos de comunicaciones: estos son nodos dedicados específicamente a soportar la red de comunicaciones de la vivienda. Entre ellos podemos destacar:

Nodo repetidor: Se utiliza para extender en longitud la red de comunicaciones de la vivienda, cuando esta supere los 1000m, o para aislar galvánicamente sectores de la red. Por ejemplo, cuando la red de comunicaciones sale al exterior de la vivienda, es conveniente que tanto la alimentación como los datos queden aislados de la red interior.

Nodos Routers: El nodo router realiza una adaptación física y lógica de dos medios de transmisión diferentes. ISDE tiene desarrollados dos routers, uno de RS485 a línea de potencia, y otro de RS485 a par trenzado de 78Kbps.

1.1.4 Valoración de sistemas domóticos

1.1.4.1 Grado de complejidad

Para evaluar la complejidad de un sistema se han de atender a determinadas variables que entrarán a formar parte de una ecuación que determine la bondad de este sistema. Las variables son:

- Complejidad de instalación (CI). [0...10]
- Complejidad de puesta en marcha (CPM). [0...10]
- Complejidad de cara al usuario (CCU). [0...10]

Las variables con valores [0, X] sólo pueden tomar los valores '0' o con 'X'.

Las variables con valores [0, X] pueden tomar los valores de '0' a '10'.

La ecuación resultante se obtendría por la suma de la ponderación de las variables:

$$\text{Grado de Complejidad} = \text{CI} + \text{CPM} + \text{CCU}$$

Dando un valor que a medida que sea más alto el sistema es más complejo y por tanto más difícil de instalar y usar.

1.1.4.2 Potencia del sistema

Para evaluar la potencia de un sistema se han de tener en cuenta las siguientes variables:

- Facilidad de ampliación (FA). [0, 10]
- Facilidad de incorporar nuevas funciones (INF). [0...10]
- Velocidad de transmisión (VT). [0...10]
- Protocolo estándar (PE). [0, 5]
- Capacidad de multimedia (CM). [0...5]
- Preinstalación compatible con la instalación tradicional (PC). [0, 10]

Las variables con valores [0, X] sólo pueden tomar los valores '0' o con 'X'.
Las variables con valores [0...X] pueden tomar los valores de '0' a '10'.

Una vez ponderadas las variables anteriores se insertarían en la siguiente ecuación:

$$\text{Potencia del sistema} = \text{FA} + \text{INF} + \text{VT} + \text{PE} + \text{CM} + \text{PC}$$

Dando un valor que a medida que sea más alto el sistema es más potente.

1.1.4.3 Fiabilidad del sistema

Para evaluar la fiabilidad de un sistema se han de tener en cuenta las siguientes variables:

- Control distribuido (CD). [0, 5, 10]
- Fuentes de Alimentación redundantes (FAR). [0, 10]

Las variables con valores [0, X] sólo pueden tomar los valores '0' o con 'X'.
Las variables con valores [0...X] pueden tomar los valores de '0' a '10'.

Una vez ponderadas las variables anteriores se insertarían en la siguiente ecuación:

$$\text{Fiabilidad del sistema} = \text{CD} + \text{FAR}$$

Dando un valor que a medida que sea más alto el sistema es más fiable.

1.2 DESCRIPCIÓN DEL PROYECTO

1.2.1 Características de los sistemas de alarma

Los sistemas de alarma consisten en una red de sensores distribuidos por una vivienda, instalación,... conectados a un panel de control central que los controla y responde con un sistema de alarma cuando alguno de los sensores detecta la presencia de algún intruso.

El panel de control está programado con un sistema de contraseñas grabadas en la memoria EEPROM del microcontrolador para que después de un número determinado de intentos erróneos de desactivar la alarma, se active la sirena durante un tiempo determinado.

1.2.2 Características de los simuladores de presencia

Los simuladores de presencia consisten en encender y apagar luces y/o electrodomésticos dentro del hogar en horarios pre programados o aleatorios, simulando que el lugar está realmente habitado cuando no hay residentes en el mismo.

El encendido de las luces se realiza únicamente en la noche por un tiempo predeterminado, es decir que se programa el encendido y apagado de las mismas para así dar apariencia de que realmente alguien se encuentra en la vivienda.

Este concepto brinda mayor seguridad al hogar, evitando que posibles intrusos opten por irrumpir en el mismo ya que tiene la apariencia de estar habitado. En este caso solo procederemos al encendido de luces.

1.3 CARACTERÍSTICAS GENERALES DEL PROYECTO

El sistema a desarrollar comprende básicamente de dos servicios; el primero es un sistema de alarma y el segundo un simulador de presencia. Tanto el sistema de alarma como el simulador de presencia están pensados para la seguridad del hogar, teniendo en cuenta que son sistemas preventivos, haciendo que las personas que habitan en el mismo se sientan tranquilos al momento de abandonar su vivienda, ya sea por trabajo, viaje o cualquier otro motivo. Lo que hace que este sistema se base en el ámbito de la protección patrimonial.

El sistema de alarma, mediante la red de sensores de movimiento, de contactos magnéticos en puertas y ventanas y el panel de control, previene la entrada a la vivienda de posibles intrusos cuando los usuarios no están en ella. El panel de control actuará, a través de su sistema de claves, en caso de que los intrusos quisieran entrar a la vivienda desactivando la alarma desde el

panel de control. El sistema está programado para que después de tres intentos erróneos de ingreso de la clave de seguridad se active la alarma. Mientras que los contactos magnéticos detectarán la entrada en la vivienda por alguna de las puertas o ventanas y los sensores de movimiento la presencia de intrusos en la casa en caso de que el panel de control o los contactos magnéticos no hayan sido efectivos.

Por otra parte, el simulador de presencia, como su nombre indica, permite simular la presencia de usuarios en una vivienda mediante la activación de luces durante un determinado horario.

Este prototipo es diseñado mediante el uso de un controlador, el cual actúa con las señales de entrada de los distintos sensores y manipula los diferentes dispositivos electrónicos a encender. El sistema se basa en soporte metálico, lo que quiere decir que cada sensor, actuador, panel de control, entre otros, es cableado hacia el panel de control, teniendo cada dispositivo su propio cableado. También la conexión hacia el panel se podría hacer de manera inalámbrica mediante diversos protocolos como posteriormente explicaré.

La activación del sistema de alarma se realiza mediante el ingreso de una clave personal en el panel de control. Cuando se activa por primera vez se usa una clave por defecto, la cual se puede cambiar por otra que el usuario desee en cualquier momento. En el caso de que saltara la alarma se encendería una sirena y al usuario le llegaría un SMS avisándole de este hecho.

Por motivos económicos y logísticos no ha sido posible comprar los sensores y montarlos en una vivienda o instalación. Por lo que se he optado por simular los sensores con un interruptor y simular la sirena y las luminarias de la vivienda con diodos LED.

CAPÍTULO II

SISTEMAS DE COMUNICACIÓN

En el presente capítulo se analizarán los distintos tipos de módems, ya que con éste dispositivo se realizará la comunicación entre el microcontrolador y el usuario. Además se tratarán las tecnologías móviles celulares, debido a que se podría utilizar un teléfono móvil como módem para la realización del proyecto. Por último se muestra el módem GSM que elegiría para este proyecto y un ejemplo de lo que podría ser una comunicación entre el microcontrolador y un teléfono móvil en caso de que se produjera una alarma.

2.1. SISTEMAS MÓVILES CELULARES

A los sistemas móviles celulares se los puede dividir en tres distintas generaciones, teniendo cada una diferentes características y protocolos. Estas generaciones son las siguientes:

Primera generación (1G)

Es la más antigua, también conocida como TACS (Total Access Communications System), y engloba a todas aquellas tecnologías de comunicaciones móviles analógicas. El sistema TACS es un sistema de comunicaciones para telefonía móvil celular dúplex en la banda de 900 MHz, con capacidad para transmitir voz pero no datos. Dado que el estándar TACS sólo define el protocolo de acceso radio entre una estación móvil y su correspondiente estación base, y no cubre aspectos relativos a la gestión de la movilidad, surgieron toda una serie de estándares diferentes en diversos países: NTT-MTS (Japón), MNT (Escandinavia) y C450 (Alemania), con los correspondientes problemas de incompatibilidades. En la actualidad esta generación está en desuso y pronto se la dejará de utilizar definitivamente.

Segunda generación (2G)

Las limitaciones de la primera generación de telefonía móvil condujeron al desarrollo del sistema GSM (Global System for Mobile Communications). GSM es una tecnología digital cuya primera funcionalidad es la transmisión de voz, pero que también permite la transmisión de datos a baja velocidad: 9,6 kbit/s. Esta velocidad es baja pero permite el servicio de envío de mensajes cortos (SMS), superando con creces las mejores expectativas que podían imaginar los operadores. El sistema europeo GSM opera en torno a los 900 MHz, si bien surgió también una variante conocida como DCS (Digital Cellular System) que trabaja a 1800 MHz y cuyo objetivo es proporcionar mayor capacidad en zonas urbanas. A diferencia del sistema TACS, GSM define un sistema completo que incluye no sólo la interfaz radio, sino también una

completa arquitectura de red, lo que permite el desarrollo de multitud de nuevos servicios sobre el estándar GSM.

Segunda generación y media (2.5G)

Aquí se incluyen todas aquellas tecnologías de comunicaciones móviles digitales que permiten una mayor capacidad de transmisión de datos y que surgieron como paso previo a las tecnologías 3G. Una de estas tecnologías 2.5G es GPRS (General Packet Radio System), basada en la transmisión de paquetes y donde los canales de comunicación se comparten entre los distintos usuarios de forma dinámica. La velocidad teórica máxima que puede alcanzar GPRS es de 171,2 kbit/s. Las principales ventajas que aporta GPRS respecto a GSM son, además de una mayor velocidad de transmisión, la conexión permanente y la tarificación por tráfico, convirtiéndolo en el portador ideal para los servicios WAP (Wireless Application Protocol), el acceso a Internet (web browsing, ftp, e-mail) y el acceso a intranets de empresas.

Por otro lado, el HSCSD (High Speed Circuit Switched Data) es una especificación de la fase 2+ de GSM homologada por el ETSI. Con esta tecnología, el número de canales equivalentes de GSM utilizados en cada instante por una comunicación de datos es variable, permitiendo velocidades de transmisión de hasta 57,6 kbit/s. La ventaja de HSCSD es que la inversión para ser implantado es mínima, aunque tiene como desventaja un mayor coste para los usuarios, pues pagan por el uso de cada canal. Por último, la tecnología EDGE (Enhanced Data Rates for Global Evolution) es otro de los desarrollos de las redes GSM, permitiendo teóricamente velocidades de datos de hasta 384 kbit/s. Se trata de una tecnología que mejora el ancho de banda de la transmisión de los datos en GSM y GPRS.

Tercera generación (3G)

Las tecnologías 3G se encuentran contenidas dentro del IMT-2000 (International Mobile Telecommunications-2000) de la ITU, el cual puede considerarse como la guía que marca los puntos en común que deben cumplirse para conseguir el objetivo de la itinerancia global, es decir, que un terminal de usuario de 3G pueda comunicarse con cualquier red 3G del mundo.

Los servicios que ofrecen las tecnologías 3G son básicamente: acceso a Internet, servicios de banda ancha, roaming internacional e interoperabilidad. Pero fundamentalmente, estos sistemas permiten el desarrollo de entornos multimedia para la transmisión de vídeo e imágenes en tiempo real, fomentando la aparición de nuevas aplicaciones y servicios tales como videoconferencia o comercio electrónico.

Criterio	Primera Generación	Segunda Generación	Tercera Generación
Servicios	Voz	Voz y mensajería corta	Voz y datos
Calidad de servicio (QoS)	Baja	Alta	Alta
Nivel estandarización	Bajo	Fuerte	Fuerte
Velocidad de transmisión	Baja	Baja	Alta
Tipo de conmutación	Circuitos	Circuitos	Paquetes (IP)

Tabla 1. Comparativa entre diferentes generaciones de servicios móviles

2.1.1. Sistemas GSM

El sistema de comunicaciones móviles GSM (Global System for Mobile Communications) es un sistema ampliamente utilizado que representa el estándar europeo de comunicaciones móviles de segunda generación.

Aunque el enorme incremento en el número de usuarios de comunicaciones móviles ha venido de la mano de este sistema, todavía existe cierta preocupación por dos aspectos fundamentales; esto es, la privacidad y la autenticación, que son la pieza fundamental en la seguridad de las comunicaciones, pero que en el caso de GSM se ha demostrado poco robusta con respecto a lo indicado anteriormente. A pesar de ello, la considerable mejora en la calidad del servicio ofrecido a los usuarios y el abaratamiento de los costes de los terminales no ha impedido la masiva utilización del sistema.

GSM es un sistema de conmutación de circuitos, diseñado originalmente para voz al que posteriormente se le adicionaron algunos servicios de datos: servicio de mensajes cortos, un servicio de entrega de mensajes de texto de hasta 160 caracteres y un servicio de datos GSM, que permite una tasa de transferencia a 2.4, 4.8 y 9.6 kbp/s.

En España el sistema GSM tiene frecuencias reservadas en las bandas de 900 y 1800 MHz Utiliza la división de frecuencia para obtener comunicación full-dúplex, con dos portadoras separadas 45 MHz entre sí para ambos sentidos de transmisión por canal. Dichas portadoras se encuentran en el rango entre 890 y 915 y 1710 y 1785 MHz para el enlace ascendente entre móvil y base, y entre 935 y 960 y 1805 y 1880 para un enlace descendente entre base y móvil.

La infraestructura proporcionada por la red GSM le permite la prestación de los siguientes servicios:

- Localización de los terminales.
- Autenticación y cifrado.
- Establecimiento y liberación de llamadas de voz, datos y fax entre otros tipos.
- Transmisión de mensajes cortos.
- Tarificación.
- Itinerancia de las llamadas al cambiar de estación controladora.
- Servicios avanzados como el desvío de llamadas, o el grupo cerrado de usuarios.

Por otro lado, GPRS coexiste con GSM, compartiendo gran parte de la infraestructura desplegada en el mismo, pero ofreciendo al usuario un servicio portador más eficiente para las comunicaciones de datos, especialmente en el caso de los servicios de acceso a redes IP como Internet.

La arquitectura del sistema se compone de cuatro bloques o subsistemas que engloban el conjunto de elementos de la jerarquía del sistema. Cada uno de estos subsistemas desempeña funciones específicas para ofrecer el servicio de telefonía móvil al usuario final. Estos son:

- Estación móvil (MS)
- Estación base (BSS)
- Subsistema de conmutación y de red (NSS)
- Subsistema de operación y mantenimiento (OMC)

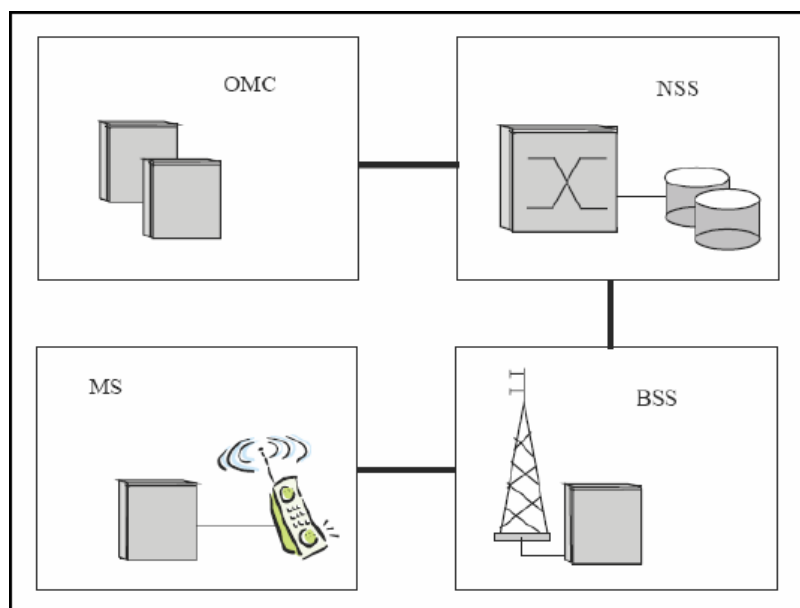


Ilustración 4. Arquitectura del sistema GSM

La estación móvil comprende todos los elementos utilizados por el abonado del servicio. La estación base engloba los elementos que desempeñan las funciones de interconexión entre la radio con la estación móvil. El subsistema de conmutación y de red realiza las operaciones de interconexión con otras redes de telefonía y de gestión de la información del abonado. Finalmente el subsistema de operación y mantenimiento se encarga de supervisar el funcionamiento del resto de bloques.

El módem telefónico posee dos modos de funcionamiento:

- **Modo comando:** Es cuando el módem responde a los comandos que le envía la terminal local, es decir, que la información que recibe el módem la procesa sin transmitirla por la línea y le envía su respuesta a la terminal. En este modo es posible configurar el módem o realizar operaciones de marcado y conexión. Antes de que se pueda enviar un comando al módem, este debe estar en el presente modo.



Ilustración 5. Modo comando

- **Modo en línea:** Es cuando el módem se conecta con otro. Aquí, cualquier información que envíe el terminal local al módem será transmitida al módem remoto. En este caso el módem no procesa ningún tipo de información y simplemente la transmite a través de la línea. Lo que si puede hacer es añadir a los datos información adicional para la corrección de errores y verificar si los datos que recibe del otro módem no han sido adulterados.

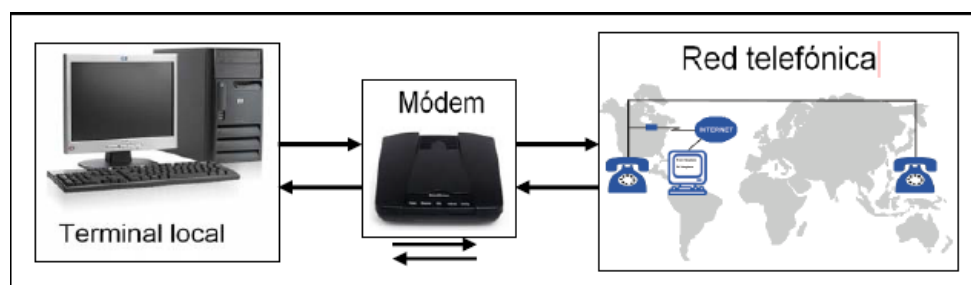


Ilustración 6. Modo en línea

2.2. MODEMS DE DATOS

Un módem es un equipo que sirve para modular y demodular (en amplitud, frecuencia, fase u otro sistema) una señal llamada portadora mediante otra señal de entrada llamada moduladora.

2.2.1. Funcionamiento

El modulador emite una señal denominada portadora, que, generalmente, se trata de una simple señal eléctrica sinusoidal de mucha mayor frecuencia que la señal moduladora. La señal moduladora constituye la información que se prepara para una transmisión (un módem prepara la información para ser transmitida, pero no realiza la transmisión). La moduladora modifica alguna característica de la portadora (que es la acción de modular), de manera que se obtiene una señal, que incluye la información de la moduladora. Así el demodulador puede recuperar la señal moduladora original, quitando la portadora.

Las características que se pueden modificar de la señal portadora son:

- **Amplitud**, dando lugar a una modulación de amplitud (AM/ASK).
- **Frecuencia**, dando lugar a una modulación de frecuencia (FM/FSK).
- **Fase**, dando lugar a una modulación de fase (PM/PSK).

También es posible una combinación de demodulaciones o modulaciones más complejas como la modulación de amplitud en cuadratura.

2.2.2. Estándares de modulación

Dos módems para comunicarse necesitan emplear la misma técnica de modulación. La mayoría de los módem son full-dúplex, lo cual significa que pueden transferir datos en ambas direcciones. Hay otros módems que son half-duplex y pueden transmitir en una sola dirección al mismo tiempo. Algunos estándares permiten sólo operaciones asíncronas y otros síncronas o asíncronas con el mismo módem.

A continuación se enlistan los tipos de modulación más frecuentes:

TIPO	CARACTERISTICAS
Bell 103	Especificación del Sistema Bell para un módem de 300 baudios, asíncrono y full-dúplex
Bell 201	Especificación del Sistema Bell para un módem de 2400 BPS, síncrono y full-dúplex
Bell 212	Especificación del Sistema Bell para un módem de 2400 BPS, asíncrono y full-dúplex

V.22 bis	Módem de 2400 BPS, síncrono/asíncrono y full-dúplex
V.29	Módem de 4800/7200/9600 BPS, síncrono y full-dúplex
V.32	Módem de 4800/9600 BPS, síncrono/asíncrono y full-dúplex
V.32 bis	Módem de 4800/7200/9600/12000/14400 BPS, síncrono/asíncrono y full-dúplex
Hayes Express	Módem de 4800/9600 BPS, síncrono/asíncrono y half-dúplex
USR-HST	Módem de USRobotics de 9600/14400 BPS. Sólo compatibles con los del mismo modelo aunque los más modernos soportan V.32
Vfast	Es una recomendación de los fabricantes de la industria de módems. La norma Vfast permite velocidades de transferencia de hasta 28800 BPS.
V34	Estándar del CCITT para comunicaciones de módem de en velocidades de hasta 28800 BPS

Tabla 2. Estándares de modulación

2.2.3. Tipos de modulación

Dependiendo de si el módem es digital o analógico se usa una modulación de la misma naturaleza. Para una modulación digital, por ejemplo, se emplean los siguientes tipos de modulación:

- **ASK**, (Amplitude Shift Keying, Modulación en Amplitud): la amplitud de la portadora se modula a niveles correspondientes a los dígitos binarios de entrada 1 ó 0.
- **FSK**, (Frequency Shift Keying, Modulación por Desplazamiento de Frecuencia): la frecuencia portadora se modula sumándole o restándole una frecuencia de desplazamiento que representa los dígitos binarios 1 ó 0. Es el tipo de modulación común en módems de baja velocidad en la que los dos estados de la señal binaria se transmiten como dos frecuencias distintas.
- **PSK**, (Phase Shift Keying, Modulación de Fase): tipo de modulación donde la portadora transmitida se desplaza cierto número de grados en respuesta a la configuración de los datos.

En el canal telefónico existen perturbaciones que el módem debe enfrentar para poder transmitir la información. Estos trastornos se pueden enumerar en:

- Distorsiones
- Deformaciones
- Ecos

- Ruidos aleatorios
- Ruidos impulsivos
- Interferencias

Para una modulación analógica se tienen, por ejemplo, los siguientes tipos de modulación:

- **AM** Amplitud Modulada: la amplitud de la portadora se varía por medio de la amplitud de la moduladora.
- **FM** Frecuencia Modulada: la frecuencia de la portadora se varía por medio de la amplitud de la moduladora.
- **PM** Phase Modulation. Modulación de fase: en este caso el parámetro que se varía de la portadora es la fase de la señal, matemáticamente es casi idéntica a la modulación en frecuencia. Igualmente que en AM y FM, es la amplitud de la moduladora lo que se emplea para afectar a la portadora.

2.2.4. Comandos Hayes

Los comandos Hayes fueron creados por el fabricante de módems analógicos del mismo nombre para el control de sus productos desde un terminal de datos. La denominación de “comandos AT” surgió a través de su formato, en el que un comando va siempre precedido por el prefijo “AT”, de “ATtention”, para que el modem pueda sincronizarse y recibir correctamente la cadena de texto. Este método de control de módems pronto fue imitado por otros fabricantes y se constituyó en un estándar de facto en la industria del sector. Finalmente, los comandos AT fueron recogidos en la Recomendación V.25ter de la International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), que posteriormente fue renombrada como V.250. En esta recomendación se fijaba una serie de comandos básicos, pero no los comandos de formato extendido, que han seguido sin regulación. Cada fabricante tenía la libertad de implementar el subconjunto de comandos extendidos que creyese conveniente, así como de añadir otros comandos específicos para sus productos.

Originalmente existían tres grupos principales de comandos AT: los comandos básicos, extendidos y de gestión de registros. Los primeros eran los recogidos en la norma V.25ter y se identificaban por una o varias letras que seguían inmediatamente al prefijo “AT”. Los segundos representaban ampliaciones de comandos básicos, así como comandos adicionales específicos del fabricante. Se distinguían de los primeros en que un carácter “&” o “/” o “+” se anteponían a las letras que identificaban al comando. Por último, los comandos de gestión de los registros estaban identificados por la letra “S” y eran utilizados para variar el valor de uno de los registros del modem que contenían parámetros relativos a su funcionamiento, como por ejemplo el carácter usado para el retorno del carro o el número de tonos de llamada que se debía esperar como máximo al intentar establecer una llamada.

Con la aparición del sistema GSM, se pensó inmediatamente en la adaptación de los comandos AT utilizados en los módems analógicos para el control de los equipos específicos para la transmisión de datos. De esta forma se definieron una serie de comandos basados en los definidos en la recomendación V.25ter, pero adaptados y ampliados para su uso en un entorno celular. Ciertos comandos dentro de esta definición recogida en los estándares técnicos GM 07.07 y 07.05 de la ETSI, son marcados como obligatorios, dejando la implementación de los demás como opcional dependiendo de las características y las capacidades incluidas en cada modem. Se deja abierta la posibilidad de incluir comandos específicos, siempre y cuando no entren en conflicto con los comandos definidos en la norma ya sean obligatorios o no. En esta norma, los comandos extendidos son identificados con el prefijo “+”.

2.2.4.1 Sintaxis

Una línea de comando debe comenzar con la cadena “AT” o “at”, excepto los comandos “A/” y “+++”. Las cadenas “aT” y “At” no son aceptadas. En una sola línea pueden aparecer varios comandos separados por un punto y coma.

Cada formato puede tener hasta tres modos de funcionamiento:

- Básico: realiza la funcionalidad definida en el comando.
- Lectura: devuelve el valor actual de los parámetros del comando. Se identifica por un signo de interrogación (?) situado detrás del identificador de comando. Presente en todos los comandos para la modificación de parámetros, pero no en los comandos de acción.
- Test: devuelve una lista de los parámetros requeridos por el comando, así como sus posibles valores. Se identifica por un signo de interrogación detrás de un signo de igual (=?) al final del identificador de un comando. Presente en todos los comandos extendidos.

Los comandos que requieren un argumento numérico pueden utilizarse sin él, en cuyo caso se entenderá que el valor de dicho argumento es cero. Todas las líneas de comandos excepto “A/” y “+++” deben terminar con el carácter <CR> para que sean procesadas. Este carácter es por lo general el carácter ASCII 13, pero puede ser cambiado modificando el valor del registro S3 mediante el comando AT correspondiente.

Cada comando existente en una línea de comandos puede dar lugar a una o varias líneas de información como respuesta, separadas por la cadena <CRLF>, que por defecto se corresponden con la unión de los códigos ASCII 10 y 13. Al final de dichas respuestas, si todos los comandos se han ejecutado correctamente se envía la cadena “<CRLF>OK<CRLF>” si se ha seleccionado el formato de respuestas verbales mediante el comando V1 o el código “0<CR>” si se ha seleccionado el formato de respuesta numérico con el comando V0. En caso de existir un error en el procesamiento de alguno de los comandos, se devuelve en formato numérico la cadena “4<CR>” y en formato verbal “<CRLF>ERROR<CRLF>”. Este mensaje puede ser sustituido por la

cadena "<CRLF>+CME ERROR: <err> <CRLF>" o bien por "<CRLF>+CMS ERROR: <err> <CRLF>" si se halla activado el modo de errores extendidos. En este último caso <err> representa un código numérico que identifica la causa del error.

Existen dos tipos de comandos que no representan este tipo de sistema. En primer lugar, el comando "AT+CMGS", utilizado para el envío de un comando AT, sigue un algoritmo distinto. Este comando admite como parámetro el número de teléfono de destino entre comillas. A continuación el modem en lugar de devolver una respuesta o un código de terminación, devuelve la cadena "<CR><LF>+" (ASCII 13, 10, 62, 32) para indicar al usuario que puede proceder a introducir el texto del mensaje. Este texto debe terminar con el carácter <CTRL-Z> (ASCII 26). Si el comando puede ejecutarse correctamente, el modem devuelve la cadena "<CR><LF>+CMGS: <id><CR><LF>", donde <id> es un entero entre 0 y 255 que identifica el mensaje enviado, antes de devolver el código de finalización OK. De lo contrario devuelve un mensaje de error en el formato usual.

La otra excepción es un grupo muy reducido de comandos, cuyo principal exponente puede ser "AT+CPIN?". Este comando, que en su modo básico de operación no presenta problema alguno, en su modo de lectura devuelve una línea de respuesta pero no ningún código de terminación confirmando su ejecución correcta.

Por último, existen los llamados códigos de resultado no solicitados. Se trata de indicaciones de sucesos enviadas por el modem sin que exista relación directa con la ejecución de un comando AT. Ejemplos de estos códigos son la indicación de llamada entrante ("RING" o "+CRING") o la indicación de llegada de un mensaje corto ("CMT" o "CMNI").

2.3. MODEM GSM Y PROGRAMACIÓN DEL MICROCONTROLADOR

2.3.1. Módem GSM

Un modem GSM (Global System for Mobile Communications) es un dispositivo que funciona en la red GSM, utilizada mundialmente para la comunicación entre teléfonos móviles. La comunicación se realiza a través de ondas de radio.

El modem GSM puede verse como un teléfono móvil al cual se le ha adaptado una interfaz serial RS-232, con el objeto de poder ser controlado a través de un ordenador o de un microcontrolador como es este caso. A través del modem GSM puede realizarse enlaces para la transmisión de voz, fax, datos, comunicación por Internet y envío mensajes SMS. También existen módems GSM que poseen una interfaz USB para ser controlados.

El modem GSM que elegiría para la realización del proyecto es el Modem Wavecom Q2303A dado que por sus características y buen precio es el que mejor se adaptaría a mis necesidades.



Ilustración 7. Modem GSM

Las características del modem elegido son las siguientes:

Modem GSM	
Modo de operación	Asíncrona
Máxima velocidad	14400 bits/s
Velocidad serie	300 - 115200 bits/s
Velocidad automática	2400 - 19200 bits/s
Frecuencia	GSM 900/1800 MHz
Alimentación	5 - 24V DC
Potencia	Class4 (2W @ 900 MHz) Class5 (1W @ 1800 MHz)

Tabla 3. Especificaciones técnicas del modem GSM

2.3.2. Programación del microcontrolador

Para poder establecer una comunicación entre el microcontrolador y el módem es necesario programar el microcontrolador. Para ello, primero se utiliza una herramienta muy útil que Windows ofrecía hasta la llegada de Windows Vista, que es Hyper-Terminal. Al desaparecer de Windows, han aparecido otros programas muy similares y potentes como Hyperterminal Private Edition.

Hyper-Terminal es un potente programa de comunicación de Windows. Con Hyper-Terminal y un módem es posible conectar el ordenador a un servicio de correo electrónico, bancos de datos, servicio SMS,...

Aunque utilizar Hyper-Terminal con un servicio de boletín electrónico para tener acceso a información de equipos remotos es una práctica que está dejando de ser habitual gracias a Internet. Hyper-Terminal sigue siendo un medio útil para probar y configurar el módem o examinar la conexión con otros sitios, que es para lo que sería útil en este proyecto.

Posteriormente, basándonos en las pruebas realizadas con HyperTerminal pasaríamos a programar el microcontrolador mediante lenguaje C como en el resto de sistemas de esta alarma.

Por diversos motivos ajenos a mi persona como los problemas económicos de la universidad, el gran retraso en la compra y envío del módem y demás problemas varios no se ha podido desarrollar en profundidad el programa de comunicación entre el microcontrolador y el módem para el envío de SMS. A continuación muestro un ejemplo que he realizado de cómo sería dicha comunicación para el envío de un SMS mediante la programación del microcontrolador:

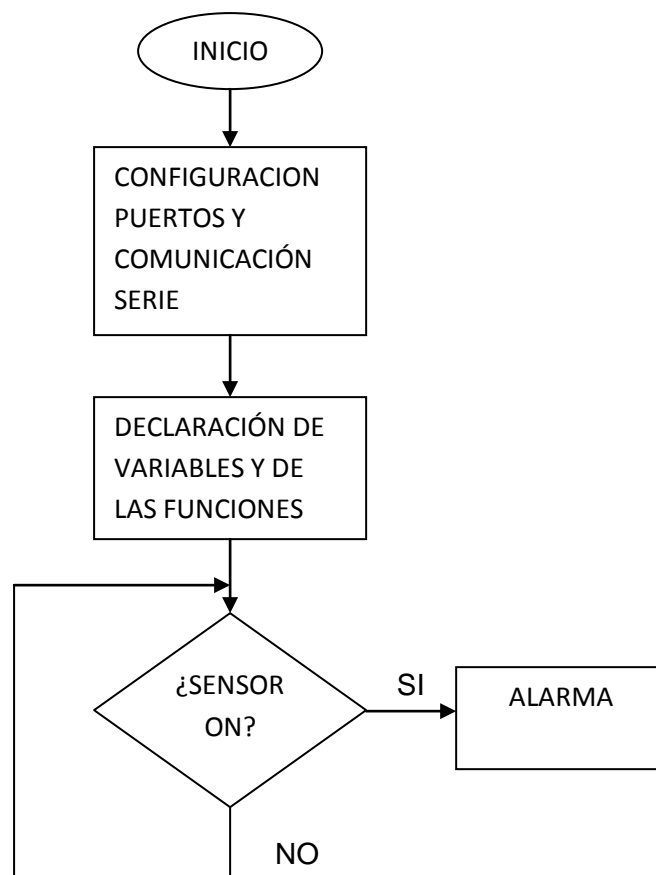


Ilustración 8. Diagrama de flujo del envío de un SMS

```
#FUSES NOWDT, HS, NOPUT, NOLVP
#use delay(clock=8000000)

//Configuración puerto serie
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

//Variables de configuración SMS
char NUMERO[] = "+34XXXXXXXXXX"; //Número de teléfono a marcar
char MENSAJE[] = "ALARMA";       //Mensaje a enviar

int flag=0; //variable auxiliar

#int_EXT
void EXT_isr(void) //función de tratamiento a la interrupción externa
{
    flag=1;
}

//Función para configurar parámetros del modem
void configurar_Modem()
{
    printf("at+cmgf=1\r"); //configura el tipo de formato de SMS en modo TEXTO
    delay_ms(200);
}

//Función para enviar mensajes SMS
void Envio_SMS()
{
    char ctrlz = 26; //combinación de teclas "ctrl+z"

    printf("at+cmgs=\"%s\r\",NUMERO);
    delay_ms(200);
    printf("%s",MENSAJE);
    delay_ms(200);
    printf("%c",ctrlz); //se envía el mensaje
    delay_ms(200);
}

////////PROGRAMA PRINCIPAL//////////
void main()
{
    int i;

    //Configuración registros PIC
    enable_interrupts(GLOBAL); //habilito interrupción global
    enable_interrupts(INT_EXT); //habilito interrupción externa
    ext_int_edge(L_TO_H);       //flanco de activación ascendente

    delay_ms(1000); //retardo para estabilizar

    configurar_Modem(); //llamada a la función de configuración del modem

    while(true) //Bucle infinito hasta interrupción
```

```
{  
  if (flag==1)           // si se produce alarma envío mensajes  
  {  
    Envio_SMS();  
    flag=0;  
  }  
}  
}
```


CAPÍTULO III

DISEÑO DEL HARDWARE DEL SISTEMA

3.1 DESARROLLO DE ESPECIFICACIONES

En el siguiente capítulo se exponen las especificaciones definitivas y detalladas que tendría el hardware del prototipo de simulador domótico y el diseño que le correspondería en el caso de que hubiera sido posible comprar todos los elementos y montarlos.

En primer lugar se realizaría una preinstalación domótica, con lo que se dota a la vivienda con los elementos necesarios (tuberías, cables, cajas, entre otros) para posteriormente poder instalar los servicios avanzados de control y seguridad.

Para poder realizar esta preinstalación, se debe haber definido el lugar exacto de cada componente dentro de la vivienda, es decir, saber la ubicación de cada sensor, controlador y otros elementos en la casa.

3.1.1 Medios de transmisión

Para este prototipo de simulador; cada componente del sistema de control domótico desarrollado, tendría como medio de transmisión líneas de cobre.

Por otro lado, el vertiginoso avance en las tecnologías inalámbricas está haciendo que los dispositivos, cada vez, mejoren sus características y disminuyan su precio. De esta forma, dichos dispositivos han reducido su consumo energético para posibilitar una mayor autonomía, en el caso de tener que instalarlos en lugares donde deban ser alimentados con baterías.

Tecnologías como Wi-Fi, Bluetooth entre otras, nos rodean hoy en día proporcionándonos servicios digitales avanzados. Además los grandes estándares domóticos (EIB-KNX y LonWorks) actualmente disponen de la posibilidad de implantar su protocolo por medio de transmisión inalámbrica, permitiendo realizar instalaciones sin tener que hacer obras.

La tecnología inalámbrica en la actualidad es una realidad y cada vez está más extendida, este hecho junto con los avances de la electrónica embebida, han propiciado una reducción en los costes de fabricación que se ha traducido en una implantación masiva de todo tipo de comunicaciones inalámbricas, que además sigue creciendo.

La tendencia de las tecnologías inalámbricas en el ámbito de los edificios terciarios ha dado lugar en la última década a la aparición de un nuevo protocolo de comunicaciones, optimizado para adaptarse perfectamente a las prestaciones necesarias para soportar los servicios digitales necesarios. ZigBee comunica una serie de dispositivos haciendo que trabajen de forma más eficiente entre sí. Es un transmisor y un receptor que usa baja potencia para trabajar y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. Es ideal para conexiones con diversos tipos de topología, lo que a su vez lo hace más seguro, barato y que no haya ninguna dificultad a la hora de su construcción porque es muy sencilla. Esta tecnología no tiene competencia fuerte con las tecnologías existentes debido a que sus aplicaciones son de automatización de edificios, residenciales e industriales, especialmente para aplicaciones con usos de sensores. Es el protocolo principal de las redes de sensores.

Por otro lado, los edificios cada vez tienen una necesidad más fuerte en lo que se refiere a equipamiento informático y el correspondiente cableado estructurado para distribuir la red local necesaria para suministrar una conexión a internet y compartir archivos. Por ello, la tarea de diseñar la red de comunicaciones interna del edificio para soportar los servicios necesarios, es cada vez más, una labor previa a la utilización del edificio.

La instalación y utilización de redes tiende a estructurarse según una jerarquía, en la cual, existen niveles en los que se sitúan cada uno de los servicios instalados.

La planificación, instalación y utilización de todas estas redes, se realiza hoy en día según unas normas y unos estándares que intentan facilitar la implantación de todas las tecnologías necesarias en el edificio en cuestión.

Las redes de sensores, se encuentran hoy en día en período de inserción dentro de todas las redes comentadas anteriormente. Están ideadas para participar en la gestión del edificio recogiendo los parámetros necesarios para participar en cada uno de los cuatro pilares básicos de la domótica e inmótica hoy en día: eficiencia energética, confort, seguridad y telecomunicaciones.

3.1.2 Elementos

A continuación detallo los elementos necesarios para montar este prototipo con sus correspondientes características y modelos elegidos. Como anteriormente he citado, ha sido imposible comprar los sensores de movimiento, rotura de cristales y contactos magnéticos por motivos económicos y la imposibilidad de montarlos en una vivienda. Por lo que han sido sustituidos por un interruptor.

· **Alimentación:**

Debido a que por problemas económicos no se va a poder montar el dispositivo, para alimentar el prototipo utilizaría una fuente de alimentación. En electrónica, una fuente de alimentación es un dispositivo que convierte la tensión alterna, en una o varias tensiones, prácticamente continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta (ordenador, televisor, impresora, router, etc.).

Estudíé una serie de posibilidades para la elección de la fuente de alimentación. Me decanté por las de banco por su sencillez. Entre las fuentes estudiadas estaban: la fuente PSU-A400-010 del fabricante Laboratory PSU, la fuente ALF2902M del fabricante ELC y la fuente digital KAT 10VD del fabricante Kert.

Dado que el prototipo sólo necesita tensiones de 5 voltios, tampoco es necesaria una fuente de alimentación muy potente y compleja. Por sus características, porque permite una tensión de salida ajustable de 5 a 29 V en incrementos de 2 V, y ajuste preciso de ± 1 V y por su precio, el modelo elegido sería el modelo ALF2902M del fabricante ELC.



Ilustración 9. Fuente de alimentación

A continuación detallo las características de la fuente de alimentación con la cual realizaría el proyecto:

Fuente de alimentación	
Tensión de salida	De 5 a 29V
Corriente de salida	De 2,1 a 4A
Rizado	<3mV ef.
Tensión de entrada	230 Vac a 50-60Hz
Potencia nominal	60W
Dimensiones	106 x 92 x 58mm

Tabla 4. Especificaciones técnicas de la fuente de alimentación

· **Sensor de movimiento:**

Los sensores de movimiento son uno de los tipos de sensores más usados en viviendas y edificios. Permiten controlar la iluminación, la climatización y el sistema de seguridad.

Estos sensores no deben activarse por sí solos con señales falsas como pueden ser variaciones en la luminosidad natural, presencia de animales o variaciones bruscas de temperatura. Los sensores de movimiento deben ser seleccionados según el área que se desee supervisar, el tipo de movimiento que se desee sensar y el tipo de carga que se desee comandar.

Muchos sensores de movimiento tienen integrados dispositivos como por ejemplo sensores de luminosidad, receptores de IR, temporizadores y controladores horarios.

Los sensores de movimiento utilizados serían de tipo PIR (Passive Infra Red), los cuales son los más adecuados en cuanto a un sistema de seguridad se refiere, ya que ofrecen una garantía en la detección de intrusiones gracias a su amplio radio de acción.

Analicé diversos modelos de sensor de presencia como son: el sensor de movimiento 821917017 de la compañía Kopp, el modelo MX+ & QX+ de Gardtec y el sensor Xevox 360° PIR Motion Detector de Abus Security Center.

El modelo elegido sería el Gardtec MX+ & QX+, por su relación calidad-precio y características que se adaptan a nuestras necesidades ya que no son necesarios ángulos de cobertura tan amplios como los posibles en los otros dos modelos.



Ilustración 10. Sensor de movimiento

A continuación se muestran los lóbulos de detección del sensor de movimiento Gardtec MX+ & QX+, a ser utilizado en este prototipo de simulador de control domótico:

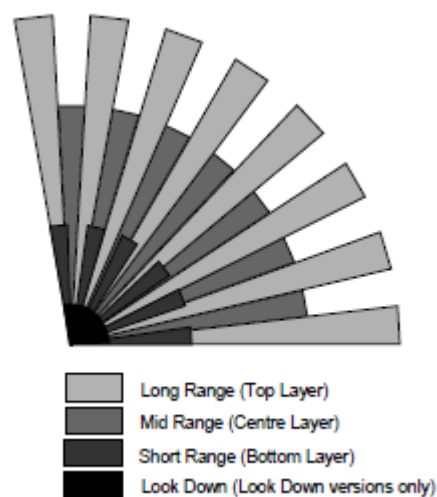


Ilustración 11. Ángulo de detección de la lente del sensor de movimiento

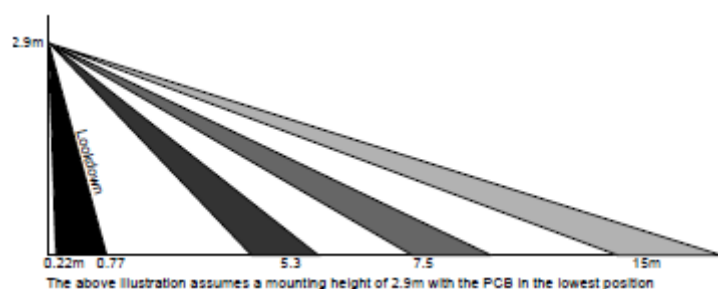


Ilustración 12. Cobertura del alcance del sensor de movimiento

A continuación detallo las características del sensor de movimiento con el cual voy a realizar el proyecto:

Sensor de movimiento	
Modelo	Gardtec MX+ & QX+
Tipo	PIR (Passive Infrared Sensor)
Alimentación	9.5 a 16 VDC
Corriente	13mA
Tipo de lazo	Normalmente cerrado
Alcance	15m
Ángulo	110º

Tabla 5. Especificaciones técnicas de los sensores de movimiento

· Contactos magnéticos (sensores de puertas y ventanas):

Un contacto magnético es un dispositivo que se utiliza para indicar el estado abierto o cerrado de un punto de acceso. Normalmente, es un dispositivo sin contacto que contiene un imán posicionado en la puerta o ventana y un receptor magnético posicionado en el marco que activa un interruptor interno. El controlador es capaz de interpretar la situación de la puerta mediante el cambio de estado del dispositivo.

El contacto magnético es el más común de los contactos de puerta y ventana y puede adquirirse en múltiples formatos (montaje en superficie, empotrado, y otros). Este dispositivo no determina si la puerta está bloqueada o no, sino solamente si está abierta o cerrada. En el control de accesos, se usa para comprobar si un punto de acceso ha sido forzado o ha sido mantenido abierto durante más tiempo del necesario.

A continuación se muestran distintos tipos de sensores de puertas y ventanas, siendo la primera imagen desde la izquierda de la Ilustración 13, el tipo de sensor que sería utilizado en este proyecto.



Ilustración 13. Contactos magnéticos



Ilustración 14. Contacto magnético

Se escogió este tipo de sensores debido a su facilidad para la instalación, su funcionamiento, y la distancia a la que pueden ser colocados con respecto a la central.

Dentro de este tipo de sensores estudié una serie de ellos para decantarme por el cual más se ajustara a mis necesidades. Entre los estudiados se encuentran: el contacto 337-396 de RS, el interruptor de proximidad 190-1011 de RS y por ultimo el MK4200 de Abus Security Center.

A pesar de que el modelo elegido no sea el más barato ni sencillo, el modelo concreto elegido sería el 190-1011 de RS por ser un interruptor normalmente cerrado que es el tipo de interruptor que necesito.

A continuación detallo las características del contacto magnético con el cual voy a realizar el proyecto:

Contactos de puertas y ventanas	
Dimensiones	58 mm x 20 mm x 15 mm
Contacto magnético	Normalmente cerrado
Corriente de conmutación	20mA
Tensión de conmutación	30V d.c.
Espacio de funcionamiento máximo	20 mm
Tipo de activación	Interruptor de proximidad

Tabla 6. Especificaciones técnicas de los contactos magnéticos

· **Sensor de rotura de cristales:**

Un sensor de rotura de cristales es un dispositivo sensor de fractura, con componente sónico de impacto, quiebre y caída de un paño vidriado, integrado por varias unidades o modelos con diferentes principios de funcionamiento. Es muy sensible a ciertas frecuencias de sonido, particularmente la de rotura de cristales y astillado de madera. Hay dos tipos principales de sensores de rotura de cristales: los de choque y los acústicos. Los primeros detectan vibraciones generadas por vidrios rotos, mientras que los acústicos se activan por ondas de sonido.

Cuando el cristal se rompe, la información llega a los sensores de maneras distintas. En los de choque, reciben una alerta en forma de vibración de la ventana misma. Cuando el dispositivo detecta el choque generado por el cristal rompiéndose, se activa. Los sensores acústicos se diseñan específicamente para detectar la frecuencia sonora de los cristales rompiéndose y la madera astillándose. Se activan tan pronto como los sensores cuando la alarma detecta el cristal destrozándose. Protegen un área más grande que los de choque, pero también pueden ser activados por ruidos que confunden con cristales rompiéndose. Ambos tipos de sensores de rotura de cristales están diseñados para detectar una intrusión antes de que el intruso tenga tiempo de entrar a la estructura.

Para la elección del sensor que necesito analicé tres sensores distintos que fueron: el sensor RG71FM-RS de Rokonet, el modelo GB2000 de Abus Security Center y por último el sensor FU7300B de la compañía ABUS.

El dispositivo elegido sería el FU7300B de la compañía ABUS ya que de los tres anteriormente citados es el único sensor de choque. Por su simpleza y para evitar que la alarma se dispare en casos de vibración alta sin ruptura como podría ser el paso de un avión rompiendo o acercándose a la barrera del sonido, es el que mejor se adapta a mis necesidades. Además de ser el más económico de los tres, que para una vivienda con tantas ventanas como ésta es un factor a tener en cuenta.

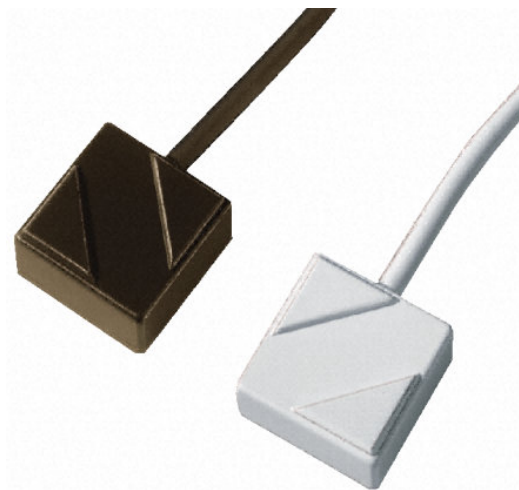


Ilustración 15. Sensor de rotura de cristales

Las características del sensor elegido son las que muestro a continuación:

Sensor de rotura de cristales	
Modelo	FU7300B
Dimensiones	18mm x 18mm x 9mm
Tipo de contacto	Normalmente Cerrado
Tensión de activación máxima	15V DC
Corriente de activación máxima	15mA
Temperatura máxima de operación	55°C
Longitud de cable	2m
Peso	40g
Tipo de sensor	Piezoeléctrico

Tabla 7. Especificaciones técnicas de los sensores de rotura de cristales

· **Microcontrolador:**

El campo de los microcontroladores es muy amplio en cuanto a compañías, tipos y plataformas que los desarrollan y fabrican. A grandes rasgos podemos definir un microcontrolador como un circuito integrado programable para desempeñar funciones completamente personalizadas, según lo que el usuario desee; es como un pequeño ordenador. Para programarlo, suele ser necesario adquirir también una tarjeta de evaluación o una placa de desarrollo la cual se conecta al ordenador.

Durante el estudio de la gama de microcontroladores y plataformas en el mercado, analicé la posibilidad de trabajar con Arduino. Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (*boot loader*) que corre en la placa.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Al ser open-hardware, tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Otra de las opciones que analicé fue trabajar con microcontroladores PIC. Los **PIC** son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El nombre actual no es un acrónimo. En realidad, el nombre completo es **PICmicro**, aunque generalmente se utiliza como *Peripheral Interface Controller* (controlador de interfaz periférico).

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada y salida, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba micro código simple almacenado en ROM para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador.

El PIC, sin embargo, se mejoró con EEPROM para conseguir un controlador de canal programable. Hoy en día multitud de PIC's vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una *palabra* corresponde a una instrucción en lenguaje ensamblador, y puede ser de 12, 14, 16 ó 32 bits, dependiendo de la familia específica de PICmicro).

Al final me decanté por el uso de un PIC ya que de esta manera se estudia mejor el funcionamiento del microcontrolador de manera independiente y cómo programarlo. Con Arduino la cantidad de librerías y programas ya hechos es inmensa por lo que no aprendería un lenguaje de programación, que es algo que busco con la realización de este proyecto. Otro de los motivos de mi elección, es que el departamento de Ingeniería Eléctrica, con el que realizo el proyecto, acostumbra a trabajar con este tipo de microcontroladores.

Dentro de la amplia gama de PIC's estudié cual se adaptaba mejor a mis necesidades para el proyecto y en el tema económico. Con la placa de desarrollo EasyPIC v7 que se compró para la realización del proyecto venía incorporado un PIC18F45K22 y el departamento de Ingeniería Eléctrica suele trabajar con un PIC16F876, así que estudié ambos.

El microcontrolador elegido fue el PIC16F876, porque tanto por su número de puertos como de capacidad de memoria era suficiente para el control de los periféricos y la grabación del programa y las contraseñas. Era el que mejor optimizaba todos sus recursos de los dos. El PIC18F45K22 es un microcontrolador demasiado potente para el proyecto que voy a realizar ya que dispone de 5 puertos y una cantidad de memoria excesiva para lo que necesito.

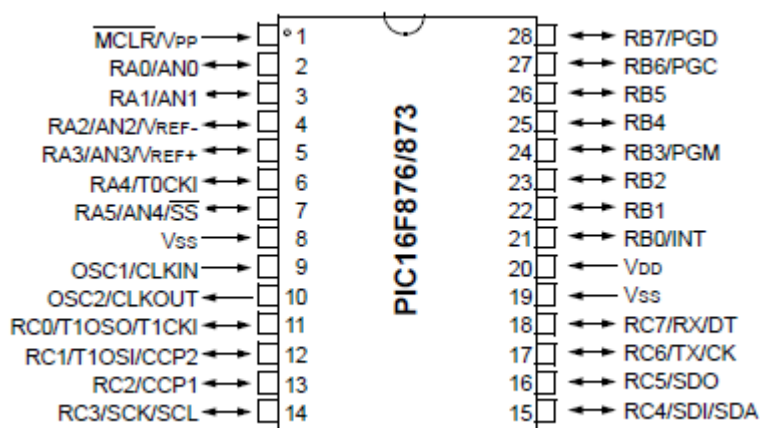


Ilustración 16. Diagrama general del microcontrolador PIC16F876

A continuación detallo las características del PIC16F876 con el cual voy a realizar el proyecto:

Microcontrolador		
Memoria de programa (flash)	Bytes	14336
	Palabras	8192x14
Memoria de datos	Bytes EEPROM	256
	Bytes RAM	368
Conversor A/D		8 (10 bits)

BOR (detección de baja tensión)	SÍ
Líneas I/O	22 Líneas 3 Puertos
Comunicación Serie	USART/MSSP
CCP (Captura, comparación y ancho de pulso)	2
Temporizadores	1 de 16 bits 2 de 8 bits 1 de WDT
Frecuencia Máxima	20Mhz
ICSP (Programación serie en circuito)	SÍ
Resets	POR, BOR (PWRT, OST)
Encapsulado	PDIP, SOIC
Fuentes de interrupción	13
Comunicación Paralelo	NO
Voltaje	2,0V a 5,5V
Corriente máxima	25mA

Tabla 8. Especificaciones técnicas del microcontrolador PIC16F876

· Reloj de tiempo real:

El reloj en tiempo real que necesitamos para la realización del simulador de presencia se puede implementar de dos maneras: por software o por hardware.

Una vez analizadas las dos maneras me decanté por hacerlo mediante hardware dado que por software el programa ha realizar es complicado. El integrado es muy económico y fácil de conectar con el microcontrolador mediante comunicación I2C, ya que tanto el integrado como el microcontrolador disponen de dicha tecnología.

Analicé tres relojes en tiempo real dentro de la amplia gama que existe. El primero de ellos fue el modelo MCP79410-I/MS de Microchip, después el DS1307 de Maxim y por último el reloj ISL1219IUZ de la compañía Intersil.

Después de analizar las tres opciones y de acuerdo a lo recomendado en las fuentes consultadas el reloj de tiempo real elegido es el DS1307 de Maxim por ser el más completo de los tres. Es un dispositivo de bajo consumo de energía, completo con código binario decimal (BCD), reloj/calendario más 56 bytes de NV SRAM. Dirección y datos son transferidos a través de 2 hilos serie, bus bi-direccional. El reloj/calendario provee información de, segundos, minutos, horas, día, fecha, mes y año. El final de fecha de mes se ajusta automáticamente durante meses menores de 31 días, incluyendo correcciones para el año bisiesto. El reloj funciona en cualquiera formato de 24 horas o en 12 horas con indicador AM/PM. El DS1307 tiene incorporado un circuito de

sensor de tensión que detecta fallas de energía y cambia automáticamente al suministro de batería de respaldo.

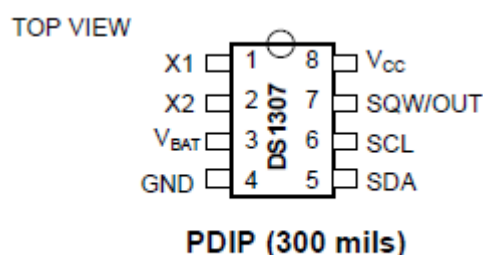


Ilustración 17. Diagrama general del reloj de tiempo real DS1307

A continuación detallo las características del reloj en tiempo real con el cual voy a realizar el proyecto:

RECOMMENDED DC OPERATING CONDITIONS

($T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		$V_{CC} + 0.3$	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 4.5\text{V}$ to 5.5V ; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	μA
Logic 0 Output ($I_{OL} = 5\text{mA}$)	V_{OL}				0.4	V
Active Supply Current ($f_{SCL} = 100\text{kHz}$)	I_{CCA}				1.5	mA
Standby Current	I_{CCS}	(Note 3)			200	μA
V_{BAT} Leakage Current	I_{BATLKG}			5	50	nA
Power-Fail Voltage ($V_{BAT} = 3.0\text{V}$)	V_{PF}		$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 0\text{V}$, $V_{BAT} = 3.0\text{V}$; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON); SQW/OUT OFF	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}			480	800	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATDR}			10	100	nA

Tabla 9. Especificaciones técnicas del reloj en tiempo real DS1307

· Display:

Para la visualización de los distintos menús, estado de la alarma y demás mensajes es necesario un display. He estudiado el funcionamiento de dos tipos de displays: un LCD de 16x2 caracteres y un display táctil gráfico de 128x64 caracteres. Dado que no vamos a representar gráficos ni figuras, ni necesitamos una pantalla táctil he optado por el LCD de 16x2 caracteres ya que se adapta mejor a mis necesidades y es más asequible económicamente.

La gama de displays es muy amplia por su variedad de tipos según número de filas y columnas, tamaño de los caracteres, calidad de la pantalla,... Entre las muchas opciones estudié tres: el modelo 162D-BA-BC de Displaytech, el display LM016L de Hitachi y por último el display FDCC1602P-RNNYBW-66SE del fabricante Fordata.

Elegí el modelo LM016L de Hitachi por su sencillez y por ser el display que suele utilizar el departamento de Ingeniería Eléctrica con el que realizo el proyecto.



Ilustración 18. Display LM016L de Hitachi

Las características básicas del display con las que voy a realizar el proyecto son las siguientes:

Display	
Modelo	LM016L
Dimensiones	84mm x44mm x10.5mm
Tamaño carácter	2.96mm x 4.86mm
Tamaño punto	0.56mm x 0.66mm
Alimentación	De 0 a 6,5V
Voltaje de entrada alto	2.2V min
Voltaje de entrada bajo	0.6V máx.
Voltaje de salida alto	2.4V min

Voltaje de salida bajo	0.4V máx.
Corriente de suministro con Vdd= 5V	1.0 mA típica
	3.0mA máx.

Tabla 10. Especificaciones técnicas del display

· Teclado:

Para la introducción de las contraseñas y navegación por el menú he elegido un teclado de 4 filas por 3 columnas. Me he decantado por uno de este tipo y no por uno de 4x4, de uso también muy común, debido a que por el número de pines del puerto del microcontrolador y como lo tengo configurado, este se adapta mejor a mi esquema y a mis necesidades. Ya que el teclado lo gestiono por la interrupción del nibble alto del puerto B y que el pin 0 del puerto B lo tengo ocupado, este tipo de teclado puedo agruparlo todo en el puerto B. Un número de columnas mayor ofrece más posibilidades para la introducción de contraseñas pero con las teclas de las que dispongo lo considero suficiente.

Existen dos tipos de teclados: matricial y de punto común. He elegido uno matricial debido a que el número de pines necesarios en el microcontrolador es menor, ya que solo son necesarios 7 pines en este caso, 4 pines para las filas y 3 para las columnas. Esta característica me resulta muy útil para continuar con la idea de optimizar los recursos del microcontrolador y que no queden una gran cantidad de pines sin utilizar. En los de punto común sería necesario un pin para cada tecla del teclado.

El teclado elegido es el modelo ECO.12150.06 de la compañía EOZ porque cumple con las características que necesito y por su bajo coste.



Ilustración 19. Teclado

Las características del teclado son las siguientes:

Teclado	
Modelo	ECO.12150.06
Dimensiones	64mm x 51mm x 10,6mm
Tensión nominal	24V

Corriente nominal	20mA
Voltaje máximo	24V
Voltaje mínimo	500mV
Corriente mínima	10mA

Tabla 11. Especificaciones técnicas del teclado

· **Sirena:**

Para el aviso de la entrada de intrusos cuando los sensores detecten su presencia o cuando el número de intentos de ingreso de la clave de seguridad supere los permitidos, es necesario el uso de una sirena o avisador.

Las posibilidades para ello son infinitas pero, una vez estudiadas las posibilidades como: uso de una luz rotativa, uso de un zumbador o avisador acústico, la combinación de la luz rotativa y el zumbador,... Al final optaría por la combinación de ambos tipos de aviso por ser el más completo y por la posibilidad de encontrarlo por un precio económico.

Dentro de la opción de combinación de avisador acústico y luminoso existe una amplia gama modelos dependiendo del tipo de luz, avisador luminoso continuo o intermitente, número de decibelios,... Analicé tres modelos distintos que son: PSC-0002 del fabricante Klaxon, el modelo 85710R0/HB del fabricante Hosiden Besson y para finalizar el SYAV/RL/R de la compañía Fulleon.

La sirena elegida sería la baliza acústica destellante 85710R0/HB del fabricante Hosiden Besson ya que se adapta a los requisitos que buscamos, ofrece 32 tipos distintos de tonos y por su económico precio.



Ilustración 20. Sirena

Las características de la sirena elegida son las siguientes:

Sirena	
Velocidad	1 destello/seg
Corriente a 12V	90mA
Corriente a 24V	40mA
Volumen máximo	112 dB

Tabla 12. Especificaciones técnicas de la sirena

3.2 ESTUDIO DE LOS PLANOS DE LA VIVIENDA

En función del nivel de automatización y sus futuras ampliaciones el proyecto domótico cambia, así como las diferentes instalaciones a integrar. Es por esto que para cada sistema se tiene que realizar un estudio previo de las características de la vivienda en donde se va a instalar el proyecto, teniendo en cuenta las necesidades del usuario y la topología física de la casa.

Por lo tanto, el prototipo desarrollado sería un sistema centralizado, con servicio domótico básico, el cual brinda una gestión y control de alarmas técnicas y de seguridad básico, que cuenta con:

- Gestión de seguridad
- Gestión de las comunicaciones
- Interfaces de usuario.

Si por ejemplo el plano de la vivienda fuera el siguiente, estudiaríamos el número de habitaciones con ventana al exterior, el número de puertas al exterior también, la cantidad de luminarias exteriores, el lugar donde iría el panel de control y los sensores de movimiento y los posibles puntos sensibles por los que podrían entrar los intrusos.



Ilustración 21. Plano de la vivienda

He elegido el plano de una vivienda de una planta con porche exterior ya que ofrece más posibilidades de diseño y la necesidad de un sistema de alarma es mayor que en el caso de una vivienda de un bloque de pisos. En este tipo de viviendas hay ventanas que dan al exterior a baja altura y un mayor número de puertas por las que acceder a la vivienda.

3.2.1 Determinación del número de circuitos de iluminación

Para determinar el número de luminarias a encender para el simulador de presencia, se tomarían en cuenta las luminarias que dan a los exteriores, es decir, todas aquellas luminarias que se pueden observar desde fuera de la vivienda, esto es para ahorrar energía, y que personas ajenas a la vivienda puedan percibir que se encuentra alguien en la misma. Dichas luminarias se encenderán a una hora determinada del día, generalmente cuando anochezca y se apagaran a otra hora que fijemos al amanecer. Por lo tanto, en este caso, se escogerían cuatro luminarias, las cuales están situadas en la parte exterior de la casa (dos en el porche 1 y las otras dos en el porche 2).



Ilustración 22. Distribución de luminarias en la vivienda

3.2.2 Determinación de la ubicación de sensores de movimiento, contactos magnéticos y sensores de rotura de cristales.

Los sensores de movimiento se instalan normalmente en todas las habitaciones, el salón y la cocina. Su ubicación depende de la tipología de la habitación, pero una vez estudiados los planos de la vivienda, los sensores se colocarían tal y como se muestran en la Ilustración 23. El sensor de movimiento del salón se colocaría de tal forma que cuando alguien entrara por la puerta principal, pueda manejar el panel de control sin que se active ya que dicha persona puede ser un habitante de la vivienda.

Los contactos magnéticos de ventanas y los sensores de rotura de cristales los colocaríamos en todas las ventanas que dieran al exterior. Los contactos magnéticos de puertas serían colocados uno en la puerta que une el garaje con la cocina, otro en la puerta que une el garaje con la despensa y el último en la puerta que del porche 2. La puerta del porche 2 suponemos que es de cristal ya que une el porche con el salón por lo que también llevaría un sensor de rotura de cristales.

Cuando uno de los sensores se activara, ya sea porque se haya abierto una puerta o ventana o se haya detectado movimiento dentro de la vivienda, saltaría la alarma y se encendería la sirena durante el tiempo fijado. Este tiempo no puede ser elevado para no molestar a los vecinos.

Para la ubicación de estos sensores, se tomarían en cuenta las siguientes especificaciones:

- Evitar la ubicación en superficies donde reciban directamente la luz solar.
- Evitar ubicarlos en áreas donde puedan existir cambios bruscos de temperatura.
- Evitar la ubicación de los sensores en áreas con conductos de aire.
- Elegir una ubicación donde se pueda interceptar rápidamente una intrusión.
- Tener en cuenta que los sensores de movimiento reaccionan mejor ante un ambiente estable.

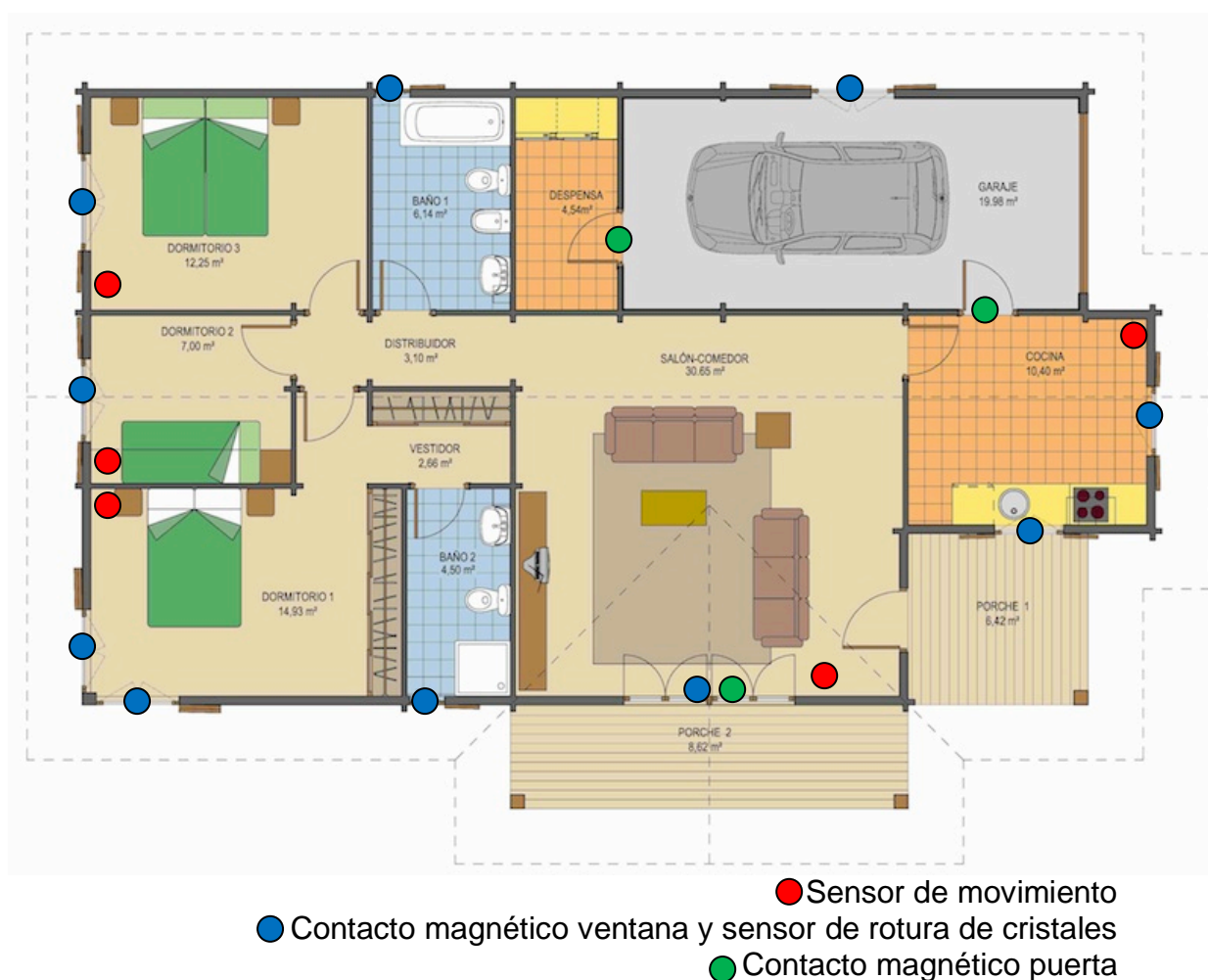
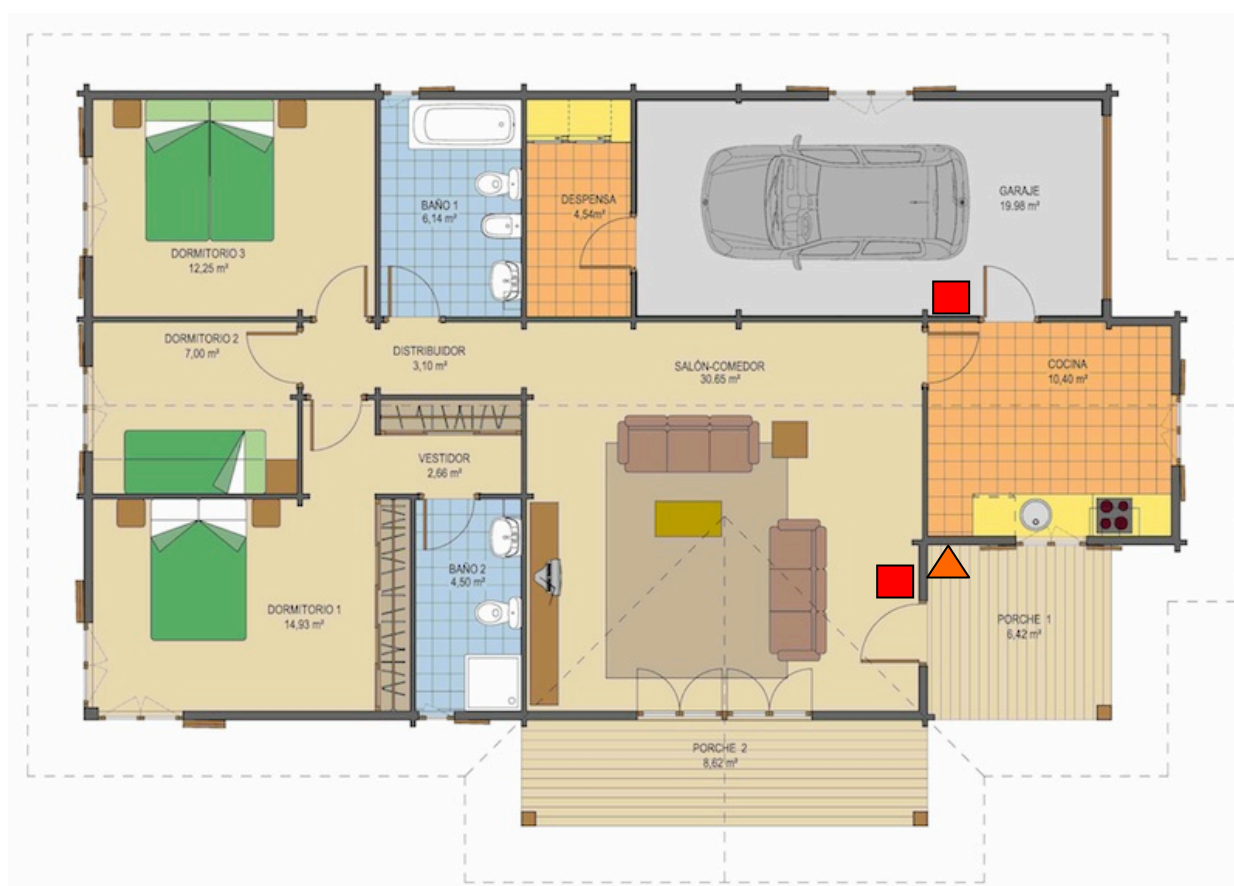


Ilustración 23. Distribución de sensores dentro de la vivienda.

3.2.3 Estudio de planos eléctricos y otras instalaciones de interés a instalar en la vivienda

El estudio de los planos eléctricos sirve para tener en cuenta donde realizar las conexiones y poder ubicar los elementos. Con esto, se ubicarían los siguientes elementos:

- Dos paneles de control: uno cerca de la puerta principal de entrada, para que de esta forma el usuario tenga la facilidad de activar o desactivar el sistema al momento de abandonar o llegar a su residencia sin que se active el sensor de movimiento del salón y otro situado en el garaje.
- Una sirena; cuya ubicación habitual es en la fachada principal de la vivienda, en un lugar visible y de difícil acceso desde la calle.



■ Panel de control
▲ Sirena

Ilustración 24. Distribución del panel de control y de las sirenas.

3.3 DISEÑO DEL HARDWARE DE CONTROL

3.3.1 Herramientas de desarrollo

Se utilizan varias herramientas de desarrollo para la elaboración de este proyecto, las cuales intervienen en diferentes aspectos del mismo. Todas las herramientas a continuación mencionadas se aplican en este prototipo, pero se utilizan dependiendo de las características que cada una posee.

Cabe destacar la diferencia entre un compilador y un IDE. Un IDE es un entorno de desarrollo integrado (siglas en inglés, Integrated Development Environment). Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Un compilador traduce el lenguaje o código fuente original a código de máquina.

Compilador:

- PCW CCS

Las características que presenta el compilador PCW CCS y que hacen de él una buena opción para elegirlo como compilador de C para programar Microcontroladores PIC son: al compilar genera un código máquina muy compacto y eficiente, se integra perfectamente con MPLAB y otros simuladores/emuladores como PROTEUS para el proceso de depuración, incluye una biblioteca muy completa de funciones pre compiladas para el acceso al hardware de los dispositivos (entrada/salida, temporizaciones, conversor A/D, transmisión RS-232, bus I2C..., etc.) , incorpora drivers para dispositivos externos, tales como pantallas LCD, teclados numéricos, memorias EEPROM, conversores A/D, relojes en tiempo real, etc.(los drivers son pequeños programas que sirven de interfaz entre los dispositivos hardware y nuestro programa) y permite insertar partes de código directamente en Ensamblador, manteniendo otras partes del programa en C.

Grabador del microcontrolador:

- MikroProg Suite for PIC

MikroProg Suite for PIC es el software que requiere el programador MikroProg que va incluido en la placa de desarrollo EasyPic v7 con la que vamos a programar el microcontrolador.

Este software nos permite programar todos los microcontroladores de la familia Microchip, incluyendo los PIC10, PIC12, PIC16, PIC18, dsPIC30/33, PIC24 y PIC32. Dispone de una interfaz muy intuitiva y tecnología SingleClick de programación.

Simulador:

- Proteus

Proteus es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra.

El Programa ISIS, Intelligent Schematic Input System (*Sistema de Enrutado de Esquemas Inteligente*) permite diseñar el plano eléctrico del circuito que se desea realizar con componentes muy variados, desde simples resistencias, hasta alguno que otro microprocesador o microcontrolador, incluyendo fuentes de alimentación, generadores de señales y muchos otros componentes con prestaciones diferentes. Los diseños realizados en Isis pueden ser simulados en tiempo real, mediante el módulo VSM, asociado directamente con ISIS. Una de las prestaciones de Proteus, integrada con ISIS, es VSM, el Virtual System Modeling (*Sistema Virtual de Modelado*), una extensión integrada con ISIS, con la cual se puede simular, en tiempo real, con posibilidad de más rapidez; todas las características de varias familias de microcontroladores, introduciendo nosotros mismos el programa que controlará el microcontrolador y cada una de sus salidas, y a la vez, simulando las tareas que queramos que lleve a cabo con el programa. Se pueden simular circuitos con microcontroladores conectados a distintos dispositivos, como motores, displays LCD, teclados en matriz, etc. Incluye, entre otras, las familias de PIC's PIC10, PIC12, PIC16, PIC18, PIC24 y dsPIC33. ISIS es el corazón del entorno integrado PROTEUS. Combina un entorno de diseño de una potencia excepcional con una enorme capacidad de controlar la apariencia final de los dibujos.

ARES, o Advanced Routing and Editing Software (*Software de Edición y Ruteo Avanzado*); es la herramienta de enrutado, ubicación y edición de componentes, se utiliza para la fabricación de placas de circuito impreso, permitiendo editar generalmente, las capas superficial (Top Copper), y de soldadura (Bottom Copper). Ejecutando ARES directamente de forma manual ubicando cada componente en el circuito. Tener cuidado al DRC, Design Rules Checker (Verificador de Reglas de Diseño) Proteus es una potente herramienta usada por los mecatrónicos. El propio programa puede trazar las pistas de manera automática, si se guarda previamente el circuito en ISIS, y haciendo clic en el ícono de ARES, en el programa, el programa compone la Netlist.

Método 1 (Autorouter)

1. Poner SOLO los componentes en la board
2. Especificar el área de la placa (con un rectángulo, tipo "Board Edge")
3. Hacer clic en "Autorouter", en la barra de botones superior
 1. Editar la estrategia de ruteo en "Edit Strategies"
4. Hacer clic en "OK"

Método 2 (Electra Autorouter)

Utilizando el módulo Electra (Electra Auto Router), el cual, una vez colocados los componentes trazará automáticamente las pistas realizando varias pasadas para optimizar el resultado.

Con Ares además se puede tener una visualización en 3D del PCB que se ha diseñado, al haber terminado de realizar la ubicación de piezas, capas y ruteo, con la herramienta "3D Visualization", en el menú output, la cual se puede demorar, solo haciendo los trazos un periodo de tiempo un poco más largo que el de los componentes, los cuales salen al empezar la visualización en 3D.

3.3.2 Requerimientos de funcionamiento

Los diferentes tipos de sistemas domóticos ofrecen distintas características de funcionamiento. La variedad de sensores y las características propias del sistema, hacen que cada uno tenga requerimientos diferentes. Por cual, este sistema cumple con los siguientes requerimientos:

- Posee una central, la cual es la encargada de controlar todo el sistema en general, como: la activación, programación, encendido de luces y control del panel.
 - Debe controlar los periféricos, tanto los de entrada (sensores) como los de salida (sirena, iluminación).
 - Una vez que esté activado el sistema, empezar el simulador de presencia, el cual es el encargado de encender las luces a partir de una hora establecida.
 - Activar la sirena, una vez que ha habido intrusión dentro de la vivienda y el envío de un SMS al propietario de la vivienda.
- Poseer dos paneles de control, desde los cuales se pueda activar / desactivar el sistema, cambiar la clave y otros. Por lo general, cada sensor de movimiento pertenece a una zona diferente, asimismo la sirena también pertenece a otra zona ya que tiene una función específica muy distinta a las demás.
- Se deben instalar sensores de movimiento y sensores de puertas y ventanas en los lugares necesarios para la seguridad de la vivienda, para que estos sensores sean los encargados de determinar alguna intrusión en la misma.

3.4 PROGRAMACIÓN DEL CONTROLADOR

Programar es hacer una lista de instrucciones en un lenguaje que nos permita indicarle al PIC lo que deseamos que haga. Existen varios lenguajes como: Ensamblador, Basic, C, etc. Todos ellos pretenden acercarse a nuestra manera de pensar y de hablar. Sin embargo los PIC no conocen más que unos y ceros. Por eso es necesario un compilador.

Compilar es traducir el programa al lenguaje de máquina que si “entiende” el PIC. Para realizar esta traducción hacemos uso de un software que transforma el “Programa Fuente”, en otro que si podemos comunicarle al PIC. Existen compiladores “especiales” en los que se puede editar en el lenguaje Basic o C y obtener un archivo .hex listo para que su grabador de PIC's lo utilice.

Dentro del amplio abanico de lenguajes de programación, analicé los dos más conocidos y utilizados, mas una variante de uno de ellos, para posteriormente decantarme por uno de ellos:

- BASIC

La sintaxis mínima de BASIC sólo necesita los comandos LET, INPUT, PRINT, IF y GOTO. Un intérprete que ejecuta programas con esta sintaxis mínima no necesita una pila. Algunas de las primeras implementaciones eran así de simples. Si se le agrega una pila, se pueden agregar también ciclos FOR anidados y el comando GOSUB.

BASIC no tiene una biblioteca externa estándar como otros lenguajes como C. En cambio, el intérprete (o compilador) contiene una biblioteca incorporada de procedimientos intrínsecos. Estos procedimientos incluyen la mayoría de las herramientas que un programador necesita para aprender a programar y escribir aplicaciones sencillas, así como funciones para realizar cálculos matemáticos, manejar cadenas, entrada desde la consola, gráficos y manipulación de archivos.

BASIC, como Pascal, hace una distinción entre un procedimiento que no devuelve un valor (llamado subrutina) y un procedimiento que lo hace (llamado función). Muchos otros lenguajes (como C) no hacen esa distinción y consideran todo como una función (algunas que devuelven un valor “void” [vacío]).

BASIC es reconocido por tener muy buenas funciones para manipular cadenas de caracteres. En BASIC las variables no necesitan forzosamente ser declaradas antes de usarse, excepto los arreglos de más de 10 elementos; aunque versiones BASIC relativamente modernas poseen la opción (considerada buena práctica de programación) para obligar al programador a declarar toda variable antes de su uso (una directiva como OPTION EXPLICIT). La declaración de variables en BASIC se hace usando la palabra clave DIM.

Muchos lenguajes también soportan tipos numéricos adicionales, como enteros de 16 y 32 bits (simple y long, respectivamente), además de sus números de coma flotante. Adicionalmente algunos permiten la utilización de tipos de datos definidos por el usuario, similar a los "records" de Pascal, o las "structs" de C.

· C

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

Este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina concreta. Se le suele llamar lenguaje de programación de sistemas debido a su utilidad para escribir compiladores y sistemas operativos, aunque de igual forma se puede desarrollar cualquier tipo de aplicación.

C trabaja con tipos de datos que son directamente tratables por el hardware de la mayoría de computadoras actuales, como son los caracteres, números y direcciones. Estos tipos de datos pueden ser manipulados por las operaciones aritméticas que proporcionan las computadoras. No proporciona mecanismos para tratar tipos de datos que no sean los básicos, debiendo ser el programador el que los desarrolle. Esto permite que el código generado sea muy eficiente y de ahí el éxito que ha tenido como lenguaje de desarrollo de sistemas. No proporciona otros mecanismos de almacenamiento de datos que no sea el estático y no proporciona mecanismos de entrada ni salida. Ello permite que el lenguaje sea reducido y los compiladores de fácil implementación en distintos sistemas. Por contra, estas carencias se compensan mediante la inclusión de funciones de librería para realizar todas estas tareas, que normalmente dependen del sistema operativo.

· MikroC

Este lenguaje es muy similar al C estándar, no obstante en determinados aspectos difiere del ANSI estándar en algunas características. Algunas de estas diferencias se refieren a las mejoras, destinadas a facilitar la programación de los microcontroladores PIC, mientras que las demás son la consecuencia de la limitación de la arquitectura del hardware de los PIC.

Después de un periodo estudiando y probando ambos lenguajes de programación me decante por el lenguaje C debido a las siguientes ventajas que presentaba respecto al otro lenguaje de programación:

- Lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas.
- A pesar de su bajo nivel es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos.

- Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes.

3.4.1 Lógica de programación

Dado que por motivos económicos no se va a realizar la construcción de todo el sistema de seguridad, hemos simulado el sistema a menor escala, siendo los sensores representados mediante un interruptor y la sirena y las luminarias por diodos LED.

A continuación muestro el esquema eléctrico del prototipo de simulación que he podido montar.

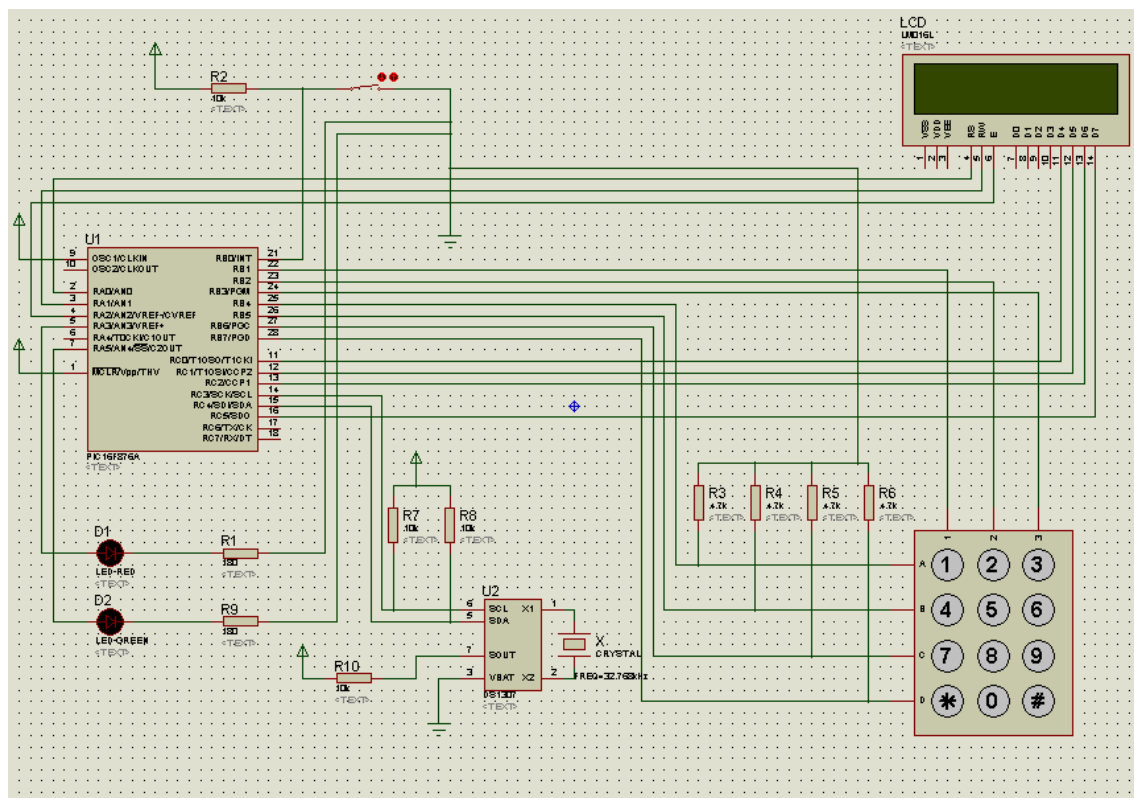


Ilustración 25. Esquema eléctrico del sistema de alarma

Los sensores estarían conectados al PortB.0 para trabajar con ellos mediante la interrupción externa del microcontrolador, la cual estará habilitada cuando la alarma esté activada. El teclado también trabaja mediante interrupciones, pero en este caso es mediante la interrupción del cambio de nivel del nibble alto del puerto B.

Los procesos de atención a interrupciones tienen la ventaja de que se implementan por hardware ubicado en el procesador, así que es un método rápido de hacer que el procesador se dedique a ejecutar un programa especial

para atender eventos que no pueden esperar por mecanismos lentos como el de encuesta. En términos generales, un proceso de interrupción y su atención por parte del procesador, tiene la siguiente secuencia de acciones:

1. En el mundo real se produce el evento para el cual queremos que el procesador ejecute un programa especial, este proceso tiene la característica de que no puede esperar mucho tiempo antes de ser atendido o no sabemos en qué momento debe ser atendido.
2. El circuito encargado de detectar la ocurrencia del evento se activa, y como consecuencia, activa la entrada de interrupción del procesador.
3. La unidad de control detecta que se ha producido una interrupción y “levanta” una bandera para registrar esta situación; de esta forma si las condiciones que provocaron el evento desaparecen y el circuito encargado de detectarlo desactiva la entrada de interrupción del procesador, ésta se producirá de cualquier modo, porque ha sido registrada.
4. La unidad de ejecución termina con la instrucción en curso y justo antes de comenzar a ejecutar la siguiente comprueba que se ha registrado una interrupción
5. Se desencadena un proceso que permite guardar el estado actual del programa en ejecución y saltar a una dirección especial de memoria de programas, donde está la primera instrucción de la subrutina de atención a interrupción.
6. Se ejecuta el código de atención a interrupción, esta es la parte “consciente” de todo el proceso porque es donde se realizan las acciones propias de la atención a la interrupción y el programador juega su papel.
7. Cuando en la subrutina de atención a interrupción se ejecuta la instrucción de retorno, se desencadena el proceso de restauración del procesador al estado en que estaba antes de la atención a la interrupción.

Como podemos observar, el mecanismo de interrupción es bastante complicado, sin embargo tiene dos ventajas que obligan a su implementación: la velocidad y su capacidad de ser asíncrono. Ambas de conjunto permiten que aprovechemos al máximo las capacidades de trabajo de nuestro procesador.

Los mecanismos de interrupción no solo se utilizan para atender eventos ligados a procesos que requieren atención inmediata sino que se utilizan además para atender eventos de procesos asíncronos.

Las interrupciones son tan eficaces que permiten que el procesador actúe como si estuviese haciendo varias cosas a la vez cuando en realidad se dedica a la misma rutina de siempre, ejecutar instrucciones una detrás de la otra.

Las dos líneas que necesitamos del reloj de tiempo real están conectadas con sus correspondientes del microcontrolador.

Los pines del microcontrolador están distribuidos de la siguiente manera:

Microcontrolador		
Pin	Estado	Función
PortA.0	1/0	Línea RS del display
PortA.1	1/0	Línea RW de lectura y escritura del display
PortA.2	1/0	Línea E del display que habilita la señal
PortA.3	0	Led que simula la sirena
PortA.4	Libre	
PortA.5	0	Led que simula las luminarias
PortB.0	1	Señal que indica presencia de intrusos
PortB.1	0	Conexión con columnas del teclado para detectar tecla pulsada
PortB.2	0	
PortB.3	0	
PortB.4	1	Conexión con filas del teclado para que mediante interrupciones detectemos la fila en que se encuentra la tecla pulsada
PortB.5	1	
PortB.6	1	
PortB.7	1	
PortC.0	1	Línea D4 de datos del display
PortC.1	1	Línea D5 de datos del display
PortC.2	1	Línea D6 de datos del display
PortC.3	1	SCL del ds1307 para sincronizar el movimiento de datos en la interfaz de serie.
PortC.4	1	SDA pin entrada/salida del ds1307 para el interfaz I2C
PortC.5	1	Línea D7 de datos del display
PortC.6	1	
PortC.7	1	

Tabla 13. Pines del microcontrolador PIC16F876

Para el programa del panel, se establece primero la comunicación con el LCD, teclado y microcontrolador.

Estos datos son los que indican que acción debe realizar el microcontrolador, ya que desde el panel se puede modificar la clave y lo más importante se puede activar y desactivar el sistema.

El teclado es conectado de forma paralela al controlador, y se maneja con interrupciones, al presionar cada tecla se genera una interrupción y el controlador realiza la acción solicitada.

Como se puede observar en la Ilustración 26, primero se realiza una inicialización general del sistema, para inicializar los periféricos y mostrar el mensaje de bienvenida. Esto se realiza mediante el seteo o reseteo de varios

pinos del microcontrolador. A continuación comprobamos el estado de los sensores. Si se ha detectado una intrusión se activara la alarma y si no, procederemos a cambiar la clave que viene por defecto y después iremos al menú principal. Al mismo tiempo, el reloj en tiempo real del simulador de presencia estará funcionando. Dicho reloj interno, nos dice en qué momento debe empezar y terminar de funcionar el simulador, ya que este solo debe funcionar en el rango de horas que fijemos.

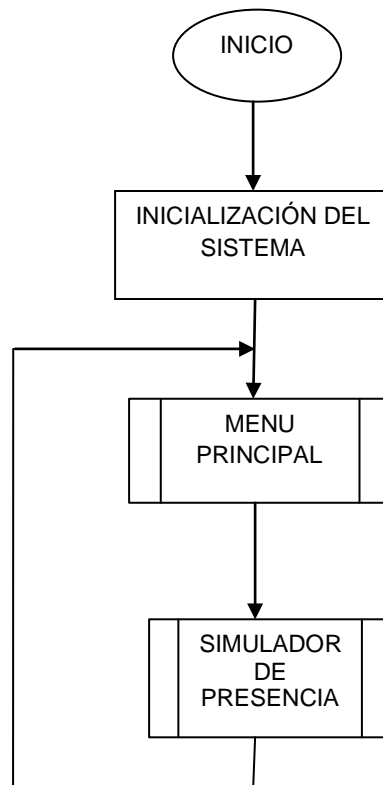


Ilustración 26. Diagrama de flujo del programa principal

```
//////////PROGRAMA PRINCIPAL//////////
void main()
{
char alarmas[14];
char alarma1[ ]="221415";    //hora fijada para que se encienda el led
char alarma2[ ]="221430";    //hora fijada para que se apague el led
char *leido=alarmas;

leer_clave_en_eeprom(d);
write_string_eeprom(0x04,alarma1); //escribo las horas fijadas en la eeprom
write_string_eeprom(0x10,alarma2);

set_tris_a(0b11100000);    //configuro el puerto a
set_tris_b(0b11110001);    //configuro el puerto b
set_tris_c(0b11111111);    //configuro el puerto c
enable_interrupts(INT_RB); //habilito la interrupción por cambio de nibble alto
```

```
ext_int_edge(L_TO_H);    //habilito la interrupción externa por cambio de flanco
enable_interrupts(INT_EXT); //habilito la interrupción externa
enable_interrupts(GLOBAL); //habilito las interrupciones

disable_interrupts(INT_EXT); //deshabilito la interrupción externa

lcd_init();             //inicializa lcd
delay_ms(1000);
lcd_putc (" Bienvenido ");
delay_ms(1000);

ds1307_init(DS1307_ALL_DISABLED);
    //Set date for -> 21-Abril-2010 Lunes
    //Set time for -> 22:14:00
ds1307_set_date_time(21,4,10,1,22,14,00);    //hora que le fijo al reloj

while(TRUE)
{
    lcd_putc("\f0-Activa Alarma\n1-Cambia Clave");
    delay_ms(1000);
    if(flag_teclado==1)    //Si se ha pulsado una tecla
    {
        flag_teclado=0;
        menu_principal();    //voy al menú principal
    }
    if(bandera==1)    //Si se ha producido una interrupción externa
    {
        bandera=0;
        alarma_general();    //activo la sirena
        regreso_menu();
    }
    else
    {
        delay_ms(100);
        ds1307_get_time(hr,min,sec);    //capturo la hora del reloj

        read_string_eeprom(0x04,leido);    //leo y comparo la primera hora fijada
        sprintf(ValoresLeidos,"%02u%02u%02u",hr,min,sec);
        if(strcmp(ValoresLeidos,leido)==0) //si la primera hora fijada coincide con el reloj
        {
            output_high(pin_A5);    //...enciendo el led
        }
        read_string_eeprom(0x10,leido);    //leo y comparo la segunda hora fijada
        sprintf(ValoresLeidos,"%02u%02u%02u",hr,min,sec);
        if(strcmp(ValoresLeidos,leido)==0) //si la segunda hora fijada coincide con la
del reloj...
        {
            output_low(PIN_A5);    //...apago el led
            set_tris_a(0b11100000);
        }
    }
}
}
```

Una vez inicializado el sistema iremos al menú principal. En la Ilustración 27, se detalla el diagrama de flujo del Menú Principal. En él se muestra un menú con dos opciones. Si se presiona la tecla '0' nos lleva a la parte del programa que activa la alarma y si presionamos la tecla '1' nos lleva a la rutina de cambio de la clave administrativa. En el supuesto de presionar cualquier otra tecla, el menú permanecerá inalterado.

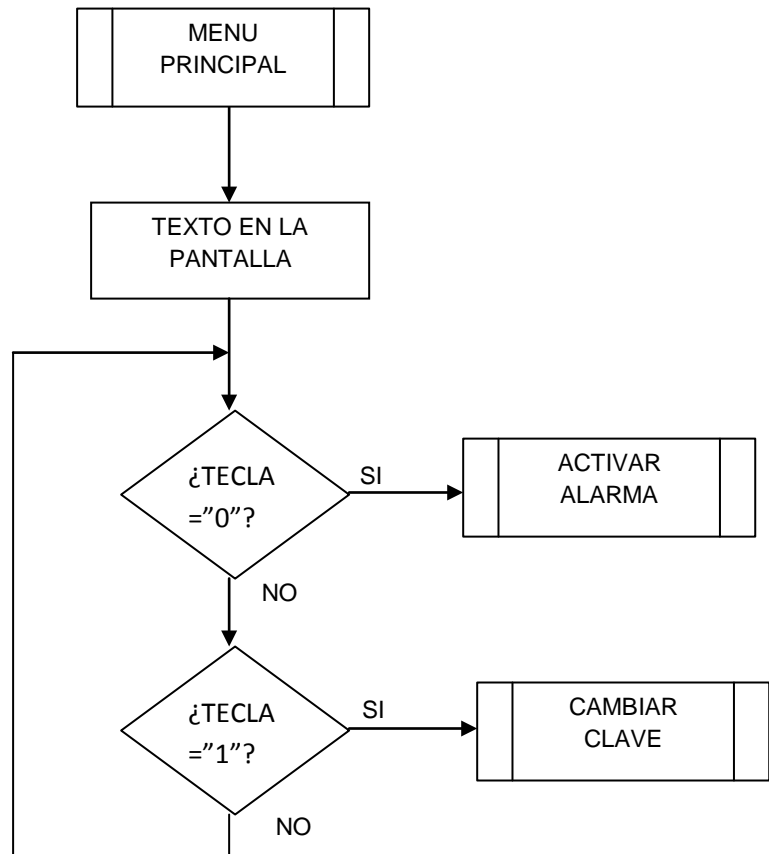


Ilustración 27. Diagrama de flujo del menú principal

```
////////FUNCION MENU PRINCIPAL
void menu_principal (void)      //función que muestra el menú principal
{
    d=0;
    switch (tecla)
    {
        case '0':      //si '0' va a activar la alarma
            activar_alarma();
            break;
        case '1':      //si '1' va a comprobar clave para después cambiar la clave
            cambiar_clave();
            break;
    }
}
```


Se recomienda lo primero cambiar la contraseña por defecto con la que el sistema de alarma viene programado, grabada en la memoria EEPROM del microcontrolador, la cual podemos cambiar por otra que sea más del gusto del usuario. Por lo tanto, en el Menú Principal se pulsaría la tecla '1' para entrar en la subrutina "Cambio de clave" cuyo funcionamiento nos explica el diagrama de flujo de la Ilustración 28. En esta subrutina se escribe la clave administrativa y se compara con la grabada en la memoria EEPROM del microcontrolador, que en este caso, la clave por defecto sería '1','2','3'. Si no son iguales, nos llevara el programa a la subrutina de "Conteo de intentos" y a los tres intentos fallidos saltará la alarma. Si son iguales, iremos a la subrutina de "Nueva clave administrativa" para realizar el cambio de clave.

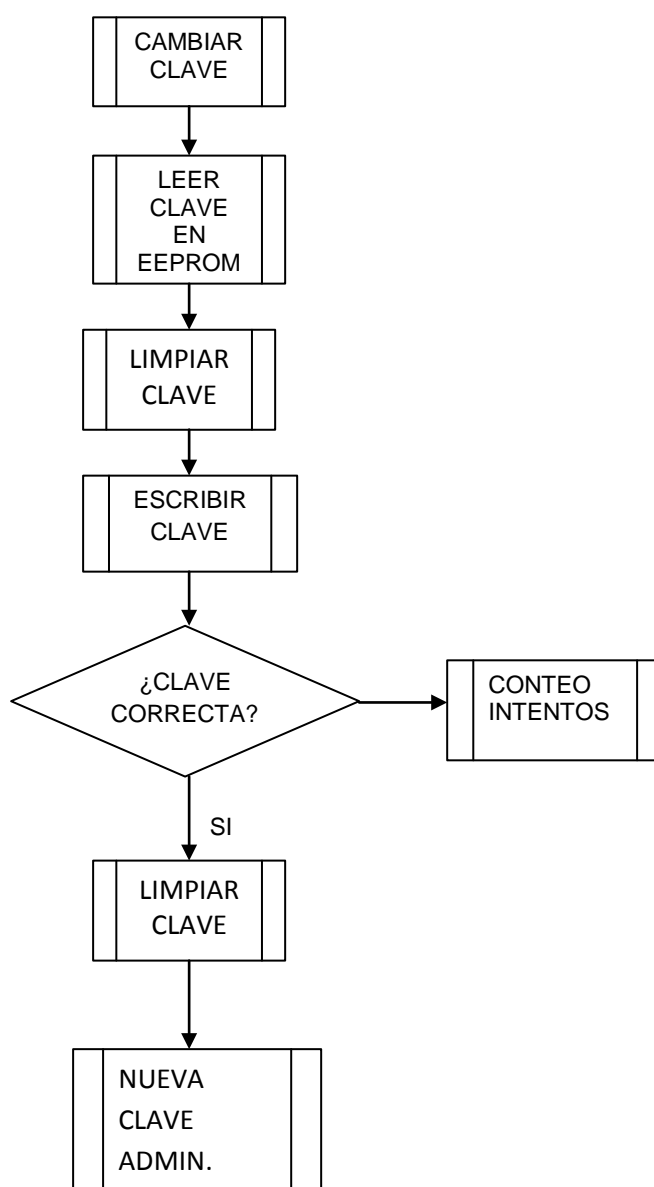


Ilustración 28. Diagrama de flujo de la rutina "Cambiar clave"

//////////FUNCION CAMBIAR CLAVE

```
void cambiar_clave(void)      //función que cambia clave administrativa
{
    int p;
    tecla=0;
    d=0;
    int aux=0;

    leer_clave_en_eeprom(d);
    limpiar_clave();
    lcd_putc("\fEscribir clave:\n");
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
        {
            flag_teclado=0;
            lcd_putc("*");
            clave[d]=tecla;
            d++;
            if (d==3)                //Si ya se han introducido todos los valores de la clave
            {
                p=comparar_claves(d);        //compara ambas claves
                if (p==0)                    //si son iguales...
                {
                    lcd_putc("\fClave Correcta");
                    delay_ms(2000);
                    alarm_cont = 0;
                    limpiar_clave();        //pongo a '0' la variable clave
                    nueva_clave_admin();    //introduzco la nueva clave
                }
                if (p!=0)                    //si son distintas...
                {
                    lcd_putc("\fClave Incorrecta");
                    delay_ms(2000);
                    limpiar_clave();        //pongo a '0' la variable clave
                    conteo_intentos();
                }
            }
        }
        else
        {
            tecla=0;
        }
    }
}
```

//////////FUNCION COMPARAR CLAVES

```
int comparar_claves(int) //función que compara la clave administrativa y la introducida
{
    d=0;
    int s=0;
    for (d=0;d<3;d++) //comparo carácter a carácter las claves
    {
        if (clave[d]==clave_eeprom_admin[d]) //si el carácter es igual en las dos claves
            s++;
    }
}
```

```
else //si es distinto la contraseña es incorrecta
{
    s=0;
    lcd_putc("\fClave Incorrecta");
    break;
}
}
if (s==3) //si todos los tres caracteres son iguales
    return 0; //devuelvo un '0'
}

/////////LEER_CLAVE_EN_EEPROM
void leer_clave_en_eeprom(int d) //función que lee las claves y las guarda en su
correspondiente variable
{
    for (d=0;d<3;d++)
    {
        clave_eeprom_admin[d] = read_eeprom(d); //guardamos cada valor de la
direccion i + 0 (para no pisar datos)
        delay_ms(20); //de la eeprom en la variable global
    }
}
```

Mediante la función limpiar clave ponemos a cero la clave tecleada una vez que hemos trabajado con ella para tenerla disponible para próximos usos. En la Ilustración 29 tenemos el diagrama de flujo que explica su funcionamiento.

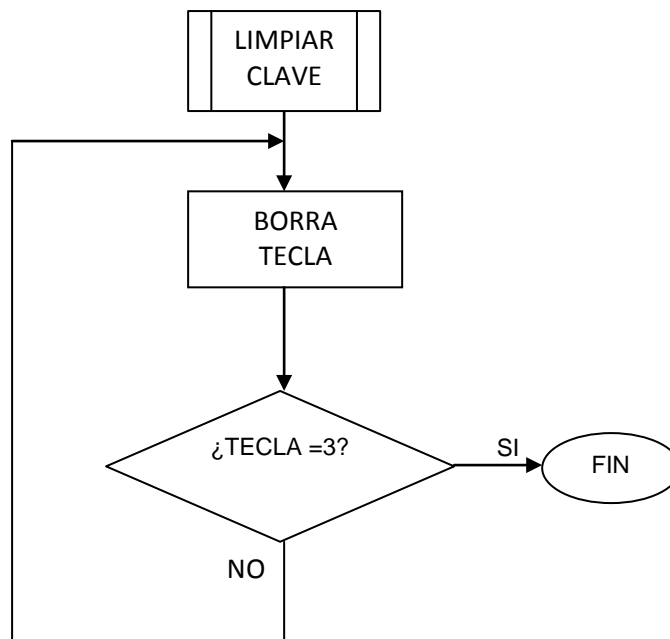


Ilustración 29. Diagrama de flujo de la rutina “Limpiar clave”

```
//////////LIMPIAR CLAVE
void limpiar_clave() //función que pone a '0' la variable clave
{
    int k;
    for(k=0;k<3;k++) //limpiamos uno a uno cada valor de la variable
    {
        clave[k]=0x00;
    }
}
```

A continuación, en la Ilustración 30, se muestra el diagrama de flujo de la rutina “Conteo intentos” anteriormente mencionada. Se dispone de 3 intentos para introducir la contraseña. Si los superamos se activará la alarma de la vivienda para alertar que ha habido una intrusión y el intruso está intentando desactivar el sistema de alarma.

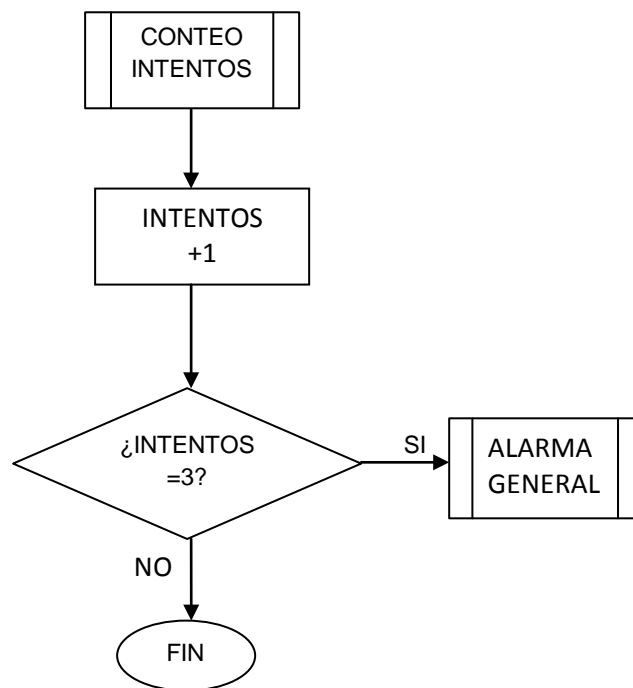


Ilustración 30. Diagrama de flujo de la rutina “Conteo intentos”

```
//////////FUNCION CONTEO INTENTOS
void conteo_intentos(void) //función que cuenta el numero de intentos de
ingreso de contraseña
{
    if (alarm_cont==2)
    {
```

```
alarm_cont = 0;           //pongo a cero el contador de errores
alarma_general();
}
else
{
    alarm_cont++;
    tecla=0;               //limpio las variables
    d=0;
}
}
```

Si se llegase a los tres intentos fallidos, se enciende la sirena y se muestra en la pantalla un mensaje de alarma. Una vez transcurridos 10 segundos, la sirena se apaga y desaparece el mensaje de alarma. No se deja todo el tiempo encendida la sirena hasta que alguien desactive el sistema ya que esto perturba a los vecinos debido al elevado sonido que la misma emite. En la Ilustración 31 se muestra el diagrama de flujo explicativo.

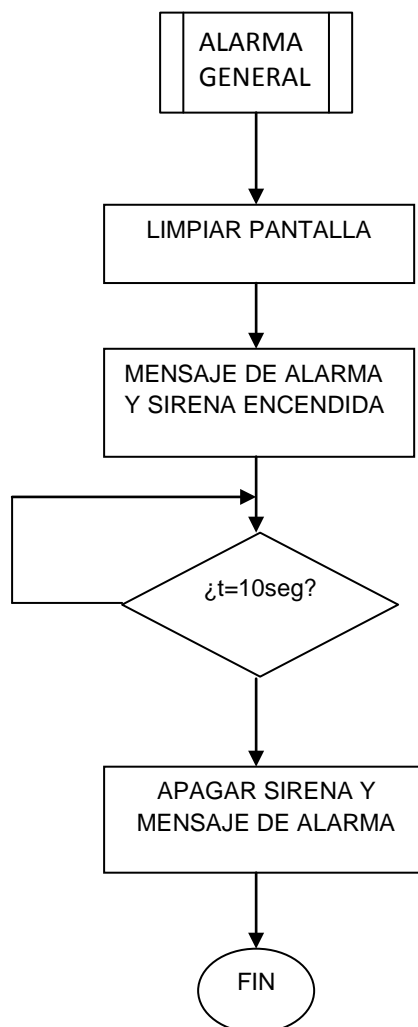


Ilustración 31. Diagrama de flujo de la rutina “Alarma general”

/////////FUNCION ALARMA GENERAL

```
void alarma_general (void)      //función que enciende la sirena
{
    set_tris_a(0b11110111);      //pin A3 como salida
    lcd_putc("\f" );              //borra pantalla
    delay_ms(100);
    output_high(pin_A3);          //enciende la sirena durante 10 seg y muestra mensaje
    lcd_putc("\f  ALARMA  ");
    delay_ms(10000);
    output_low(pin_A3);           //apaga la sirena
    alarm_cont = 0;
}
```

En la Ilustración 32, se muestra el diagrama de flujo de la rutina “Nueva clave administrativa” con la que activaremos y desactivaremos la alarma y realizaremos los futuros cambios de clave administrativa. En esta rutina escribimos la nueva contraseña y la vamos grabando número a número en la memoria EEPROM del microcontrolador.

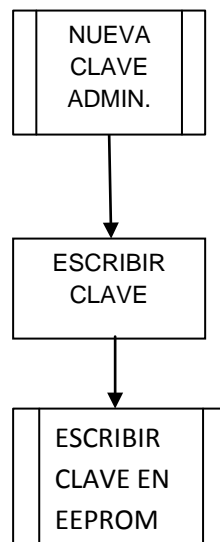


Ilustración 32. Diagrama de flujo de la rutina “Limpiar clave”

/////////FUNCION NUEVA CLAVE

```
void nueva_clave_admin(void)      //función que añade la nueva clave administrativa
{
    tecla=0;                        //limpio las variables
    int aux=0;
    d=0;

    lcd_putc("\fNueva clave:\n");  //presentación
```

```
delay_ms(500);
for(aux=0;aux<3;aux++)
{
    while(tecla==0 && bandera==0);
    {
        flag_teclado=0;
        lcd_putc("*");
        clave[d]=tecla;
        if(d<3)
        {
            escribir_clave_en_eeprom(d); //guardamos la nueva clave en la memoria
EEPROM
            tecla=0;
            d++;
        }
        else
        {
            d=0;
            limpiar_clave();
        }
    }
}

/////////ESCRIBIR_CLAVE_EN_EEPROM
void escribir_clave_en_eeprom(int d) //función que escribe las claves en la memoria
eeprom
{
    write_eeprom (d,tecla);           //guardamos cada valor de la variable global
correspondiente
    delay_ms(20);                     //en la dirección d + 0 (para no pisar datos) de la
eeprom
    d++;                             //en una palabra escribe de la posición 0 a la 2 (3 lugares)
}
```

Si en el Menú Principal se ha presionado la tecla '0', entraremos en la subrutina de "Activar la alarma" cuyo diagrama de flujo (Ilustración 33) tenemos a continuación. Su funcionamiento es muy similar a la de "Cambio de clave". Primero introducimos nuestra clave administrativa y la comprara con la grabada en la memoria EEPROM. Si son iguales iremos a la subrutina de "Desactivar alarma" y si son distintas a la de "Conteo de intentos" para controlar el numero de intentos realizados.

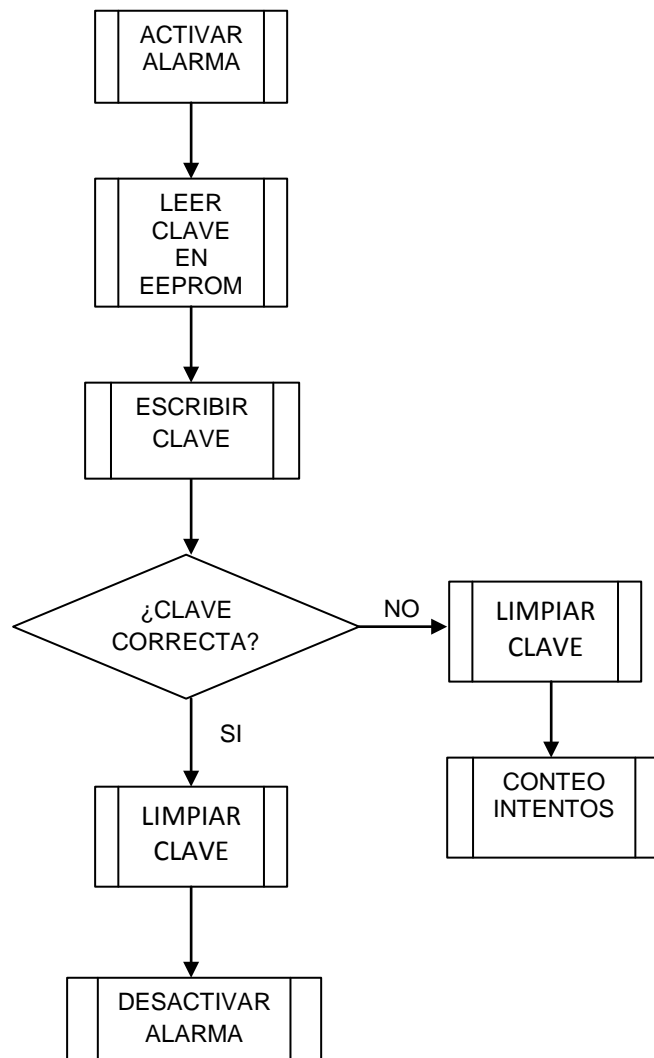


Ilustración 33. Diagrama de flujo de la rutina "Activar alarma"

```
/////////ACTIVAR ALARMA
void activar_alarma(void)      //función que activa la alarma
{
  int p;
  tecla=0;
  d=0;
  int aux=0;
  leer_clave_en_eeprom(d);
  lcd_putc("\fEscribir clave:\n");
  delay_ms(500);
  for(aux=0;aux<3;aux++)
  {
    while(tecla==0 && bandera==0);
    {
      lcd_putc("*");
      clave[d]=tecla;
      d++;
      if (d==3)                      //Si ya se han introducido todos los valores de la clave
```



```
{
p=comparar_claves(d);           //compara ambas claves
if (p==0)                       //si son iguales...
{lcd_putc("\fAlarma Activada");
delay_ms(2000);
alarm_cont = 0;
limpiar_clave();               //pongo a '0' la variable clave
desactivar_alarma();          //vamos a la rutina para desactivar la alarma
}
if (p!=0)                       //si son distintas...
{lcd_putc("\fClave Incorrecta");
delay_ms(2000);
limpiar_clave();               //pongo a '0' la variable clave
conteo_intentos();
}
}
else
{
tecla=0;
}
}
}
```

En la figura Ilustración 34, se muestra el diagrama de flujo de la subrutina “Desactivar alarma” cuyo funcionamiento es similar al de las dos anteriores. Introducimos la clave administrativa y la compara con la grabada en la memoria EEPROM. Si son iguales iremos de vuelta al Menú Principal y si son distintas a la subrutina “Conteo de intentos” para controlar el numero de intentos realizados. La diferencia radica en que mientras estemos en esta función la interrupción externa estará activada y por tanto los contactos magnéticos y los sensores de movimiento estarían en funcionamiento para detectar si algún intruso entra en la vivienda.

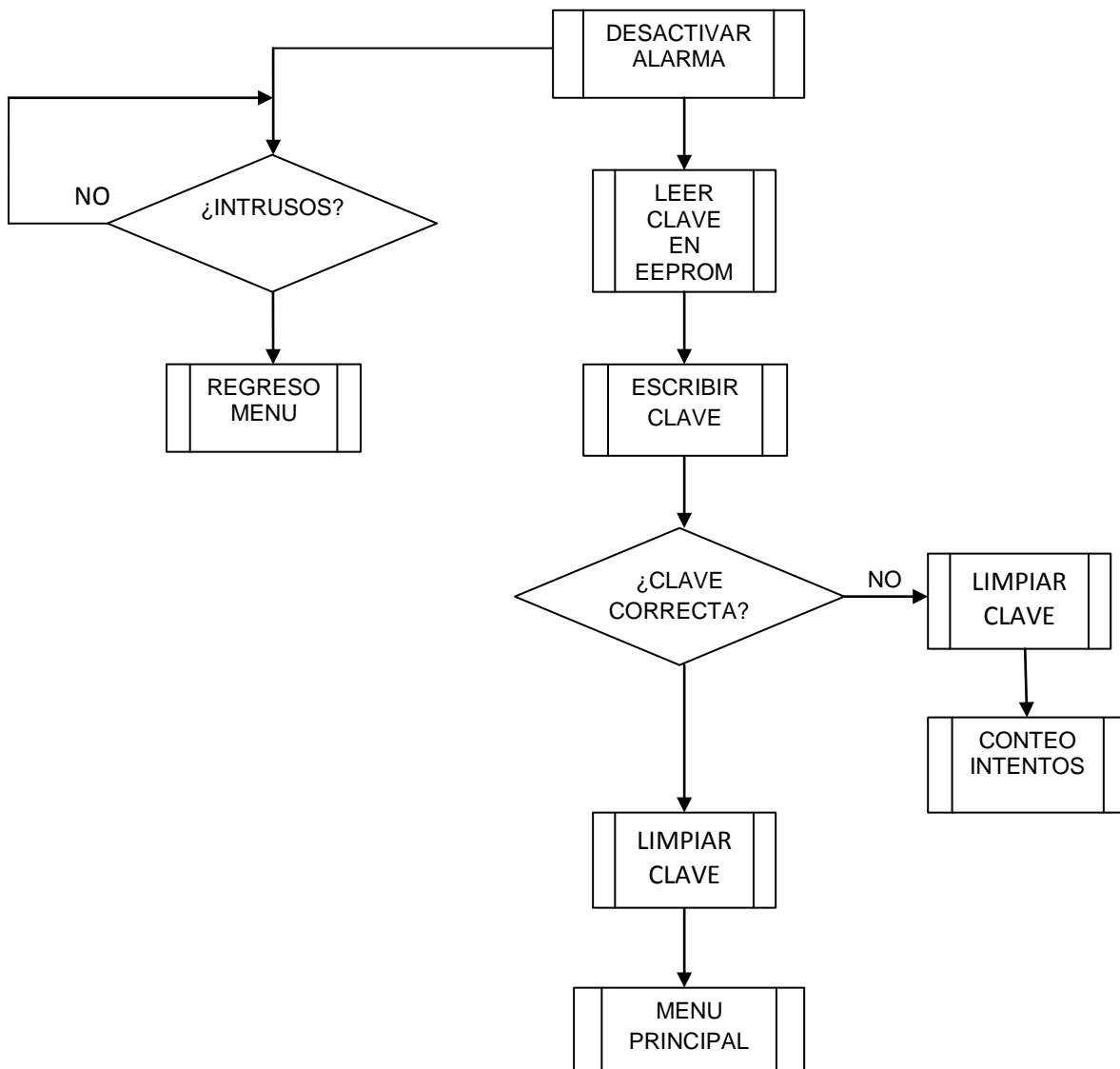


Ilustración 34. Diagrama de flujo de la rutina “Desactivar alarma”

```
/////////DESACTIVAR ALARMA
void desactivar_alarma(void)    //función que desactiva la alarma
{
    int p;
    tecla=0;
    d=0;
    int aux=0;
    enable_interrupts(INT_EXT);
    lcd_putc("\fEscribir clave:\n");
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
    }
}
```

```
lcd_putc("");
clave[d]=tecla;
d++;
if (d==3) //Si ya se han introducido todos los valores de la clave
{
    leer_clave_en_eeprom(d); //lee la contraseña que antes hemos cambiado
    p=comparar_claves(d); //compara ambas claves
    if (p==0) //si son iguales...
    {
        lcd_putc("\fAlarma\nDesactivada");
        delay_ms(2000);
        alarm_cont = 0;
        limpiar_clave(); //pongo a '0' la variable clave
        disable_interrupts(INT_EXT);
    }
    if (p!=0) //si son distintas...
    {
        lcd_putc("\fClave Incorrecta");
        delay_ms(2000);
        limpiar_clave(); //pongo a '0' la variable clave
        conteo_intentos();
    }
}
else
{
    tecla=0;
}
}
}
```

Si el sistema ha sido penetrado, Ilustración 35, lo que significa que se activó algún sensor de movimiento o contacto magnético estando el sistema activado, el programa lo detectaría mediante una interrupción. A continuación se iría a una función para comprobar la clave que teníamos y si escribimos la correcta cambiamos dicha clave ya que si no se hiciera esto cualquiera podría modificar la clave y entrar en la vivienda. En caso de que no se tecleara la correcta se volvería a encender la alarma.

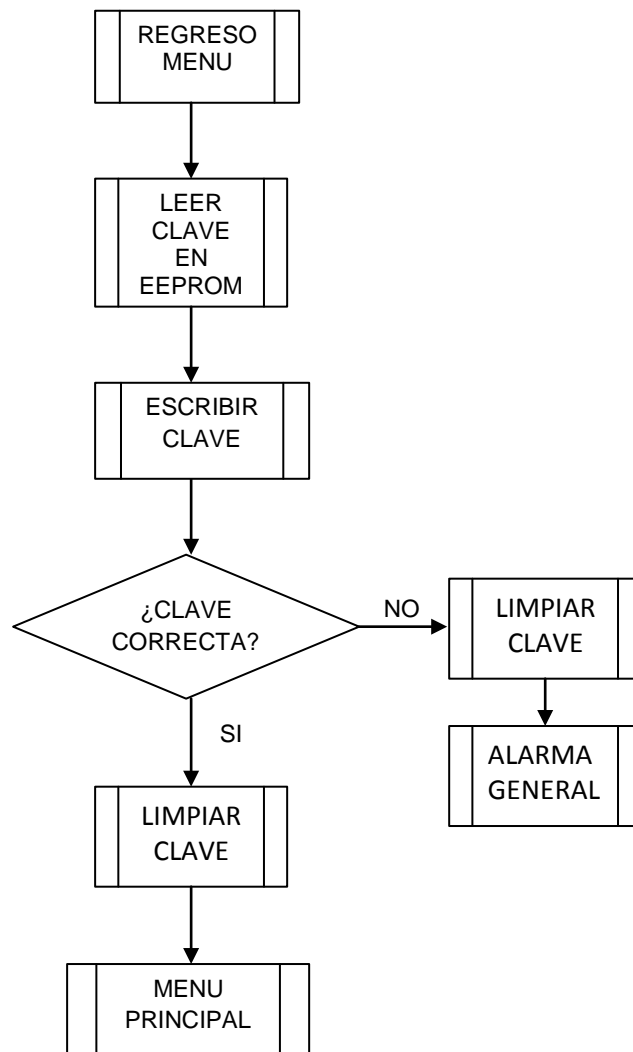


Ilustración 35. Diagrama de flujo de la rutina “Regreso menú”

```
//////////FUNCION REGRESO MENU
void regreso_menu(void)
{
int p;
tecla=0;
d=0;
int aux=0;

limpiar_clave();
lcd_putc("\fEscribir clave:\n");
delay_ms(500);
for(aux=0;aux<3;aux++)
{
while(tecla==0 && bandera==0);
{
flag_teclado=0;
lcd_putc("*");
clave[d]=tecla;
d++;
}
```

```
if (d==3)                //Si ya se han introducido todos los valores de la clave
{
    p=comparar_claves(d);    //compara ambas claves
    if (p==0)                //si son iguales...
    {
        lcd_putc("\fClave Correcta");
        delay_ms(2000);
        limpiar_clave();    //pongo a '0' la variable clave
        disable_interrupts(INT_EXT);
    }
    if (p!=0)                //si son distintas...
    {
        lcd_putc("\fClave Incorrecta");
        delay_ms(2000);
        limpiar_clave();    //pongo a '0' la variable clave
        alarma_general();
    }
}
else
{
    tecla=0;
}
}
```

3.4.2 Hardware de programación

Para la programación del PIC16F876, se ha utilizado la placa de desarrollo EasyPic v7, de MicroElektronika, la cual se conecta serialmente al ordenador, y mediante su software permite la fácil selección del programa a grabar y en que dispositivo.

Dicha placa de desarrollo también nos permite realizar pruebas sobre ella ya que dispone de una serie de recursos y dispositivos como por ejemplo: zócalos para PIC's de 40, 28, 20, 18, 14 y 8 pines; diodos LED y pulsadores conectados a cada uno de los pines de los puertos; comunicación UART vía RS-232 y USB; zócalos para la conexión de diversos accesorios; posibilidad de conectar display LCD de 16x2 caracteres o display GLCD de 128x64 caracteres; también dispone de un display de 4 dígitos de 7 segmentos; sensor digital de temperatura DS1820; sensor de temperatura analógico LM35; memoria EEPROM externa 24C08 con conexión I2C; zumbador piezoeléctrico y tres pines de tierra adicionales en la placa que permiten conectar fácilmente la masa del osciloscopio.

La alimentación de la placa de desarrollo puede ser a través del cable USB que viene con la placa y que conectaríamos al ordenador, mediante un adaptador de tensión o mediante una fuente de alimentación.

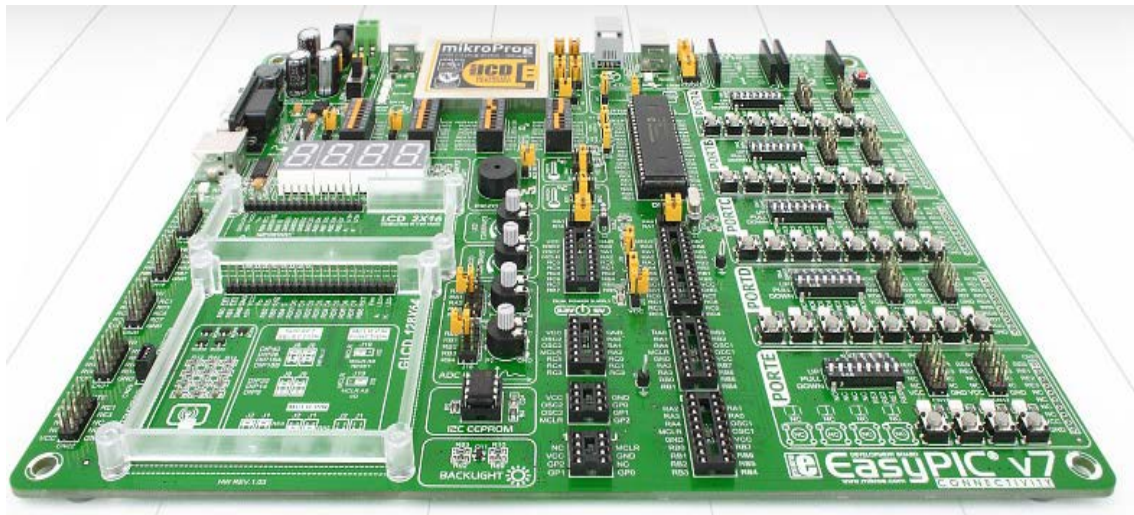


Ilustración 36. Placa de desarrollo EasyPic v7

CAPITULO IV

PRUEBAS AL SISTEMA Y RESULTADOS

4.1 PRUEBAS AL SISTEMA

Una vez finalizada la instalación de elementos (sensores de movimiento, contactos magnéticos, sensor de rotura de cristales, sirena,...) del prototipo de control domótico, se procede a realizar las pruebas necesarias para comprobar el funcionamiento correcto del mismo.

Panel de control:

Para la verificación del panel de control del sistema se tiene que tomar en cuenta que el óptimo funcionamiento de éste depende de la conexión de todos y cada uno de los componentes del sistema. Por este motivo, las pruebas básicas que se hicieron al sistema son:

- Comprobación del correcto funcionamiento del teclado y LCD.
- Realizar pruebas de comunicación entre el panel de control y la central del sistema:
 - Activación / desactivación del sistema.
 - Cambios de clave.
 - Funcionamiento del simulador de presencia.
 - Funcionamiento de los avisadores de alarma.

Sensores:

- Verificar la activación / desactivación de sensores tanto de movimiento como de puertas y ventanas.

4.2 RESULTADOS

Los resultados obtenidos en las pruebas realizadas se detallan a continuación:

Panel de control:

- Comprobación del correcto funcionamiento del teclado y LCD.
- Realizar programa de prueba para la verificación en pantalla del LCD si los datos son transmitidos correctamente.



Ilustración 37. Verificación funcionamiento del display

- Verificar el funcionamiento del teclado.



Ilustración 38. Verificación funcionamiento del teclado

- Activación / desactivación del sistema.



Ilustración 39. Activación de la alarma



Ilustración 40. Desactivación de la alarma

- Cambios de clave.

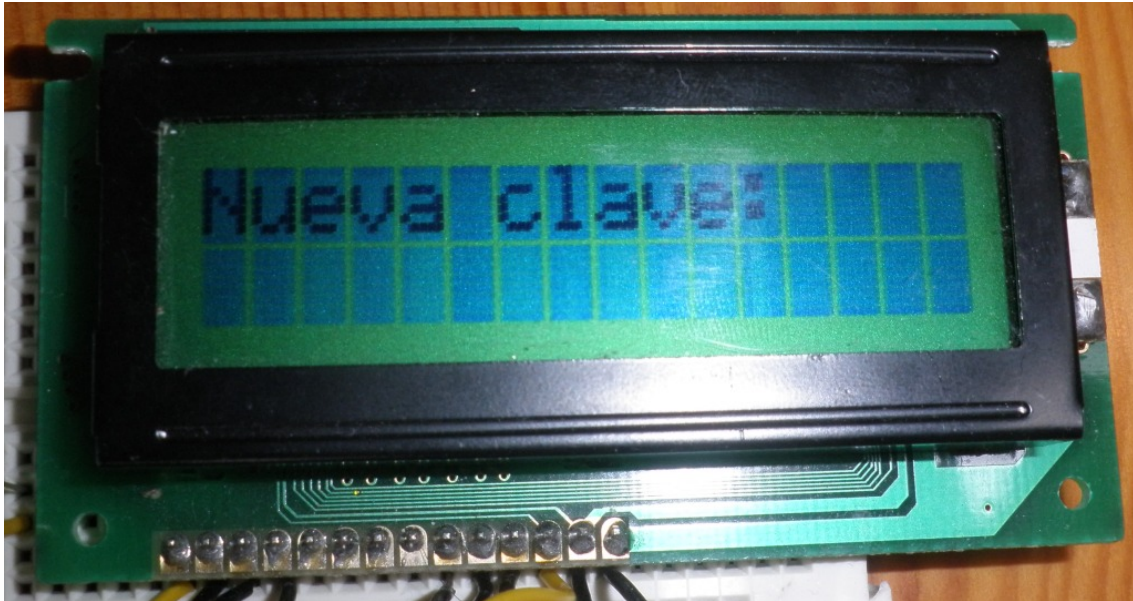


Ilustración 41. Cambio de clave

- Alarma.

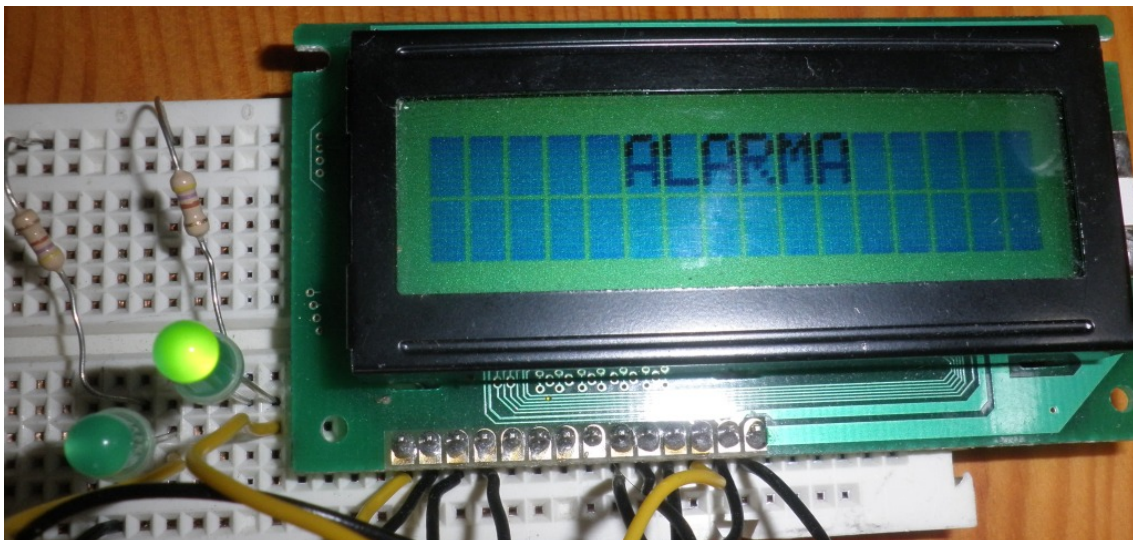


Ilustración 42. Activación de la sirena. (Simulación mediante un LED)

- Simulador de presencia.



Ilustración 43. Puesta en marcha del simulador de presencia

Sensores:

- Verificar la activación / desactivación de sensores tanto de movimiento como de puertas y ventanas.

En este caso, ante la imposibilidad de compra de los sensores, la prueba sería la simulación de ellos mediante un interruptor. En la Ilustración 44 se muestra como saltaría la alarma al activar el interruptor.

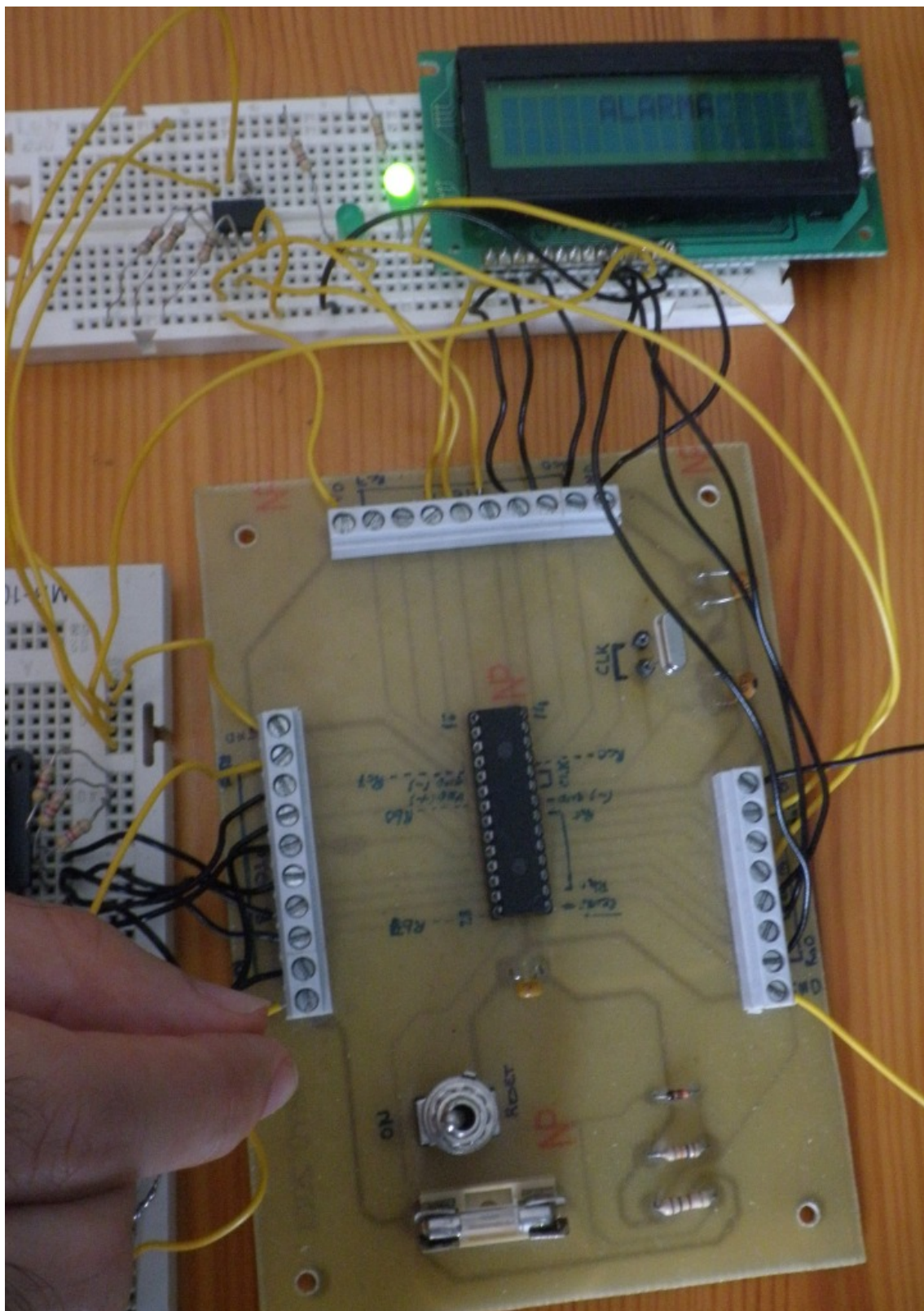


Ilustración 44. Activación de los sensores

CAPITULO V

CONCLUSIONES, RECOMENDACIONES Y FUTUROS DESARROLLOS

5.1 CONCLUSIONES

- La utilización de un LCD en el panel de control es de mayor comodidad y facilidad para el usuario, ya que los tradicionales paneles de control únicamente muestran mediante diodos LED la configuración del sistema y ésto hace que sea confuso para el usuario.
- El uso de SMS hace que el sistema sea de confort para el usuario, ya que tendrá conocimiento de que se ha activado la alarma sin importar el lugar en el que se encuentre el propietario de la vivienda.
- La implementación del reloj en tiempo real como simulador de presencia es de gran utilidad porque previene la intrusión de ladrones evitando daños y molestias en la vivienda si se da el caso de que intentaran entrar en ella

5.2 RECOMENDACIONES

- Realizar un sistema distribuido en lugar de un sistema centralizado, esto reduce el cableado, ya que en un sistema distribuido no se necesita que todos los componentes del sistema estén cableados hacia la central. Además en un sistema distribuido no existe un punto único de fallo, lo que hace un sistema más confiable y rápido.
- No olvidar de poner un reinicio en el sistema y en cada placa, para evitar el tener que desconectar el sistema completo al momento de una fallo, prueba, o cambio de componentes en el mismo.
- Si se requiere que los sensores de puertas y ventanas no se vean, se puede utilizar sensores empotrados, los cuales se insertan en el interior del marco de la puerta o ventana, lo que hace que los mismos no se observen.
- Es importante familiarizar al usuario con el sistema, para que no tenga inconvenientes con el uso del mismo, y así evitar posibles daños que se puedan generar por el mal manejo del sistema.

5.3 FUTUROS DESARROLLOS

- Posibilidad de modificar mediante el panel de control la hora de activación y desactivación del simulador de presencia o establecer un sistema para que se fijen dichas horas de manera aleatoria.
- Dotar al sistema de sensores de otro tipo, por ejemplo: humedad, humo, incendio,... para completar y mejorar el sistema de alarma y de esta manera abarcar mas campos.
- Profundizar en el desarrollo del programa de envío de SMS por parte del microcontrolador y del módem
- Envío de un SMS en caso de fallo de la red eléctrica o de ruptura del cableado por parte de intrusos.
- Poder activar y desactivar el sistema de alarma mediante el envío de un SMS o que el SMS de alarma indique también la causa de la misma: robo, incendio, inundación,...
- Dado que el modem que elegiría es capaz de realizar y recibir llamadas, se podría completar el sistema con un demoportero que esté conectado con el portero automático de la vivienda y de ésta forma, en el caso de que alguien timbrara, simular la presencia de personas en la vivienda.
- Utilizar un teléfono móvil como modem en lugar de un modem GSM para realizar la comunicación entre el sistema de alarma y el usuario.

CAPITULO VI

DEFINICIONES Y ABREVIATURAS

6.1 DEFINICIONES

- **Array**

En programación, una matriz o vector (llamados en inglés arrays) es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo, los elementos de la matriz. Desde el punto de vista lógico, una matriz se puede ver como un conjunto de elementos ordenados en fila (o filas y columnas si tuviera dos dimensiones).

En principio, se puede considerar que todas las matrices son de una dimensión, la dimensión principal, pero los elementos de dicha fila pueden ser a su vez matrices (un proceso que puede ser recursivo), lo que nos permite hablar de la existencia de matrices multidimensionales, aunque las más fáciles de imaginar son los de una, dos y tres dimensiones.

Estas estructuras de datos son adecuadas para situaciones en las que el acceso a los datos se realice de forma aleatoria e impredecible. Por el contrario, si los elementos pueden estar ordenados y se va a utilizar acceso secuencial sería más adecuado utilizar una lista, ya que esta estructura puede cambiar de tamaño fácilmente durante la ejecución de un programa.

- **ASCII**

ASCII (acrónimo inglés de **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange — *Código Estándar Estadounidense para el Intercambio de Información*), pronunciado generalmente [áski] o [ásci], es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales. Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía. Más tarde, en 1967, se incluyeron las minúsculas, y se redefinieron algunos códigos de control para formar el código conocido como **US-ASCII**.

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión. A menudo se llama incorrectamente ASCII a otros códigos de caracteres de 8 bits, como el estándar ISO-8859-1 que es una extensión que utiliza 8 bits para proporcionar caracteres adicionales usados en idiomas distintos al inglés, como el español.

ASCII fue publicado como estándar por primera vez en 1967 y fue actualizado por última vez en 1986. En la actualidad define códigos para 32

caracteres no imprimibles, de los cuales la mayoría son caracteres de control obsoletos que tienen efecto sobre cómo se procesa el texto, más otros 95 caracteres imprimibles que les siguen en la numeración (empezando por el carácter espacio).

Casi todos los sistemas informáticos actuales utilizan el código ASCII o una extensión compatible para representar textos y para el control de dispositivos que manejan texto como el teclado. No deben confundirse los códigos ALT+número de teclado con los códigos ASCII.

- **BCD**

En sistemas de computación, Binary-Coded Decimal (BCD) o Decimal Codificado en binario es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits. Con esta codificación especial de los dígitos decimales en el sistema binario, se pueden realizar operaciones aritméticas como suma, resta, multiplicación y división de números en representación decimal, sin perder en los cálculos la precisión ni tener las inexactitudes en que normalmente se incurre con las conversiones de decimal a binario puro y de binario puro a decimal. La conversión de los números decimales a BCD y viceversa es muy sencilla pero los cálculos en BCD se llevan más tiempo y son algo más complicados que con números binarios puros.

- **Bit**

Es el acrónimo de **B**inary **d**igit (dígito binario). Un bit es un dígito del sistema de numeración binario. Mientras que en el sistema de numeración decimal se usan diez dígitos, en el binario se usan sólo dos dígitos, el 0 y el 1. Un bit o dígito puede representar uno de esos dos valores, 0 o 1.

El bit es la unidad mínima de información empleada en informática, en cualquier dispositivo digital, o en la teoría de la información. Con él, podemos representar dos valores cuales quiera, como verdadero o falso, abierto o cerrado, blanco o negro, norte o sur, masculino o femenino, rojo o azul, etc. Basta con asignar uno de esos valores al estado de “apagado” (0) y otro al estado de “encendido” (1).

- **Byte**

Es una secuencia de bits contiguos, cuyo tamaño depende del código de información o código de caracteres en que sea definido.

Se usa comúnmente como unidad básica de almacenamiento de datos en combinación con los prefijos de cantidad. Originalmente el byte fue elegido para ser un submúltiplo del tamaño de la palabra de un ordenador, desde cinco a doce bits. Aunque la popularidad de las microcomputadoras y los microprocesadores de 8 bits en los años 1980 ha hecho obsoleta la utilización de otra cantidad que no sean 8 bits. El término octeto se utiliza ampliamente como un sinónimo preciso donde la ambigüedad es indeseable (por ejemplo, en definiciones de protocolos).

La unidad byte no tiene símbolo establecido internacionalmente, aunque en países anglosajones es frecuente la “B” mayúscula, mientras que en los francófonos es la “o” minúscula (de octet); la ISO y la IEC recomiendan restringir el empleo de esta unidad a los octetos (bytes de 8 bits).

- **Comandos AT**

Instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem/GSM.

- **Corriente alterna AC**

Se denomina corriente alterna (abreviada CA en español y AC en inglés, de *alternating current*) a la corriente eléctrica en la que la magnitud y el sentido varían cíclicamente. La forma de oscilación de la corriente alterna más comúnmente utilizada es la de una oscilación sinusoidal, puesto que se consigue una transmisión más eficiente de la energía. Sin embargo, en ciertas aplicaciones se utilizan otras formas de oscilación periódicas, tales como la triangular o la cuadrada.

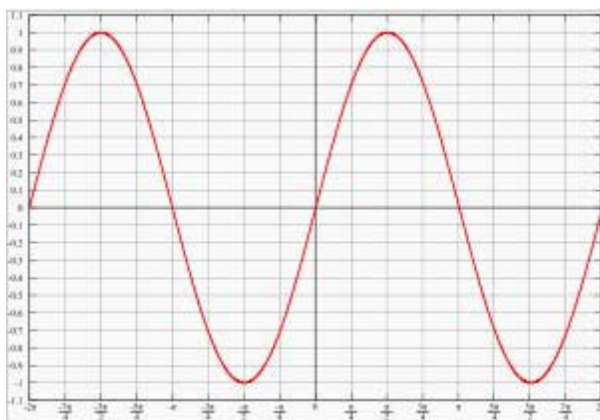


Ilustración 45. Corriente alterna AC

Utilizada genéricamente, la CA se refiere a la forma en la cual la electricidad llega a los hogares y a las empresas. Sin embargo, las señales de audio y de radio transmitidas por los cables eléctricos, son también ejemplos de corriente alterna. En estos usos, el fin más importante suele ser la transmisión y recuperación de la información codificada (o modulada) sobre la señal de la CA.

- **Corriente continua DC**

La corriente continua o corriente directa (CC en español, en inglés DC, de *Direct Current*) es el flujo continuo de electrones a través de un conductor entre dos puntos de distinto potencial. A diferencia de la corriente alterna (CA en español, AC en inglés), en la corriente continua las cargas eléctricas circulan siempre en la misma dirección (es decir, los terminales de mayor y de menor potencial son siempre los mismos). Aunque comúnmente se identifica la corriente continua con la corriente constante (por ejemplo la suministrada por una batería), es corriente continua toda corriente que mantenga siempre la

misma polaridad. También se dice corriente continua cuando los electrones se mueven siempre en el mismo sentido, el flujo se denomina corriente continua y va (por convenio) del polo positivo al negativo.

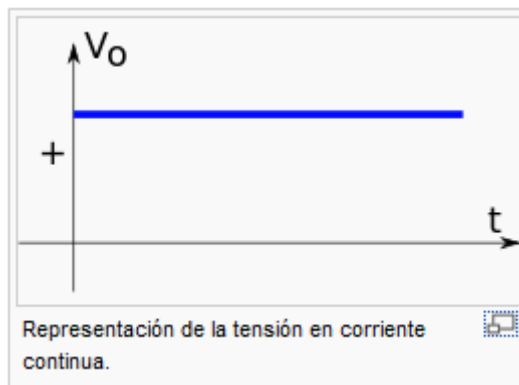


Ilustración 46. Corriente continua DC

- **Datasheet**

Un datasheet es un documento que resume el funcionamiento y otras características de un componente (por ejemplo, un componente electrónico) o subsistema (por ejemplo, una fuente de alimentación) con el suficiente detalle para ser utilizado por un ingeniero de diseño y diseñar el componente en un sistema.

Comienza típicamente con una página introductoria que describe el resto del documento, seguido por los listados de componentes específicos, con la información adicional sobre la conectividad de los dispositivos. En caso de que haya código fuente relevante a incluir se une cerca del final del documento o se separa generalmente en otro archivo.

La información típica que suele aparecer es:

- Datos del fabricante
- Número y denominación
- Lista de formatos con imágenes y códigos
- Propiedades
- Breve descripción funcional
- Esquema de conexiones. Habitualmente es un anexo con indicaciones detalladas.
- Tensión de alimentación, consumo.
- Condiciones de operación recomendadas
- Tabla de especificaciones, tanto en corriente continua como alterna
- Esquema de la onda de entrada-salida
- Medidas
- Circuito de prueba
- Información sobre normas de seguridad y uso.

- **Display**

Se llama visualizador, display en inglés, a un dispositivo de ciertos aparatos electrónicos que permite mostrar la información al usuario de manera visual. Un visualizador de una señal de vídeo se lo llama más comúnmente pantalla; los dos ejemplos más comunes son el televisor y la pantalla de ordenador. Un visualizador es un tipo de dispositivo de salida.

- **EEPROM**

EEPROM o E²PROM son las siglas de *Electrically Erasable Programmable Read-Only Memory* (ROM programable y borrada eléctricamente). Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioleta. Son memorias no volátiles.

Las celdas de memoria de una EEPROM están constituidas por un transistor MOS, que tiene una compuerta flotante (estructura SAMOS), su estado normal está cortado y la salida proporciona un 1 lógico.

Aunque una EEPROM puede ser leída un número ilimitado de veces, sólo puede ser borrada y reprogramada entre 100.000 y un millón de veces.

Estos dispositivos suelen comunicarse mediante protocolos como I²C, SPI y Microwire. En otras ocasiones, se integra dentro de chips como microcontroladores y DSPs para lograr una mayor rapidez.

- **GSM**

Sistema global para las comunicaciones móviles. Proviene del francés (Groupe Spécial Mobile). Es un sistema digital de telefonía móvil que provee un estándar común para los usuarios, permitiendo el roaming internacional y la capacidad de ofrecer a alta velocidad servicios avanzados de transmisión de voz, datos y vídeo.

- **Hardware**

Término del inglés que se utiliza generalmente para describir los artefactos físicos de una tecnología. El hardware puede ser equipo militar importante, equipo electrónico, o equipo informático. En electrónica se denomina hardware o soporte físico al conjunto de elementos materiales que componen una computadora. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

- **Interrupción**

Las interrupciones son esencialmente llamadas a subrutinas generadas por los dispositivos físicos, al contrario de las subrutinas normales de un programa en ejecución. Como el salto de subrutina no es parte del hilo o secuencia de ejecución programada, el controlador guarda el estado del procesador en la pila de memoria y entra a ejecutar un código especial llamado "manejador de interrupciones" que atiende al periférico específico que generó

la interrupción. Al terminar la rutina, una instrucción especial le indica al procesador el fin de la atención de la interrupción. En ese momento el controlador restablece el estado anterior, y el programa que se estaba ejecutando antes de la interrupción sigue como si nada hubiese pasado. Las rutinas de atención de interrupciones deben ser lo más breves posibles para que el rendimiento del sistema sea satisfactorio, porque normalmente cuando una interrupción es atendida, todas las demás interrupciones están en espera.

- **I2C**

Es un bus de comunicaciones en serie. Su nombre viene de Inter-Integrated Circuit (Circuitos Inter-Integrados). La versión 1.0 data del año 1992 y la versión 2.1 del año 2000, su diseñador es Philips. La velocidad es de 100Kbits por segundo en el modo estándar, aunque también permite velocidades de 3,4Mbit/s. Es un bus muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados (Embedded Systems) y generalizando más para comunicar circuitos integrados entre sí que normalmente residen en un mismo circuito impreso.

La principal característica de I2C es que utiliza dos líneas para transmitir información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria. Las líneas se llaman:

1. SDA: datos
2. SCL: reloj
3. GND: tierra

Las dos primeras líneas son drenaje abierto, por lo que necesitan resistencias de pull-up. Los dispositivos conectados al bus I2C tienen una dirección única para cada uno. También pueden ser maestros o esclavos. El dispositivo maestro inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el maestro sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que el bus I2C se le denomine bus multimaestro.

- **LED**

Un LED (Light-Emitting Diode) es un diodo semiconductor que emite luz. Se usan como indicadores en muchos dispositivos, y cada vez con mucha más frecuencia, en iluminación. Presentado como un componente electrónico en 1962, los primeros LEDs emitían luz roja de baja intensidad, pero los dispositivos actuales emiten luz de alto brillo en el espectro infrarrojo, visible y ultravioleta.

Cuando un LED se encuentra en polarización directa, los electrones pueden recombinarse con los huecos en el dispositivo, liberando energía en forma de fotones. Este efecto es llamado electroluminiscencia y el color de la luz (correspondiente a la energía del fotón) se determina a partir de la banda de energía del semiconductor. Por lo general, el área de un LED es muy pequeña

(menor a 1 mm²), y se pueden usar componentes ópticos integrados para formar su patrón de radiación. Los LEDs presentan muchas ventajas sobre las fuentes de luz incandescente y fluorescente, principalmente con un consumo de energía mucho menor, mayor tiempo de vida, tamaño más pequeño, gran durabilidad, resistencia a las vibraciones, no es frágil, reduce considerablemente la emisión de calor que produce el efecto invernadero en nuestro planeta, no contienen mercurio (el cual al exponerse en el medio ambiente es altamente venenoso) a comparación de la tecnología fluorescente o de inducción magnética que sí contienen mercurio, no crean campos magnéticos altos como la tecnología de inducción magnética con los cuales se crea mayor radiación hacia el ser humano, reducen ruidos en las líneas eléctricas, son especiales para utilizarse con sistemas fotovoltaicos (paneles solares) a comparación de cualquier otra tecnología actual, no les afecta el encendido intermitente (es decir pueden funcionar como luces estroboscópicas) y esto no reduce su vida promedio, son especiales para sistemas anti-exploración ya que no es fácil quebrar un diodo emisor de luz y cuentan con una alta fiabilidad. Los LEDs con la potencia suficiente para la iluminación de interiores son relativamente caros y requieren una corriente eléctrica más precisa, por su sistema electrónico para funcionar con voltaje alterno y requieren de disipadores de calor cada vez más eficientes a comparación de las bombillas fluorescentes de potencia equiparable.

- **Masa/tierra**

La toma de tierra, también denominada hilo de tierra, toma de conexión a tierra, puesta a tierra, pozo a tierra, polo a tierra, conexión a tierra, conexión de puesta a tierra, tierra o simplemente masa, se emplea en las instalaciones eléctricas/electrónicas para evitar el paso de la corriente al usuario por un fallo del aislamiento de los conductores activos.

La puesta a tierra es una unión de todos los elementos metálicos que, mediante cables de sección suficiente entre las partes de una instalación y un conjunto de electrodos, permite la desviación de corrientes de falta o de las descargas de tipo atmosférico y consigue que no se pueda dar una diferencia de potencial peligrosa en los edificios, instalaciones y superficie próxima al terreno.

- **Microcontrolador**

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado. Son diseñados para disminuir el coste económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la CPU, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación.

Un microcontrolador difiere de una CPU normal, debido a que es más fácil convertirla en una computadora en funcionamiento, con un mínimo de chips externos de apoyo. La idea es que el chip se coloque en el dispositivo, enganchado a la fuente de energía y de información que necesite, y eso es todo. Un microprocesador tradicional no le permitirá hacer esto, ya que espera

que todas estas tareas sean manejadas por otros chips. Hay que agregarle los módulos de entrada/salida (puertos) y la memoria para almacenamiento de información.

Por ejemplo, un microcontrolador típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria RAM y ROM/EPROM/EEPROM/FLASH, significando que para hacerlo funcionar, todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidores de analógico a digital, temporizadores, UARTs y buses de interfaz serie especializados, como I2C y CAN. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados.

- **Pin/terminal/patilla**

En electrónica se denomina pin, palabra inglesa que significa clavija, terminal o patilla a cada uno de los contactos metálicos de un conector o de un componente fabricado de un material conductor de la electricidad. Estos se utilizan para conectar componentes sin necesidad de soldar nada, de esta manera se logra transferir electricidad e información. Para determinar la misión de cada uno de los pines de un dispositivo se deberán consultar sus respectivas hojas de datos o datasheet.

- **Precisión**

Es la capacidad del sensor o instrumento para dar un mismo valor al medir varias veces. A corto plazo se habla de *repetitividad* (dar el mismo valor al tomar medidas consecutivas), y a largo plazo de *estabilidad* (dar el mismo valor al tomar medidas en largos espacios de tiempo); la estabilidad se relaciona con las derivas.

- **Pull-Down**

Al contrario que pull-up, lo que se pretende con una resistencia de pull-down es polarizar una señal a un nivel de tensión bajo, normalmente masa, la referencia del circuito.

- **Pull-Up**

En electrónica se denomina pull-up a la acción de elevar la tensión de salida de un circuito lógico a la tensión que, por lo general mediante un divisor de tensión se pone a la entrada de un amplificador con el fin de desplazar su punto de trabajo.

- **Reloj en tiempo real**

Un reloj en tiempo real (en inglés, real-time clock, RTC), es un reloj de un ordenador, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en ordenadores personales, servidores y sistemas embebidos, los RTC's están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto. Su uso tiene beneficios como: bajo consumo de energía (importante cuando

está funcionando con una pila), libera de trabajo al sistema principal para que pueda dedicarse a tareas más críticas y algunas veces es más preciso que otros métodos.

Los RTC's a menudo tienen una fuente de alimentación alternativa, por lo que pueden seguir midiendo el tiempo mientras la fuente de alimentación principal está apagada o no está disponible.

La mayoría de los RTC's usan un oscilador de cristal, pero algunos usan la frecuencia de la fuente de alimentación. En muchos casos la frecuencia del oscilador es 32.768 kHz. Ésta es la misma frecuencia usada en los relojes de cuarzo, y por las mismas razones, que la frecuencia es exactamente 2^{15} ciclos por segundo, que es un ratio muy práctico para usar con circuitos de contadores binarios simples.

- **Resistencia**

La resistencia eléctrica de un objeto es una medida de su oposición al paso de corriente. La unidad de la resistencia en el sistema Internacional de Unidades es el ohmio (Ω).

Para su medición en la práctica existen diversos métodos, entre los que se encuentra el uso de un ohmímetro. Además, su cantidad recíproca es la conductancia, medida en Siemens.

La resistencia de cualquier objeto depende únicamente de su geometría y de su resistividad, por geometría se entiende a la longitud, el área del objeto mientras que la resistividad es un parámetro que depende del material del objeto y de la temperatura a la cual se encuentra sometido. Esto significa que, dada una temperatura y un material, la resistencia es un valor que se mantendrá constante. Además, de acuerdo con la ley de Ohm, la resistencia de un material puede definirse como la razón entre la caída de tensión y la corriente en dicha resistencia, así: $R=V/I$. Donde R es la resistencia en ohmios, V es la diferencia de potencial medida en voltios e I es la intensidad de corriente en amperios.

Según sea la magnitud de esta medida, los materiales se pueden clasificar en conductores, aislantes y semiconductor. Existen además ciertos materiales en los que en determinadas condiciones de temperatura aparece un fenómeno denominado superconductividad, en el que el valor de la resistencia es prácticamente nulo.

- **Resolución**

La resolución viene determinada por la longitud de una palabra digital (número de bits), es decir, por las agrupaciones de ceros y unos con que se va componiendo (codificando) la señal.

- **ROM**

ROM es la sigla de Read Only Memory, que significa “Memoria de Solo Lectura”: Una memoria de semiconductor destinada a ser leída y no destructible, es decir, que no se puede escribir sobre ella y que conserva intacta la información almacenada, incluso en el caso de que se interrumpa la corriente(memoria no volátil). La ROM suele almacenar la configuración del sistema o el programa de arranque de la computadora.

Las memorias de sólo lectura o ROM son utilizadas como medio de almacenamiento de datos en las computadoras. Debido a que no se puede escribir fácilmente, su uso principal reside en la distribución de programas que están estrechamente ligados al soporte físico de la computadora, y que seguramente no necesitarán actualización. Por ejemplo una tarjeta gráfica puede realizar algunas funciones básicas a través de los programas contenidos en la ROM.

- **RS-232**

RS232 (Recommended Standard 232, también conocido como Electronic Industries Alliance RS-232C) es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (*Data Communication Equipment*, Equipo de Comunicación de datos), aunque existen otras en las que también se utiliza la interfaz RS-232.

En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos, como pueden ser computadores. Evidentemente, en el caso de interconexión entre los mismos, se requerirá la conexión de un DTE (*Data Terminal Equipment*) con otro DTE. Para ello se utiliza una conexión entre los dos DTE sin usar módem, por ello se llama: null módem ó módem nulo.

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9, o popularmente también denominados DB-9), más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC).

- **Rutina**

En informática, una subrutina o subprograma (también llamada procedimiento, función o rutina), como idea general se presenta como un sub-algoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica.

Una subrutina al ser llamada dentro de un programa hace que el código principal se detenga y se dirija a ejecutar el código de la subrutina, en cambio cuando se llama a una macro, el compilador toma el código de la macro y lo implanta donde fue llamado, aumentando así el código fuente y por consiguiente el objeto.

Las declaraciones de subrutinas generalmente son especificadas por:

- Un nombre único en el ámbito: nombre de la función con el que se identifica y se distingue de otras. No podrá haber otra función ni procedimiento con ese nombre (salvo sobrecarga o polimorfismo en programación orientada a objetos).
- Un tipo de dato de retorno: tipo de dato del valor que la subrutina devolverá al terminar su ejecución.
- Una lista de parámetros: especificación del conjunto de argumentos (pueden ser cero, uno o más) que la función debe recibir para realizar su tarea.
- El código u órdenes de procesamiento: conjunto de órdenes y sentencias que debe ejecutar la subrutina.

- **Sensor**

Un sensor es un dispositivo que detecta manifestaciones de cualidades o fenómenos físicos, como la energía, velocidad, aceleración, tamaño, cantidad, temperatura, etc. y convierte estos fenómenos físicos en un cambio de alguna de las siguientes variables, por ejemplo: resistencia eléctrica (como una RTD), capacidad eléctrica (como un sensor de humedad), tensión eléctrica (como un termopar), corriente eléctrica (como un fototransistor), etc. Es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro elemento. Un sensor es un tipo de transductor que transforma la magnitud que se quiere medir, en otra, que facilita su medida.

- **Señal analógica**

Una señal analógica es un tipo de señal generado por algún tipo de fenómeno electromagnético y que es representable por una función matemática continua en la que es variable su amplitud y período (representando un dato de información) en función del tiempo. Algunas magnitudes físicas comúnmente portadoras de una señal de este tipo son eléctricas como la intensidad, la tensión y la potencia pero también pueden ser hidráulicas como la presión, térmicas como la temperatura, mecánicas, etc. La magnitud también puede ser cualquier objeto medible como los beneficios o pérdidas de un negocio.

En la naturaleza, el conjunto de señales que percibimos son analógicas, así la luz, el sonido, la energía etc., son señales que tienen una variación continua. Incluso la descomposición de la luz en el arco iris vemos como se realiza de una forma suave y continúa. Una onda senoidal es una señal analógica de una sola frecuencia. Los voltajes de la voz y del vídeo son señales analógicas que varían de acuerdo con el sonido o variaciones de luz que corresponden a la información que se está transmitiendo.

- **Señal digital**

La señal digital es un tipo de señal generada por algún tipo de fenómeno electromagnético en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representen

valores discretos, en lugar de valores dentro de un cierto rango. Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada. Esto no significa que la señal físicamente sea discreta ya que los campos electromagnéticos suelen ser continuos, sino que en general existe una forma de discretizarla unívocamente.

Los sistemas digitales, como por ejemplo el ordenador, usan lógica de los dos estados representados por dos niveles de tensión eléctrica, uno alto, H y otro bajo, L (de High y Low, respectivamente, en inglés). Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria. Si el nivel alto se representa por 1 y el bajo por 0, se habla de lógica positiva y en caso contrario de lógica negativa.

Cabe mencionar que, además de los niveles, en una señal digital están las transiciones de alto a bajo y de bajo a alto, denominadas flanco de bajada y de subida, respectivamente. En la ilustración se muestra una señal digital donde se identifican los niveles y flancos.

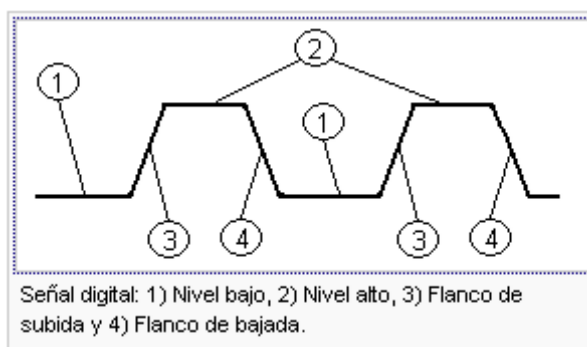


Ilustración 47. Señal digital

- **Software**

Se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de una computadora o microcontrolador, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

- **Tolerancia**

Se aplica a los sensores como componente electrónico (no a los instrumentos), y se relaciona con la dispersión en los parámetros del sensor debidos a su proceso de fabricación.

- **UART**

Su misión principal es convertir los datos recibidos del bus del PC/ μ C en formato paralelo, a un formato serie que será utilizado en la transmisión hacia el exterior. También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus. Universal

indica que el formato de los datos y la velocidad de transmisión son configurables y los niveles de tensión son adaptados por un circuito externo.

- **USB**

El Universal Serial Bus (Bus Universal en Serie) es un estándar industrial desarrollado en los años 1990 que define los cables conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores, periféricos y dispositivos electrónicos. La iniciativa del desarrollo partió de Intel, en conjunto con una decena de compañías, pero actualmente agrupa más de 685.

USB fue diseñado para estandarizar la conexión de periféricos. Su éxito ha sido total y ha desplazado a conectores como el puerto serie, puerto paralelo, puerto de juegos, Apple Desktop Bus o PS/2 a la consideración de dispositivos obsoletos a eliminar de los modernos ordenadores.

En la siguiente ilustración podemos ver los distintos tipos y su patillaje:

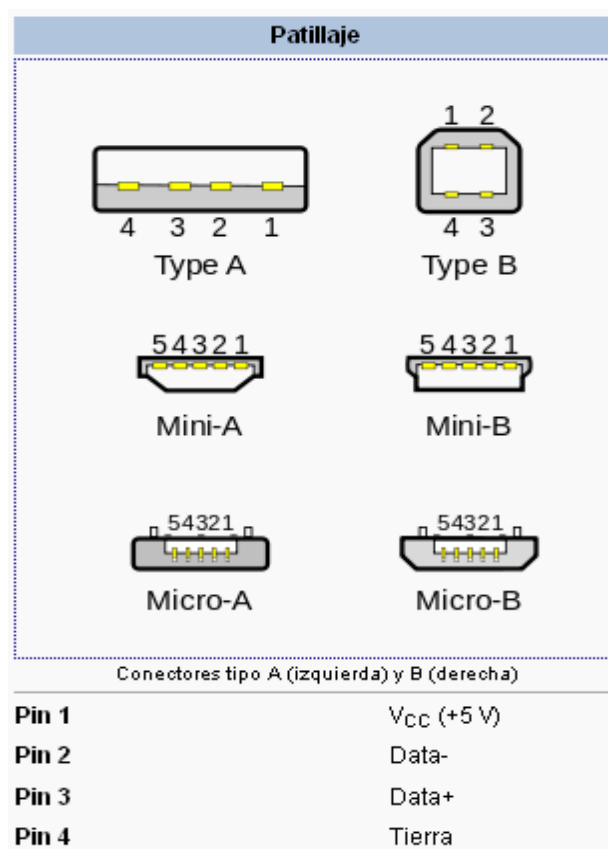


Ilustración 48. Tipos de USB y su patillaje

6.2 ABREVIATURAS

- *LED*: Light-Emitting Diode.
- *DC*: Discontinuous Current.
- *AC*: Alternating Current.
- *BCD*: Binary-Coded Decimal.
- *V, mV, μ V*: Voltios, milivoltios, microvoltios.
- *A, mA, μ A*: Amperios, miliamperios, microamperios.
- *s, ms, μ s, ns*: Segundos, milisegundos, microsegundos, nanosegundos.
- *M, cm, mm*: Metros, centímetros, milímetros.
- *I2C*: Inter-Integrated Circuit.
- *IDE*: Development Environment
- *RTC*: Real-Time Clock
- *UART*: Universal Asynchronous Receiver/Transmitter.
- *SMS*: Short Message Service

CAPITULO VII

NORMATIVA

En la actualidad se están desarrollando trabajos de normalización relacionados con la domótica tanto en organismos europeos (CENELEC, CEN), en organismos internacionales (ISO/IEC) como en organismos nacionales (AENOR). A continuación se resumen estos trabajos.

7.1 CENELEC

7.1.1 Comité técnico 205

El Comité Técnico 205 “Sistemas electrónicos para viviendas y edificios”, se encarga de preparar normas para todos los aspectos de sistemas electrónicos domésticos y en edificios en relación a la sociedad de la información.

En más detalle, preparar normas para asegurar la integración de un espectro amplio de aplicaciones y aspectos de control y gestión de otras aplicaciones en y entorno a viviendas y edificios, incluyendo las pasarelas residenciales a diferentes medios de transmisión y redes públicas, teniendo en cuenta todo lo relativo a EMC y seguridad eléctrica y funcional.

TC 205 no preparará normas de producto sino los requisitos de actuación necesarios y los interfaces de hardware y software necesarios. Las normas deberán especificar ensayos de conformidad.

7.1.2 Normas publicadas

La estructura normativa del TC 205 se basa en dos series de normas, la EN 50491, que especifica los requisitos generales comunes para todos los sistemas domóticos, y la EN 50090, que especifica los requisitos para el protocolo Konnex, dicho protocolo trata comunicaciones estándar, multimedia y abierto, Normalizado en Europa cuyo ámbito actuación se reduce a viviendas y, en menor medida, edificios.

7.2 CEN

7.2.1 Comité técnico 247

El Comité Técnico 247 “Automatización de Edificios, Controles y Gestión de Edificios”, se encarga de la normalización de automatización de edificios, controles y gestión de edificios y servicios para edificios residenciales y no residenciales.

Estas normas incluyen definiciones, requisitos, funciones y métodos de ensayo de los productos de automatización de edificios y sistemas para control automático de instalaciones de servicios en edificios.

Las medidas de integración primarias incluyen interfaces de aplicación, sistemas y servicios para asegurar una gestión técnica de edificios eficiente en cooperación con la gestión comercial y de infraestructuras del edificio.

Se excluyen de su campo de aplicación las áreas de automatización de edificios bajo la responsabilidad de otros comités de CEN/CENELEC.

7.2.2 Normas publicadas

La estructura normativa del TC 247 se basa en dos series de normas, la EN 14908, que trata sobre la comunicación abierta de datos para automatización, control y gestión de edificios, y la EN 13321, que especifica la comunicación de datos en automatización de edificios, los controles y la construcción de sistemas de gestión.

7.3 ISO/IEC

7.3.1 Subcomité 25

El Subcomité 25 “interconexión en la tecnología de la información” es el responsable de la interconexión en la tecnología de la información. Dentro de su campo de aplicación está la normalización de sistemas microprocesadores, así como de interfaces, protocolos y medios de interconexión asociados para equipos de tecnología de la información, generalmente para entornos comerciales y residenciales. Se excluye el desarrollo de normas para redes de telecomunicaciones e interfaces a redes de comunicación.

7.3.2 Normas publicadas

La estructura normativa del subcomité 25 trabaja las normativas: ISO/IEC 15045, ISO/IEC 14762, ISO/IEC 15067 y ISO/IEC 14543. Todas estas normativas tratan los sistemas electrónicos en casas domóticas.

7.4 AENOR

7.4.1 Subcomité 205

El Subcomité 205 es el encargado de la instalación y evaluación de las instalaciones domóticas. Tiene como objetivos: impulsar el desarrollo del mercado, aclarar la confusión en el mercado (un detector de gas por si solo no es un sistema domótico), establecer unos requisitos mínimos que debe cumplir un sistema domótico y la posible certificación de instalaciones domóticas.

7.4.2 Norma EA 0026

El subcomité 205 trabaja la norma EA 0026 que tiene como objeto establecer los requisitos mínimos que deben cumplir las instalaciones de sistemas domóticos para su correcto funcionamiento y las prescripciones generales para la evaluación de aptitud en viviendas. Esta norma es la base para la Certificación de Sistemas Domóticos en Viviendas (AEN/CTC030 de AENOR)

La norma EA 0026 va dirigida tanto al usuario, como al instalador del sistema domótico y al promotor de la vivienda. Al usuario le sirve para: definir sus necesidades permitiendo escoger la opción más adecuada a su realidad, valorar presupuestos que le permitirá conocer y comparar las opciones del mercado y como herramienta para la comparación de las distintas ofertas domóticas. Al instalador le será útil como: proceso de certificación voluntario que le permite una diferenciación respecto al resto de instaladores; optimización de instalaciones tradicionales y como diversificación de sus actividades: apuesta por nuevas oportunidades de negocio. Finalmente el promotor podrá utilizar esta norma para: satisfacer al usuario, reducir el período de venta y evolucionar hacia la sostenibilidad en el sector residencial.

CAPITULO VIII

BIBLIOGRAFÍA

8.1 PÁGINAS WEB

Información sobre domótica. Historia, desarrollo, conceptos básicos:

www.upcommons.upc.edu, Sistemas Domóticos

www.theslogan.es, Domótica, la Revolución del Bienestar

www.monografias.com/trabajos5/edin/edin.shtml, Edificios Inteligentes y Casa Domóticas.

www.webdehogar.com/decoracion/domotica-hogar-casa-inteligente.htm

Información acerca de sensores de movimiento y contactos magnéticos. Tipo de sensores, características, precios,...:

www.homesecuritystore.com

Páginas web para la compra de los distintos componentes y sensores con información acerca de los mismos a través de sus datasheets:

www.es.rs-online.com/web/

www.microchip.com/

www.cooking-hacks.com

www.diotronic.com

www.arduino.cc

Página web en la cual encontrar el datasheet de los componentes:

www.alldatasheet.es

Consultas sobre programación y placa de desarrollo. Páginas web con foros donde poder resolver dudas sobre programación:

www.foro.elhacker.net/electronica

www.todopic.com.ar/foros/

www.mikroe.com

www.ccsinfo.com/forum

www.ucontrol.com.ar/forosmf/

www.forosdeelectronica.com

Información sobre normativa y normalización de domótica en organismos nacionales, europeos e internacionales :

www.cedom.es/normativa.php

Información en general:

www.es.wikipedia.org

8.2 LIBROS

Conocimientos básicos para manejar el compilador CCS y el programa de simulación Proteus y ayuda sobre lenguaje C:

García Breijo, Eduardo. "Compilador C CCS Y Simulador Proteus Para Microcontroladores Pic". Alfaomega, 2008.

Manual de funcionamiento del compilador PCW de CCS. Explicación de las distintas funciones, declaración de variables,...:

Cánovas López, Andrés. "Manual de usuario del compilador PCW de CCS".

Información sobre los microcontroladores PIC. Funcionamiento, estructura, posibles usos y ejemplos explicativos:

Tavernier, Christian. "Microcontroladores PIC". Paraninfo, 1997.

Lenguaje Basic. Desarrollo de las funciones, declaración de variables, etc. en este lenguaje de programación:

Ibrahim, Dogan. "Programación de microcontroladores PIC. Desarrollo de 30 proyectos con PIC BASIC y PIC BASIC Profesional". Marcombo, 2007.

Programación de microcontroladores. Funcionamiento, estructura, posibles usos y ejemplos explicativos:

Carretero Pérez, Jesús; García Carballeira, Félix; Pérez Lobato, José Manuel; Fernández Muñoz, Javier; Calderón Mateos, Alejandro. "Fundamentos de Programación". Thompson Paraninfo

CAPÍTULO IX

ANEXOS

9.1 CÓDIGO DEL PROGRAMA

```
#fuses HS           //Crystal osc <= 4mhz for PCM/PCH , 3mhz to 10 MHz for PCD
#fuses NOWDT        //Deshabilita el Watch Dog Timer
#fuses NOPROTECT    //Deshabilita la protección del código del PIC
#fuses NOLVP        //No Low Voltage Programming on B3(PIC16) or B5(PIC18)
#fuses PUT          //Habilita le reseteo de power up timer
#fuses NOBROWNOUT   //No brownout reset
```

```
#use delay (CLOCK=8000000)
```

```
#use fast_io(B)
#byte port_b = 0x06
#byte tris_b = 0xff
```

```
#include <lcd5.c>      //librería manejo lcd 8 bits
#include <string.h>     //librería manejo cadenas
#include <_DS1307.c>
```

```
#rom 0x2100={ '1','2','3' } //la EEPROM se carga con los valores 00h desde la dirección
00h hasta 06h
```

```
//////////DEFINICION DE FUNCIONES//////////
```

```
void captura(void);           //función que captura la clave tecleada
void nueva_clave_admin(void); //función que añade la nueva clave administrativa
void limpiar_clave(void);     //función que pone a '0' la variable clave
void escribir_clave_en_eeprom(int d); //función que escribe las claves en la memoria
eeprom
void menu_principal(void);    //función que muestra el menú principal
void cambiar_clave(void);     //función que cambia clave administrativa
int comparar_claves(int);    //función que compara las claves
void activar_alarma(void);    //función que activa la alarma
void desactivar_alarma(void); //función que desactiva la alarma
void leer_clave_en_eeprom(int); //función que lee las claves y las guarda en su
correspondiente variable
void conteo_intentos(void);   //función que cuenta el numero de intentos de
ingreso de contraseña
void alarma_general(void);     //función que enciende la sirena
void regreso_menu(void);      //función que después de una intrusión recupera la
alarma
```

```
void write_string_eeprom(int address,char *data); //función que graba en la eeprom
las horas fijadas
```

```
void read_string_eeprom(int address,char *sl);    //función que lee de la eeprom las
horas fijadas
```

```
//////////VARIABLES GLOBALES//////////
```

```
int flag_teclado=0;           //flag que señala que una tecla ha sido pulsada
char bandera=0;               //flag que señala que se ha producido una
interrupción externa
int tecla;                    //variable que recoge la tecla pulsada
int fila,columna;             //variables que recogen la fila y columna de la tecla
pulsada
char clave[3]={};            //variable usada para recoger la clave tecleada
int d=0;                      //variable usada para movernos por las posiciones de la
memoria
char clave_eeprom_admin[3];   //variable usada para comparar los datos de la
eeprom contra los datos ingresados
int alarm_cont=0;             //variable global que cuenta los fallos
```

```
//////////variables del ds1307
```

```
int hr;
int min;
int sec;
char ValoresLeidos[15];
```

```
/*****
```

Rutina de Interrupción:

- Antirrebote.
- Llama a rutina "captura" para saber la tecla presionada.
- Activa la bandera "flag_teclado" a usar en el main o en otra parte del programa.

```
*****/
```

```
#INT_RB
```

```
void tecla_presionada()
```

```
{
```

```
int lectura_1,lectura_2;           //variables locales para detectar la tecla pulsada
```

```
lectura_1 = port_b & 0b11110000;    //lee portb4:7 y lo guarda en lectura_1
```

```
delay_ms(10);                      //Función AntiRebotes: Espero 10ms.
```

```
lectura_2 = port_b & 0b11110000;    //Vuelvo a leer el puerto.
```

```
if(lectura_1==lectura_2 && lectura_1!=0) //Si el valor es igual y distinto de 0 es
porque en realidad se presiono una tecla.
```

```
{
```

```
    flag_teclado=1;                //marcamos que se ha pulsado una tecla
```

```
    captura();                     //capturamos dicha tecla
```

```
}
```

```
    set_tris_b(0b11110000);
```

```
    output_b(0b00001111);
```

```
    clear_interrupt(INT_RB);
```

```
    set_tris_b(0b11110001);
```

```
}
```

```
//////////INTERRUPCION EXTERNA//////////
#INT_EXT
void externa(void)
{
    bandera=1;        //marcamos que se ha producido una interrupción externa
}

//////////PROGRAMA PRINCIPAL//////////
void main()
{

    char alarmas[14];
    char alarma1[]="221415";    //hora fijada para que se encienda el led
    char alarma2[]="221430";    //hora fijada para que se apague el led
    char *leido=alarmas;

    leer_clave_en_eeprom(d);
    write_string_eeprom(0x04,alarma1); //escribo las horas fijadas en la eeprom
    write_string_eeprom(0x10,alarma2);

    set_tris_a(0b11100000);    //configuro el puerto a
    set_tris_b(0b11110001);    //configuro el puerto b
    set_tris_c(0b11111111);    //configuro el puerto c
    enable_interrupts(INT_RB);  //habilito la interrupción por cambio de nibble alto
    ext_int_edge(L_TO_H);       //habilito la interrupción externa por cambio de flanco
    enable_interrupts(INT_EXT); //habilito la interrupción externa
    enable_interrupts(GLOBAL);  //habilito las interrupciones

    disable_interrupts(INT_EXT); //deshabilito la interrupción externa

    lcd_init();                //inicializa lcd
    delay_ms(1000);
    lcd_putc (" Bienvenido ");
    delay_ms(1000);

    ds1307_init(DS1307_ALL_DISABLED);
                //Set date for -> 21-Abril-2010 Lunes
                //Set time for -> 22:14:00
    ds1307_set_date_time(21,4,10,1,22,14,00);    //hora que le fijo al reloj

    while(TRUE)
    {
        lcd_putc("\f0-Activa Alarma\n1-Cambia Clave");
        delay_ms(1000);
        if(flag_teclado==1)    //Si se ha pulsado una tecla
        {
            flag_teclado=0;
            menu_principal();    //voy al menú principal
        }
        if(bandera==1)        //Si se ha producido una interrupción externa
```

```
{
    bandera=0;
    alarma_general();    //activo la sirena
    regreso_menu();
}
else
{
    delay_ms(100);
    ds1307_get_time(hr,min,sec);    //capturo la hora del reloj

    read_string_eeprom(0x04,leido);    //leo y comparo la primera
hora fijada
    sprintf(ValoresLeidos,"%02u%02u%02u",hr,min,sec);
    if(strcmp(ValoresLeidos,leido)==0)    //si la primera hora fijada
coincide con la del reloj...
    {
        output_high(pin_A5);    //...enciendo el led
    }
    read_string_eeprom(0x10,leido);    //leo y comparo la segunda
hora fijada
    sprintf(ValoresLeidos,"%02u%02u%02u",hr,min,sec);
    if(strcmp(ValoresLeidos,leido)==0)    //si la segunda hora fijada
coincide con la del reloj...
    {
        output_low(PIN_A5);    //...apago el led
        set_tris_a(0b11100000);
    }
}
}
```

/*****

Esta Rutina permite rastrear el teclado y devuelve el valor de la tecla que se ha
presionado

*****/

void captura(void) //función que captura la clave tecleada

```
{
int const teclado[4][4] = {{0,'1','2','3'},
                           {0,'4','5','6'},
                           {0,'7','8','9'},
                           {0,'*','0','#'}};
```

```
int lectura,i;
for(i=0;i<4;i++)
{
    output_b(0);
    bit_set(port_b,i);
    lectura = port_b & 0b11110000;
    switch(lectura)
    {
        case 0b00010000: fila=0;
        columna=i;
        break;
```

```
case 0b00100000: fila=1;
columna=i;
break;
case 0b01000000: fila=2;
columna=i;
break;
case 0b10000000: fila=3;
columna=i;
break;
}
}
tecla = teclado[fila][columna];
}
```

////////FUNCION MENU PRINCIPAL

```
void menu_principal (void)          //función que muestra el menú principal
{
d=0;
    switch (tecla)
    {
        case '0':          //si '0' va a activar la alarma
            activar_alarma();
            break;
        case '1':          //si '1' va a comprobar clave para después cambiar la clave
            cambiar_clave();
            break;
    }
}
```

//////////FUNCION CAMBIAR CLAVE

```
void cambiar_clave(void)            //función que cambia clave administrativa
{
int p;
tecla=0;
d=0;
int aux=0;

    leer_clave_en_eeprom(d);
    limpiar_clave();
    lcd_putc("\fEscribir clave:\n");
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
        {
            flag_teclado=0;
            lcd_putc("*");
            clave[d]=tecla;
            d++;
            if (d==3)                //Si ya se han introducido todos los valores de la clave
            {
                p=comparar_claves(d);          //compara ambas claves
            }
        }
    }
}
```

```
    if (p==0)                //si son iguales...
    {lcd_putc("\fClave Correcta");
      delay_ms(2000);
      alarm_cont = 0;
      limpiar_clave();        //pongo a '0' la variable clave
      nueva_clave_admin();    //introduzco la nueva clave
    }
    if (p!=0)                //si son distintas...
    {lcd_putc("\fClave Incorrecta");
      delay_ms(2000);
      limpiar_clave();        //pongo a '0' la variable clave
      conteo_intentos();
    }
  }
  else
  {
    tecla=0;
  }
}
}

//////////FUNCION COMPARAR CLAVES
int comparar_claves(int)    //función que compara la clave administrativa y la
introducida

{
  d=0;
  int s=0;
  for (d=0;d<3;d++) //comparo carácter a carácter las claves
  {
    if (clave[d]==clave_eeprom_admin[d]) //si el carácter es igual en las claves
      s++;                               //paso al siguiente carácter
    else                               //si es distinto la contraseña es incorrecta
    {s=0;
      lcd_putc("\fClave Incorrecta");
      break;
    }
  }
  if (s==3)                //si todos los tres caracteres son iguales
    return 0;              //devuelvo un '0'
}

//////////LIMPIAR CLAVE
void limpiar_clave() //función que pone a '0' la variable clave
{
  int k;
  for(k=0;k<3;k++) //limpiamos uno a uno cada valor de la variable
  {
    clave[k]=0x00;
  }
}
```

```
//////////FUNCION NUEVA CLAVE
void nueva_clave_admin(void)      //función que añade la nueva clave administrativa
{
    tecla=0;                      //limpio las variables
    int aux=0;
    d=0;

    lcd_putc("\fNueva clave:\n"); //presentación
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
        {
            flag_teclado=0;
            lcd_putc("*");
            clave[d]=tecla;
            if(d<3)
            {
                escribir_clave_en_eeprom(d); //guardamos la nueva clave en la
                memoria EEPROM
                tecla=0;
                d++;
            }
            else
            {
                d=0;
                limpiar_clave();
            }
        }
    }
}
```

```
//////////ACTIVAR ALARMA
void activar_alarma(void)      //función que activa la alarma
{
    int p;
    tecla=0;
    d=0;
    int aux=0;
    leer_clave_en_eeprom(d);
    lcd_putc("\fEscribir clave:\n");
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
        {
            lcd_putc("*");
            clave[d]=tecla;
            d++;
            if (d==3)          //Si ya se han introducido todos los valores de la clave
            {
                p=comparar_claves(d);          //compara ambas claves
            }
        }
    }
}
```



```
    if (p==0)                //si son iguales...
    {lcd_putc("\fAlarma Activada");
      delay_ms(2000);
      alarm_cont = 0;
      limpiar_clave();        //pongo a '0' la variable clave
      desactivar_alarma();    //vamos a la rutina para desactivar la alarma
    }
    if (p!=0)                //si son distintas...
    {lcd_putc("\fClave Incorrecta");
      delay_ms(2000);
      limpiar_clave();        //pongo a '0' la variable clave
      conteo_intentos();
    }
  }
  else
  {
    tecla=0;
  }
}
}
```

/////////DESACTIVAR ALARMA

```
void desactivar_alarma(void)    //función que desactiva la alarma
{
  int p;
  tecla=0;
  d=0;
  int aux=0;
  enable_interrupts(INT_EXT);
  lcd_putc("\fEscribir clave:\n");
  delay_ms(500);
  for(aux=0;aux<3;aux++)
  {
    while(tecla==0 && bandera==0);
    {
      lcd_putc("");
      clave[d]=tecla;
      d++;
      if (d==3)                //Si ya se han introducido todos los valores de la clave
      {
        leer_clave_en_eeprom(d); //lee la contraseña que antes hemos cambiado
        p=comparar_claves(d);    //compara ambas claves
        if (p==0)                //si son iguales...
        {lcd_putc("\fAlarma\nDesactivada");
          delay_ms(2000);
          alarm_cont = 0;
          limpiar_clave();        //pongo a '0' la variable clave
          disable_interrupts(INT_EXT);
        }
        if (p!=0)                //si son distintas...
        {lcd_putc("\fClave Incorrecta");
          delay_ms(2000);
        }
      }
    }
  }
}
```

```
        limpiar_clave();           //pongo a '0' la variable clave
        conteo_intentos();
    }
}
else
{
    tecla=0;
}
}
}

/////////ESCRIBIR_CLAVE_EN_EEPROM
void escribir_clave_en_eeprom(int d)    //función que escribe las claves en la memoria
eeprom
{
    write_eeprom(d,tecla);           //guardamos cada valor de la variable global
    delay_ms(20);                    //en la dirección d + 0 (para no pisar datos) de la eeprom
    d++;                             //en una palabra escribe de la posición 0 a la 2 (3 lugares)
}

/////////LEER_CLAVE_EN_EEPROM
void leer_clave_en_eeprom(int d)       //función que lee las claves y las guarda en su
correspondiente variable
{
    for (d=0;d<3;d++)
    {
        clave_eeprom_admin[d] = read_eeprom(d);    //guardamos cada valor de la
        direccion i + 0 (para no pisar datos)
        delay_ms(20);                             //de la eeprom en la variable global correspondiente
    }
}

/*****
FUNCIONES ESCRITURA Y LECTURA EEPROM INTERNA
*****/
void write_string_eeprom(int address,char *data)
{
    while(*data!=0)
    {
        write_eeprom(address++,*data);
        data++;
    }
    write_eeprom(address,0);
}

void read_string_eeprom(int address,char *sl)
{
    sl--;
    do
    {
```

```
    sl++;
    *sl=read_eeprom(address++);
    }while(*sl!=0);
}
```

//////////FUNCION CONTEO INTENTOS

void conteo_intentos(void) //función que cuenta el numero de intentos de
ingreso de contraseña

```
{
    if (alarm_cont==2)
    {
        alarm_cont = 0; //pongo a cero el contador de errores
        alarma_general();
    }
    else
    {
        alarm_cont++;
        tecla=0; //limpio las variables
        d=0;
    }
}
```

//////////FUNCION ALARMA GENERAL

void alarma_general (void) //función que enciende la sirena

```
{
    set_tris_a(0b11110111); //pin A3 como salida
    lcd_putc("\f" ); //borra pantalla
    delay_ms(100);
    output_high(pin_A3); //enciende la sirena durante 10 seg y muestra
    mensaje
    lcd_putc("\f ALARMA ");
    delay_ms(10000);
    output_low(pin_A3); //apaga la sirena
    alarm_cont = 0;
}
```

//////////FUNCION REGRESO MENU

void regreso_menu(void)

```
{
    int p;
    tecla=0;
    d=0;
    int aux=0;

    limpiar_clave();
    lcd_putc("\fEscribir clave:\n");
    delay_ms(500);
    for(aux=0;aux<3;aux++)
    {
        while(tecla==0 && bandera==0);
    }
}
```

```
flag_teclado=0;
lcd_putc("*");
clave[d]=tecla;
d++;
if (d==3)           //Si ya se han introducido todos los valores de la clave
{
    p=comparar_claves(d);           //compara ambas claves
    if (p==0)           //si son iguales...
    {
        lcd_putc("\fClave Correcta");
        delay_ms(2000);
        limpiar_clave();           //pongo a '0' la variable clave
        disable_interrupts(INT_EXT);
    }
    if (p!=0)           //si son distintas...
    {
        lcd_putc("\fClave Incorrecta");
        delay_ms(2000);
        limpiar_clave();           //pongo a '0' la variable clave
        alarma_general();
    }
}
else
{
    tecla=0;
}
}
}
```

9.2 MANUAL DE USUARIO DEL SISTEMA DE ALARMA

Este manual explica todo lo que se necesita saber acerca del sistema de alarma y explica paso a paso las instrucciones para todas las funciones del sistema.

El sistema de seguridad está compuesto de un display, un interruptor que simula los sensores, un interruptor de reset y un teclado para la introducción de las contraseñas y navegación por el menú. A continuación se muestra una imagen del sistema de alarma.

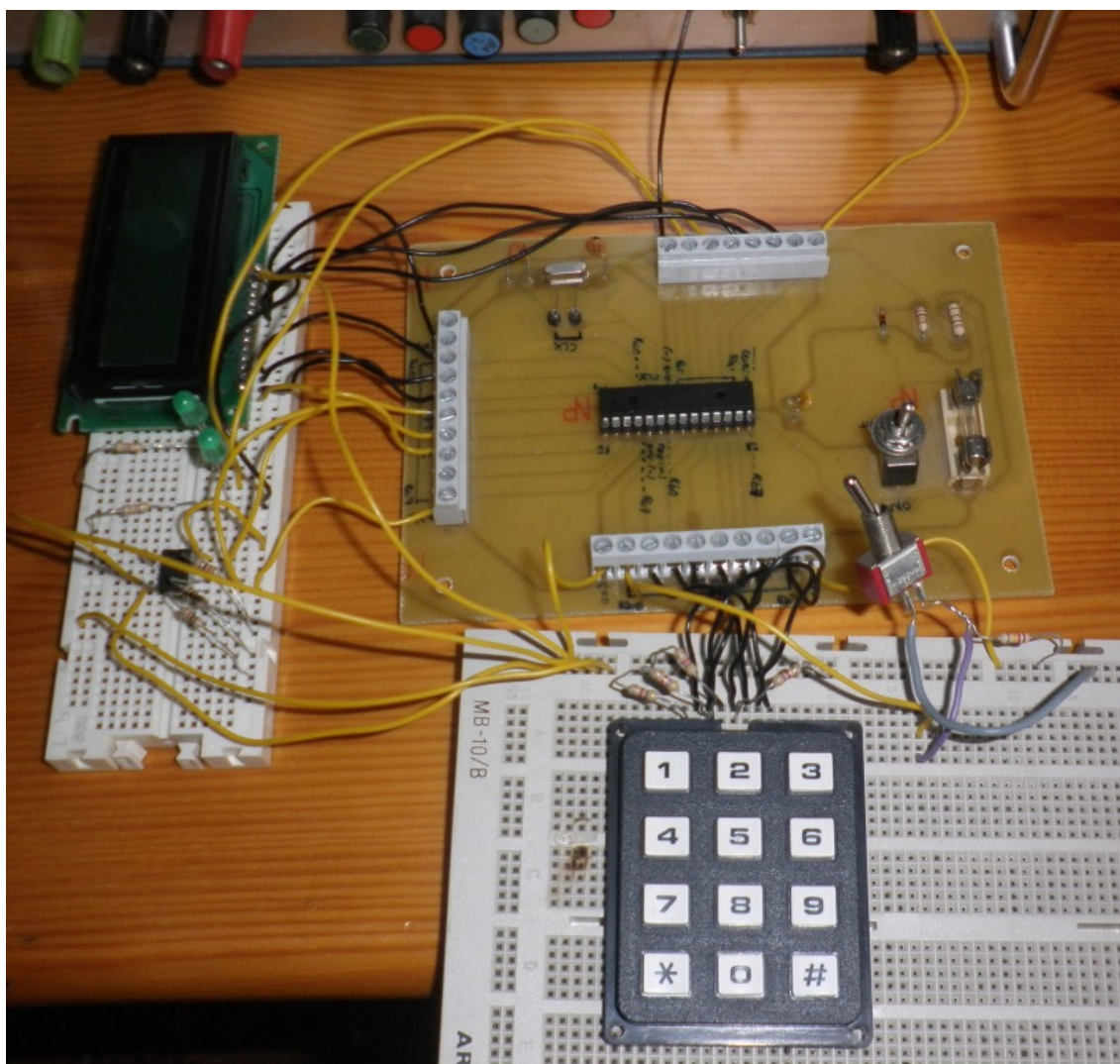


Ilustración 49. Sistema de alarma

Una vez encendido el sistema de alarma mediante la fuente de alimentación lo primero que se visualiza es un mensaje de bienvenida.



Ilustración 50. Mensaje de bienvenida

A continuación el sistema nos lleva al menú principal en el que aparece las dos opciones principales del sistema de alarma: activar alarma y cambiar clave.



Ilustración 51. Menú principal

El sistema por defecto viene con la contraseña de seguridad “123” por lo que se recomienda al usuario cambiarla por otra para mayor seguridad. La contraseña de seguridad se utiliza tanto para cambiar de contraseña como para activar y desactivar la alarma por lo que tiene una gran importancia. Para cambiar la contraseña primero pulsáramos la tecla “1” para entrar en el submenú “Cambiar clave”. Después introduciríamos la contraseña de seguridad (“123” en el caso de que se realice la primera vez). Este procedimiento es el mismo tanto se cambie la contraseña por primera vez como en posteriores cambios.



Ilustración 52. Cambio de clave

Si la clave introducida es la correcta aparece un mensaje confirmándolo y si no, aparecerá un mensaje avisándonos de que la clave es incorrecta y se retornara al menú principal sumando un intento de los tres que se disponen antes de que salte la alarma.

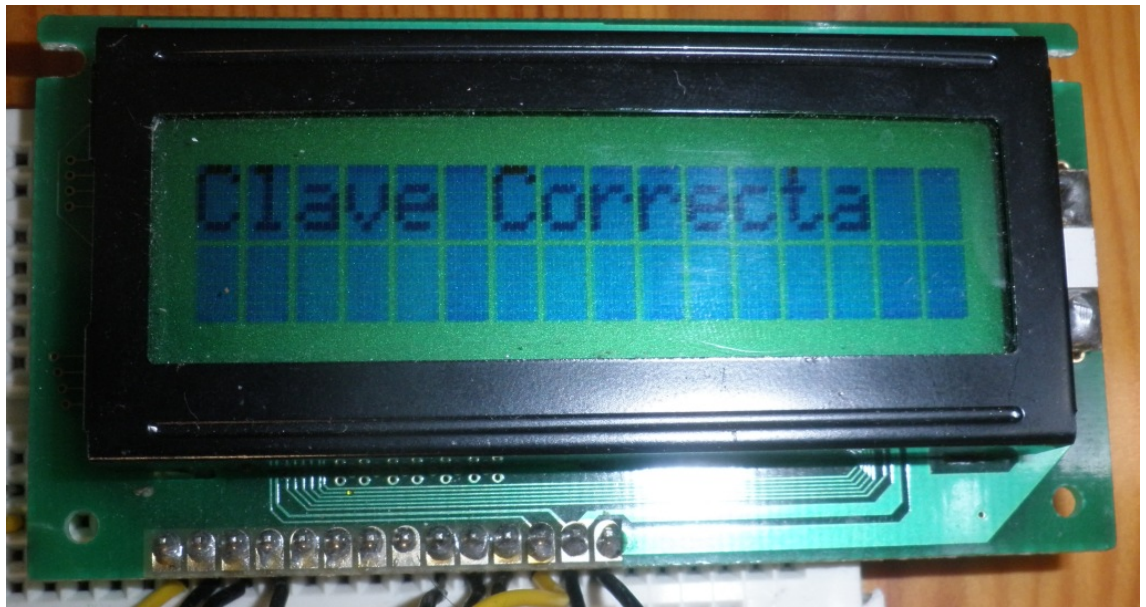


Ilustración 53. Mensaje de “Clave correcta”

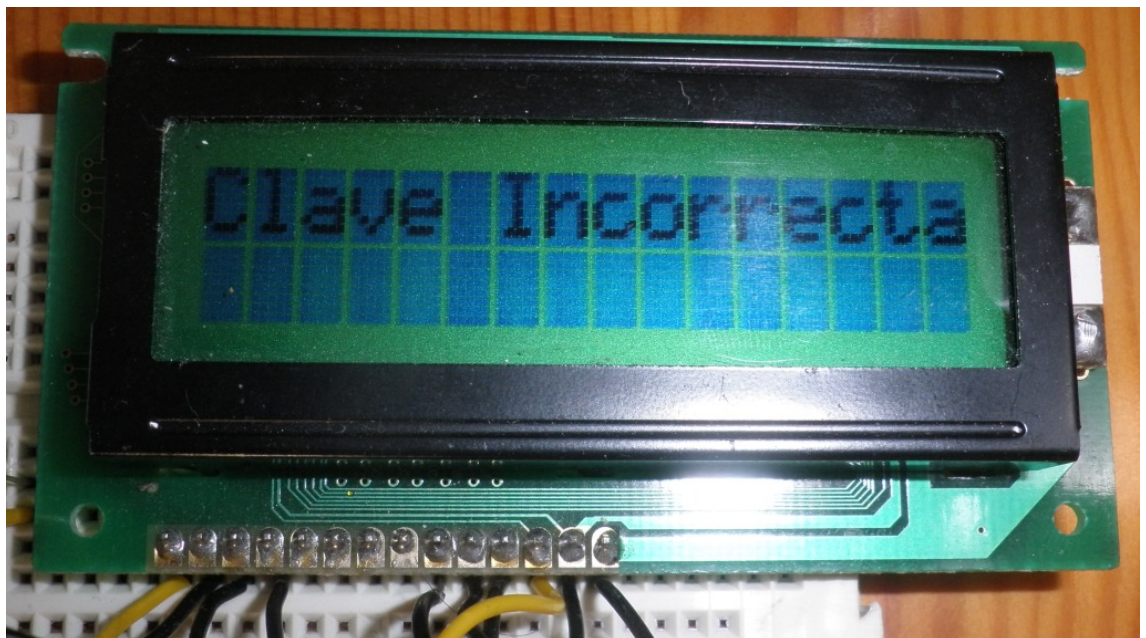


Ilustración 54. Mensaje de “Clave incorrecta”

Cuando se introduce la clave correcta, aparece un mensaje “Nueva clave:” e introduciríamos la nueva contraseña de seguridad.

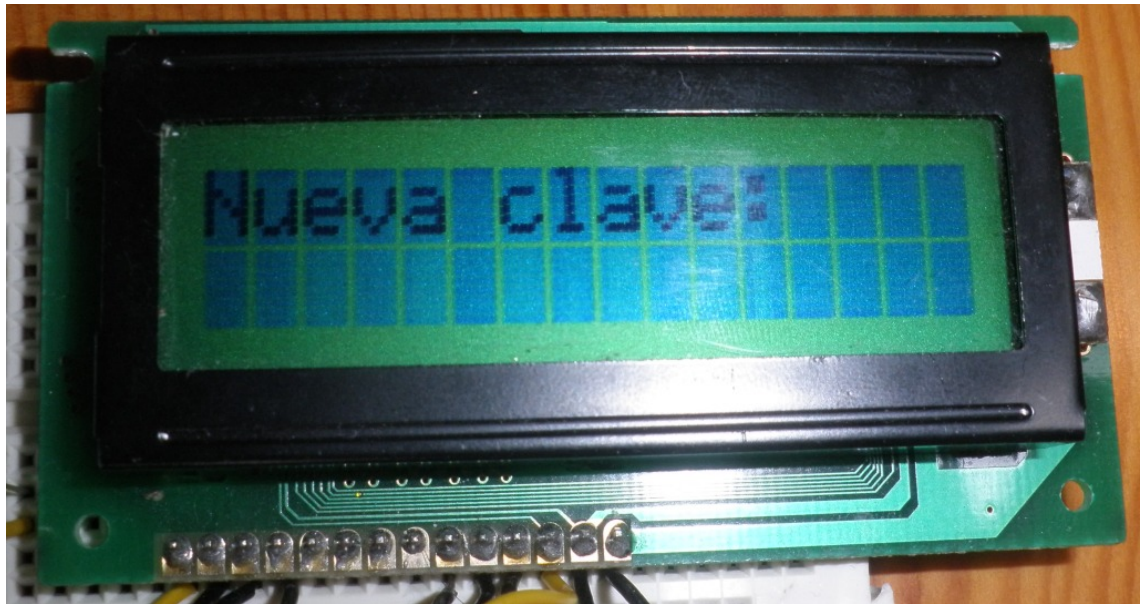


Ilustración 55. Nueva clave

Para activar y desactivar la alarma, el procedimiento es muy similar al del cambio de clave. Primero se pulsaría la tecla “0” cuando se estuviera en el menú principal. De esta manera se entra en el submenú “Activar alarma”. Para activarla se tiene que introducir la contraseña de seguridad.

Si la clave introducida es la correcta aparece un mensaje confirmándolo y si no, aparecerá un mensaje avisándonos de que la clave es incorrecta y se retornara al menú principal sumando un intento de los tres que se disponen antes de que salte la alarma, al igual que ocurría en el caso del cambio de clave.



Ilustración 56. Mensaje de "Alarma activada"

Al introducir la contraseña correctamente, los sensores entrarían en funcionamiento (en este caso, el interruptor que los simula) y permanecerían en funcionamiento hasta que desactiváramos la alarma. Para desactivarla habría que volver a introducir la contraseña de seguridad correctamente o se sumaría otro intento fallido al contador de intentos.



Ilustración 57. Alarma desactivada

Hay tres maneras de que salte la alarma: que actúe uno de los sensores cuando este activada, que se lleguen a los tres intentos fallidos de introducción de la contraseña y que una vez que ha saltado la alarma se introduzca mal la contraseña de seguridad, ya que en ese caso volvería a saltar.

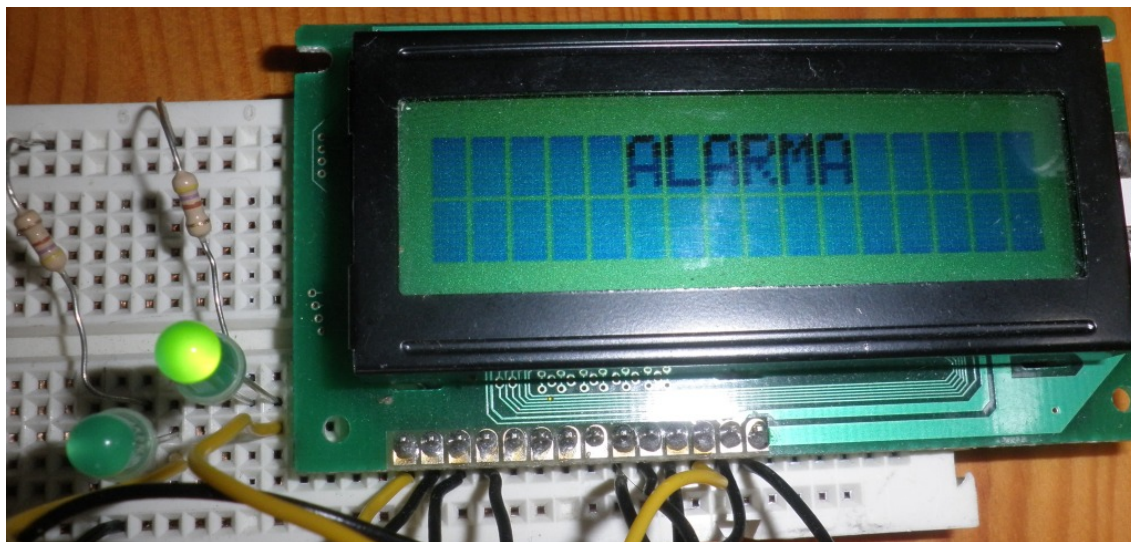


Ilustración 56. Alarma

9.3 DATASHEET Y CATÁLOGOS

A continuación se muestran los datasheets y catálogos más relevantes de los diversos dispositivos utilizados en la fabricación de este proyecto.



PIC16F87X

Data Sheet

28/40-Pin 8-Bit CMOS FLASH
Microcontrollers

"All rights reserved. Copyright © 2001, Microchip Technology Incorporated, USA. Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights."

Trademarks

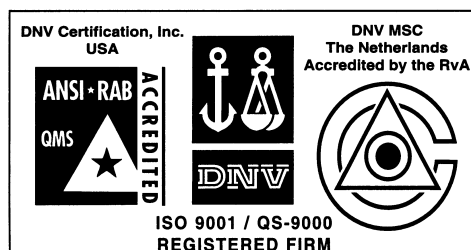
The Microchip name, logo, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, KEELOQ, SEEVAL, MPLAB and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Total Endurance, ICSP, In-Circuit Serial Programming, Filter-Lab, MXDEV, microID, *FlexROM*, *fuzzyLAB*, MPASM, MPLINK, MPLIB, PICDEM, ICEPIC, Migratable Memory, FanSense, ECONOMONITOR and SelectMode are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Term Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2001, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoq® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

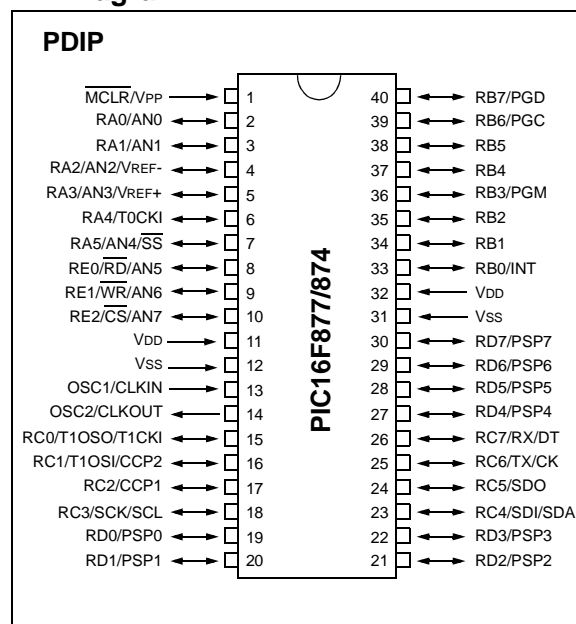
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



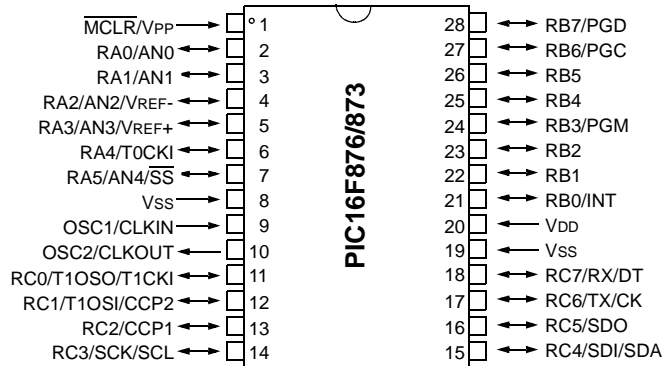
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external \overline{RD} , \overline{WR} and \overline{CS} controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

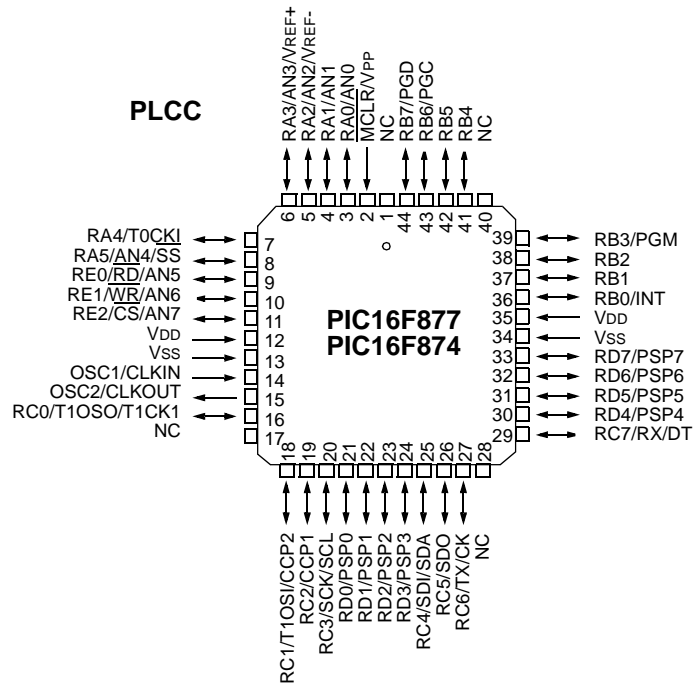
PIC16F87X

Pin Diagrams

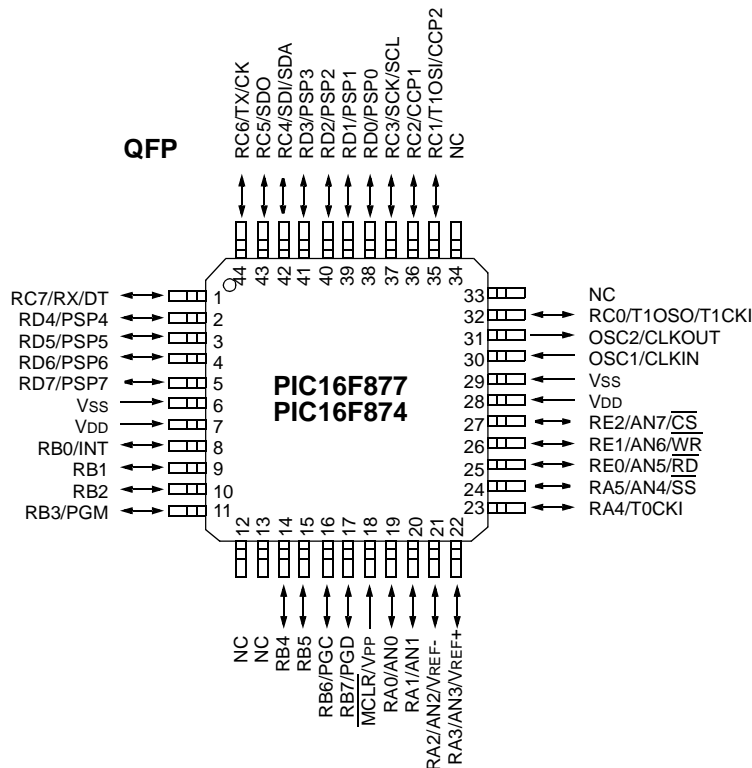
PDIP, SOIC



PLCC



QFP



Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

PIC16F87X

Table of Contents

1.0	Device Overview	5
2.0	Memory Organization.....	11
3.0	I/O Ports	29
4.0	Data EEPROM and FLASH Program Memory.....	41
5.0	Timer0 Module	47
6.0	Timer1 Module	51
7.0	Timer2 Module	55
8.0	Capture/Compare/PWM Modules	57
9.0	Master Synchronous Serial Port (MSSP) Module.....	65
10.0	Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART)	95
11.0	Analog-to-Digital Converter (A/D) Module.....	111
12.0	Special Features of the CPU.....	119
13.0	Instruction Set Summary.....	135
14.0	Development Support	143
15.0	Electrical Characteristics.....	149
16.0	DC and AC Characteristics Graphs and Tables.....	177
17.0	Packaging Information	189
	Appendix A: Revision History	197
	Appendix B: Device Differences	197
	Appendix C: Conversion Considerations	198
	Index	199
	On-Line Support.....	207
	Reader Response	208
	PIC16F87X Product Identification System	209

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@mail.microchip.com or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on all of our products.

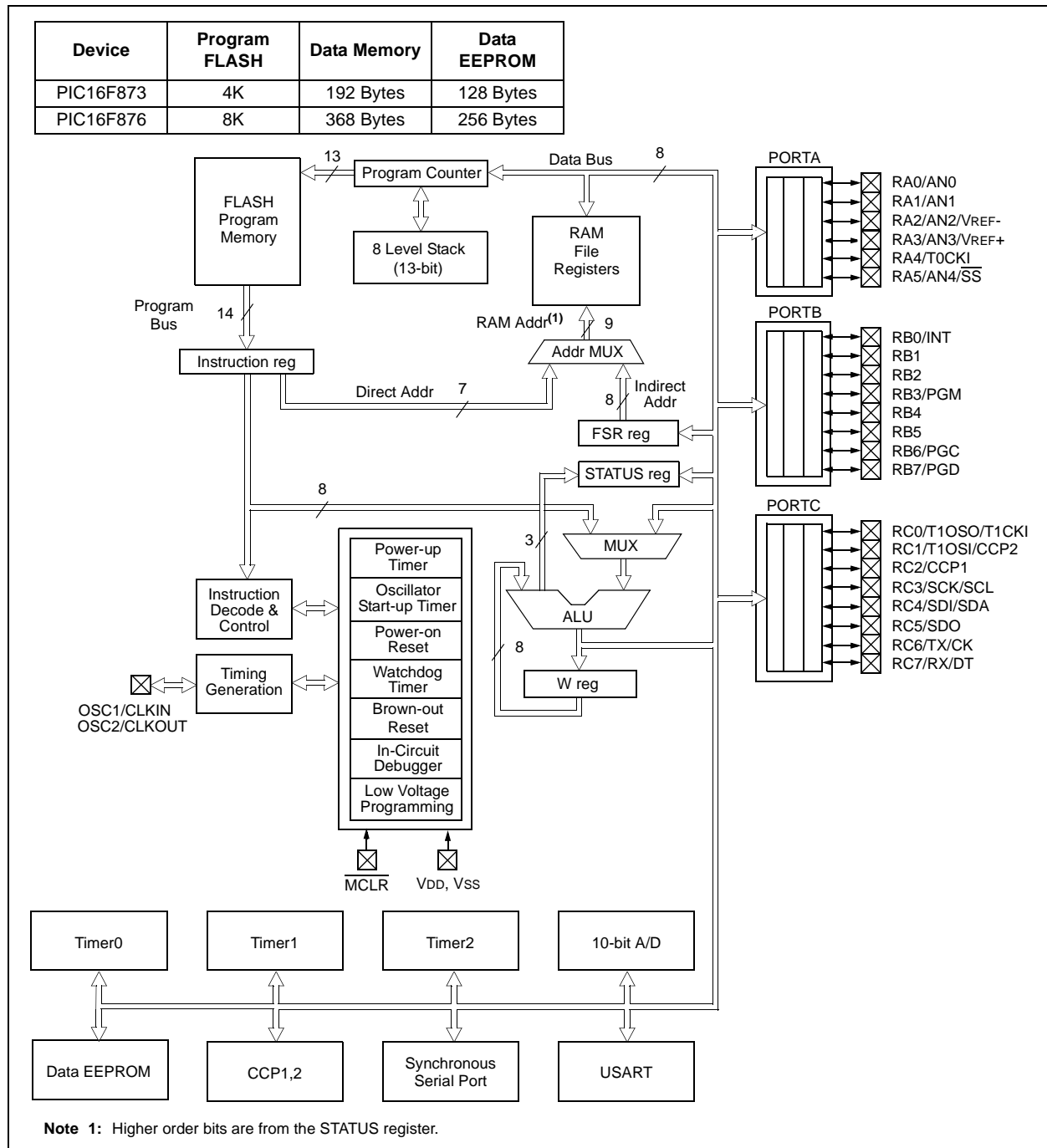
1.0 DEVICE OVERVIEW

This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM



PIC16F87X

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

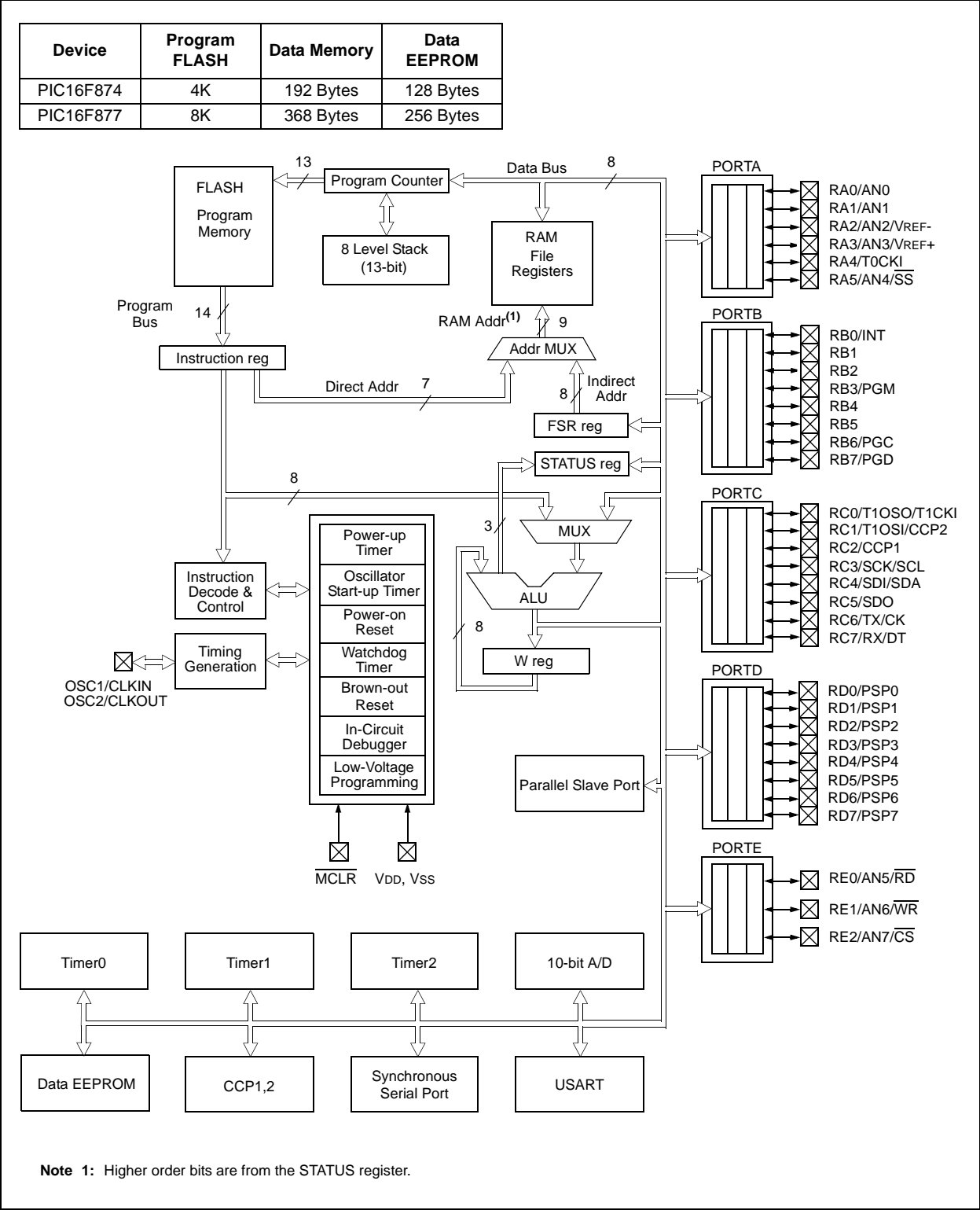


TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0.</p> <p>RA1 can also be analog input1.</p> <p>RA2 can also be analog input2 or negative analog reference voltage.</p> <p>RA3 can also be analog input3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 module. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ⁽²⁾	
RB7/PGD	28	28	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I²C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I²C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p>
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F87X

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS ⁽⁴⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	3	19	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0.</p> <p>RA1 can also be analog input1.</p> <p>RA2 can also be analog input2 or negative analog reference voltage.</p> <p>RA3 can also be analog input3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/VREF-	4	5	21	I/O	TTL	
RA3/AN3/VREF+	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST ⁽²⁾	
RB7/PGD	40	44	17	I/O	TTL/ST ⁽²⁾	

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as an external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	
RC2/CCP1	17	19	36	I/O	ST	
RC3/SCK/SCL	18	20	37	I/O	ST	
RC4/SDI/SDA	23	25	42	I/O	ST	
RC5/SDO	24	26	43	I/O	ST	
RC6/TX/CK	25	27	44	I/O	ST	
RC7/RX/DT	26	29	1	I/O	ST	
RD0/PSP0	19	21	38	I/O	ST/TTL ⁽³⁾	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	23	40	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	24	41	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	30	2	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	31	3	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	32	4	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	33	5	I/O	ST/TTL ⁽³⁾	
RE0/ $\overline{\text{RD}}$ /AN5	8	9	25	I/O	ST/TTL ⁽³⁾	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5. RE1 can also be write control for the parallel slave port, or analog input6. RE2 can also be select control for the parallel slave port, or analog input7.
RE1/ $\overline{\text{WR}}$ /AN6	9	10	26	I/O	ST/TTL ⁽³⁾	
RE2/ $\overline{\text{CS}}$ /AN7	10	11	27	I/O	ST/TTL ⁽³⁾	
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F87X

NOTES:

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory, and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK

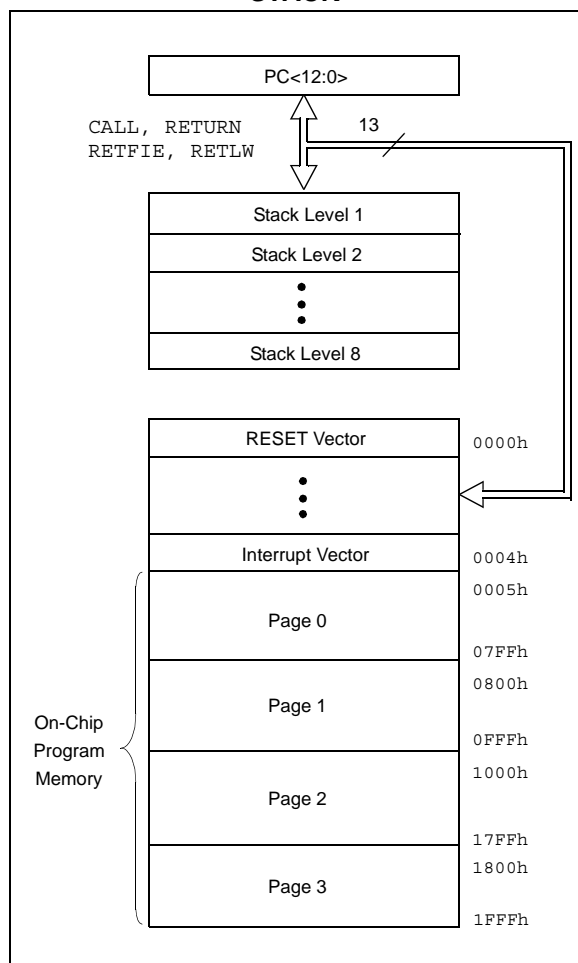
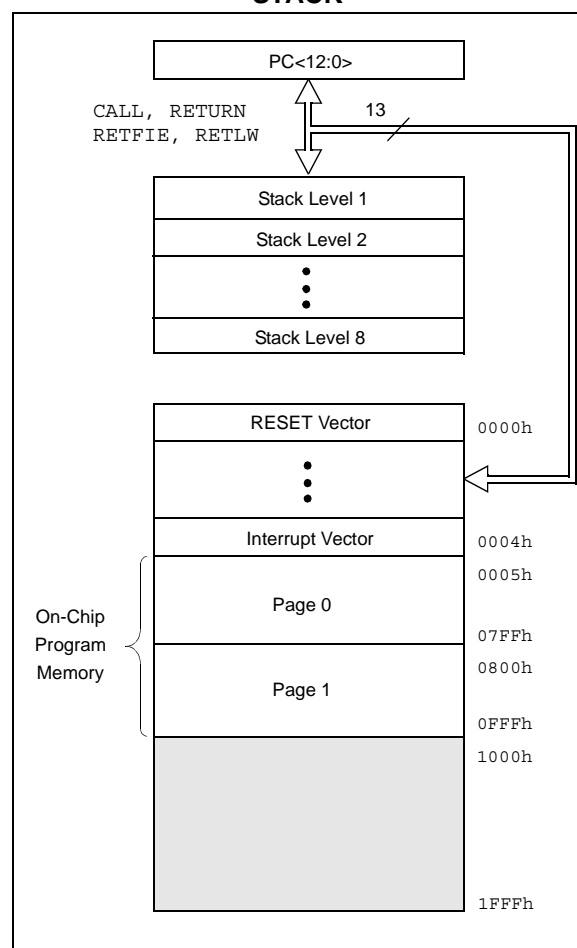


FIGURE 2-2: PIC16F874/873 PROGRAM MEMORY MAP AND STACK



PIC16F87X

2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Note: EEPROM Data Memory description can be found in Section 4.0 of this data sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register (FSR).

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. ^(*) 00h	Indirect addr. ^(*) 80h	Indirect addr. ^(*) 100h	Indirect addr. ^(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h - 7Fh
Bank 0	Bank 1	Bank 2	Bank 3


■ Unimplemented data memory locations, read as '0'.
 * Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
Note 2: These registers are reserved, maintain these registers clear.

PIC16F87X

FIGURE 2-4: PIC16F874/873 REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. ^(*) 00h	Indirect addr. ^(*) 80h	Indirect addr. ^(*) 100h	Indirect addr. ^(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 96 Bytes	accesses 20h-7Fh	accesses A0h - FFh
7Fh	FFh	16Fh 170h	1EFh 1F0h
Bank 0	Bank 1	Bank 2	Bank 3
		17Fh	1FFh

 Unimplemented data memory locations, read as '0'.
 * Not a physical register.

Note 1: These registers are not implemented on the PIC16F873.
Note 2: These registers are reserved, maintain these registers clear.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 0											
00h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
01h	TMR0	Timer0 Module Register								xxxx xxxx	47
02h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
03h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxxx	18
04h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	33
08h ⁽⁴⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	35
09h ⁽⁴⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	36
0Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
0Bh ⁽³⁾	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	20
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	22
0Dh	PIR2	—	(5)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	24
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	51
11h	TMR2	Timer2 Module Register								0000 0000	55
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	55
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	70, 73
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	67
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	58
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	96
19h	TXREG	USART Transmit Data Register								0000 0000	99
1Ah	RCREG	USART Receive Data Register								0000 0000	101
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	57
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	57
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	58
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	116
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ \overline{DONE}	—	ADON	0000 00-0	111

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.
- 3:** These registers can be addressed from any bank.
- 4:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.
- 5:** PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

PIC16F87X

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:	
Bank 1												
80h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27	
81h	OPTION_REG	RBP \overline{U}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19	
82h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26	
83h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	18	
84h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27	
85h	TRISA	—	—	PORTA Data Direction Register							--11 1111	29
86h	TRISB	PORTB Data Direction Register								1111 1111	31	
87h	TRISC	PORTC Data Direction Register								1111 1111	33	
88h ⁽⁴⁾	TRISD	PORTD Data Direction Register								1111 1111	35	
89h ⁽⁴⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	37	
8Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						---0 0000	26
8Bh ⁽³⁾	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20	
8Ch	PIE1	PSPIE ⁽²⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	21	
8Dh	PIE2	—	(5)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	23	
8Eh	PCON	—	—	—	—	—	—	\overline{POR}	\overline{BOR}	---- --qq	25	
8Fh	—	Unimplemented								—	—	
90h	—	Unimplemented								—	—	
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	68	
92h	PR2	Timer2 Period Register								1111 1111	55	
93h	SSPADDD	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	73, 74	
94h	SSPSTAT	SMP	CKE	D/ \overline{A}	P	S	R/ \overline{W}	UA	BF	0000 0000	66	
95h	—	Unimplemented								—	—	
96h	—	Unimplemented								—	—	
97h	—	Unimplemented								—	—	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	95	
99h	SPBRG	Baud Rate Generator Register								0000 0000	97	
9Ah	—	Unimplemented								—	—	
9Bh	—	Unimplemented								—	—	
9Ch	—	Unimplemented								—	—	
9Dh	—	Unimplemented								—	—	
9Eh	ADRESL	A/D Result Register Low Byte								xxxxx xxxxx	116	
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	112	

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.

Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 2											
100h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
101h	TMR0	Timer0 Module Register								xxxx xxxx	47
102h ⁽³⁾	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26
103h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	18
104h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
10Bh ⁽³⁾	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	20
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	41
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	41
10Eh	EEDATH	—	—	EEPROM Data Register High Byte					xxxx xxxx	41	
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	41
Bank 3											
180h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
181h	OPTION_REG	RBPV	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	19
182h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
183h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	18
184h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	31
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
18Bh ⁽³⁾	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	20
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	41, 42
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	41
18Eh	—	Reserved maintain clear								0000 0000	—
18Fh	—	Reserved maintain clear								0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

PIC16F87X

2.2.2.1 STATUS Register

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable, therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits, see the "Instruction Set Summary."

Note: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7							bit 0

- bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
 1 = Bank 2, 3 (100h - 1FFh)
 0 = Bank 0, 1 (00h - FFh)
- bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)
 11 = Bank 3 (180h - 1FFh)
 10 = Bank 2 (100h - 17Fh)
 01 = Bank 1 (80h - FFh)
 00 = Bank 0 (00h - 7Fh)
 Each bank is 128 bytes
- bit 4 **$\overline{\text{TO}}$:** Time-out bit
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
 0 = A WDT time-out occurred
- bit 3 **$\overline{\text{PD}}$:** Power-down bit
 1 = After power-up or by the `CLRWDT` instruction
 0 = By execution of the `SLEEP` instruction
- bit 2 **Z:** Zero bit
 1 = The result of an arithmetic or logic operation is zero
 0 = The result of an arithmetic or logic operation is not zero
- bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
 (for borrow, the polarity is reversed)
 1 = A carry-out from the 4th low order bit of the result occurred
 0 = No carry-out from the 4th low order bit of the result
- bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
 1 = A carry-out from the Most Significant bit of the result occurred
 0 = No carry-out from the Most Significant bit of the result occurred
- Note:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high, or low order bit of the source register.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

2.2.2.2 OPTION_REG Register

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2: OPTION_REG REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP $\overline{\text{U}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBP $\overline{\text{U}}$:** PORTB Pull-up Enable bit
 1 = PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG:** Interrupt Edge Select bit
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS:** TMR0 Clock Source Select bit
 1 = Transition on RA4/T0CKI pin
 0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE:** TMR0 Source Edge Select bit
 1 = Increment on high-to-low transition on RA4/T0CKI pin
 0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA:** Prescaler Assignment bit
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0:** Prescaler Rate Select bits

Bit Value TMR0 Rate WDT Rate

000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 - n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

Note: When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device

PIC16F87X

2.2.2.3 INTCON Register

The INTCON Register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

bit 7	GIE: Global Interrupt Enable bit 1 = Enables all unmasked interrupts 0 = Disables all interrupts
bit 6	PEIE: Peripheral Interrupt Enable bit 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts
bit 5	TOIE: TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 interrupt 0 = Disables the TMR0 interrupt
bit 4	INTE: RB0/INT External Interrupt Enable bit 1 = Enables the RB0/INT external interrupt 0 = Disables the RB0/INT external interrupt
bit 3	RBIE: RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt
bit 2	TOIF: TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	INTF: RB0/INT External Interrupt Flag bit 1 = The RB0/INT external interrupt occurred (must be cleared in software) 0 = The RB0/INT external interrupt did not occur
bit 0	RBIF: RB Port Change Interrupt Flag bit 1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software). 0 = None of the RB7:RB4 pins have changed state

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

DS1307

64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, Nonvolatile (NV) RAM for Data Storage
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratory (UL) Recognized

Typical Operating Circuit and Pin Configurations appear at end of data sheet.

ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z/T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN/T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+ Denotes a lead-free/RoHS-compliant device.

* A "+" anywhere on the top mark indicates a lead-free device.

Note: Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, click here: www.maxim-ic.com/errata.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground.....-0.5V to +7.0V

Operating Temperature Range (Noncondensing)

Commercial.....0°C to +70°C

Industrial.....-40°C to +85°C

Storage Temperature Range.....-55°C to +125°C

Soldering Temperature (DIP, leads).....+260°C for 10 seconds

Soldering Temperature (surface mount).....See JPC/JEDEC Standard J-STD-020

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V _{CC}		4.5	5.0	5.5	V
Logic 1 Input	V _{IH}		2.2		V _{CC} + 0.3	V
Logic 0 Input	V _{IL}		-0.3		+0.8	V
V _{BAT} Battery Voltage	V _{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I _{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I _{LO}		-1		1	μA
Logic 0 Output (I _{OL} = 5mA)	V _{OL}				0.4	V
Active Supply Current (f _{SCL} = 100kHz)	I _{CCA}				1.5	mA
Standby Current	I _{CCS}	(Note 3)			200	μA
V _{BAT} Leakage Current	I _{BATLKG}			5	50	nA
Power-Fail Voltage (V _{BAT} = 3.0V)	V _{PF}		1.216 x V _{BAT}	1.25 x V _{BAT}	1.284 x V _{BAT}	V

DC ELECTRICAL CHARACTERISTICS(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{BAT} Current (OSC ON); SQW/OUT OFF	I _{BAT1}			300	500	nA
V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I _{BAT2}			480	800	nA
V _{BAT} Data-Retention Current (Oscillator Off)	I _{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t _{BUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HD:STA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SU:STA}		4.7			μs
Data Hold Time	t _{HD:DAT}		0			μs
Data Setup Time	t _{SU:DAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _R				1000	ns
Fall Time of Both SDA and SCL Signals	t _F				300	ns
Setup Time for STOP Condition	t _{SU:STO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{IO}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{IH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{HD:DAT} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

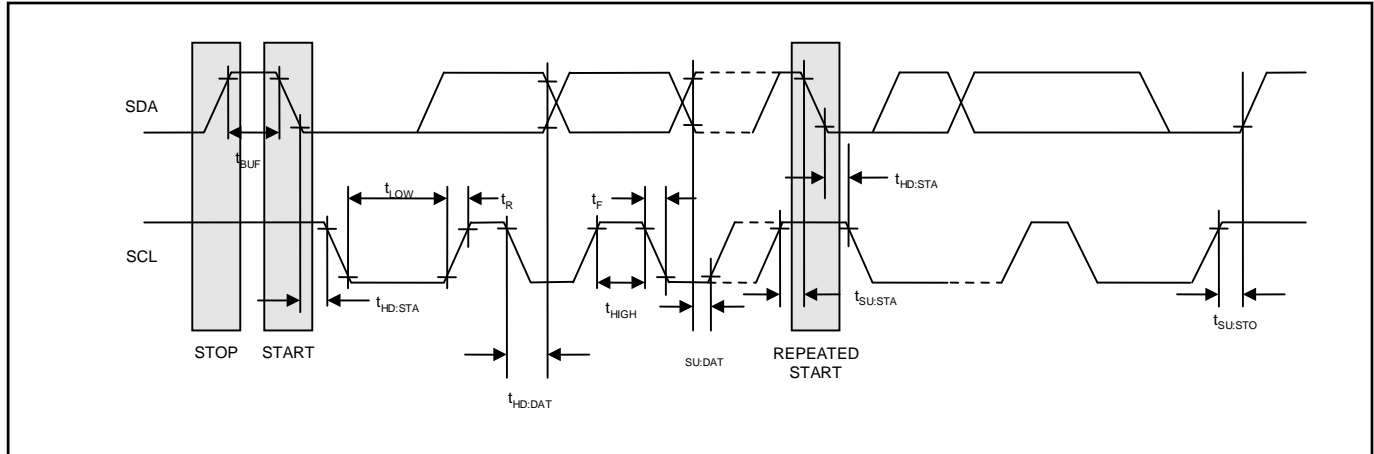
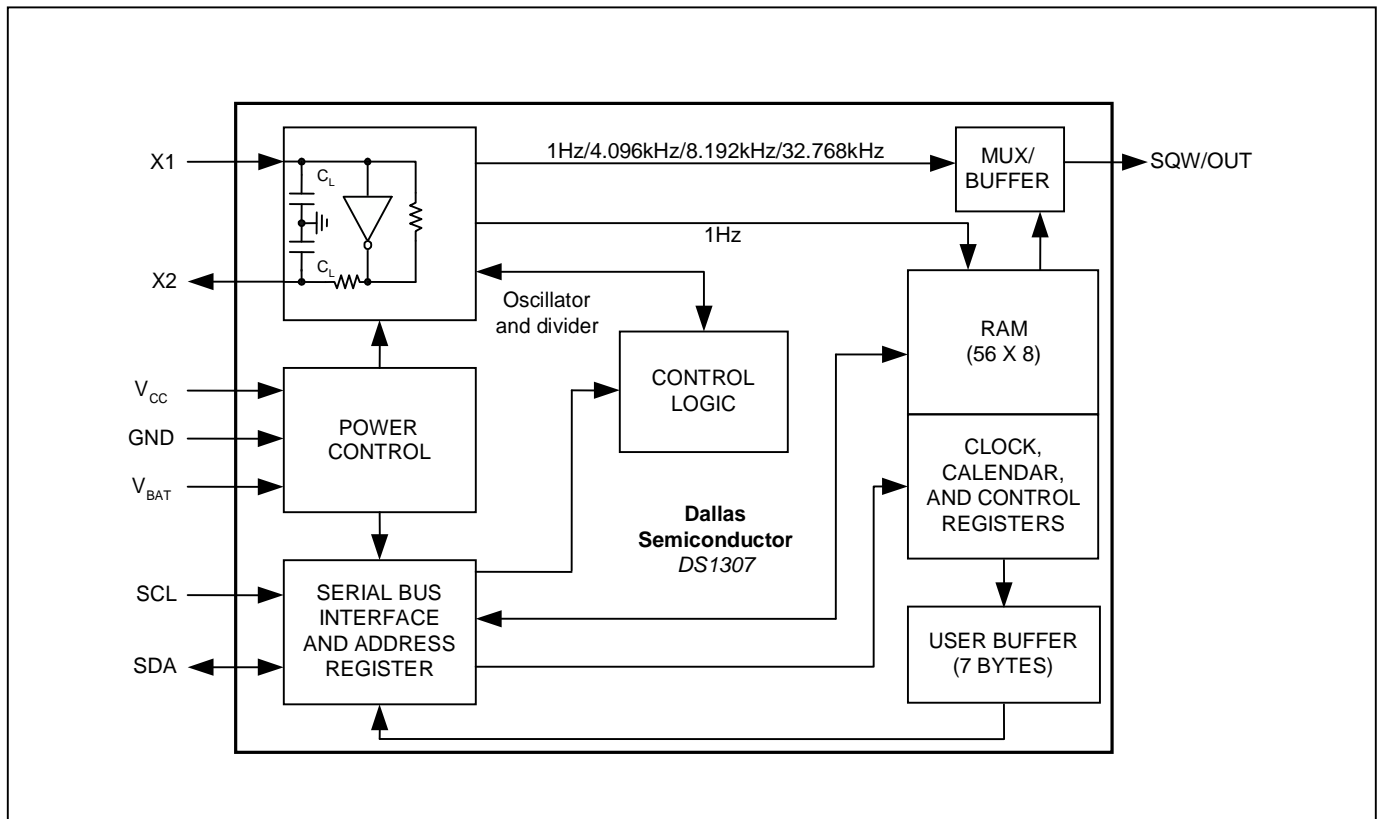
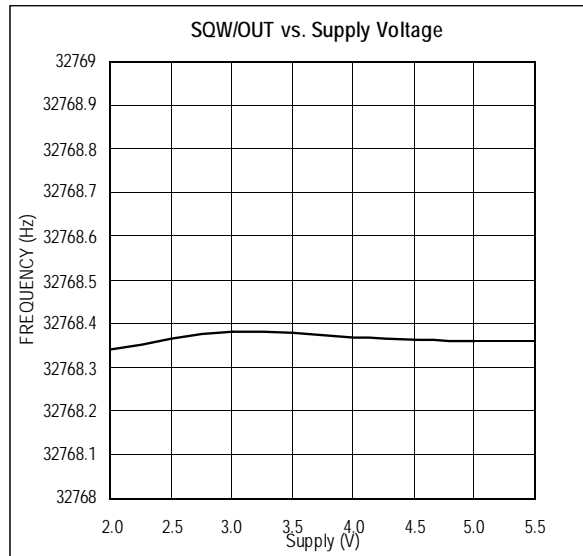
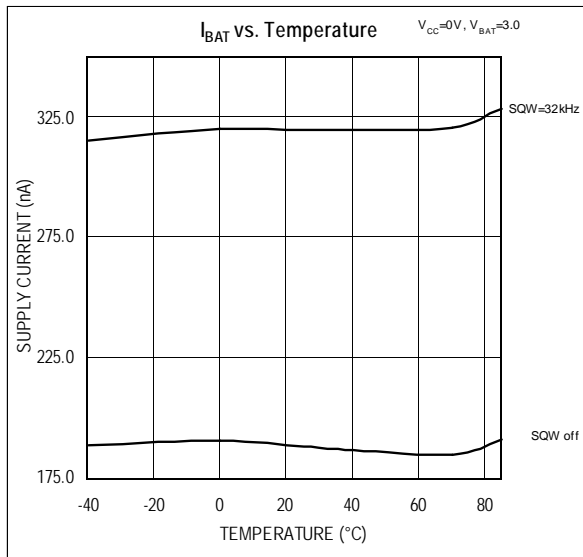
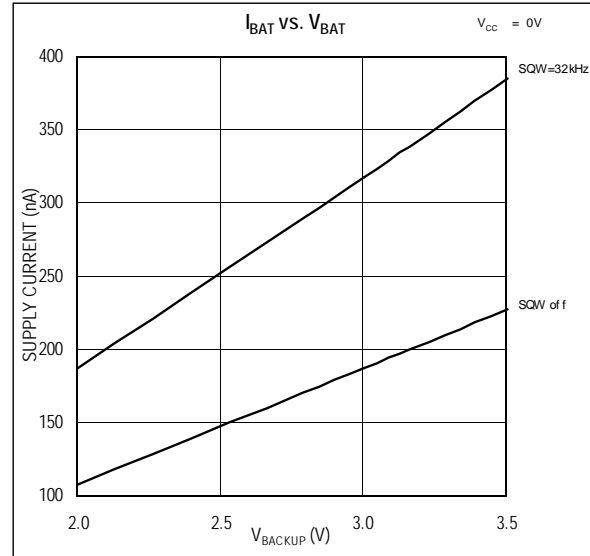
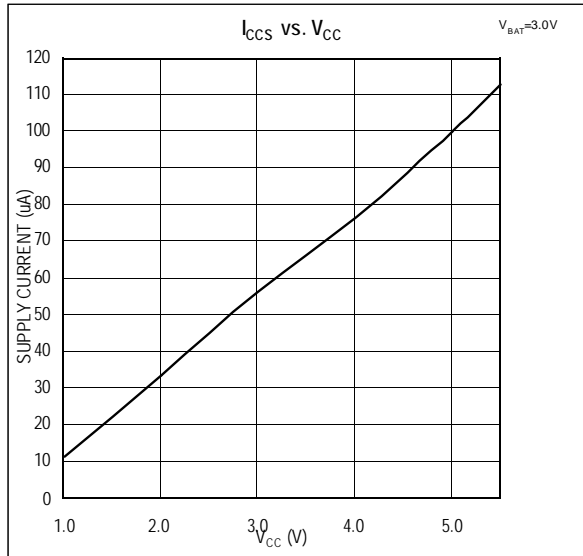


Figure 1. Block Diagram



TYPICAL OPERATING CHARACTERISTICS(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)

PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1. Note: For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
2	X2	
3	V _{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V _{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/ .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor.
6	SCL	Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface.
7	SWQ/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied.
8	V _{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{TP} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT}+0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1. shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

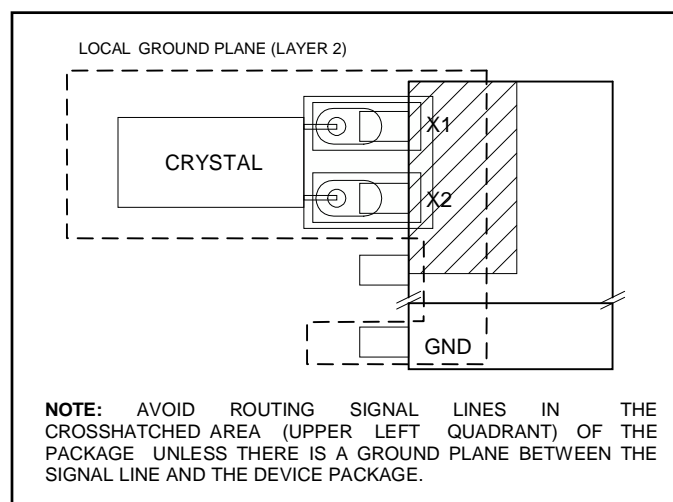
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to *Application Note 58: Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_0		32.768		kHz
Series Resistance	ESR			45	k Ω
Load Capacitance	C_L		12.5		pF

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: Crystal Considerations for Dallas Real-Time Clocks for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled.

Note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds				Seconds	00–59
01H	0	10 Minutes			Minutes				Minutes	00–59
02H	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03H	0	0	0	0	0	DAY			Day	01–07
04H	0	0	10 Date		Date				Date	01–31
05H	0	0	0	10 Month	Month				Month	01–12
06H	10 Year				Year				Year	00–99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08H-3FH									RAM 56 x 8	00H–FFH

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave.

Bits 1, 0: Rate Select (RS1, RS0). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

I²C DATA BUS

The DS1307 supports the I²C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I²C bus.

Figures 3, 4, and 5 detail how data is transferred on the I²C bus.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

Start data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

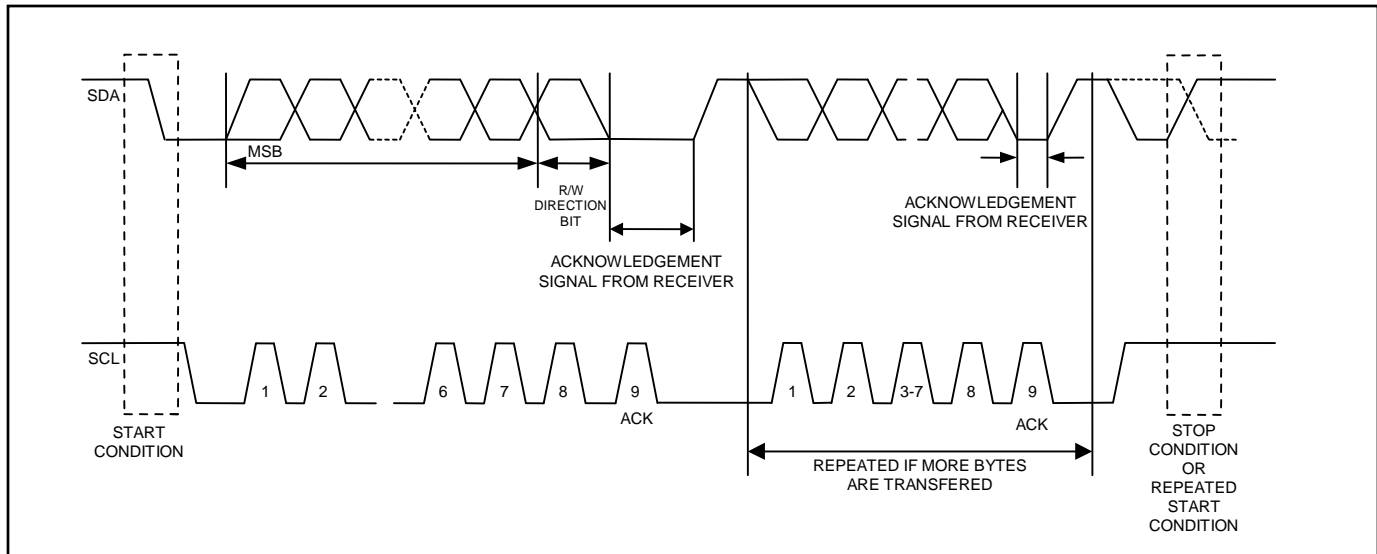
Stop data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I²C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I²C Serial Bus

Depending upon the state of the R/W bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 may operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/W), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/W), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

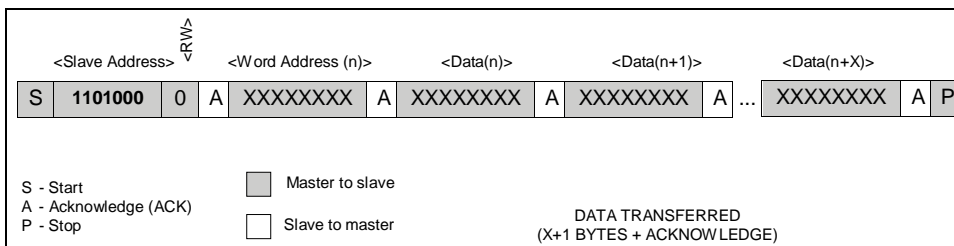


Figure 5. Data Read—Slave Transmitter Mode

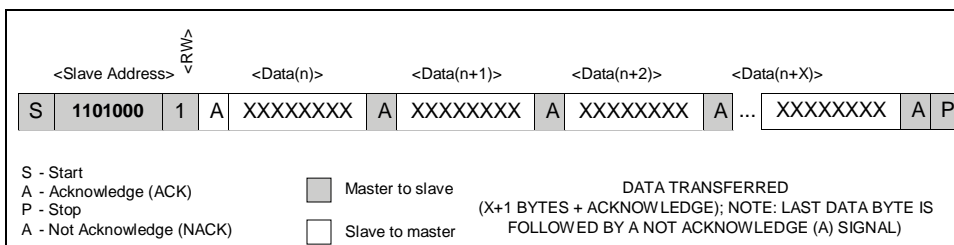
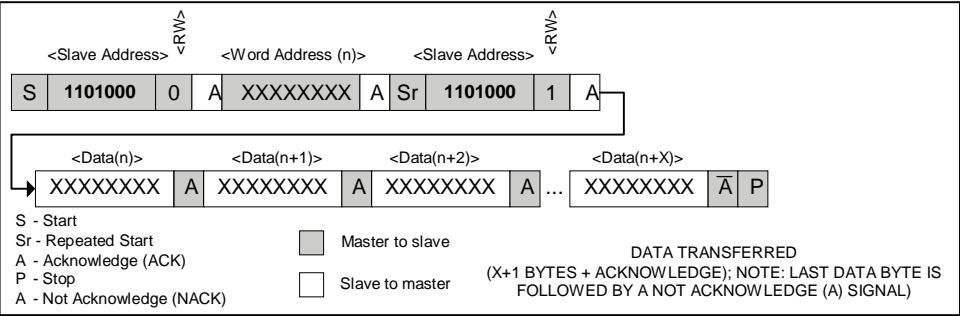
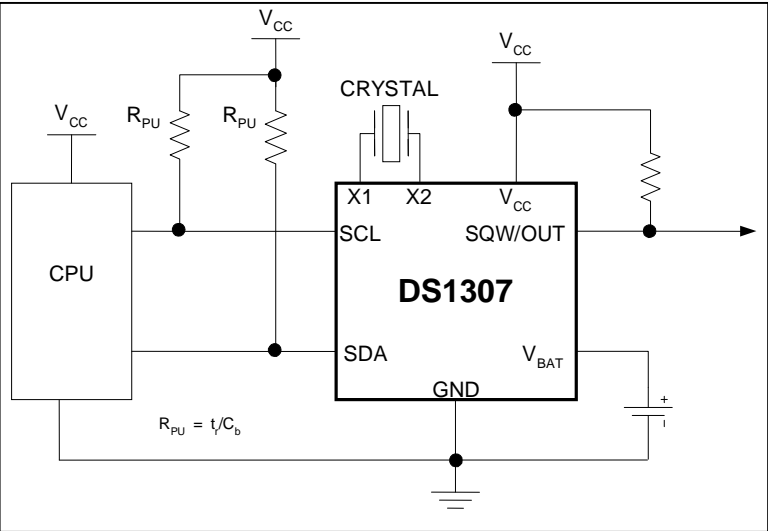


Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit



TYPICAL OPERATING CIRCUIT

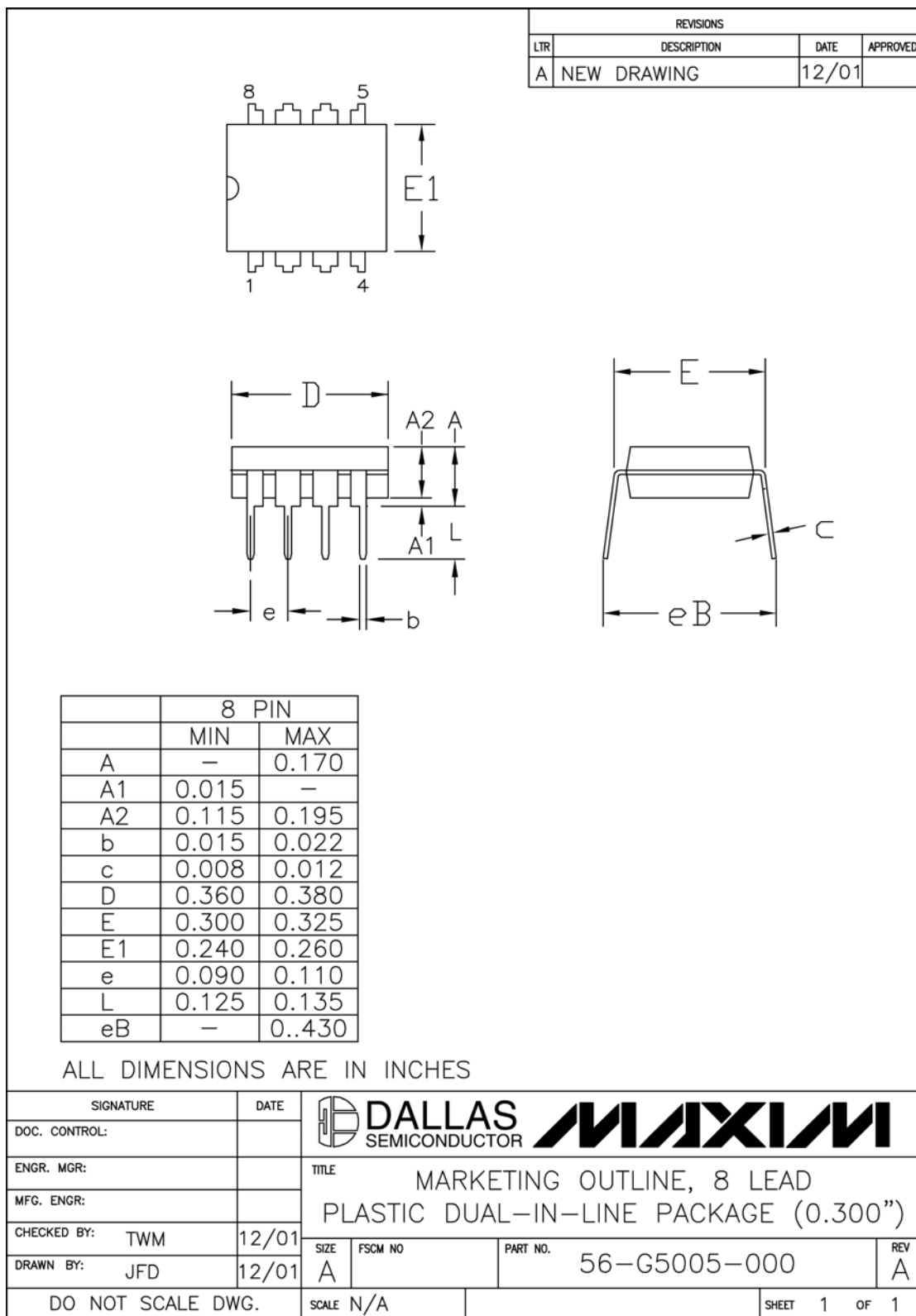


PIN CONFIGURATIONS



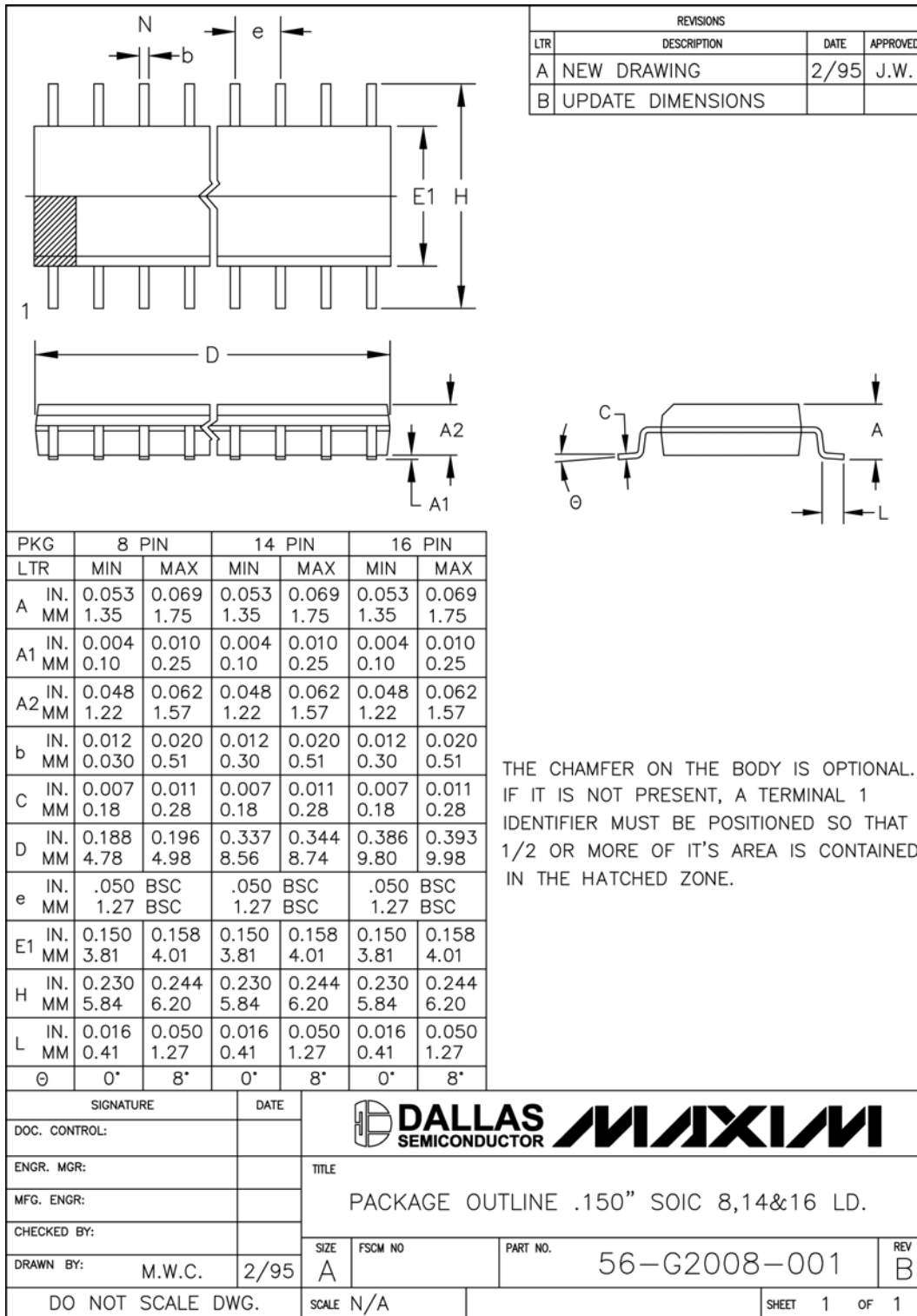
PACKAGE INFORMATION

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to www.maxim-ic.com/DallasPackInfo.)



PACKAGE INFORMATION (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to www.maxim-ic.com/DallasPackInfo.)



Maxim/Dallas Semiconductor cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim/Dallas Semiconductor product. No circuit patent licenses are implied. Maxim/Dallas Semiconductor reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600

© 2006 Maxim Integrated Products

The Maxim logo is a registered trademark of Maxim Integrated Products, Inc. The Dallas logo is a registered trademark of Dallas Semiconductor Corporation.

LM016L·LM016XMBL

- 16 character x 2 lines
- Controller LSI HD44780 is built-in (See page 79).
- +5V single power supply
- Display color: LM016L : Gray
LM016XMBL : New-gray

MECHANICAL DATA (Nominal dimensions)

Module size	84W x 44H x 10.5T (max.) mm
Effective display area	61W x 15.8H mm
Character size (5 x 7 dots)	2.96W x 4.86H mm
Character pitch	3.55 mm
Dot size	0.56W x 0.66H mm
Weight	about 35 g

ABSOLUTE MAXIMUM RATINGS

	min.	max.
Power supply for logic ($V_{DD}-V_{SS}$)	0	6.5 V
Power supply for LCD drive ($V_{DD}-V_O$)	0	6.5 V
Input voltage (V_i)	V_{SS}	V_{DD} V
Operating temperature (T_a)	0	50 40*°C
Storage temperature (T_{stg})	-20	70 60*°C

* Shows the value of type LM016XMBL.

ELECTRICAL CHARACTERISTICS

$T_a = 25^\circ\text{C}$, $V_{DD} = 5.0 \text{ V} \pm 0.25 \text{ V}$

Input "high" voltage (V_{IH})	2.2 V min.
Input "low" voltage (V_{IL})	0.6 V max.
Output "high" voltage (V_{OH}) ($-I_{OH} = 0.2 \text{ mA}$)	2.4 V min.
Output "low" voltage (V_{OL}) ($I_{OL} = 1.2 \text{ mA}$)	0.4 V max.
Power supply current (I_{DD}) ($V_{DD} = 5.0 \text{ V}$)	1.0 mA typ. 3.0 mA max.

POWER SUPPLY FOR LCD DRIVE (Recommended) ($V_{DD}-V_O$)

	Duty = 1/16
Range of $V_{DD}-V_O$	1.5~5.25 V
$T_a = 0^\circ\text{C}$	4.6 V typ.
$T_a = 25^\circ\text{C}$	4.4 V typ.
$T_a = 50^\circ\text{C}$	4.2 V typ.

OPTICAL DATA See page 7

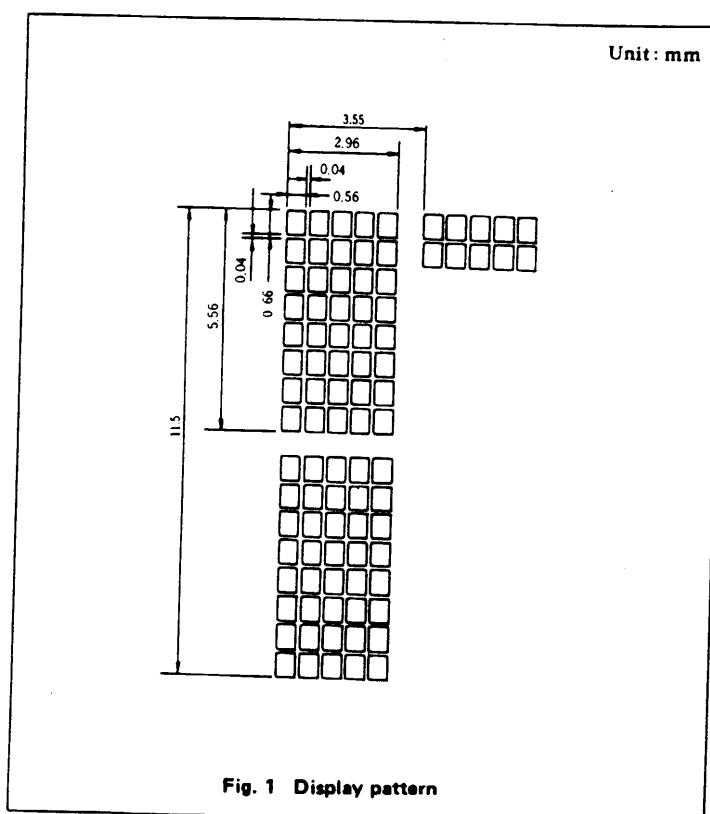
INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function
1	V_{SS}	—	0V
2	V_{DD}	—	+5V
3	V_O	—	—
4	RS	H/L	L: Instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module→MPU) L: Data write (LCD module←MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	

Notes:

In the HD44780, the data can be sent in either 4-bit 2-operation or 8-bit 1-operation so that it can interface to both 4 and 8 bit MPU's.

- (1) When interface data is 4 bits long, data is transferred using only 4 buses of $DB_4 \sim DB_7$ and $DB_0 \sim DB_3$ are not used. Data transfer between the HD44780 and the MPU completes when 4-bit data is transferred twice. Data of the higher order 4 bits (contents of $DB_4 \sim DB_7$ when interface data is 8 bits long) is transferred first and then lower order 4 bits (contents of $DB_0 \sim DB_3$ when interface data is 8 bits long).
- (2) When interface data is 8 bits long, data is transferred using 8 data buses of $DB_0 \sim DB_7$.



Unit: mm

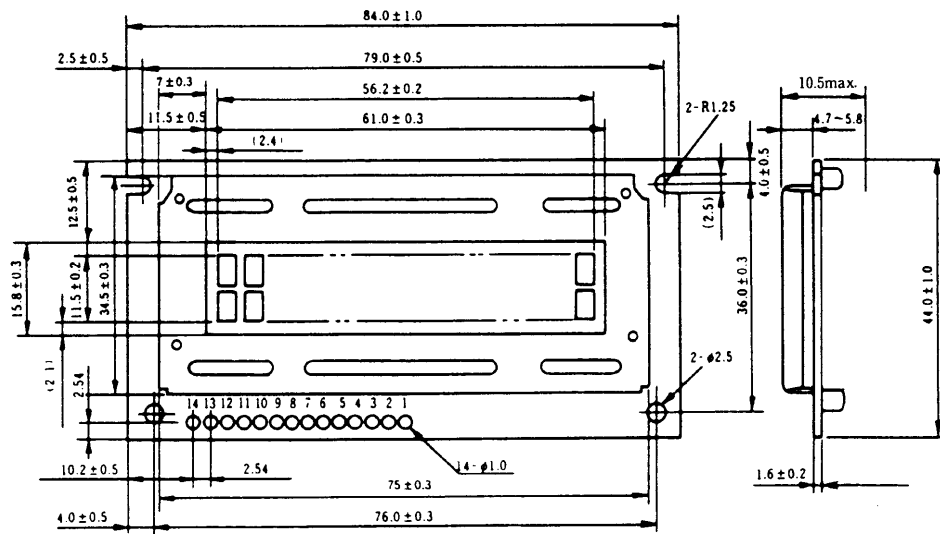


Fig. 2 External dimensions

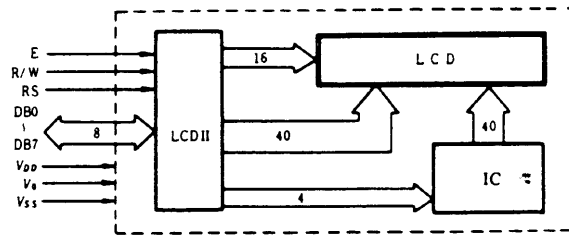


Fig. 3 Block diagram

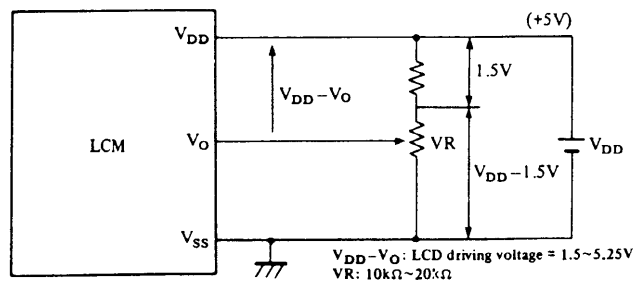
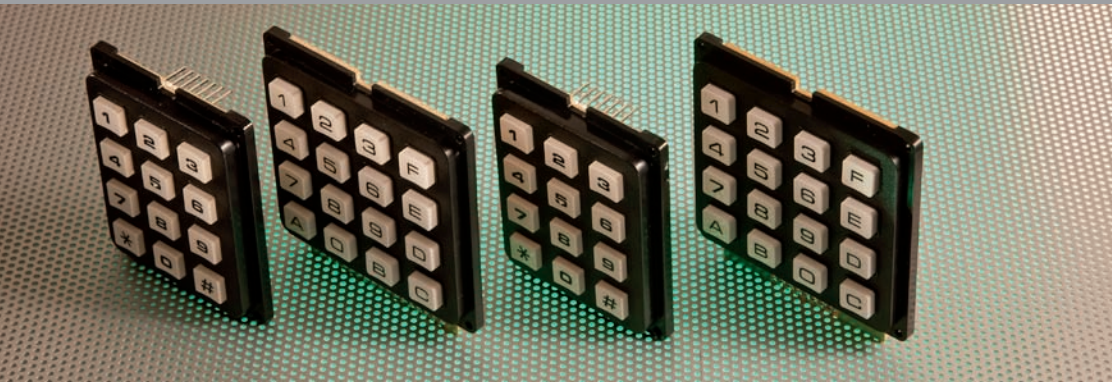


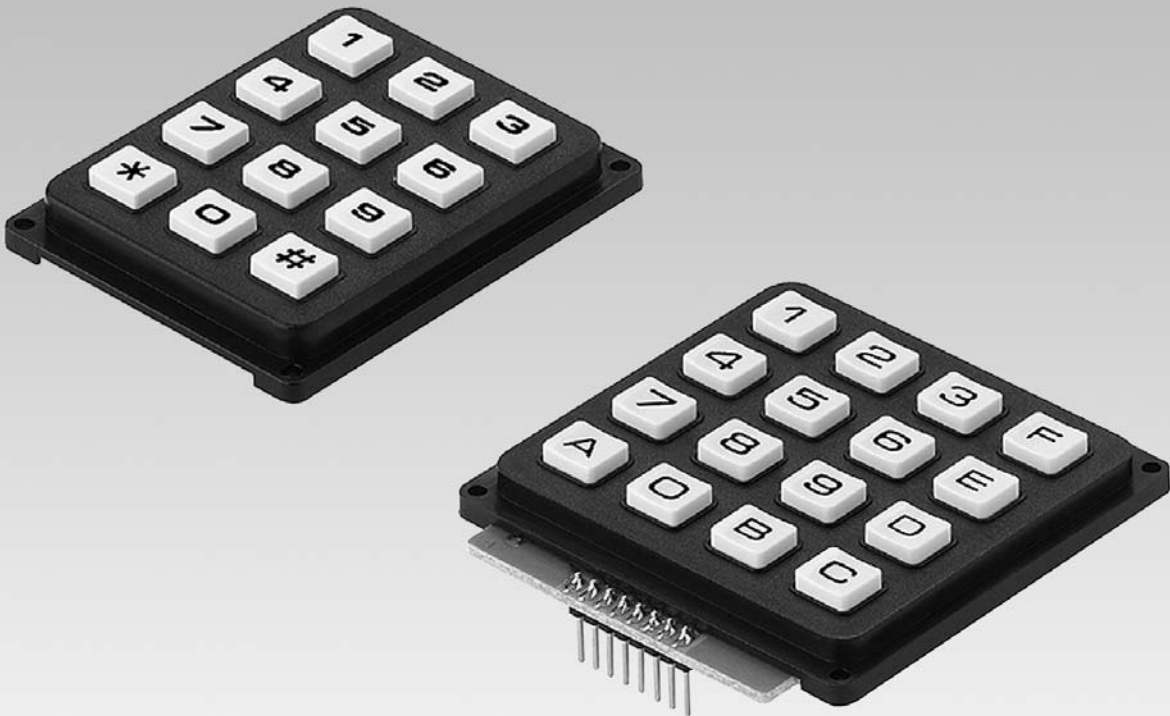
Fig. 4 Power supply

EAO – Your Expert Partner for
Human Machine Interfaces



EAO Product Information

ECO



Description 3

Keypads 4

Technical Data..... 5

Drawings..... 6

Index..... 7

Product Information

General notes

The EAO ECO keypads are suited to all indoor applications including data-entry systems, remote controls, telephone, point of sales terminals or alarm systems.

The ECO range of low-cost, flush-mount keypads are rugged, economical devices available in 12-key telephone style and 16-key hexadecimal II layouts.

Mounting

Mounting is from the back of a panel using fixed studs.

Protection degree of the keypad is IP 40 (front side).

Contacts

To ensure integrity of contacts and lower switch ratings, the contacts are gilded and external connection to the keypad is by means of a pin header on the back.

The electrical keypad circuit can be supplied in a choice of either matrix or common point configuration.

Keys

The keys are made of polycarbonate.

Marking




Standard markings are hot stamped. On request customized symbols and markings are available.

We reserve the right to modify technical data

All dimensions in mm

Keypad



	Front protection	Terminals	Key cap	No. of keys	Marking	Circuit	 60 x 57 Typ-Nr.	 46 x 57 Typ-Nr.	Component layout	Technical drawing	
Keypad Keys for indoor use	IP 40	PH	Plastic white	16	Hexadecimal II	P	ECO.16200.06		2	2	0.024
						M	ECO.16250.06		2	2	0.024
				12	Telephone	P		ECO.12100.06	1	1	0.020
						M		ECO.12150.06	1	1	0.020

Packaging of 10 pcs.
Terminals: PH = Pin header
Circuit: P = Common point, M = Matrix
Component layout from page 6, Technical drawing from page 6

Keypad

Material

Keys

Polycarbonate (PC)

Housing

Polycarbonate (PC)

Contacts

Carbon/Gold

Mechanical characteristics

Actuating force

1.2 N \pm 35 %

Actuating travel

1.4 mm \pm 0.1 N

Rebound time

\leq 2 ms

Marking height

3.5 mm

Electrical characteristics

Operating voltage/-current

Nominal 24 V, 20 mA

Maximum voltage 24 V

Minimum voltage 500 mV

Minimum current 10 mA

Contact resistance

$<200 \Omega$ of electric circuit

Isolation resistance

$>1000 \text{ M}\Omega$ at 100 VDC

Life time

>1 Million cycles operation per key at nominal break rating

Switch rating

0.5 W

ESD-protection

5 kV

Environmental conditions

Storage temperature

-40 °C ... +65 °C

Operating temperature

-20 °C ... +60 °C

Approvals

Declaration of conformity

CE

Component layout

1 Keypad page 4

12 KEYS – CONNECTOR TERMINALS
(rear view)

Matrix XY

Common point

CONTACTS

CONTACTS

PC = Common point

Connection by female connector HE 14
(not supplied)

X1	X2	X3	
Y1	1	2	3
Y2	4	5	6
Y3	7	8	9
Y4	*	0	#

2 Keypad page 4

16 KEYS – CONNECTOR TERMINALS
(rear view)

Matrix XY

Common point

CONTACTS

CONTACTS

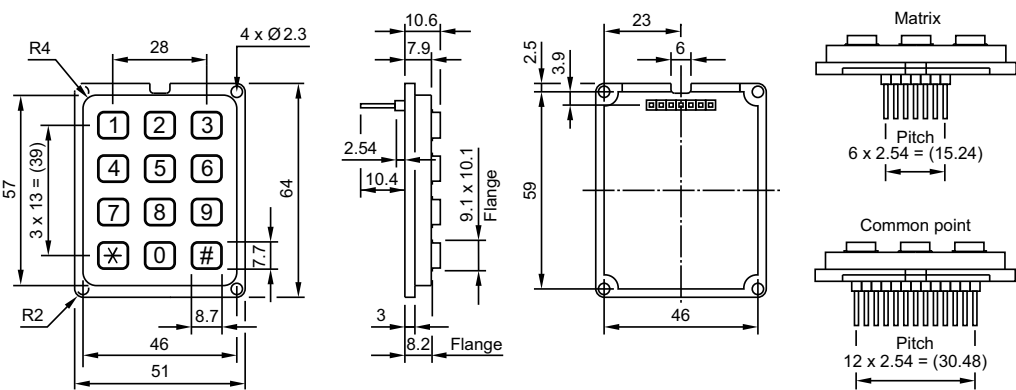
PC = Common point

Connection by female connector HE 14
(not supplied)

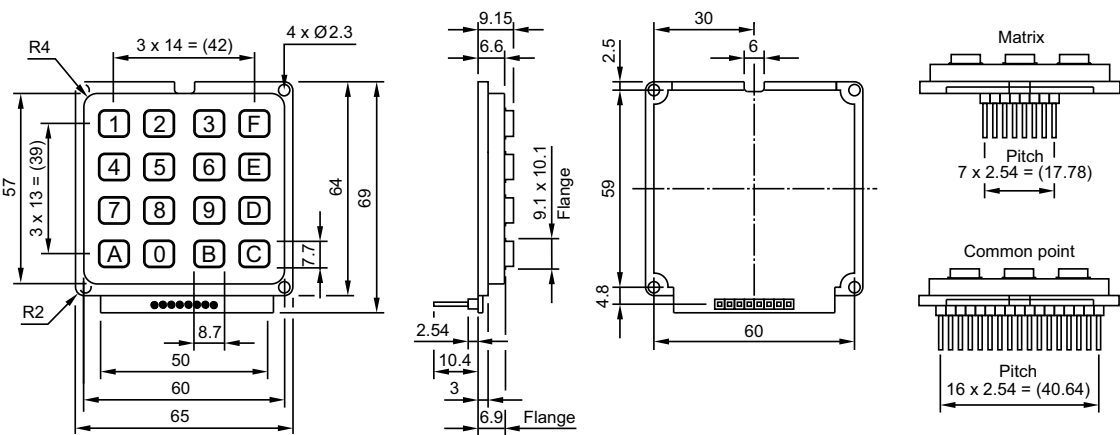
X1	X2	X3	X4	
Y1	1	2	3	F
Y2	4	5	6	E
Y3	7	8	9	D
Y4	A	0	B	C

Technical drawing

1 Keypad page 4



2 Keypad page 4



Index from Typ-Nr.

Typ-Nr.	Page	Typ-Nr.	Page	Typ-Nr.	Page
ECO.12100.06	4				
ECO.12150.06	4				
ECO.16200.06	4				
ECO.16250.06	4				

	EAO AG
	Tannwaldstrasse 88
	4601 Olten, Switzerland
E-mail	info@eao.com
Website	www.eao.com
	Austria
Phone	+49 201 85 87 0
Fax	+49 201 85 87 210
E-mail	sales.ede@eao.com
	Belgium
Phone	+32 3 777 82 36
Fax	+32 3 777 84 19
E-mail	sales.ebl@eao.com
	China
Phone	+852 27 86 91 41
Fax	+852 27 86 95 61
E-mail	sales.ehk@eao.com
	France
Phone	+33 1 64 43 37 37
Fax	+33 1 64 43 37 49
E-mail	sales.esa@eao.com
	Germany
Phone	+49 201 85 87 0
Fax	+49 201 85 87 210
E-mail	sales.ede@eao.com
	Italy
Phone	+39 035 481 0189
Fax	+39 035 481 3786
E-mail	sales.eit@eao.com
	Japan
Phone	+81 3 5444 5411
Fax	+81 3 5444 0345
E-mail	sales.esj@eao.com
	Netherlands
Phone	+31 78 653 17 00
Fax	+31 78 653 17 99
E-mail	sales.enl@eao.com
	Sweden
Phone	+46 8 683 86 60
Fax	+46 8 724 29 12
E-mail	sales.esw@eao.com
	Switzerland
Phone	+41 62 388 95 00
Fax	+41 62 388 95 55
E-mail	sales.ech@eao.com
	United Kingdom
Phone	+44 1444 236 000
Fax	+44 1444 236 641
E-mail	sales.euk@eao.com
	USA
Phone	+1 203 877 4577
Fax	+1 203 877 3694
E-mail	sales.eus@eao.com
	Other Countries
Phone	+41 62 286 92 10
Fax	+41 62 296 21 62
E-mail	info@eao.com

5 - 29 Volts continu
2,1A à 29V / 2,5A à 24V
3,5A à 12V / 4A à 5V

PRECISE : ondulation < 3mV efficace.
UNIVERSELLE : 12 positions au pas de 2V avec réglage fin de $\pm 1V$.
COMPLETE : Fonction chargeur de batterie au plomb 12 ou 24V.
PRATIQUE : Interrupteur M/A, Témoins de fonctionnement et de positions chargeur.
PROTEGEE : Contre les courts-circuits et les inversions de polarité.

60 watts



PRECISE : Switching power supply offering a ripple <3mV rms.
UNIVERSAL : 12 settings in 2V steps with $\pm 1V$ adjustment range.
COMPLETE : 12 or 24V lead-acid battery charger function.
PRACTICAL : ON/OFF switch, charger position and status indicators.
PROTECTED : against short circuits and reverse polarity.



FESTGELEGT : Unterbrechungsfreie Versorgung mit Welligkeit < 3mV effektiv.
UNIVERSELL : 12 Positionen in Schritten von 2V mit Einstellung von $\pm 1V$.
VOLLSTÄNDIG : Ein-/Ausschalter, Funktion Ladegerät für 12 und 24 V Bleibatterien.
PRAXIS : Kontrollleuchten für Betrieb und Positionen des Ladegeräts.
GESCHÜTZT : Schutz gegen Kurzschlüsse und falsche Polung.



Caractéristiques techniques

Tension

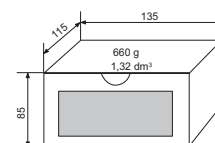
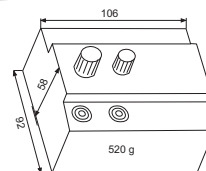
- Sorties flottantes sur douilles de sécurité de 4mm.
- Tension de sortie : ajustable de 5 à 29V par commutateur à 12 positions et réglage fin. positions du commutateur : 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 Volts. variation du réglage fin : ± 1 Volt quelle que soit la position du commutateur. positions chargeur de batterie au plomb 12 et 24V repérées par deux LED
- Précision : $\pm 1\%$
- Régulation : < 1mV à 5V 4A et < 5mV à 29V 2,1A pour une variation de charge de 0 à 100%.
 < 1mV à 29V 2,1A et < 4mV à 5V 4A pour une variation secteur de 190V à 253V.
- Ondulation : < 3mV efficace comprenant :
 < 3mV crête à crête du signal à 100KHz
 < 4mV crête à crête du signal à 100Hz
 < 10mV crête à crête de pics de commutations
- Résistance interne : < 3m Ω
- Temps de maintien : 25ms à 50% de charge et 12ms à 100% (secteur à 190V)
- Visualisation : Led verte "alimentation en fonctionnement"
 Leds jaune "position chargeur de batterie Pb 12V et 24V"
 Led rouge "status, fusible de sortie coupé"

Intensité

- I maxi : 4,2A au court-circuit
 4A à 5V
 3,5 A à 12V
 2,5A à 24V
 2,1A à 29V

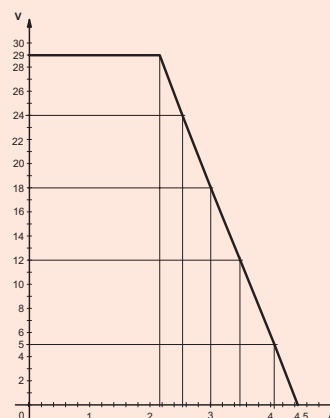
Chargeur de batterie

- Capacité nominale des batteries au plomb à électrolyte libre (charge 14h) : 35Ah pour les 12V et 20Ah pour les 24V.
 - Capacité minimale des batteries au plomb scellées (I charge = 1/3 I nominal) : 10Ah pour les 12V et 7Ah pour les 24V.
- (Dans tous les cas, se reporter à la notice du fabricant des batteries)



Puissance

- Varie linéairement de 60W à 20W en fonction de la tension (29 à 5 Volts).



Protections

- Contre les courts-circuits par limitation de courant.
- Contre les surintensités sur la source, par fusible.
- Contre les inversions de polarité batterie, par fusible en sortie.

Autres caractéristiques

- Sécurité : Classe II, double isolation, conforme à la norme **EN 61010-1** et **EN 60335-2-29**.
- CEM : Conforme aux normes **EN 61326-1**, critère d'aptitude B, **EN 55011**, ISM Groupe I, Classe B.
- Indice de protection : IP 30.
- Alimentation : Secteur 190 à 253 Volts, 50 / 60Hz.
- Entrée secteur : cordon 2 pôles double isolation.
- Consommation : 71W maxi.
- Mise sous tension : Interrupteur à bascule.
- Rigidité diélectrique : 3000V entre entrée et sortie.
- Présentation : Boîtier métal avec peinture époxy.



Specifications

Voltage

- Floating outputs on 4 mm safety sockets.
- Output voltage :
adjustable from 5 to 29V by 12-position switch, and fine adjustment switch positions : 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 Volts
fine adjustment range : ± 1 Volt, whatever the switch setting
12 and 24V lead-acid battery charger positions identified by two LED indicators
- Accuracy : $\pm 1\%$
- Regulation : $< 1\text{mV}$ at 5V 4A and $< 5\text{mV}$ at 29V 2,1A for a load change from 0 to 100%.
 $< 1\text{mV}$ at 29V 2,1A and $< 4\text{mV}$ at 5V 4A for a line change from 190V to 253V.
- Ripple : $< 3\text{mV}$ rms including:
 $< 3\text{mV}$ peak to peak of the 100kHz signal
 $< 4\text{mV}$ peak to peak of the 100kHz signal
 $< 10\text{mV}$ peak to peak of switching transients
- Internal resistance : $< 3\text{m}\Omega$.
- Hold-up time : 25 ms at half load and 12 ms at full load. (190V line input)
- Indicators : Green LED indicator: "power supply operating"
Yellow LED indicator: "12V and 24V charger position"
Red LED indicator: "Status, output fuse broken"

Current

- Max I : 4,2A in short circuit condition
4A to 5V
3,5A to 12V
2,5A to 24V
2,1A to 29V

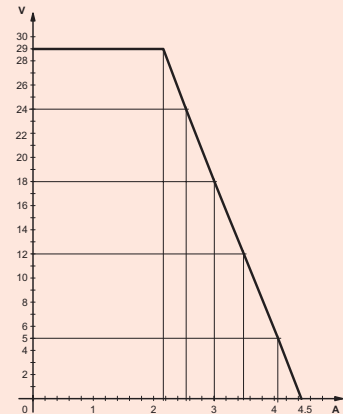
Charger of the battery

- Rated capacity of the lead-acid batteries with electrolyte free: 35 Ah for 12 V and 20 Ah for 24 V.
- Minimum capacity of the lead-acid batteries sealed: 10 Ah for 12 V and 7 Ah for the 24V.

(In all the cases, to refer to the note of the batteries manufacturer)

Power

- A linear function of voltage from 60W to 20W (29 to 5 Volts).



Protection

- Short circuit protection, by current limiting.
- Transformer primary overcurrent protection, by fuse.
- Battery reverse polarity protection by output fuse.

Other specifications

- Safety : Classe II, double insulation, according to **EN 61010-1** and **EN 60335-2-29**.
- EMC : Complies with **EN 61326-1**, performance criteria B, and **EN 55011**, ISM Group I, Class B.
- Protection level : IP 30.
- Input voltage : 190 to 253 Volts, 50 / 60 Hz.
- Mains input : double insulation 2 poles cable.
- Power consumption : 71 W max.
- ON/OFF control : toggle switch
- Dielectric strength : 3000V.
- Presentation : metal case with epoxy finish.



Technische Daten

Spannung

- Ausgänge von Masse getrennt (floating) auf 4-mm-Schutzbuchsen
- Ausgangsspannung :
einstellbar zwischen 5 und 29V durch Schalter mit 12 Positionen und Feineinstellung
Schalterpositionen : 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 Volt
Feineinstellbereich : ± 1 Volt, unabhängig von der Schalterposition
Position des 12- und 24-V-Batterieladegeräts an zwei LED ablesbar
- Genauigkeit : $\pm 1\%$
- Regelung : $< 1\text{mV}$ bei 5V 4A und $< 5\text{mV}$ bei 29V 2,1A bei Laständerungen von 0 bis 100%.
 $< 1\text{mV}$ bei 29V 2,1A und $< 4\text{mV}$ bei 5V 4A bei Schwankungen der Netzversorgung zwischen 190V und 253V.
- Welligkeit : $< 3\text{mV}$ effektiv mit:
 $< 3\text{mV}$ Spitze-Spitze des Signals bei 100kHz
 $< 4\text{mV}$ Spitze-Spitze des Signals bei 100Hz
 $< 10\text{mV}$ Spitze-Spitze von Schaltspitzen
- Innenwiderstand : $< 3\text{m}\Omega$
- Haltezeit: 25ms bei 50% der Last und 12ms bei 100% (Netzversorgung bei 190V)
- Anzeige : Grüne LED "Versorgung bei Betrieb"
Gelbe LED "Bleibatterie-Ladegerät, Position 12V"
Gelbe LED "Bleibatterie-Ladegerät, Position 24V"

Stromstärke

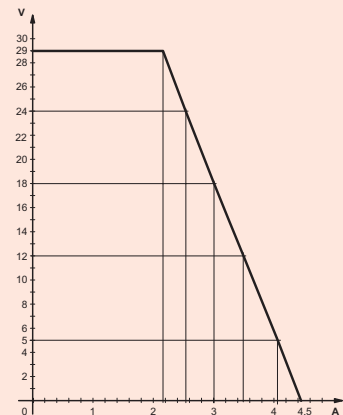
- I max : 4,2A bei Kurzschluss
4A bei 5V
3,5A bei 12V
2,5A bei 24V
2,1A bei 29V

Ladegerät der Batterien

- Nominalkapazität der Bleiakкумуляtoren an electrolyte freiem : 35 Ah für die 12 V und 20 Ah für die 24 V.
 - Minimale Kapazität der versiegelten Bleiakкумуляtoren : 10 Ah für die 12 V und 7 Ah für die 24V.
- (In allen Fällen sich auf die Notiz des Herstellers der Batterien zu beziehen)

Leistung

- Lineare Änderung von 60W bis 20W in Abhängigkeit von der Spannung (29 bis 5 Volt).



Schutzvorrichtungen

- Gegen Kurzschlüsse durch Strombegrenzung.
- Gegen Überströme auf dem Primärkreis des Transformators durch Sicherung.
- Gegen falsche Polung der Batterie durch Sicherung am Ausgang.

Andere Eigenschaften

- Schutz : Klasse II, schutzisoliert, entspricht den Normen **EN 61010-1** und **EN 60335-2-29**.
- EMC : Entspricht den Normen **EN 61326-1**, Eignungskriterium B, und **EN 55011**, ISM Gruppe I, Klasse B.
- Schutzart : IP 30.
- Versorgung: Netzversorgung 190 bis 253 Volt, 50 / 60 Hz.
- Netzversorgungseingang : schutzisoliertes 2-Phasen-Netzkabel.
- Leistungsaufnahme : max. 71W.
- Einschaltung : Kippschalter.
- Durchschlagsfestigkeit : 3000V.
- Erscheinungsbild : Metallgehäuse mit Epoxid-Lackierung.

Fig.6 PCB Position v Range @ 2.9m (Ground strike)

Position	Down Angle	Range
1.	5°	15m+
2.	10°	15m
3.	15°	8m
4.	20°	6m
5.	25°	4m

DIL Switch Functions

Control of the unit is through a bank of six DIL switches situated on the rear of the PCB. The switches are marked 1-6 (see Fig 7).

DIL Switch 1 Shadow Algorithm System (SAS) Disable



SAS EnableSAS Disable

With SAS Enabled the unit will provide greater stability in areas with increased thermal noise, improved signal to noise ratio and greater discrimination against small targets. Note there will be a reduction in range when SAS is enabled.

DIL Switches 2&3 Pulse Count



Pulse Count1 Pulse Count2 Pulse Count3 Pulse Count4

DIL Switches 4, 5 & 6

These switches control the display modes of the Yellow and Red LEDs. If the DIS(able) terminal is to be used for external control of the LEDs, switches 4, 5, & 6 should be in the OFF position.

The following terminology is used for the LEDs.

Gard LED A state when the Yellow LED flashes once every 2 seconds (typical)

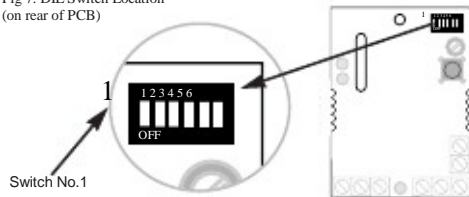
Pre AlarmA 2 second (typical) flash of the Yellow LED when Warning System a signal not great enough to count as a pulse or alarm has been processed.

Pulse CountA brief pulse of the relevant LED as a Pulse Count is processed.

Full AlarmA 2 second (typical) illumination of the Red LED and the activation of the alarm relay when a Full Alarm is processed.

ON if SETWhen +12V applied to Latch Terminal

Fig 7. DIL Switch Location
(on rear of PCB)



LED Configuration Switches

LED Displays			
	Gard LED	Pre Alarm	Pulse Count
	OFF	ON	Red LED
	OFF	OFF	Red LED
	Gard LED	Pre Alarm	Pulse Count
	OFF	OFF	Yellow LED
	OFF	OFF	Red LED
	Gard LED	Pre Alarm	Pulse Count
	OFF	OFF	Red LED
	Gard LED	Pre Alarm	Pulse Count
	ON	OFF	Red LED
	Gard LED	Pre Alarm	Pulse Count
	ON	OFF	OFF
	Gard LED	Pre Alarm	Pulse Count
	ON if SET	OFF	Yellow LED
	Gard LED	Pre Alarm	Pulse Count
	ON if SET	OFF	Red LED

LED Displays with Latch

When the latch terminal is used, the Red LED indicates if a detector was triggered whilst the system was set. A Red LED flashes on the first detector to be triggered in a group with subsequent detectors in that group having the Red LED permanently lit*. Any Latched LEDs will be reset when the system is next Set/Unset. All displays are inhibited whilst the system is set.*Not on Any to Alarm.

Specification	MX+	QX+
Voltage Range 9.5-16V d.c	x	x
Current Consumption 13mA	x	x
Max Ripple 4V Pk to Pk	x	x
N/C Alarm Output	x	x
2 Second Alarm Period (typical)	x	x
N/C Tamper Output (0.5A @24V)	x	x
-10°C to +50°C Operating Temp	x	x
15 metres Nominal Range	x	x
110°Coverage Angle	x	x
20V/m to 1000MHz RFI Protection	x	x
1, 2, 3, 4 Pulse Count	x	x
Optical Light Pipe		x
Advanced Temperature Compensation (ATC)	x	x
Pre-Alarm Warning System (PAWS)	x	x
Intelligent Detection Algorithms (IDA)	x	x
Shadow Algorithm System (SAS)	x	x

GARDINER TECHNOLOGY

The
Gardscan MX+ & QX+
Microprocessor
Controlled PIR

Installation
Instructions

Gardiner Technology Ltd
Queensway, Rochdale, OL11 1TQ.
United Kingdom

Tel: 01706 510000 Fax: 01706 510100
Int Tel: +44 1706 510000 Fax: 44 1706 510100
Internet: <http://www.gardiner-technology.com>
e-mail: sales@gardiner-technology.com
Technical Support Tel:01706 510200
Technical Support Fax: 01706 510010
(Mon - Fri 09:00-17:30)



Introduction

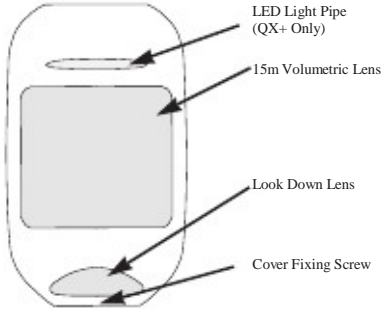
The Gardscan MX+ & QX+ Infra-Red detectors are high quality latching PIRs incorporating the latest SMT (surface mount technology) production techniques. Microprocessor based Intelligent Detection Algorithms and Advanced Temperature Compensation give you the ultimate PIR detector that is efficient, reliable and compact.

Control of the advanced Microprocessor features that are available is achieved through a bank of six DIL switches situated on the rear of the PCB.

The unit offers a 110° field of view and is supplied with volumetric lens giving a maximum range of 15 metres. Incorporated into the moulding is a look down lens that utilises mirrored optics to give detection directly beneath the detector. The QX+ also offers the benefit of an optical light pipe for the main LED giving a wider angle of view than the standard LED configuration.

Range adjustment of the unit is achieved via five preset notches on the PCB and a sensitivity pot giving simple but accurate range adjustment. The units are suitable for corner or wall mounting and an optional ceiling mount bracket is also available.

Fig1. Gardscan QX



Mounting Location

The surface that the PIR is to be mounted to should be of firm construction and free from vibration.
Do notMount the unit in a position where direct or reflected sunlight can fall on the lens.
Do notMount the unit in draughty locations.
Do notMount the unit over or facing heat sources.
Do notRun cabling to the unit parallel to mains wiring.

Mounting

To mount the PIR to the wall proceed as shown in the steps below:-

- a) Slacken (but do not remove) the cover fixing screw and remove the cover.
- b) Unscrew the PCB retaining screw.

- c) Remove the PCB and store in a safe place.
- d) Referring to Fig 2. remove the desired cable entry and fixing blanks (only remove necessary blanks).
- e) Using the PIR base as a template mark the wall (do not drill through the base under any circumstances).
- f) Using a suitable drill bit, drill the wall and fit the rawl plugs (supplied).
- g) Whilst feeding the cable through the cable entry offer the base to the wall then screw the base to the wall using the screws provided.
- h) After wiring and adjusting the PCB (see relevant sections) replace the cover and tighten the cover retaining screw.

Fig 2. Gardscan QX+ & MX+ Base

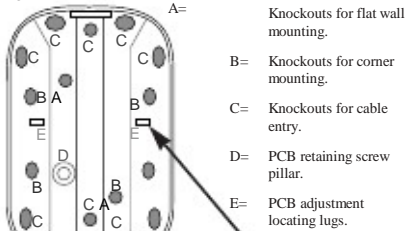
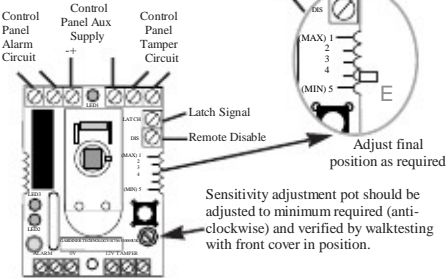


Fig 3. Gardscan QX+ & MX+ Wiring



Note: LED positions will vary dependant on model

Wiring

Wiring to the PIR should be six core 7/02 security cable. This cable should be fitted in accordance with good wiring practice. Connection details are given in Fig 3.

Latch Terminal

Wiring to the latch terminal should be from the control panel SW+ terminal for any to alarm indication. For first / other to alarm indication the 47k resistor (supplied) should be placed in series with the SW+ wire. Only one resistor should be used for each group of PIRs using the first to alarm feature.

DIS (Disable) Terminal

This terminal acts in conjunction with the DIL switches. Provided the correct mode of operation is selected through the DIL switches, the LEDs may be disabled by a remote signal from the control equipment that provides 12V+ to this terminal.

Coverage

The coverage pattern of the Gardscan MX+ & QX+ is shown in Fig 4. & Fig 5.

Fig 4. Coverage Pattern (plan view)

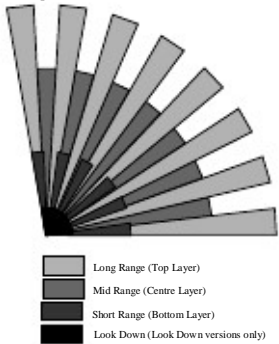


Fig 5. Coverage Pattern (side view)



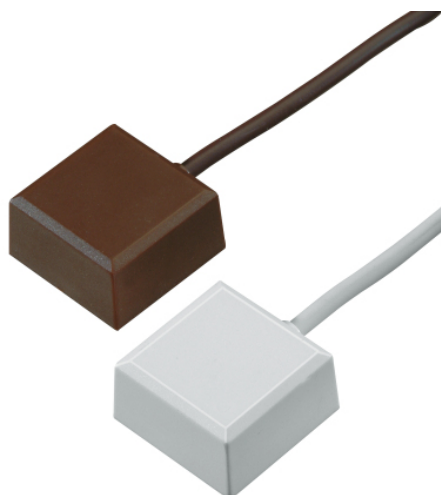
Range Adjustment

The Gardscan MX+ & QX+ range may be adjusted by moving the PCB into one of the five preset notches. With the unit mounted at the recommended 2.9 metres and counting the top notch as No.1 the ranges shown in Fig 6 may be achieved. If necessary use the PCB screw to hold the PCB in position. Final adjustment is via the sensitivity adjustment pot to minimum required (see Fig 3) for details.

Potential-free glass breakage detector, brown

Art. no. FU7300B

Page 1 of 1



- No operating voltage necessary
- Ideal for use with wireless opening detectors FU8320, FU8321 and FU9025
- Detection range: 2 m radius

Use the passive glass breakage detector to monitor individual window panes. The detector is fixed to the window pane to be protected. The ultrasound signals resulting from glass breakage are detected and reported to the alarm center. The passive glass breakage detector is fixed between 20 and 50 mm from the window frame. Important: The detector should be fixed in such a way that you notice immediately if there is no contact to the pane. The cable should not be in danger of being damaged when the window is opened and closed. If fixed in a corner of the window, the passive glass breakage detector monitors an area of up to 4m². Its maximum monitoring radius is 2 meters. The passive glass breakage detector works with an integrated piezo element. It reacts to the sound frequency caused by breaking glass. In the event of an alarm, a CMOS relay is opened. The passive glass breakage detector is potential-free, i.e. no voltage supply is required. This makes the detector ideal for wireless systems. The passive glass breakage detector is available in white (FU7300W) and brown (FU7300B).

Technical Data:

Dimensions	(WxHxD) 18x18x9 mm
Connections	NC switch contact max. switching voltage 15 V DC / max. switching current: 15 mA
Width	18 mm
Type of detection	Audible
Detection range of detector (m ²)	4 m ²
Housing material	ABS
Height	18 mm
Cable type	2-core

Cable length	2 m
Length	9 mm
Max. operating temperature	55 °C
Installation location	On glass areas
Net weight	0.04 kg
IP protection class	65
Sensor type	Piezo sensor
AC voltage supply	Not required V
DC voltage supply	Not required V



Hosiden Besson Ltd
St. Joseph's Close
Brighton & Hove
East Sussex BN3 7EZ
England
Tel: +44 1273 861166
Fax: +44 1273 777501
Email: info@hbl.co.uk
www.hbl.co.uk

product

Flashtone_{MT}

Multi-Tone Sounder / Beacon

sheet

- EN54 part 3* approved LPC + VdS
- Ultra low current consumption
- 32 user selectable tones
- Choice of 3 volume levels
- Rated IP66 or IP45
- Installation via side & rear cable entry
- Easy push & twist lockable bayonet mounting
- Temperature range from - 40 to +70 °C
- Simple in & out wiring blocks offering continuous override and dual signalling
- Full range of accessories & colours available

The Flashtone_{MT} combines the modern traits of the Banshee_{MT} multi-tone sounder with the very latest strobe technology to create an extremely efficient Zenon Beacon/sounder combination. Available in a wide range of colours, the Flashtone_{MT} will suit the majority of fire and security needs where conventional sound requires the backup of a visual alarm.

Manufactured in flame retardant polymer in a choice of and lens colours, the Flashtone_{MT} is suitable for many applications and environments.

Designed and constructed by utilising the easy push and twist bayonet fitting, the Flashtone_{MT} can be mounted on a common bracket used on the Banshee Multitone.



Product Order no:
8571000/HB Red / Red Std Base
8571150/HB Red / Clear Deep Base
8571300/HB Red / Blue Std Base
8571450/HB Red / Green Deep Base
8571600/HB White / Clear Std Base
8571750/HB White / Amber Deep Base
8571900/HB White / Green Std Base

8571050/HB
8571200/HB
8571350/HB
8571500/HB
8571650/HB
8571800/HB
8571950/HB

Red / Red Deep Base
Red / Amber Std Base
Red / Blue Deep Base
White / Red Std Base
White / Clear Deep Base
White / Blue Std Base
White / Green Deep Base

8571100/HB
8571250/HB
8571400/HB
8571550/HB
8571700/HB
8571850/HB

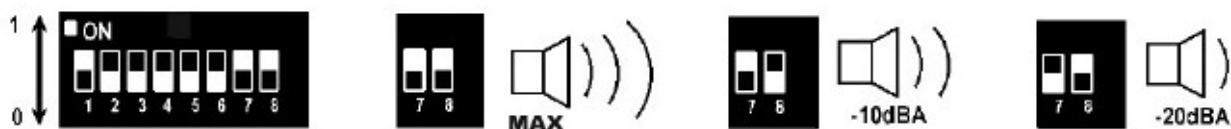
Red / Clear Std Base
Red / Amber Deep Base
Red / Green Std Base
White / Red Deep Base
White / Amber Std Base
White / Blue Deep Base

*Pending

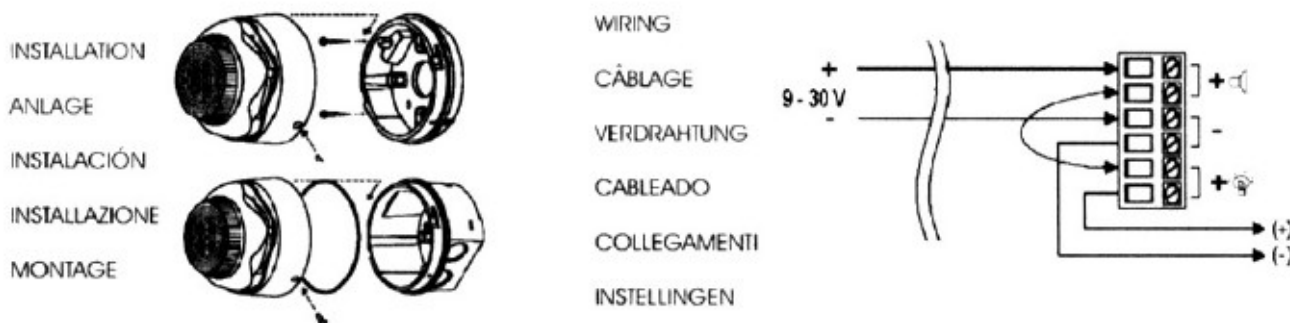
Flashtone_{MT}

Multi-Tone Sounder / Beacon

Switch bank



Sound selection 1 to 5, output level 7 & 8, 6 not used



Flasher Information: Flash rate: 1/sec Current @ 12V: 90mA Current @ 24V: 40mA

Tone Information			Banshee 12v				Banshee 24v			
No.	Description	Frequency and Timing	2nd tone	dBA (Typ)	mA (Typ)	dBA (Typ)	mA (Typ)			12345
1	Banshee LF Buzz	800Hz to 950Hz swept at 120Hz*	4	94	6	100	12			00000
2	Banshee LF Fast Sweep	800Hz to 950Hz swept at 9Hz*	4	94	6	100	12			10000
3	Banshee LF Slow Sweep	800Hz to 950Hz swept at 3Hz*	4	94	6	100	12			01000
4	Banshee LF Continuous	Continuous at 850Hz*	4	95	6	101	12			11000
5	Banshee LF Fast Sweep (New)	830Hz to 970Hz swept at 9Hz*	4	95	6	101	12			00100
6	Medium Sweep LF	800Hz to 970Hz swept at 1Hz*	7	95	6	102	12			10100
7	Continuous LF	Continuous at 970Hz*	7	97	6	103	12			01100
8	Backup Alarm LF	Intermittant at 950Hz 1s on, 1s off*	4	97	4	103	7			11100
9	Alternate LF	Alternating 800Hz/1000Hz, 0.5s each sound*	4	94	8	101	14			00010
10	Medium Sweep LF	800Hz/1000Hz swept at 0.5s*	4	95	5	101	12			10010
11	Alternate LF	Alternating tones 800/950Hz*	4	96	6	103	14			01010
12	Banshee/Bedlam HF buzz	2400Hz to 2900Hz at 120Hz	15	102	12	110	25			11010
13	Banshee/Bedlam HF Fast Sweep	2400Hz to 2900Hz at 9Hz	15	103	12	110	25			00110
14	Banshee/Bedlam HF Slow Sweep	2400Hz to 2900Hz at 3Hz	15	103	12	110	25			10110
15	Banshee/Bedlam HF Continuous	Continuous 2900Hz at 3Hz	15	105	12	112	30			01110
16	Banshee/Bedlam HF Fast Sweep (New)	2450Hz to 3100Hz swept at 9Hz	15	103	12	110	25			11110
17	Backup Alarm LF	Intermittant at 2900Hz 1s on, 1s off	15	105	7	112	15			00001
18	Alternate LF	Alternating tones 2400/2900Hz at 3Hz	15	104	12	110	25			10001
19	Slow Whoop	500Hz rising to 1200Hz over 3.5, silence 0.5s, repeat	4	96	6	101	12			01001
20	Din Tone (DK)	1200Hz falling to 500Hz over 1s, silence 10ms, repeat	4	93	6	101	14			11001
21	French Fire Sound	554Hz for 100ms and 440Hz for 400ms	4	90	5	96	10			00101
22	Australian Alert Signal	420Hz repeating 0.625s on, 0.625s off	4	88	3	94	6			10101
23	Australian Evacuation Signal	500Hz rising to 1200Hz over 3.75s on, 0.25s off	4	96	6	103	12			01101
24	US Temporal Tone LF	950Hz for 0.5s on 0.5s off, for 3 phases, silence for 1.5s & repeat	4	96	3	103	6			11101
25	US Temporal Tone HF	2900Hz for 0.5s on 0.5s off, for 3 phases, silence for 1.5s & repeat	15	105	6	112	12			00011
26	Swedish Tone (Fire)	Intermittant 660Hz 150ms on, 150ms off	26	81	4	87	7			10011
27	Swedish Tone (All Clear)	Continuous 660Hz	27	84	5	89	14			01011
28	ISO8201 LF	Intermittant 970Hz 500ms on, 500ms off*	28	96	4	103	7			11011
29	ISO8201 HF	Intermittant 2900Hz 500ms on, 500ms off	29	106	8	112	15			00111
30	BT Banshee (FP1063.1)	Yodel 800Hz/1000Hz, 0.25s each frequency*	31	93	8	101	14			10111
31	BT Banshee (FP1063.1)	Continuous 1000Hz*	31	95	8	102	17			01111
32	Bell Tone	Bell Tone*	32	96	8	101	17			11111

*Tone meets frequency requirements of BS 5839 pt. 1 1988