

Despradel Novas Peña

ID #:UB5436SEL11522

Microcontroladores

Arquitectura, programación y aplicación

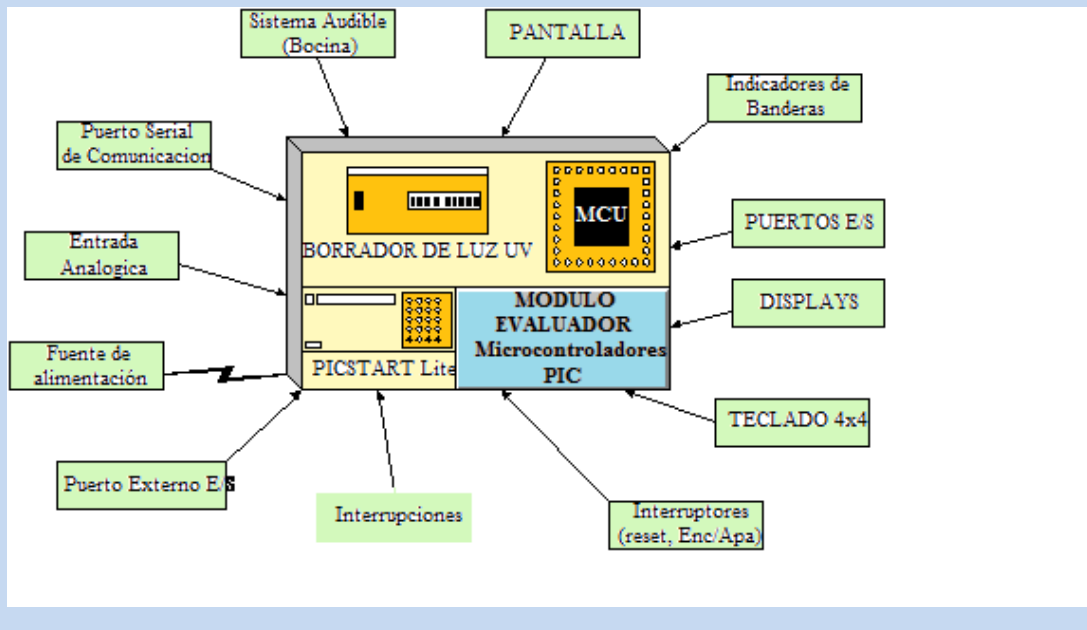
**ATLANTIC INTERNATIONAL UNIVERSITY Honolulu,
Hawai**

April 23 2008

Atlantic International University

MicroControladores

Arquitectura, Programación y Aplicación



Índice

Presentación	1
Portada	2
Índice	3
Introducción	4
Que es un microcontrolador?	5
Diferencia entre un microprocesador y un microcontrolador	6
Arquitectura interna de un microcontrolador	7
Memorias	13
Tipos de microcontroladores PIC	16
Herramientas para desarrollar sistemas con PIC	18
El MPLAB-IDE	20
Aplicaciones de los microcontroladores	24
Sistemas con microcontroladores	26
Conclusión	30
Bibliografía	31

Introducción

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos. Entre ellos los microprocesadores y los microcontroladores, los cuales son básicos en las carreras de ingeniería electrónica.

Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea. El número de productos que funcionan en base a uno o varios microcontroladores aumenta de forma exponencial.

La industria Informática acapara gran parte de los microcontroladores que se fabrican. Casi todos los periféricos del computador, desde el ratón o el teclado hasta la impresora, son regulados por el programa de un microcontrolador.

Los electrodomésticos de línea blanca (lavadoras, hornos, lavavajillas, etc.) y de línea marrón (televisores, vídeos, aparatos musicales, etc.) incorporan numerosos microcontroladores. Igualmente, los sistemas de supervisión, vigilancia y alarma en los edificios utilizan estos chips. También se emplean para optimizar el rendimiento de ascensores, calefacción, aire acondicionado, alarmas de incendio, robo, etc.

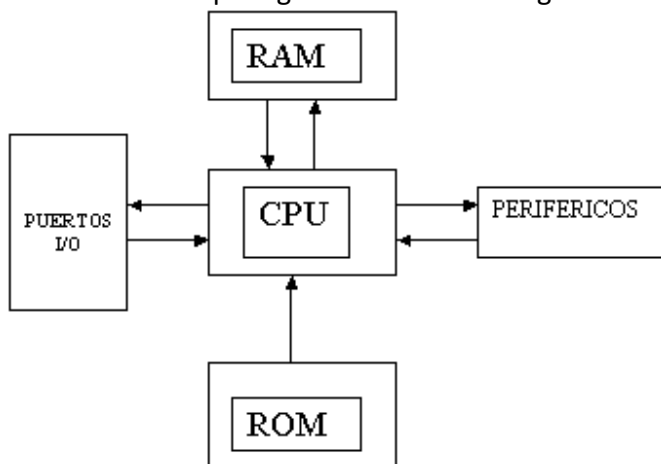
Las aplicaciones de los microcontroladores son vastas, se puede decir que solo están limitadas por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y el automatismo, en la industria del entretenimiento, en las telecomunicaciones, en la instrumentación, en la industria automotriz, etc.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores electrónicos se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un solo circuito integrado, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo ordenador contenido en un circuito integrado.

¿QUÉ ES UN MICROCONTROLADOR?

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.

El microcontrolador es un computador dedicado a diversas aplicaciones. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.



Toda microcomputadora requiere de un programa para que realice una función específica. Este se almacena normalmente en la memoria ROM. No está de más mencionar que sin un programa, los microcontroladores carecen de utilidad.

El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario le escribe, es por esto que la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. El carácter programable de los microcontroladores simplifica el diseño de circuitos electrónicos. Permiten modularidad y flexibilidad, ya que un mismo circuito se puede utilizar para que realice diferentes funciones con solo cambiar el programa del microcontrolador.

Diferencias entre un microprocesador y un microcontrolador

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (CPU), también llamado procesador, de un computador. El CPU está formado por la Unidad de Control, que interpreta las instrucciones, y el BUS de Datos, que los ejecuta. Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de (ENTRADA / SALIDA) E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos. Usted podría pensar que las características de un sistema cerrado representan una desventaja con relación a los Microprocesadores, pero en la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es difícil no encontrar uno que se adapte a nuestros requerimientos del momento.

Es posible seleccionar la capacidad de las memorias, el número de líneas de (ENTRADA / SALIDA) E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

Podemos concluir con que la diferencia fundamental entre un Microprocesador y un Microcontrolador: es que el Microprocesador es un sistema abierto con el que se puede construirse un computador con las características que se desee, acoplándole los módulos necesarios.

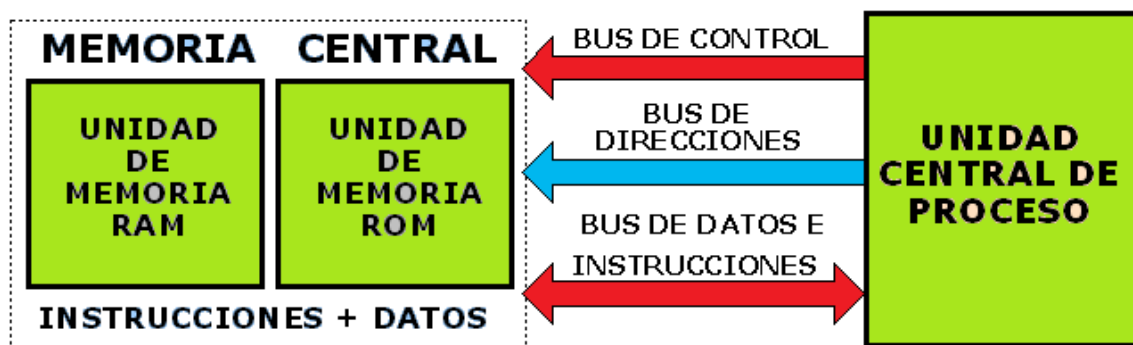
Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas: Aumento de prestaciones, un mayor control sobre un determinado elemento representa una mejora considerable en el mismo. Aumento de la habilidad al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes. Reducción del tamaño en el producto acabado. La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks. Mayor flexibilidad: las características de control están programadas por lo que su medicación sólo necesita cambios en el programa de instrucciones.

Arquitectura interna de un microcontrolador

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard.

La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control). La arquitectura Harvard dispone de dos memorias independientes, una que contiene sólo instrucciones y otra, sólo datos.

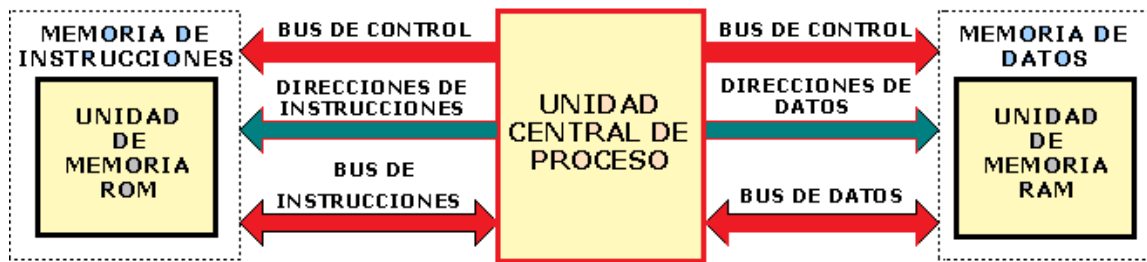
ARQUITECTURA VON NEUMANN



La repercusión más importante del empleo de la arquitectura Harvard en los microcontroladores PIC se manifiesta en la organización de la memoria del sistema. La memoria de programa es independiente de la de los datos, teniendo tamaños y longitudes de palabra diferentes.

En los PIC16C5X el formato de todas las instrucciones es de 12 bits y, en consecuencia, la longitud de las palabras de la memoria de programa también. Este tamaño permite codificar en una palabra el código OP de la instrucción junto al operando o su dirección. Para adaptarse a las necesidades de las aplicaciones del usuario hay modelos de la gama baja con 512 posiciones para la memoria de instrucciones y otros que tienen 1 k y 2 k posiciones de 12 bits.

ARQUITECTURA HARVARD



El tamaño de los buses que direccionan la memoria de datos y la de programa son diferentes. Lo mismo pasa con el bus que transfiere las instrucciones y el que lo hace con los datos. La total independencia entre los accesos a las dos memorias permite realizar accesos simultáneos.

La memoria de programa siempre está direccionada desde el Contador de Programa (PC), mientras que la memoria de datos pueda direccionarse directamente desde parte del código OP de la instrucción o indirectamente a través de un registro denominado FSR (Registro de Selección del Banco).

Un microcontrolador posee todos los componentes de un computador, pero con unas características fijas que no pueden alterarse.

A continuación detallaremos las principales partes de un microcontrolador:

-Procesador

La necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones ha desembocado en el empleo generalizado de procesadores de arquitectura Harvard frente a los tradicionales que seguían la arquitectura de von Neumann. Esta última se caracterizaba porque la UCP (Unidad Central de Proceso) se conectaba con una memoria única, donde coexistían datos e instrucciones, a través de un sistema de buses.

En la arquitectura Harvard son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso. Esta dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos. También la capacidad de cada memoria es diferente.

- Memoria de programa

El microcontrolador está diseñado para que en su memoria de programa se almacenen todas las instrucciones del programa de control. No hay posibilidad de utilizar memorias externas de ampliación.

La memoria de instrucciones puede tener una capacidad mínima de 512 palabras de 12 bits hasta una máxima de 2.048 palabras de la misma longitud. Durante la fase de búsqueda, la dirección de la instrucción la proporciona el PC (contador de Programa), el cual normalmente se autoincrementa en la mayoría de las instrucciones, excepto en las de salto.

Hasta 512 posiciones la memoria se direcciona directamente con 9 bits, denominándose a dicho tamaño página. Las 512 primeras posiciones de la memoria configuran la página 0, el siguiente conjunto de otras 512 posiciones forman la página 1, etc.

El contador de programa

El PC proporciona la dirección de la memoria de programa y su longitud puede oscilar entre 9 y 11 bits, según la capacidad de la misma.

El comportamiento del PC para las diversas instrucciones del repertorio es el siguiente:

1º. Para todas las instrucciones que no entrañen saltos en el programa, el PC se autoincrementa para apuntar la siguiente instrucción a la que se está ejecutando.

2º. Las instrucciones de salto GOTO cargan en el PC directamente los 9 bits de menos peso desde el código OP de la propia instrucción. Cuando el área de programa es superior a las 512 posiciones, los 2 bits de más peso del PC (A10 y A9) se cargan con el valor de los bits PA1 y PA0 del Registro de Estado.

3º. Las instrucciones CALL contienen en su código OP el valor de los 8 bits de menos peso del PC. El bit A8 toma en este caso el valor 0 lo que obliga a restringir la ubicación de las rutinas a las 256 primeras posiciones de cada página. Cuando son necesarios los bits A10 y A9, se cargan con los bits PA1 y PA0 del Registro de Estado.

4º. En las instrucciones de "salto relativo", la ALU suma al PC inicial (A7-A0), el valor del salto que se obtiene de los 8 bits de menos peso del código OP. El resultado de la operación se carga en la parte baja del PC (A7-A0). El bit A8 toma el valor 0 y los bits A10 y A9 se cargan con el valor de PA1 y PA0, respectivamente.

Cuando el PC apunta la última posición de una página y se autoincrementa pasa a direccionar a la siguiente instrucción que estará situada en la primera posición de la página siguiente. Cuando se produce un cambio de página, se deben actualizar los bits PA1 y PA0 del Registro de Estado porque, en caso contrario, cuando se ejecute una instrucción de salto, éste seguirá controlando la página en la que se halla esta instrucción de salto.

Cuando se genera un reset, los bits PA1 y PA0 del Registro de Estado toman el valor 0. El PC queda apuntando la última posición de la última página que exista en la EPROM. Si en dicha posición se coloca una instrucción de GOTO, se pasa a direccionar una posición de la página 0.

En el caso de llevarse a cabo un reset en un microcontrolador con 512 posiciones de programa, la primera instrucción que se ejecuta es la que ocupa la dirección 1FFh en el PIC16C54/55. Si tiene 1 k posiciones, como sucede al PIC16C56, la primera instrucción que se ejecuta tras un reset es la que ocupa la dirección 3FFh y, si tiene 2 k (PIC16C57/58), ocupa la dirección 7FFh.

Como el programa a ejecutar siempre es el mismo, debe estar grabado de forma permanente.

Los tipos de memoria adecuados para soportar esta función admiten cinco versiones diferentes:

- ROM
- EPROM
- EEPROM, entre otras

- Memoria de datos

Los datos que manejan los programas varían continuamente, y esto exige que la memoria que les contiene deba ser de **lectura y escritura**, por lo que la memoria RAM estática (SRAM) es la más adecuada, aunque sea volátil.

Hay microcontroladores que también disponen como memoria de datos una de lectura y escritura no volátil, del tipo EEPROM. De esta forma, un corte en el suministro de la alimentación no ocasiona la pérdida de la información, que está disponible al reiniciarse el programa. El PIC16C84, el PIC16F83 y el PIC16F84 disponen de 64 bytes de memoria EEPROM para contener datos.

Los microcontroladores PIC funcionan con datos de 8 bits por lo que las posiciones de la memoria de datos tienen esa longitud. La capacidad de esta memoria varía entre 25 y 73 posiciones de un byte, en los modelos de la gama baja.

La memoria de datos funciona de forma similar al **banco de registros** de un procesador por lo cual sus posiciones implementan registros de propósito especial y propósito general. Las ocho primeras posiciones del banco 0 soportan registros específicos.

-Líneas de E/S para los controladores de periféricos:

A excepción de dos pines destinadas a recibir la alimentación, otras dos para el cristal de cuarzo, que regula la frecuencia de trabajo, y una más para provocar el Reset, los restantes pines de un microcontrolador sirven para soportar su comunicación con los periféricos externos que controla.

Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo y se agrupan en conjuntos de ocho, que reciben el nombre de Puertas. Hay modelos con líneas que soportan la comunicación en serie; otros disponen de conjuntos de líneas que implementan puertas de comunicación para diversos protocolos.

Un recurso imprescindible en los microcontroladores es el que soporta las Entradas y Salidas con los periféricos del mundo exterior. Los PIC16C5X sólo disponen de líneas de E/S digitales, cada una de las cuales tiene su correspondencia con una patita de la cápsula. Dichas líneas se agrupan en tres puertos Puerto A, Puerto B y Puerto C. El Puerto A actúa lo mismo que un registro de E/S de lectura y escritura. Sólo son válidos los 4 bits de menos peso que corresponden con las patitas RA3:RA0. Los 4 bits de más peso no están implementados y cuando se leen siempre valen 0.

En los modelos PIC16C55/57, con 28 patitas, el Puerto C funciona como un registro de E/S de 8 bits. En los PIC16C54/CR54/C56, este registro se comporta como uno de propósito general al no existir suficientes patitas en las cápsulas para soportar estas líneas de E/S.

Los bits de cada puerto se configuran mediante los bits correspondientes de un registro de control asociado que recibe el nombre de TRIS. En realidad cada puerto soporta dos registros:

1º El registro de datos, al que se denomina Puerto X (PortX). Siendo X= A o B o C.

2º El registro de control TRISX, con el que se programa el sentido (Entrada o Salida) de las líneas del puerto X.

Los Puertos A, B y C se corresponden con las posiciones 5, 6 y 7 del área de datos. Cada uno de sus bits puede programarse como una línea de Entrada o de Salida, según se ponga un 1 ó un 0, respectivamente, en el bit del registro de control TRIS correspondiente.

Cuando una patita de E/S funciona como salida, el bit que proviene del bus de datos se guarda en el Biestable del dato con lo cual la información que ofrece esta patita

permanece invariable hasta que se reescriba otro bit. Para configurar la patita como Entrada, hay que cargar un 1 en el Biestable de control de E/S mientras que hay que cargar un 0 si se desea que sea Salida.

Cada línea de E/S de los puertos se programa de forma independiente y puede ser Entrada o Salida. Cuando se produce un reset, todos los bits de los registros TRIS pasan a tener el valor 1 y todas las líneas de E/S actúan como Entrada por evidentes motivos de seguridad para evitar daños irreparables.

Memorias

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio circuito integrado. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los PC's:

1. No existen sistemas de almacenamiento masivo como disco duro o disquetes.
2. Como el microcontrolador sólo se destina a una tarea en la memoria de programa, sólo hay que almacenar un único programa de trabajo.

La memoria de datos (RAM) en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la memoria de programa (ROM).

El usuario de PC está habituado a manejar Megabytes de memoria, pero los diseñadores con microcontroladores trabajan con capacidades de memoria de programa de 512 bytes, 1K, 2K (hasta unos 64K) y de RAM de 20 bytes, 68 bytes, 512 bytes (hasta unos 4K).

Según el tipo de memoria de programa que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado:

-ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip.

Máscara viene de la forma cómo se fabrican los circuitos integrados. Estos se fabrican en obleas que contienen varias decenas de chips. Estas obleas se obtienen a partir de procesos fotoquímicos, donde se impregnan capas de silicio y óxido de silicio, y según convenga, se erosionan al exponerlos a la luz. Como no todos los puntos han de ser erosionados, se sitúa entre la luz y la oblea una máscara con agujeros, de manera que donde deba incidir la luz, esta pasará. Con varios procesos similares pero más complicados se consigue fabricar los transistores y diodos que componen un circuito integrado.

El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

- OTP

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

-EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caras que los microcontroladores con memoria OTP que están hechos con material plástico. Hoy día se utilizan poco, siendo sustituidas por memorias EEPROM o Flash.

-EEPROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continúa. Hoy día están siendo sustituidas por memorias de tipo Flash.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

Este tipo de memoria es relativamente lenta.

- FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado. Son idóneas para la enseñanza y la Ingeniería de diseño.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

Tipos de microcontroladores PIC

Entre casi un centenar de fabricantes de microcontroladores que existen en el mundo es muy difícil seleccionar «el mejor». En realidad no existe, porque en cada aplicación son sus características específicas las que determinan el más conveniente.

- **Gama baja o básica: PIC16C5X con instrucciones de 12 bits**

Se trata de una serie de PIC de recursos limitados, pero con una de las mejores relaciones *Coste/prestaciones*. Sus versiones están encapsuladas con 18 y 28 patitas y pueden alimentarse a partir de una tensión de 2,5 V lo que les hace ideales en las aplicaciones que funcionan con pilas. Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo de interrupción y la pila sólo dispone de dos niveles

- **Gama media: PIC16CXXX con instrucciones de 14 bits**

Es la gama más variada y completa de los PIC. Abarca modelos con encapsulado desde 18 pines hasta 68, cubriendo varias opciones que integran abundantes periféricos. Dentro de esta gama se halla el «fabuloso PIC 16F84».

El repertorio de instrucciones es de 35 a 14 bits cada una y compatible con el de la gama baja. Sus distintos modelos contienen todos los recursos que se precisan en las aplicaciones de los microcontroladores de 8 bits. También dispone de interrupciones y una Pila de 8 niveles que permite el anidamiento de subrutinas.

- **Gama alta: PIC17CXXX con instrucciones de 16 bits**

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectorizadas muy potente. También incluyen variados controladores de periféricos, puertas de comunicación serie y paralelo con elementos externos y un multiplicador hardware de gran velocidad.

Quizás la característica más destacable de los componentes de esta gama es su *arquitectura abierta*, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos. Para este fin, las patitas sacan al exterior las líneas de los buses de datos, direcciones y control, a las que se conectan memorias o controladores de periféricos. Esta filosofía de construcción del sistema es la que se empleaba en los microprocesadores y no suele ser una práctica habitual cuando se emplean microcontroladores.

- **Gama mejorada: PIC18C(F)XXX con instrucciones de 16 bits**

En los inicios del tercer milenio de nuestra era Microchip presentó la gama mejorada de los microcontroladores PIC con la finalidad de soportar las aplicaciones avanzadas en las áreas de automoción, comunicaciones, ofimática y control industrial. Sus modelos destacaron por su alta velocidad (40 MHz) y su gran rendimiento (10 MIPS a 10 MHz). Entre las aportaciones más representativas de esta serie de modelos que crece cada año, destacan.

- A) Un espacio de direccionamiento para la memoria de programa que permite alcanzar los 2 MB, y 4 KB para la memoria de datos.
- B) Inclusión de la tecnología FLASH para la memoria de código.
- C) Potente juego de 77 instrucciones de 16 bits cada una. Permiten realizar una multiplicación 8 x 8 en un ciclo de instrucción, mover información entre las memorias y modificar el valor de un bit en un registro o en una línea de E/S.
- D) Orientación a la programación en lenguaje C con la incorporación de compiladores muy eficientes para este lenguaje.
- E) Nuevas herramientas para la emulación.

Herramientas para desarrollar sistemas con microcontroladores

Las herramientas de desarrollo están formadas por un conjunto de programas e interfaces que permiten realizar los proyectos de la forma más eficiente posible.

Algunas de las principales herramientas de ayuda para desarrollar sistemas basados en microcontroladores son las siguientes:

Ensamblador. La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema.

Compilador. La programación en un lenguaje de alto nivel (como C o Basic) permite disminuir el tiempo de desarrollo de un producto y si además está familiarizado con C o Basic es una buena opción. No obstante, cuando el compilador convierte el código del programa a un lenguaje ensamblado, cada línea de código del programa en lenguaje de alto nivel habrá generado bastantes más líneas de código en lenguaje ensamblador, normalmente en una relación de uno a tres. Esto significa que para utilizar un lenguaje de alto nivel necesitaremos un microcontrolador con una capacidad de memoria relativamente grande.

Simulador. Se trata de un software que es capaz de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador.

Placas de evaluación. Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc.

Emuladores en circuito. Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la placa de la aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo.

Programador. Es un dispositivo que conectado a un PC permite grabar en el microcontrolador el programa desarrollado.

Otra posibilidad es utilizar un **cargador de arranque**, muy útil en la etapa de desarrollo de un programa. Un cargador de arranque es un pequeño programa en el microcontrolador que está montado en la placa del circuito que se está desarrollando y que puede comunicarse con las herramientas de desarrollo a través de un enlace serie, como puede ser RS232, USB, I²C o un bus CAN.

El programa cargador de arranque debe interpretar comandos para leer, grabar y borrar la parte de memoria reservada para el programa de la aplicación. Cuando se desea verificar el programa que se está desarrollando se inicia la comunicación con el programa cargador de arranque, que carga el código de programa en la memoria de programa del microcontrolador.

Paquetes IDE

Actualmente existen paquetes de software denominados **Entornos de Desarrollo Integrado**, IDE, que suelen funcionar bajo Windows y que incluyen editores de texto para el ensamblador o el compilador, permiten la simulación del programa y también pueden integrar el control de emuladores y programadores de dispositivos. Ejemplos de estos entornos de desarrollo son MPLAB de Microchip que permite programar en lenguaje ensamblado y PCWH de la casa CCS que incluye un compilador C para los microcontroladores PIC de Microchip.

El MPLAB-IDE

Es una Plataforma de Desarrollo Integrada bajo Windows, con múltiples prestaciones, que permite escribir el programa para los PIC en **Lenguaje Ensamblador (assembler)** o en C (el compilador C se compra aparte), crear proyectos, ensamblar o compilar, simular el programa y finalmente programar el componente, si se cuenta con el programador adecuado.

MPLAB incorpora todas las utilidades necesarias para la realización de cualquier proyecto y, para los que no dispongan de un emulador, el programa permite editar el archivo fuente en lenguaje ensamblador de nuestro proyecto, además de ensamblarlo y simularlo en pantalla, pudiendo ejecutarlo posteriormente en modo paso a paso y ver como evolucionarían de forma real tanto sus registros internos, la memoria RAM y/o EEPROM de usuario como la memoria de programa, según se fueran ejecutando las instrucciones. Además el entorno que se utiliza es el mismo que si se estuviera utilizando un emulador.

Partes de MPLAB-IDE:

~**EDITOR**: Editor incorporado que permite escribir y editar programas u otros archivos de texto.

~**PROJECT MANAGER**: Organiza los distintos archivos relacionados con un programa en un proyecto. Permite crear un proyecto, editar y simular un programa. Además crea archivos objetos y permite bajar archivos hacia emuladores (**MPLAB-ICE**) o simuladores de hardware (**SIMICE**).

~**SIMULADOR**: Simulador de eventos discretos que permite simular programas con ilimitados breakpoint, examinar/modificar registros, observar variables, tiempos y simular estímulos externos.

~**ENSAMBLADOR**: Genera varios tipos de archivos objetos y relacionados, para programadores Microchip y universales.

~**LINKER**: Permite unir varios archivos objetos en uno solo, generados por el ensamblador o compiladores C como **MPAB-C18** o compiladores de terceros.

~**PROGRAMADOR**: Mplab-IDE puede trabajar con varios tipos de programadores. El usuario debe seleccionar con cual trabajará, haciendo clic en opción **Programmer/ Select programmer**, se pueden seleccionar 4 programadores distintos:

- Picstart Plus

- MPLAB ICD 2
- MPLAB PM 3

Menú View

Seguidamente analizaremos las distintas posibilidades que ofrece el menú view:

- **Project:** Esta ventana presenta la lista de archivos que actualmente hay en un proyecto. Si el proyecto se ha ensamblado o compilado la ventana de proyecto muestra una lista de todos los archivos incluidos en el proyecto.
- **Output:** Muestra la ventana **Output**, que ofrece información de salida del programa, como por ejemplo cuando se compila aquí va apareciendo la información sobre la compilación, indicando si hay errores.
- **Toolbars:** Permite seleccionar que barras de herramientas aparecerán en MPLAB.
- **Disassembly Listing:** En esta ventana se puede ver el código máquina paralelamente al código en ensamblador. Resulta muy útil para seguir paso a paso la ejecución del programa.
- **Hardware Stack:** El contenido de la pila puede verse y modificarse con el ratón o el teclado al seleccionar esta opción. Si la pila se desborda, MPLAB indica su rebosamiento con el mensaje **underflow**.
- **Program Memory:** Al seleccionar esta opción aparece una ventana en la que se puede apreciar las posiciones de memoria que ocupa cada una de las instrucciones, el código de operación de cada instrucción y la posición de memoria que se le ha dado a cada etiqueta. Puede utilizarse para seguir paso a paso la ejecución del programa y además de modificar un registro mediante el ratón y el teclado pueden modificarse varios a la vez si se pone el puntero del ratón sobre el primer registro que se quiere modificar y seguidamente se pulsa el botón derecho del ratón y se activa la opción **Fill Registers**.

Si pulsamos en cada uno de los tres botones de la parte inferior de esta ventana podemos seleccionar tres formas de ver la memoria de programa:

~ **Opcode Hex:** representa la memoria de programa con los datos en hexadecimal. Esta opción es muy útil al usar el programador del dispositivo y comprobar si se grabaron bien los datos.

~ **Machine:** esta opción presenta el código máquina ensamblado con la información de las etiquetas y direcciones de memoria que tienen asignadas.

~ **Symbolic**: despliega el código hexadecimal desensamblado con los símbolos (etiquetas) utilizados en el programa.

- **File Registers**: La lista de registros del microcontrolador, que son de memoria SRAM, se puede ver seleccionando esta opción. Esta ventana al desplegarse presenta una lista con la memoria de datos del dispositivo emulado. El listado de registros puede verse de dos maneras, según se active **Hex** o **Symbolic**.
- **EEPROM**: Si el dispositivo emulado tiene EEPROM o memoria Flash como es el caso del PIC16F84, el contenido de la memoria puede verse seleccionando esta opción. La memoria de EEPROM puede modificarse a través de esta ventana.
- **Watch**: MPLAB permite supervisar los contenidos de los registros a través de esta ventana. Para agregar los registros a visualizar, se puede hacer directamente escribiéndolos con el teclado o mediante los botones **Add SFR** o **Add Symbol** y las ventanas desplegables correspondientes.

Special Function Registers: El contenido de los registros de funciones especiales (FSR) pueden verse y modificarse seleccionando esta opción.

Simulación

Para comenzar la simulación debemos elegir la herramienta que vamos a utilizar. Para hacerlo, por ejemplo, debemos seleccionar en la barra de menús **Debugger/Select tools/MPLAB SIM**; esto dará lugar a una nueva barra de herramientas:

- **Run (F9)**: Al activarse esta opción el programa se ejecuta a toda velocidad en "modo tiempo real", aunque no hay que olvidar que se trata de una simulación por ordenador. La ejecución comienza en la dirección actual del contador de programa (PC), que se muestra en la barra de estado y en la ventana **Program Memory**. Mientras el programa está en ejecución **Step** y **Run** están deshabilitados y se detiene si se activa **Halt** o si se encuentra un **breakpoint**.
- **Animate**: Esta opción es similar a la anterior pero en este caso el programa se simula de forma automática desde la dirección que indique el PC en el instante en que se active.
- **Halt (F5)**: Detiene la ejecución del programa, el contador de programa se detiene y se actualiza la información.
- **Step Into (F7)**: Esta acción ejecuta la instrucción cuya dirección de memoria coincida con el valor al que apunta el PC antes de activarla. Si algún registro se modifica se resalta en color rojo. En definitiva, ejecuta la instrucción actual y se detiene, actualizándose la información. Si se realiza un **Step Into** sobre una instrucción **SLEEP**, como en la realidad, el dispositivo pasa a modo sleep, siendo necesario despertarlo para continuar, por ejemplo mediante un reset.

- **Step Over (F8):** Ejecuta la instrucción actual y se detiene, actualizándose la información. Aparentemente funciona como **Step Into** pero si la instrucción a ejecutar es **CALL**, ejecuta la subrutina llamada y se detiene en la instrucción siguiente a la llamada.
- **Step Out:** Si la localización actual está dentro de una subrutina, la termina y retorna al programa principal.
- **Reset:** Esta opción inicializa el sistema. El Contador de Programa (PC), que es la dirección de memoria donde se encuentra la primera instrucción que ejecuta el microcontrolador cuando se realiza un reset en el sistema, se pone a 0.

Aplicaciones de los microcontroladores

Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos.

Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, ordenadores, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

Los microcontroladores se encuentran por todas partes:

- Sistemas de comunicación: en grandes automatismos como centrales y en teléfonos fijos, móviles, fax, etc.
- Electrodomésticos: lavadoras, hornos, frigoríficos, lavavajillas, batidoras, televisores, vídeos, reproductores DVD, equipos de música, mandos a distancia, consolas, etc.
- Industria informática: Se encuentran en casi todos los periféricos; ratones, teclados, impresoras, escáner, etc.
- Automoción: climatización, seguridad, ABS, etc.
- Industria: Automatas, control de procesos, etc

- Sistemas de supervisión, vigilancia y alarma: ascensores, calefacción, aire acondicionado, alarmas de incendio, robo, etc.
- Otros: Instrumentación, electromedicina, tarjetas (smartcard), sistemas de navegación, etc.

La distribución de las ventas según su aplicación es la siguiente:

- Una tercera parte se absorbe en las aplicaciones relacionadas con los ordenadores y sus periféricos.
- La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)
- El 16% de las ventas mundiales se destinó al área de las comunicaciones.
- Otro 16% fue empleado en aplicaciones industriales.
- El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

Si vamos a emplear este equipo para controlar, por ejemplo, una heladera deberemos conectar el compresor entre los contactos Común y Normal cerrado del relé. De esta forma cuando la temperatura alcance el tope superior del seteo se encenderá el compresor y, cuando de tanto enfriar, alcance el tope inferior lo apagará.

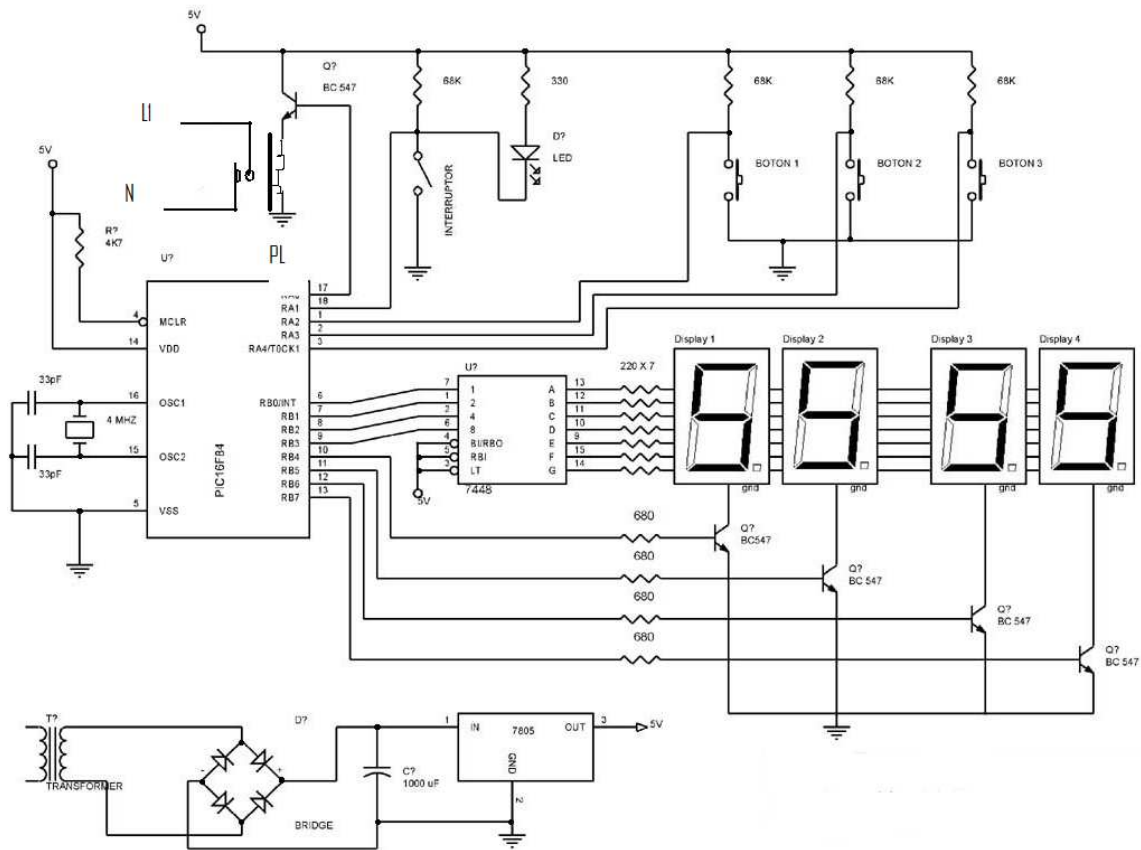
Si, en cambio, vamos a usar esto para calefaccionar, deberemos conectar el elemento calefactor entre los contactos Común y Normal Abierto del relé. De esta forma, cuando la temperatura caiga bajo el seteo inferior el calefactor arrancará y, cuando de tanto calentar, supere el tope el calefactor será desconectado.

A primera vista el programa parece algo complicado por lo extenso, pero es muy simple de entender. Primeramente se definen las posiciones de memoria a utilizar y los bits a emplear. Luego se inicializan las posiciones que así lo requieran y se lee de la EEPROM interna los seteos. Seguidamente se inicializa el LCD y se completan los caracteres fijos. Tras la primera conversión se coloca la temperatura en pantalla así como los seteos. Estos tres parámetros (Temp. actual, seteo inferior y seteo superior) son los únicos datos que se modifican en el LCD. Se compara si la temperatura está por debajo de la mínima para conectar el relé o por sobre la máxima para desconectarlos. Luego se controlan las teclas de mando, si alguna es accionada se actúa en consecuencia y por último cicla al principio donde se toma una nueva muestra de la temperatura.

PIC16F84, aplicaciones diversas

El Pic16F84 es un microcontrolador fácilmente programable mediante un ordenador. Esto lo hace muy versátil y podemos encontrar un montón de aplicaciones para este mismo chip. Para usarlo necesitas un circuito programador conectado al puerto serie de un PC y un software.

El circuito a continuación fue diseñado para un despertador, pero variando un poco el programa y haciendo ciertos cambios en el circuito, vamos intentar de usarlo para abrir y cerrar la puerta de un centro comercial.



Los displays son de cátodo común, los cátodos de todos los segmentos están unidos entre sí y se conectan de la siguiente forma: los ánodos del segmento 'a' de cada uno de los displays conectados entre sí. Los ánodos del segmento 'b' de cada uno de los displays conectados entre sí, los ánodos del segmento 'c' de cada uno de los displays conectados entre sí.... y así con todos los segmentos. Y los cátodos (comunes) de cada uno de los displays van por separado a un transistor diferente.

Según esto parece que todos los displays se tendrían que encender con el mismo número, pero no es así ya que el PIC va encendiendo cada display de uno en uno y con el número que le corresponde de forma muy rápida, de tal forma que parece que están todos los displays encendidos a la vez. Esto es una práctica muy común y es muy útil para ahorrarnos cableado y pines del PIC, imagínate la cantidad de pines que tendría que tener si manejáramos todos los segmentos de todos los displays por separado...

Usando los pulsadores del 1 al 3 del circuito lograremos poner nuestro reloj en hora, así como también, setear la hora la cual nos dará el pulso para activar el relevo PL y luego usaremos para activar otro circuito el cual nos accionara un motor para abrir o cerrar la

puerta, por ejemplo abrirla a las 7:00 a.m. y luego cerrarla a las 7:00 p.m. cuando hayan cesado las labores del centro comercial.

Conclusión

La educación es un proceso interminable, puesto que cada día se aprende cosas nuevas o se actualizan las ya conocidas o aprendido, los profesionales debemos estar constantemente actualizando nuestros conocimientos para estar a la par de los avances tecnológicos.

La elaboración de este proyecto nos sirve para comprender un poco más la teoría de los microcontroladores y su aplicación; de ahí la importancia del mismo, pues a través de él se puede apreciar lo interesante que es la programación en nuestra vida cotidiana.

Con la llegada de los PIC se abre un nuevo campo para el diseñador de sistemas automatizados, ya que podemos implementar sistemas que requieren una gran capacidad de proceso con un coste mucho menor que cualquiera de las soluciones existentes. No obstante, no tenemos que lanzarnos alocadamente a programar PIC para todas las aplicaciones, sino que debemos escoger el sistema más adecuado a nuestras necesidades.

Particularmente, creo que la importancia radica en la programación y una buena dirección del proyecto, podemos tener un sistema con microcontroladores y darle diferentes aplicaciones y funciones con solo cambiar el programa.

Después de estudiar varios autores, me atrevo a recomendar los microcontroladores PIC puesto que utilizan la arquitectura Harvard, la cual separa la memoria de programa de la memoria de datos. Esto hace que el dispositivo tenga un bus de datos y un bus de memoria de programa, hecho que permite acceder a ambos simultáneamente. En el caso de la memoria de programa es interna, ya que se encuentra en forma de EPROM o FLASH dentro del propio PIC.

Esta estructura, acompañada de un pipeline, permite que las instrucciones se ejecuten en un sólo ciclo de reloj, salvo en el caso de saltos/rupturas de programa. Lo que se hace internamente es que mientras se ejecuta la instrucción actual se carga la siguiente instrucción en el pipeline, con lo que se alcanza una alta velocidad de ejecución.

Para aprender a usar estos dispositivos, es recomendado tener una tarjeta entrenadora, esto además de los programadores y simuladores, el MPLAB es muy fácil de usar, y también lo considero muy completo, convirtiéndose así en una herramienta indispensable para los que queremos dedicar tiempo para desarrollar proyectos con microcontroladores.

Bibliografía

Microcontroladores PIC 16F84. Desarrollo de proyectos. Fernando Ramiro, Lucas J. López, Alfa omega, 2004

MICROCONTROLADORES: FUNDAMENTOS Y APLICACIONES CON PIC

Pallas, Ramón; Valdés, Fernando (MARCOMBO, EDICIONES TÉCNICAS)

1ª edición, 2007

MICROCONTROLADORES PIC 2ª PARTE. PIC 16F87X. DISEÑO PRÁCTICO DE APLICACIONES

Angulo Usategui, José María; Romero Yesa, Susana & Angulo Martínez, Ignacio (Editorial McGraw-Hill), 2ª edición, 2006

PROGRAMACIÓN DE MICROCONTROLADORES PIC: Desarrollo de 30 proyectos con pic
Primera edición, Ibrahim Dogan, 2007

PROGRAMACIÓN DE MICROCONTROLADORES PIC

IBRAHIM DOGAN (MARCOMBO), 1ª edición 2007.

MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian

User's Guide

MicroChip/ 2005

Páginas de internet

Es.wikibooks.org

Monografías.com

Es.wikipedia.org

electrónica.udea.co

apuntes.rincondelvago.com