

## Prefacio

Este documento está orientado a los estudiantes de cuarto semestre de electrónica del instituto Iutepi, que están interesados en aprender a programar un PIC y ponen todo su empeño en ello, al igual que lo hace el Docente en poner toda la información posible para la mejor comprensión de las bases teóricas. Así pues además de aprender todo sobre el mundo del PIC, es necesario saber algo de electrónica y electrónica digital, de lo contrario todo lo que escriba aquí sobre microcontroladores será difícil de comprender.

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador. Esto suena un poco complicado, pero sólo es un resumen de 3 líneas. A lo largo de este tutorial veremos algunas de las reglas y trucos de este lenguaje complicado por su sencillez

Inicialmente cuando no existían los microprocesadores las personas se ingeniaban en diseñar sus circuitos electrónicos y los resultados estaban expresados en diseños que implicaban muchos componentes electrónicos y cálculos matemáticos. Un circuito lógico básico requería de muchos elementos electrónicos basados en transistores, resistencias, etc., lo cual desembocaba en circuitos con muchos ajustes y fallos; pero en el año 1971 apareció el primer microprocesador el cual originó un cambio decisivo en las técnicas de diseño de la mayoría de los equipos. Al principio se creía que el manejo de un microprocesador era para aquellas personas con un coeficiente intelectual muy alto; por lo contrario con la aparición de este circuito integrado todo sería mucho más fácil de entender y los diseños electrónicos serían mucho más pequeños y simplificados. Entre los microprocesadores más conocidos tenemos el popular Z-80 y el 8085. Los diseñadores de equipos electrónicos ahora tenían equipos que podían realizar mayor cantidad de tareas en menos tiempo y su tamaño se redujo considerablemente; sin embargo, después de cierto tiempo aparece una nueva tecnología llamada microcontrolador que simplifica aun más el diseño electrónico

## Contenido del programa de estudios

- Microcontrolador PIC de gama media
- El Cargador de programas del PIC
- Manejo de Interruptores, Leds, Display y Potencia
- Módulos de Conversión A/D de interrupciones.
- Conclusiones
- Recomendaciones
- Bibliografía

## MICROCONTROLADORES PIC GAMA MEDIA

### Introducción a Microcontroladores

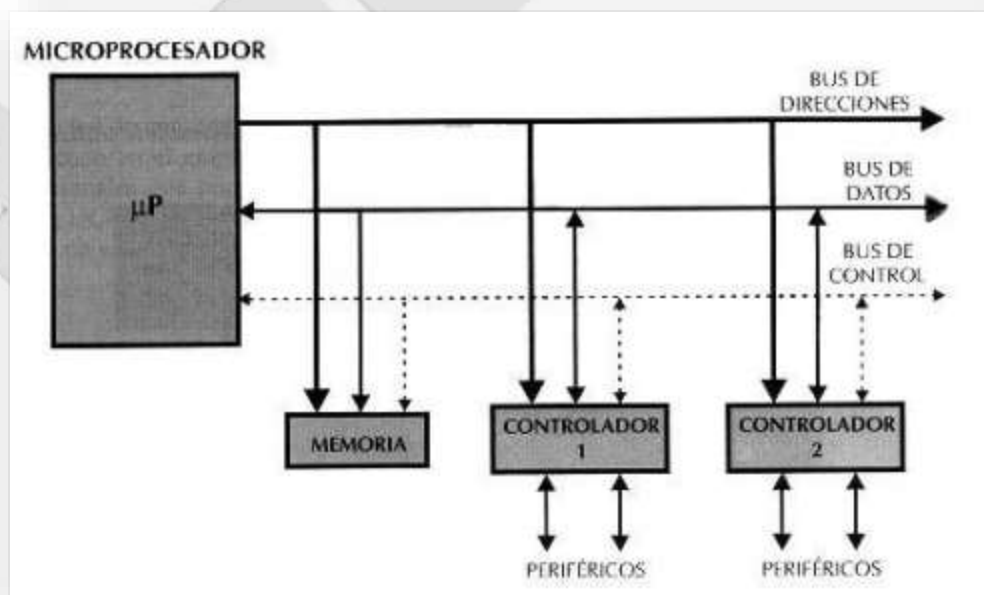
Si has tenido la oportunidad de realizar un diseño con un microprocesador pudiste observar que dependiendo del circuito se requerían algunos circuitos integrados adicionales además del microprocesador como por ejemplo: memorias RAM para almacenar los datos temporalmente y memorias ROM para almacenar el programa que se encargaría del proceso del equipo, un circuito integrado para los puertos de entrada y salida y finalmente un decodificador de direcciones

Un microcontrolador es un solo circuito integrado que contiene todos los elementos electrónicos que se utilizaban para hacer funcionar un sistema basado con un microprocesador; es decir contiene en un solo integrado la Unidad de Proceso, la memoria RAM, memoria ROM , puertos de entrada, salidas y otros periféricos, con la consiguiente reducción de espacio.

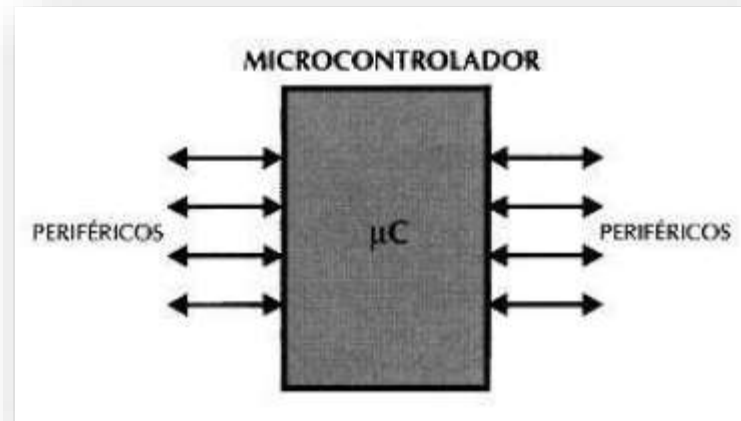
El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado o embebido (embedded controller).

### Ventajas de un microcontrolador frente a un microprocesador

Estas ventajas son reconocidas inmediatamente para aquellas personas que han trabajado con los microprocesadores y después pasaron a trabajar con los microcontroladores. Estas son las diferencias más importantes: Por ejemplo la configuración mínima básica de un microprocesador estaba constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria ROM de 28 Pines y un decodificador de direcciones de 18 pines; pero un microcontrolador incluye todo estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores: En el circuito impreso por su amplia simplificación de circuitería, el costo para un sistema basado



*Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación*



El microcontrolador es un sistema cerrado. Todas las partes del procesador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos en microcontrolador es mucho menor y, lo mejor de todo, el tiempo de desarrollo de su proyecto electrónico se disminuye considerablemente.

### Los microcontroladores hoy día

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos. 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

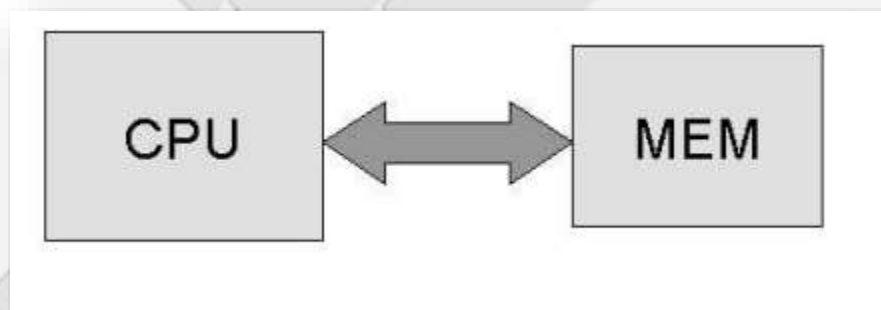
Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes. Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, 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

## TIPOS DE ARQUITECTURAS DE MICROCONTROLADORES

### Arquitectura Von Neumann

La arquitectura tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso (CPU), está conectada a una memoria única donde se guardan las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria. Y el tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior. Resumiendo todo lo anterior, las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

- La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
- La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso

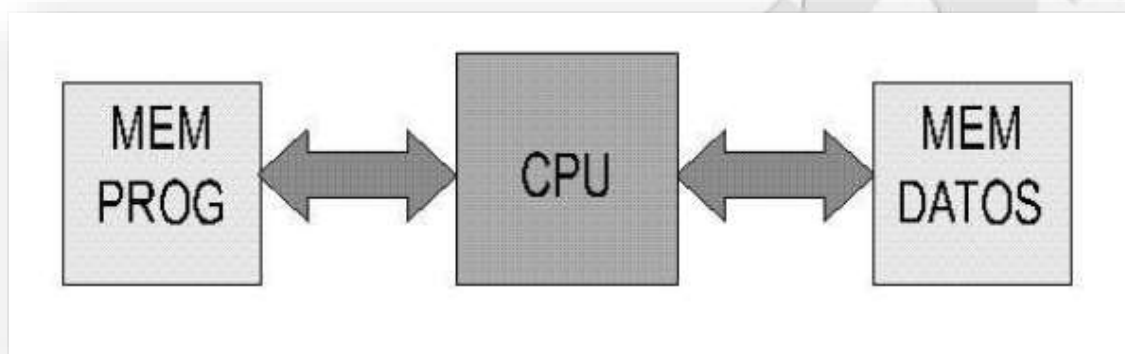


### La arquitectura Harvard

Tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y los otros sólo almacenan datos (Memoria de Datos). Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, al ser los buses independientes, la CPU puede

acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar. Ventajas de esta arquitectura:

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa
- . El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación
- Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).



El microcontrolador PIC 16F84 posee arquitectura Harvard, con una memoria de datos de 8 bits, y una memoria de programa de 14 bits. En la Figura 5 vemos la arquitectura interna organizada en bloques interconectados, en donde se incluye la memoria RAM, la memoria EEPROM, los puertos de entrada y salida (I/O), etc.

## ESTRUCTURA Y ELEMENTOS DE LOS MICROCONTROLADORES

A continuación pasamos a describir los elementos más comunes en todo tipo de microcontroladores y sistemas

### El procesador

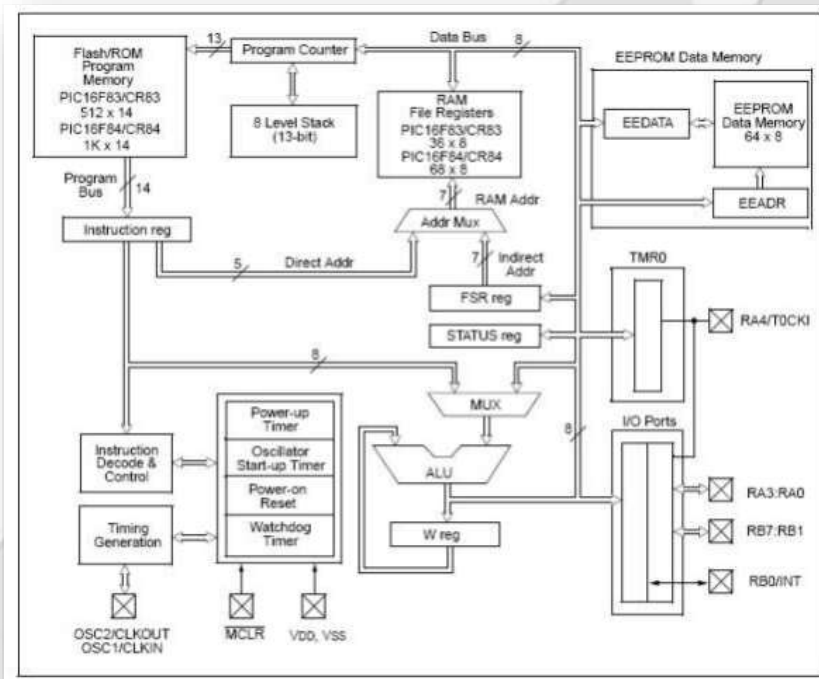
Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción,

así como la búsqueda de los operandos y el almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

### CISC

Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio.

Algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros, es decir, que si las tuviésemos que implementar con instrucciones básicas, acabaríamos con dolor de cabeza.



### RISC

Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y,

generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

## SISC

En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es específico, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

## Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. 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 computadores personales:

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

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

La 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 ROM

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes

Según el tipo de memoria ROM 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. Si tenemos idea de cómo se fabrican los circuitos integrados, sabremos de donde viene el nombre. Estos se fabrican en obleas que contienen varias decenas de chips. Estas obleas se fabrican a partir de procesos fotoquímicas, 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 micrométricos que componen un chip. Ahora ya sabes de donde viene la máscara y no te acostarás sin saber una cosa más. 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 Programable). 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 Programable 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 caros que los microcontroladores con memoria OTP que están hechos con material plástico

### EEPROM, E2PROM o E 2PROM

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 continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño. 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, como ya veremos más adelante

### 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. 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.

### Puertas de Entrada y Salida

Las puertas de Entrada y Salida (E/S) permiten comunicar al procesador con el mundo exterior, a través de interfaces, o con otros dispositivos. Estas puertas, también llamadas puertos, son la principal utilidad de las patas o pines de un microprocesador. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control

### Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Esta señal del reloj es el motor del sistema y la que hace que el programa y los contadores avancen.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía y de calor generado.

### Recursos especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

- Los principales recursos específicos que incorporan los microcontroladores son:
- Temporizadores o Timers.
- Perro guardián o Watchdog
- Protección ante fallo de alimentación o Brownout.
- Estado de reposo o de bajo consumo (Sleep mode).
- Conversor A/D (Analógico ->Digital). Conversor D/A (Digital ->Analógico).
- Comparador analógico.

- Modulador de anchura de impulsos o PWM (Pulse Wide Modulation).
- Puertas de E/S digitales.
- Puertas de comunicación

A continuación pasamos a ver con un poco más de detalle cada uno de ellos.

### Temporizadores o Timers

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos. Vaya, nos hemos metido en términos de microcontroladores demasiado pronto. Bueno, con el fin de aclarar que es un registro, anticipamos que es un valor numérico en una posición fija de memoria. Un ejemplo: esto es igual que el segundero de nuestro reloj digital, este va aumentando hasta que llega a 60 segundos, pero en la pantalla pone 00, esto quiere decir que se desborda. Pero cuando cambia da un aviso y se incrementan los minutos. En este ejemplo, el registro es el segundero; estos son fijos ya que sabemos que son los de la derecha del todo y no se van a cambiar

### Perro guardián o Watchdog

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicia el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro Guardián consiste en un contador que, cuando llega al máximo, provoca un reset automáticamente en el sistema

Se debe diseñar el programa de trabajo que controla la tarea de forma que resetee al Perro Guardián de vez en cuando antes de que provoque el reset. Si falla el programa o se bloquea (si cae en bucle infinito), no se refrescará al Perro guardián y, al completar su temporización, provocará el reset del sistema

### Protección ante fallo de alimentación o Brownout

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (brownout). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor. Esto es muy útil para evitar datos erróneos por transiciones y ruidos en la línea de alimentación

### Estado de reposo ó de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para

ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se congelan sus circuitos asociados, quedando sumido en un profundo sueño. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo. Para hacernos una idea, esta función es parecida a la opción de Suspend en el menú para apagar el equipo (en aquellos PCs con administración avanzada de energía)

### Convertor A/D (CAD)

Los microcontroladores que incorporan un Convertor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patillas del circuito integrado

### Convertor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patillas del chip. Existen muchos circuitos que trabajan con señales analógicas

### Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores

### Modulador de anchura de impulsos o PWM

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

### Puertos digitales de E/S

Todos los microcontroladores destinan parte de su patillaje a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos

Las líneas digitales de las Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración. Otra vez más nos volvemos a meter con unos, ceros y registros paciencia que pronto llega lo mejor

### Puertas de comunicación

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

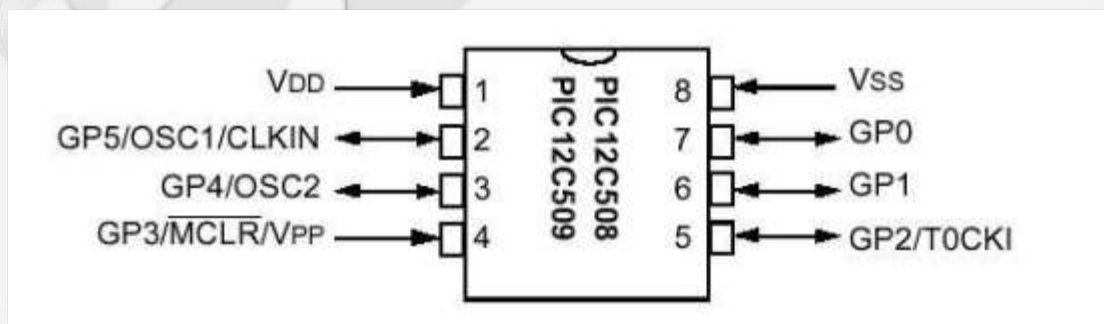
- UART, adaptador de comunicación serie asíncrona.(Ej: Puerto Serie)
- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC. Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips
- CAN (Controller Area Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850

Tanto el I2C en televisores, como el Bus CAN en automóviles, fueron diseñados para simplificar la circuitería que supone un bus paralelo de 8 líneas dentro de un televisor, así como para librar de la carga que supone una cantidad ingente de cables en un vehículo

## LA FAMILIA DE LOS PIC

La Familia PIC Esta familia, desarrollada por la casa Microchip, se divide en cuatro gamas, gamas enana, baja, media y alta. Las principales diferencias entre estas gamas radica en el número de instrucciones y su longitud, el número de puertos y funciones, lo cual se refleja en el encapsulado, la complejidad interna y de programación, y en el número de aplicaciones. En las próximas líneas pasamos a describir brevemente las cualidades de esta familia. Gama baja o gama enana, de 8 patas. Se trata de un grupo de PIC de reciente aparición que ha acaparado la atención del mercado. Su principal característica es su reducido tamaño, al disponer todos sus componentes de 8 patitas.

Se alimentan con un voltaje de corriente continua comprendido entre 2,5 V y 5,5 V, y consumen menos de 2 mA cuando trabajan a 5 V y 4 MHz. El formato de sus instrucciones puede ser de 12 o de 14 bits y su repertorio es de 33 o 35 instrucciones, respectivamente. En la Figura 6 se muestra el diagrama de conexionado de uno de estos PIC



Aunque los PIC enanos sólo tienen 8 patillas, pueden destinar hasta 6 como líneas de E/S para los periféricos porque disponen de un oscilador interno R-C, lo cual es una de sus principales características.

En la Figura 7 se presentan las principales características de los modelos de esta subfamilia, que el fabricante tiene la intención de potenciar en un futuro próximo.

Los modelos 12C5xx pertenecen a la gama baja, siendo el tamaño de las instrucciones de 12 bits; mientras que los 12C6xx son de la gama media y sus instrucciones tienen 14 bits. Los modelos 12F6xx poseen memoria Flash para el programa y EEPROM para los datos.

Se trata de una serie de PIC de recursos limitados, pero con una de la 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 teniendo en cuenta su bajo consumo (menos de 2 mA a 5 V y 4 MHz). Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo

MODELO	MEMORIA PROGRAMA	MEMORIA DATOS	FRECUENCIA MAXIMA	LINEAS E/S	ADC 8BITS	TEMPORIZADORES	PATITAS
PIC12C508	512x12	25x8	4 MHz	6		TMRO + WDT	8
PIC12C509	1024x12	41x8	4 MHz	6		TMRO + WDT	8
PIC12C670	512x14	80x8	4 MHz	6		TMRO + WDT	8
PIC12C671	1024x14	128x8	4 MHz	6	2	TMRO + WDT	8
PIC12C672	2048x14	128x8	4 MHz	6	4	TMRO + WDT	8
PIC12C680	512X12 FLASH	80x8 16x8 EEPROM	4 MHz	6	4	TMRO + WDT	8
PIC12C681	1024x14 FLASH	80x8 16x8 EEPROM	4 MHz	6		TMRO + WDT	8

Principales características de la gama baja

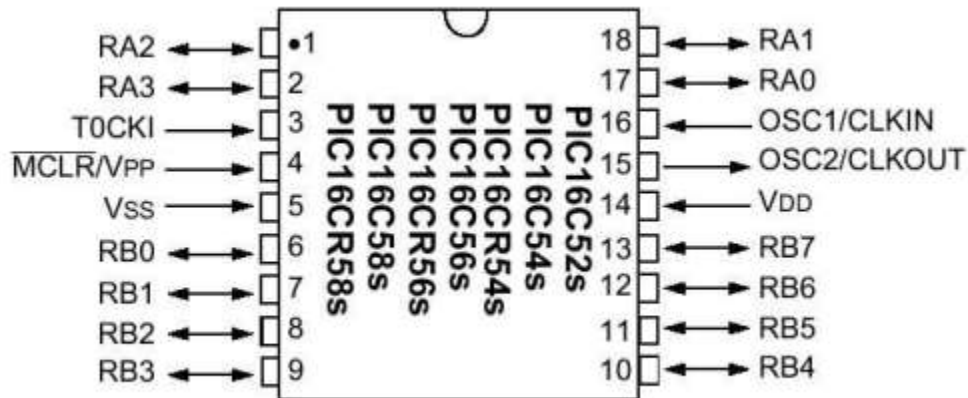


Diagrama de patas de los PIC de la gama baja que responden a la nomenclatura PIC16C54/56

Al igual que todos los miembros de la familia PIC16/17, los componentes de la gama baja se caracterizan por poseer los siguientes recursos: Sistema Power On Reset, Perro guardián (Watchdog o WDT), Código de protección, Sep, etc. Sus principales desventajas o limitaciones son que la pila sólo tiene dos niveles y

que no admiten interrupciones. En la Figura 9 se presentan las principales características de los modelos de esta subfamilia.

### Gama media. PIC16CXXX con instrucciones de 14 bit

Es la gama más variada y completa de los PIC. Abarca modelos con encapsulado desde 18 patas hasta 68, cubriendo varias opciones que integran abundantes periféricos. Dentro de esta gama se halla el «famoso PIC16X84» y sus variante

En esta gama sus componentes añaden nuevas prestaciones a las que poseían los de la gama baja, haciéndoles más adecuados en las aplicaciones complejas. Admiten interrupciones, poseen comparadores de magnitudes analógicas, convertidores A/D, puertos serie y diversos temporizadores. El repertorio de instrucciones es de 35, de 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. En la Figura 10 se presentan las principales características de los modelos de esta familia. En cuadrado en la gama media también se halla la versión PIC14C000, que soporta el diseño de controladores inteligentes para cargadores de baterías, pilas pequeñas, fuentes de alimentación ininterrumpibles y cualquier sistema de adquisición y procesamiento de señales que requiera gestión de la energía de alimentación. Los PIC 14C000 admiten cualquier tecnología de las baterías como Li-Ion, NiMH, NiCd, Ph y Zinc

El temporizador TMR1 que hay en esta gama tiene un circuito oscilador que puede trabajar asíncronamente y que puede incrementarse aunque el microcontrolador se halle en el modo de reposo (sleep), posibilitando la implementación de un reloj en tiempo real. Las líneas de E/S presentan una carga pull-up activada por software

MODELO	MEMORIA PROGRAMA (x12 BITS)		MEMORIA DATOS (bytes)	FRECUENCIA MÁXIMA	LÍNEAS E/S	TEMPORIZADORES	PATITAS
	EPROM	ROM					
PIC16C52	384		25	4 MHz	4	TMRO + WDT	18
PIC16C54	512		25	20 MHz	12	TMRO + WDT	18
PIC16C54A	512		25	20 MHz	12	TMRO + WDT	18
PIC16CR54A		512	25	20 MHz	12	TMRO + WDT	18
PIC16C55	512		24	20 MHz	20	TMRO + WDT	28
PIC16C56	1 K		25	20 MHz	12	TMRO + WDT	18
PIC16C57	2 K		72	20 MHz	20	TMRO + WDT	28
PIC16CR57B		2 K	72	20 MHz	20	TMRO + WDT	28
PIC16C58A	2 K		73	20 MHz	12	TMRO + WDT	18
PIC16CR58A		2 K	73	20 MHz	12	TMRO + WDT	18

Figura 9: Características de los modelos PIC16C(R)5X de la gama baja

MODELO	MEMORIA PROGRAMA	MEMORIA DATOS		REGISTROS ESPECÍFICOS	TEMPORIZADORES	INTERRUPTIONES	E/S	RANGO VOLTAJE	PATITAS
		RAM	EEPROM						
PIC16C84	1Kx14 EEPROM	36	64	11	TMRO + WDT	4	13	2-6	18
PIC16F84	1Kx14 FLASH	60	64	11	TMRO + WDT	4	13	2-6	18
PIC16F83	512x14 FLASH	36	64	11	TMRO + WDT	4	13	2-6	18
PIC16CB84	1Kx14 ROM	60	64	11	TMRO + WDT	4	13	2-6	18
PIC16CB83	512x14 ROM	36	64	11	TMRO + WDT	4	13	2-6	18

Figura 10: Características relevantes de los modelos PIC16X8X de la gama media.

### 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, un multiplicador hardware de gran velocidad y mayores capacidades de memoria, que alcanza los 8K palabras en la memoria de instrucciones y 454 bytes en la memoria de datos

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 facultad obliga a estos componentes a tener un elevado número de patitas comprendido entre 40 y 44. 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. En la Figura 11 se muestran las características más relevantes de los modelos de esta gama, que sólo se utilizan en aplicaciones muy especiales con grandes requerimientos

MODELO	MEMORIA PROGRAMA	MEMORIA DATOS RAM	REGISTROS ESPECIALES	TIEMPOZAJORES	CAP	PWM	CAD SERI	INTERRUPCIONES	I/O	MULTIPLICADOR HARDWARE	PATITAS
PIC17C42A	2Kx16	252	48	4 + WDT	2	2		11	33	8x8	45/44
PIC17C43	4Kx16	454	48	4 + WDT	2	2		11	33	8x8	45/44
PIC17C44	8Kx16	454	48	4 + WDT	2	2		11	33	8x8	45/44
PIC17C432	8Kx16	454	76	4 + WDT	4	3	12	18	50	8x8	64/58
PIC17C442	16Kx16	452	76	4 + WDT	4	3	12	18	50	8x8	64/58

Figura 11: Características más destacadas de los modelos PIC17CXXX de la gama alta.

## Programación

Un microcontrolador es un computador completo de limitadas prestaciones, que está contenido en un chip de un circuito integrado y se destina a gobernar una sola tarea. Para que el microcontrolador realice las operaciones que deseamos es necesario grabar en su memoria de programa un conjunto de instrucciones que constituyen el programa de aplicación. En primer lugar debemos conocer perfectamente las especificaciones de la tarea que debe ejecutar. Hay que tener en cuenta que podemos llegar al mismo resultado con distintos programas, siendo óptimo aquél que esté mejor estructurado y no realice operaciones innecesarias, ahorrando de esta forma espacio en la memoria de programa y tiempo.

Podemos citar como ejemplos de programas de aplicación a: el programa que controla un semáforo, el programa que controla un brazo de robot, el programa que controla una alarma de casa, entre otras aplicaciones. Las distintas aplicaciones requieren de ingreso y salida de datos hacia y desde la parte interna del PIC, esta tan importante función la cumplen los puertos del PIC, que dependiendo de la gama pueden tener de 2 a 5 puertos, los cuales pueden ser configurados como entrada o salida, según los requerimientos de la aplicación

Configurar correctamente los puertos del PIC es lo primero que un programador debe aprender, seguido del manejo del software de desarrollo MPLAB, y un buen simulador electrónico por ejemplo el PROTEUS que a criterio de muchos es el mejor. En la siguiente figura presentamos uno de los primeros microcontroladores en llegar a nuestro país, uno de mucha aceptación y aplicación, ideal para los principiantes.

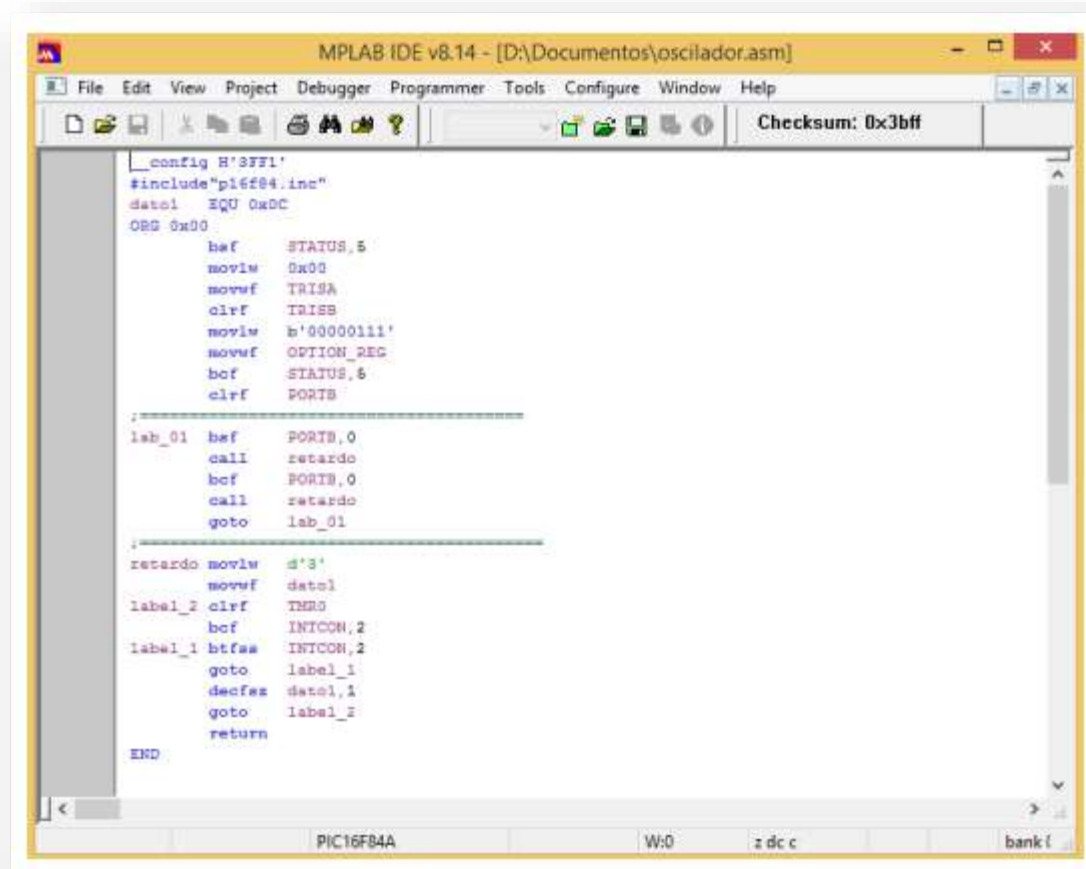
Para escribir el programa fuente de la tarea o aplicación, será necesario un editor de texto que trabaje con caracteres ASCII. El lenguaje de programación más utilizado es el lenguaje ensamblador, aunque también se usa el lenguaje Basic o el C. El lenguaje ensamblador es el lenguaje natural de los PIC, y está más cerca al lenguaje máquina, como todo lenguaje presenta ventajas y desventajas, entre otras ventajas podemos citar las siguientes:

1. Permite utilizar menos memoria de programa.
2. Permite optimizar el tiempo de ejecución.
3. Permite trabajar directamente con los registros.
4. Facilita la paginación de la memoria de programa.
5. Permite el uso directo del temporizador y contador

Recomendamos utilizar el lenguaje assembler, todas las aplicaciones desarrolladas en este texto están en dicho lenguaje, solo es una recomendación, se puede trabajar con el lenguaje C o Basic, esto depende del programador

Para desarrollar aplicaciones, proyectos entre otros, se requieren de herramientas tanto lógicas como físicas, es decir software y hardware respectivamente, respecto al software la herramienta principal es el conocido y famoso MPLAB IDE, que lo proporciona y facilita el mismo fabricante de los PIC, es decir MICROCHIP, es necesario indicar que existen diferentes marcas de PIC, podemos citar algunas como por ejemplo: ATMEL, MOTOROLA, INTEL y MICROCHIP, entonces porque elegimos la marca MICROCHIP, es simple, estos PIC son más comerciales, son económicos y hay mucha información de los mismos en INTERNET, como proyectos, manuales, tutoriales, vídeos, software libre, simuladores, sin desmerecer a las otras marcas.

Otra herramienta lógica es el simulador, que es de mucha utilidad cuando se requiere analizar un prototipo inicial, analizar los registros internos, analizar la secuencia de ejecución del programa, analizar las subrutinas e interrupciones, como vemos esta herramienta es de mucha utilidad, permite detectar y corregir errores en el programa, antes de implementar el proyecto. En la siguiente figura se ve el entorno principal del MPLAB, que como todo software tiene diferentes versiones, hace muchos años atrás inicio con la versión 5, ahora está en la 8.



```

MPLAB IDE v8.14 - [D:\Documentos\oscilador.asm]
Checksum: 0x3bff

__config H'3FF1'
#include "p16f84.inc"
datol EQU 0xDC
ORG 0x00
    bcf     STATUS,5
    movlw  0x00
    movwf  TRISA
    clrf   TRISB
    movlw  b'00000111'
    movwf  OPTION_REG
    bcf    STATUS,5
    clrf   PORTB

lab_01 bcf     PORTB,0
       call   retardo
       bcf    PORTB,0
       call   retardo
       goto  lab_01

retardo movlw  d'3'
       movwf  datol
label_2 clrf   TRISB
       bcf    INTCON,2
label_1 btfss  INTCON,2
       goto  label_1
       decfsz datol,1
       goto  label_2
       return
END
    
```

*El MPLAB versión 8.14, software que permite el desarrollo de aplicaciones, facilita un editor de texto, ensambla y compila los códigos.*

Otra herramienta lógica clave es el grabador de PIC, quien se encarga de enviar el programa hacia la memoria interna del PIC, es decir realiza la transferencia del programa de la PC hacia el PIC, esta herramienta tiene su complemento que es la parte física, es decir un circuito físico sin el cual no sirve de nada. El grabador es una tarjeta electrónica, que posee entre otros elementos un zócalo para insertar en PIC, un puerto de comunicaciones, que puede ser serial, USB, en algunos casos paralelo

Existen diferentes modelos de grabadores, desde el punto de vista del puerto pueden ser serial o paralelo, desde el punto de vista de la alimentación eléctrica, pueden ser con alimentación o sin alimentación, pueden ser universales o no universales, los grabadores actuales son universales, usan el puerto USB y no requieren alimentación externa adicional, es decir toman energía del mismo puerto USB de la PC, en la siguiente figura se observa un grabador de PIC universal, que utiliza como parte de su circuito a otro PIC que normalmente es el 18F2550



*El grabador de PIC universal, no requiere alimentación externa, hace uso de otro microcontrolador para lograr la comunicación serial*

Como ya hemos visto existen herramientas lógicas y físicas, que permiten desarrollar al 100% los proyectos basados en microcontrolador, otra herramienta física (hardware) importante es el entrenador de PIC, que es una tarjeta electrónica básicamente de desarrollo, que puede tener diferentes elementos desde los más básicos hasta los más complejos, normalmente vienen con un PIC, uno o dos display, pulsadores digitales, conectores, teclado matricial, pantalla LCD entre otros, esta tarjeta permite ver en tiempo real el funcionamiento del programa, es decir si funciona o no el programa, es de mucha utilidad cuando se desarrollan aplicaciones con PIC, en la siguiente figura se muestra un entrenador básico pero de mucha utilidad.



*El entrenador básico para el microcontrolador 16F84 de MICROCHIP*

De no contar con esta herramienta física, se puede optar por usar un protoboard, que es una base fija con diferentes conexiones internas, estas conexiones son uniformes y siguen un patrón único, el único inconveniente es el ruido electromagnético, ruido que se puede filtrar por los cables o terminales de los diferentes componentes, por lo que se recomienda reducir al máximo el tamaño de los terminales de los diferentes dispositivos y mantener un orden en el cableado. El ruido electromagnético, puede alterar el normal funcionamiento del PIC.

### Los puertos del PIC

Los puertos del PIC son un conjunto de pines, líneas, que forman en su conjunto el puerto en sí, no se debe confundir un pin con un puerto, el pin es una parte del puerto, entonces un conjunto de pines forman un puerto, por ejemplo el puerto B posee un total de 8 pines, que pueden ser configurados como entrada o salida, los mismos que cumplen una función elemental, que es permitir el ingreso o salida de datos, desde y hacia la parte interna del PIC, que no es otra cosa que comunicar la parte interna del PIC con el mundo externo, entonces un factor importante para el desarrollo de proyectos basados en microcontroladores es la interfaz de entrada y salida. A través de los pines del PIC asociados a los puertos de entrada y salida, el PIC puede interactuar con otros circuitos externos, enviando señales digitales en forma de unos y ceros, señales de control o comando, recibiendo señales digitales en forma de unos y ceros, que no son más que estímulos correspondientes a variables externas.

Los pines configurados como entrada pueden adquirir datos interpretando el valor de voltaje como un valor lógico 0 o 1, si el voltaje a la entrada es de 0 voltios el PIC lo interpreta como un 0 lógico, si el voltaje a la entrada es de 5 voltios el PIC lo interpreta como un 1 lógico, debemos indicar que el voltaje a la

entrada no tiene que ser exacto, existe un rango de valores que son aceptados por el PIC, este rango de valores lo determina el fabricante en su data sheet, para el caso de las salidas, es lo mismo, si un puerto está configurado como salida, sus pines se energizan con 5 voltios, lo que corresponde a un 1 lógico como salida, esta señal puede activar cualquier otro sistema, activar un motor, o simplemente prender un led y pone sus pines a 0 voltios cuando la salida es un 0 lógico, esto apaga cualquier sistema, motor o led

or lo general los puertos de datos son bidireccionales, es decir pueden configurarse como entrada o salida, según los requerimientos de la aplicación, para configurar un puerto como entrada o salida se utilizan los registros TRISA y TRISB, siendo esta la única función de dichos registros, el procedimiento para configurar los puertos lo veremos en los siguientes puntos.

Gracias a sus puertos el microcontrolador puede responder a eventos externos y realizar una cierta acción, como variar las señales de salida de acuerdo al valor en las entradas. Para responder a eventos externos, el microcontrolador cuenta con un recurso conocido como interrupciones. Las interrupciones son eventos aleatorios, que se generan dentro o fuera del microcontrolador, detienen la ejecución del programa principal, para atender un programa secundario o subprograma, el uso y aplicación de las interrupciones se explicará en los siguientes capítulos

Una aplicación real es por ejemplo, un botón pulsador conectado a un pin de entrada. Una vez pulsado, se genera una señal de interrupción, que iniciará la ejecución de la subrutina de interrupción, que puede ser un temporizador, que al finalizar el tiempo programado active una señal de salida, esta señal podría apagar un horno por ejemplo o simplemente encender un led

Para determinar las necesidades de entradas y salidas del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos, hardware externo o cambiar a otro microcontrolador más adecuado a ese sistema.

### Los puertos A y B

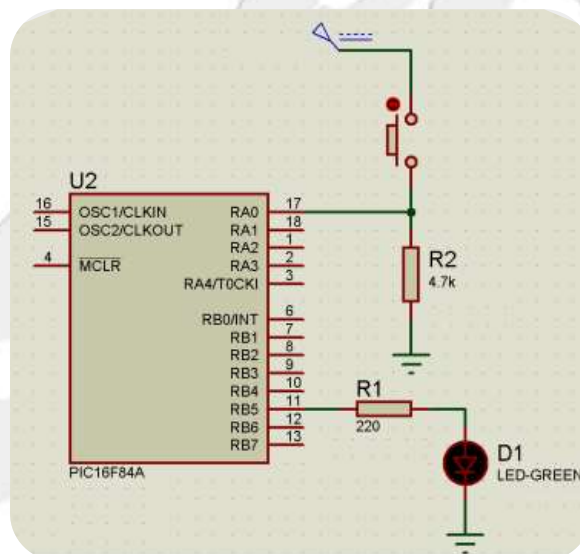
Los puertos del microcontrolador 16F84 de Microchip, son un conjunto de pines o líneas que están agrupados, cada grupo de pines o líneas conforma un puerto completo, el 16F84 tiene implementado dos puertos, el puerto A y el puerto B, ambos puertos son bidireccionales y pueden ser configurados como entrada o salida, de esta forma los puertos permiten el ingreso o salida de datos hacia o desde la parte interna del microcontrolador, el puerto A está conformado por cinco pines o líneas, que van desde el RA0 hasta el RA4, como se observa en la figura 1.5, el pin 3 está multiplexado, es decir cumple doble función, que corresponde al RA4/T0CKI, éste es el único pin del puerto A que está multiplexado, y puede trabajar como puerto RA4 o como entrada del contador de eventos externos T0CKI, para este último el pin 3 debe ser configurado como entrada

El pin 3 cumple doble función, y no se puede utilizar ambas funciones al mismo tiempo, a través de la configuración se elige que función cumplirá dicho pin, y si será de entrada o salida, el otro puerto con más pines es el puerto B, y está conformado por ocho pines o líneas, que van desde el RB0 hasta el RB7, como

se observa en la figura 1.5, el pin 6 está multiplexado, es decir cumple doble función, que corresponde al RB0/INT, éste es el único pin o línea que está multiplexado, y puede trabajar como puerto RB0 o como entrada de interrupción externa INT, pero nunca como puerto y entrada de interrupción externa al mismo tiempo, a través de la configuración se elige la función que cumplirá el pin o línea y si será de entrada o salida

Los puertos A y B del microcontrolador pueden ser configurados como entrada o salida, según los requerimientos de la aplicación, si el puerto está configurado como salida, los pines del puerto trabajan en modo fuente, suministran corriente, y la máxima corriente que puede suministrar una línea o pin es de 20mA, ya sea del puerto A o del puerto B, y la máxima corriente que puede suministrar todo un puerto es de 50mA, para el puerto A, y de 100mA, para el puerto B, si el puerto está configurado como entrada, los pines trabajan en modo sumidero, reciben corriente, y la máxima corriente que puede ingresar por una línea o pin es de

25mA, la corriente máxima de entrada para todo el puerto es de 80mA, para el puerto A y de 150mA para el puerto B. Si en la aplicación existen cargas que consumen corrientes mayores, como por ejemplo: motores paso a paso, relés, varios leds y otros periféricos que necesiten más corriente de la indicada por el fabricante, habrá que aplicar un circuito acoplador como por ejemplo: los buffers, transistores de potencia, que se encarguen de controlar la corriente.



*El pin RA0 configurado como entrada y trabajando en modo sumidero, el pin RB5 configurado como salida y trabajando en modo fuente*

Se debe activar las resistencias de pull-up cada vez que sea necesario, y no dejar pines o líneas de entrada en estado flotante, en la electrónica digital se manejan dos estados lógicos, que son el 1 y el 0, por ejemplo: un sistema basado en microcontrolador con entradas y salidas, si una de las entradas se conecta a la referencia negativa a través de un interruptor, observando la conexión, veremos que el cero ya está definido por el interruptor cuando éste está cerrado, pero si el interruptor no se activa, es decir está abierto, no hay un estado definido porque la entrada no tiene contacto con la referencia negativa, sería un pin sin conexión, esto se llama estado flotante y puede causar el mal funcionamiento del sistema.

Ese problema se soluciona activando una resistencia interna entre el pin de entrada y Vdd, llamada resistencia de pull-up, su objetivo es asegurar un 1 lógico cuando el interruptor esté abierto. Si lo que se desea es asegurar un 0 lógico, la resistencia se conecta a Vss y recibe el nombre de resistencia de pull-down. Se aconseja no dejar un pin sin conexión o en estado flotante, si un pin o línea está configurado como entrada, y está en estado flotante, por ese pin ingresan unos y ceros aleatoriamente, eso altera el normal funcionamiento del microcontrolador

### Programación PORTA y PORTB

Es necesario indicar que existen dos registros internos asociados a los puertos, no se debe confundir registro con puerto, así como existe un puerto A con 5 pines, existe un registro interno llamado PORTA, el puerto A con el registro PORTA están íntimamente relacionados, ya explicamos que un puerto es un conjunto de pines a través del cual ingresan o salen datos, pero esos datos cuando ingresan al PIC lo hacen directamente a un registro interno, o cuando salen lo hacen desde un registro interno, esa es la relación íntima a la que hacemos referencia, un registro es un conjunto de bits, para el PIC 16F84 todos sus registros internos son de 8 bits, como se observa en la figura.

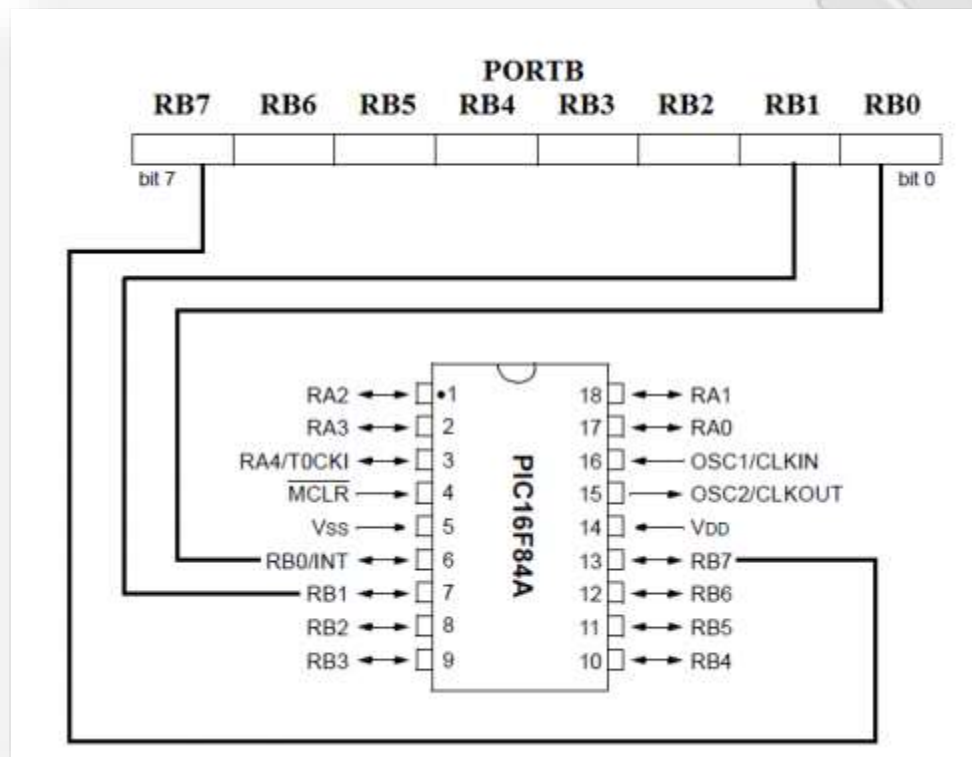
STATUS REGISTER (ADDRESS 03h, 83h)								
R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	
bit 7								bit 0

*El registro STATUS o de estado, está conformado por ocho bits y cada bit cumple una función importante*

El puerto B tiene ocho pines, y su registro correspondiente está formado por ocho bits, cada pin se corresponde con un bit, por ejemplo: el pin RB0 se corresponde con el bit 0 del registro PORTB, el pin RB1 se corresponde con el bit 1 del registro PORTB, y eso ocurre con todos los pines, como se muestra en la figura 1.8, para el puerto A es lo mismo, pero existe una diferencia, el puerto A tiene solamente 5 pines y su registro correspondiente PORTA tiene 8 bits, la relación es la siguiente, el bit 0 del registro PORTA se conecta con el pin RA0 del puerto A, el bit 1 se conecta con el RA1, el bit 2 se conecta con el RA2, como no

existen los pines RA5, RA6 y RA7, los bits 5, 6 y 7 del PORTA no se utilizan, pero si existen internamente y su valor por defecto es cero

Estos registros en algunos casos son llamados palabras o bytes, que significa lo mismo, en los registros se puede realizar las operaciones de lectura o escritura de datos, aplicando las instrucciones de movimiento de datos MOV y sus variantes, deseo precisar que hay algunos registros cuyos bits son de solo lectura, por ejemplo: el bit Z del registro STATUS, el bit RBIF del registro INTCON, entre otros bits. Esos bits son controlados por el sistema y no por el usuario.



*La relación que existe entre el puerto B y registro PORTB, la relación es de pin a bit el*

El desconocimiento de los atributos de los bits, lectura o escritura, los bancos de memoria y sus registros, el direccionamiento de los bancos de memoria, pueden generar errores en el programa y en consecuencia el mal funcionamiento de la aplicación. En el PIC existen muchos registros internos, estos registros están contenidos en bancos de memoria, un banco de memoria no es más que un conjunto de registros ordenados secuencialmente, cada registro tiene un nombre técnico y único, que lo diferencia de los otros registros, además ocupa una posición en el banco de memoria, esa posición es la dirección de memoria y se representa por un número en formato hexadecimal, hay microcontroladores que tienen dos hasta cuatro bancos de memoria, el 16F84 tiene dos bancos de memoria, a cada banco se le asigna un nombre,

banco 0 y banco 1, se debe tener muy en cuenta que registros están en el banco 0 y que registros están en el banco 1, hay registros que únicamente se encuentran en el banco 0, por ejemplo: PORTA y PORTB, y hay registros que únicamente se encuentran en el banco 1, por ejemplo: TRISA y TRISB, y hay registros que se encuentran en ambos bancos, por ejemplo: STATUS y PLC, antes de utilizar uno de estos registros se debe direccionar o seleccionar correctamente el banco de memoria

En los bancos de memoria existe un espacio de 68 bytes que el fabricante dejó libre, este espacio es conocido como la memoria RAM y es ocupado por las variables del programa, este espacio inicia en la dirección 0x0C y termina en la dirección 0x4F del banco 0, de acuerdo al espacio libre se pueden declarar hasta 68 variables, que es una cantidad suficiente para el desarrollo de aplicaciones, el procedimiento para direccionar los bancos de memoria se explicará en los siguientes puntos, en la figura se muestra la representación gráfica de los dos bancos de memoria del PIC 16F84, el cual fue copiado del manual del fabricante.



Los dos bancos de memoria del PIC 16F84, el bank 0 y el bank 1, los 68 bytes de memoria RAM para las variable

Como se observa en la figura anterior, el PIC 16F84 tiene dos bancos de memoria, llamados bank 0 y bank 1, en el bank 0 se encuentra la memoria RAM, que inicia en la posición 0Ch y termina en la posición 4Fh, también se observa que de la posición 50h hasta la 7Fh no está implementado y se lee como cero,

## Programación TRISA y TRISB

Para el desarrollo de aplicaciones debemos configurar los puertos del microcontrolador como entrada o salida, según sea el caso, si le asignamos un cero 0 a un pin éste será salida y si le asignamos un uno 1 éste será entrada, ésta asignación de pines se hace programando los registros TRISA y TRISB, el TRISA es un registro donde se almacenan los bits que definen un pin como entrada o salida del puerto A. Recordemos que el puerto A sólo tiene 5 pines, por lo tanto los tres últimos bits no se utilizan

Los registros TRISA y TRISB sirven para configurar los puertos A y B del microcontrolador, configuran cada pin de los puertos, definen si el pin será de entrada o salida, existe una correspondencia entre el registro TRISA y el puerto A, el registro TRISA configura el puerto A, bit a bit, si el bit 0 del TRISA vale 0, entonces el pin RA0 del puerto A queda configurado como salida, si el bit 3 del TRISA vale 1, entonces el pin RA3 del puerto A queda configurado como entrada, es así para todos los bits y pines, para todos los puertos existentes, siempre debemos recordar que:

- Un 1 configura el pin como entrada
- Un 0 configura el pin como salida
- TRISA para los pines del PUERTO A (5 bits)

TRISB para los pines del PUERTO B (8 bits)

- El bit 0 del TRISA configura al RA0

El bit 1 del TRISA configura al RA1

- El bit 2 del TRISA configura al RA2
- El bit 3 del TRISA configura al RA3

Cuando el PIC arranca todos sus puertos están configurados como entrada, el banco seleccionado es el 0, todo esto es por defecto, es importante verificar en que banco se encuentran los registros a utilizar, los registros TRISA y TRISB se encuentran en el banco 1, antes de leer o escribir en estos registros, se debe acceder al banco 1, esto se logra a través del registro STATUS, escribiendo un 1 en el bit 5, utilizando las instrucciones correspondientes

Por ejemplo: Si TRISA es igual a 00011110, entonces todos los pines del puerto A serán entradas, salvo RA0 que está como salida. Tal como se observa en la tabla 1.1

Si TRISB es igual a 00000001, entonces todos los pines del puerto B serán salidas, salvo RB0 que está como entrada. Tal como se observa en la tabla.

TRISA		PUERTO A	
BIT	VALOR	PIN	ESTADO
Bit 0	0	RA0	SALIDA
Bit 1	1	RA1	ENTRADA
Bit 2	1	RA2	ENTRADA
Bit 3	1	RA3	ENTRADA
Bit 4	1	RA4	ENTRADA
Bit 5	0	NO EXISTE	INDETERMINADO
Bit 6	0	NO EXISTE	INDETERMINADO
Bit 7	0	NO EXISTE	INDETERMINADO

*Configuración del puerto A, relación entre el valor del registro TRISA y el estado del puerto A, es decir entrada o salida*

TRISB		PUERTO B	
BIT	VALOR	PIN	ESTADO
Bit 0	1	RB0	ENTRADA
Bit 1	0	RB1	SALIDA
Bit 2	0	RB2	SALIDA
Bit 3	0	RB3	SALIDA
Bit 4	0	RB4	SALIDA
Bit 5	0	RB5	SALIDA
Bit 6	0	RB6	SALIDA
Bit 7	0	RB7	SALIDA

*Configuración del puerto B, relación entre el valor del registro TRISB y el estado del puerto B, es decir entrada o salida*

El código ensamblador para configurar el puerto A y el puerto B, del ejemplo anterior y de acuerdo a las tablas. Es el siguiente:

```

bsf     STATUS,5    ;Acceder al banco 1
movlw   b'00000001'
movwf   TRISB      ;TRISB igual a 00000001
movlw   b'00011110'
movwf   TRISA      ;TRISA igual a 00011110
bcf     STATUS,5    ;Acceder al banco 0
    
```

## Periféricos

El microcontrolador 16F84 es un chip con diferentes periféricos, los puertos forman parte de estos periféricos, pero existen otros que son de mucha importancia ver la figura 1.10, como se observa en la figura, el microcontrolador tiene un total de 18 pines, cada pin cumple una función elemental, por ejemplo: el reset, osc, Vss, Vdd, entre otros. Veamos

Los pines 5 y 14, son respectivamente las patitas de masa y alimentación Vss y Vdd, Vdd está comprendido entre 2V y 6V, valor típico 5V, existe un rango de tolerancia que lo indica el fabricante, no necesariamente es un valor exacto de 5V, se recomienda utilizar el regulador de voltaje LM7805 y verificar que el voltaje de rizado sea mínimo

Los pines 15 y 16, OSC1/CLKIN y OSC2/CLKOUT, son los pines de entrada para la señal de reloj, un tren de pulsos a una determinada frecuencia, esta frecuencia normalmente está en el orden de los MHz, existen diferentes fuentes que generan esta señal de reloj, por ejemplo: una red RC, un cristal, un oscilador con el timer 555, un generador de señales, en fin, un factor elemental es la precisión de la señal, una base de tiempo o periodo del pulso con una precisión absoluta, de esto depende la velocidad de trabajo del PIC y los retardos programados

El cristal es la fuente más utilizada debido a su gran precisión, se conecta directamente a estos dos pines en paralelo, cualquier otro tipo de fuente sea una red RC, oscilador externo, entre otros, se utiliza las opciones de CLKIN y CLKOUT, que son: la entrada para una señal de reloj externa y la salida para una señal de reloj respectivamente, por el pin CLKOUT sale una señal de reloj cuya frecuencia es la cuarta parte de la frecuencia inicial, esta señal de salida es opcional y se puede aplicar a otros sistemas digitales, los distintos elementos de la familia 16F8X, dependiendo de la nomenclatura que utilizan tienen distintas características de funcionamiento, que se relacionan directamente con la frecuencia máxima de funcionamiento, el tipo de oscilador utilizado, entre otros

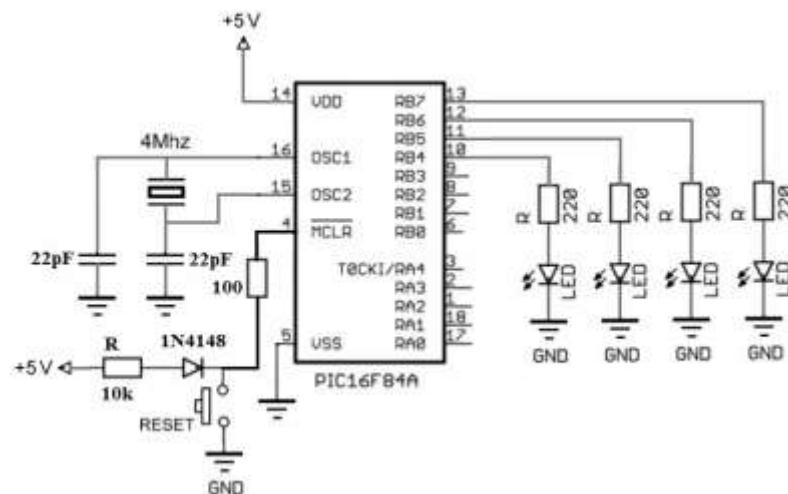
Los microcontroladores PIC, permiten cuatro tipos de osciladores externos para aplicarles la frecuencia de funcionamiento. Durante el proceso de grabación, antes de introducir el programa en memoria, debe indicarse el tipo de oscilador empleado en los bits FOSC1 y FOSC2 de la palabra de configuración. Los tipos de osciladores que puede utilizar el 16F84 son:

1. Oscilador de cristal o resonador de alta velocidad "HS" (High Speed Crystal/ Resonator): Es un oscilador de alta frecuencia, el rango de la frecuencia está comprendida entre 4MHz y 20MHz.
2. Oscilador o resonador cerámico "XT" (Crystal/Resonator): Se trata de un oscilador estándar que permite una frecuencia de reloj comprendida entre 100KHz y 4MHz, es un oscilador de frecuencia media.
3. Oscilador de cristal de cuarzo o resonador cerámico de baja potencia "LP" (Low Power Crystal): Se trata de un oscilador de bajo consumo, con un cristal o resonador diseñado para trabajar con frecuencias comprendidas entre 32KHz y 200KHz, es un oscilador de baja frecuencia.
4. Oscilador tipo "RC": Es un oscilador de bajo coste formado por una red RC. Se trata de un oscilador de baja precisión que depende de la estabilidad de la red RC, pero como contrapartida esta su bajo precio que lo hace interesante para muchas aplicaciones.

El circuito para cualquiera de las configuraciones anteriores se representa en la figura 1.10, el valor de los condensadores "22pF" depende de la frecuencia del cristal y lo indica el fabricante en su manual, no olvidar que la frecuencia máxima de trabajo del PIC es de 20MHz, considerar estos valores para el buen funcionamiento del mismo. Se recomienda ver la tabla de datos del fabricante

El pin 4, MCLR/Vpp: es la entrada de Reset, se activa con un nivel bajo, es decir con un 0 lógico, su función es reiniciar el programa, colocar a 0x00 el registro PC, también es utilizada para cambiar el modo del PIC a modo programación, aplicando una tensión de programación, que es de 12V

Los pines 17, 18, 1, 2 y 3, RA0-RA4/T0CKI: respectivamente, corresponden a cinco líneas bidireccionales de E/S del PORTA. Es capaz de entregar niveles TTL cuando la tensión de alimentación aplicada en Vdd es de 5V. El pin RA4, si se programa como salida es de colector abierto. Como entrada puede programarse en funcionamiento normal o como entrada del contador/temporizador TMRO, este pin está multiplexado y cumple doble función.



*Circuito electrónico básico para el PIC 16F84, se observa los componentes elementales como son: el cristal con sus respectivos condensadores, el botón de reset, Vdd y Vss.*

RB0-RB7: Pines 6, 7, 8, 9, 10, 11, 12 y 13 respectivamente, corresponden a ocho líneas bidireccionales de E/S del PORTB. Es capaz de entregar niveles TTL cuando la tensión de alimentación aplicada en Vdd es de 5V

El pin RB0 puede programarse además como entrada de interrupción externa INT. Los pines RB4 a RB7 pueden programarse para responder a interrupciones por cambio de estado. Los pines RB6 y RB7 corresponden con las líneas de entrada de reloj y entrada de datos respectivamente, cuando está en modo programación.

En resumen los datos técnicos del 16F84 son los siguientes:

- Memoria de programa: 1Kx14 de tecnología Flash
- Memoria de datos RAM: 68 bytes
- Memoria de datos EEPROM: 64 bytes
- Pila (stack): De 8 niveles
- Interrupciones: 4 tipos de interrupción, internas y externas
- Encapsulado: Plástico DIP de 18 patitas
- Frecuencia de trabajo: 20MHz máxima
- Temporizadores: Sólo uno, el TMR0. También tiene "Perro guardián WDT"
- Líneas de E/S Digitales: 13 (5 Puerto A y 8 Puerto B)
- Corriente máxima absorbida: 80mA Puerto A y 150mA Puerto B
- Corriente máxima suministrada: 50mA Puerto A y 100mA Puerto B
- Corriente máxima absorbida por línea: 25mA
- Corriente máxima suministrada por línea: 20mA
- Voltaje de alimentación (Vdd): De 2V a 6V D

## Set de instrucciones

El PIC16F84 es un microcontrolador de gama media y es de tipo RISC, esto quiere decir que tiene un juego de instrucciones reducido, en total 35 instrucciones o nemónicos que son la base de funcionamiento del

PIC. Al igual que los bits de los registros, sería complicado memorizar estas instrucciones, se recomienda utilizar este texto como guía, las instrucciones fundamentalmente se dividen en tres tipos. Esta división depende del tipo de datos con los que trabajan:

1. instrucciones orientadas a los registros o bytes.
2. Instrucciones orientadas a los bits.
- . Operaciones con literales y de control.

Para entender mejor cada instrucción a continuación se explicará el significado de algunos parámetros:

f : Registro al que afecta la instrucción

w : Acumulador (Working register)

b : Número de bit (hay instrucciones que afectan a un solo bit)

k : Constante (un número)

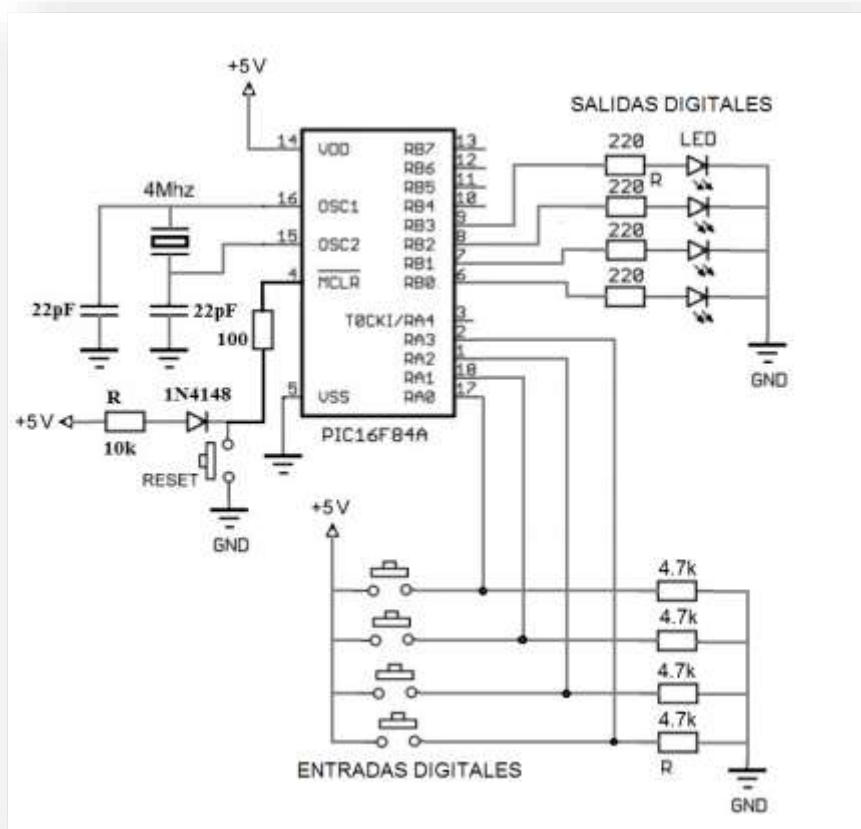
d : Selección de destino del resultado de la instrucción, puede ser "0" o "1", si es "0" el resultado se guarda en el acumulador (w) y si es "1" se guarda en el registro f al que afecta la instrucción

## Proyecto 1: Lectura de puertos

El programa consiste en ingresar un dato por el puerto A y sacarlo por el puerto B, el dato es un número en formato binario de 4 bits, en formato decimal los números van desde el 0 hasta el 15, el dato ingresado por el puerto A sale casi de inmediato por el puerto B, el tiempo que demora en salir es de 2 microsegundos, este tiempo depende de la frecuencia del cristal, se ha determinado que los datos ingresan por el puerto A, éste debe ser configurado como entrada, los pines a utilizar son: RA0, RA1, RA2 y RA3, y los datos salen por el puerto B, éste debe ser configurado como salida, los pines a utilizar son: RB0, RB1, RB2 y RB3, el esquema electrónico se muestra en la figura

Estimado lector, la importancia de este primer programa es ilustrar como se configuran los puertos a través del código assembler, es un programa básico de no más de 10 líneas, a medida que se avance en este texto los programas serán de mayor complejidad y de hasta 200 líneas, existen directivas, registros que aún no se han explicado, se explicarán durante el desarrollo de este proyecto, el código fue editado, ensamblado y compilado en el MPLAB, la simulación fue realizada en el PROTEUS, el código está comentado en las líneas más importantes, esto facilita la comprensión del mismo y permite realizar futuros cambios con éxito, se recomienda consultar con el datasheet del fabricante ante cualquier duda o consulta, cada aplicación será tratada con mucho detalle, pero no se puede abarcar todo sería algo ideal.

## Esquema Electrónico



### Instrucciones a utilizar

#### Btfsc f,b

#### bit test file skip clear

Si el bit número b del registro f está en 0, la instrucción que sigue a ésta se ignora y se trata como un NOP (skip). En este caso, y sólo en este caso, la instrucción precisa dos ciclos para ejecutarse

Ejemplo:

```

btfsc  ESTADO,3
goto   izquierda
goto   derecha
    
```

Si antes de ejecutar la instrucción el registro ESTADO vale:

ESTADO = 11111111 b

Al ejecutarse la instrucción el programa no salta y pasa a ejecutar goto izquierda, esto debido a que no se cumplió la condición de: salta si el bit 3 del registro ESTADO es 0, ese bit no es 0 es 1. Veamos otra situación.

Si antes de ejecutar la instrucción el registro ESTADO vale:

ESTADO = 00000000 b

Al ejecutarse la instrucción el programa si salta y pasa a ejecutar goto derecha, esto debido a que si se cumplió la condición de: salta si el bit 3 del registro ESTADO es 0 y efectivamente ese bit es 0

### Btfs f,b

#### Bit test file skip set

Si el bit número b del registro f está en 1, la instrucción que sigue a ésta se ignora y se trata como un NOP (skip). En este caso, y sólo en este caso, la instrucción precisa dos ciclos para ejecutarse.

Ejemplo:

```
btfs  ESTADO,3  
goto  izquierda  
goto  derecha
```

Si antes de ejecutar la instrucción el registro ESTADO vale:

ESTADO = 00000000 b

Al ejecutarse la instrucción el programa no salta y pasa a ejecutar goto izquierda, esto debido a que no se cumplió la condición de: salta si el bit 3 del registro ESTADO es 1, ese bit no es 1 es 0. Veamos otra situación

Antes de ejecutar la instrucción el registro ESTADO vale:

ESTADO = 11111111 b

Al ejecutarse la instrucción el programa si salta y pasa a ejecutar goto derecha, esto debido a que si se cumplió la condición de: salta si el bit 3 del registro ESTADO es 1 y efectivamente ese bit es 1.

### Código del Programa

*Nombre del programa: and.asm*

*Autor: Ing. Génesis Pavique*

Ensamblador: MPLAB V8.14

```

;-----
__config H'3FF1'           ;palabra de configuración
#include"p16F84.inc"       ;llamar a las librerias
ORG      0x00

    bsf      STATUS,5      ;ir al banco 1
    movlw   b'00000011'
    movwf   TRISA         ;RA0 y RA1 como entrada
    movlw   b'00000000'
    movwf   TRISB         ;RB0 como salida
    bcf      STATUS,5      ;ir al banco 0
    clrf    PORTB         ;limpiar el PORTB
inicio   btfss   PORTA,0   ;verifica RA0 es 0 o 1
         goto   lab_01
         btfss   PORTA,1   ;verifica RA1 es 0 o 1
         goto   lab_01
         bsf     PORTB,0   ;activa la salida
         goto   inicio
lab_01   bcf     PORTB,0   ;apaga la salida
         goto   inicio

END
;-----
    
```

El siguiente código de programa es para una compuerta Or:

Nombre del programa: or.asm

Autor: Ing. Genesis Pavique

Ensamblador: MPLAB V8.14

```

;
__config H'3FF1'                ;palabra de configuración
#include "p16F84.inc"           ;llamar a las librerías

ORG      0x00

      bcf      STATUS,5         ;ir al banco 1
      movlw   b'00000011'
      movwf   TRISA            ;RA0 y RA1 como entrada
      movlw   b'00000000'
      movwf   TRISE           ;RB0 como salida
      bcf      STATUS,5         ;ir al banco 0
      clrf    PORTB            ;limpiar el PORTB

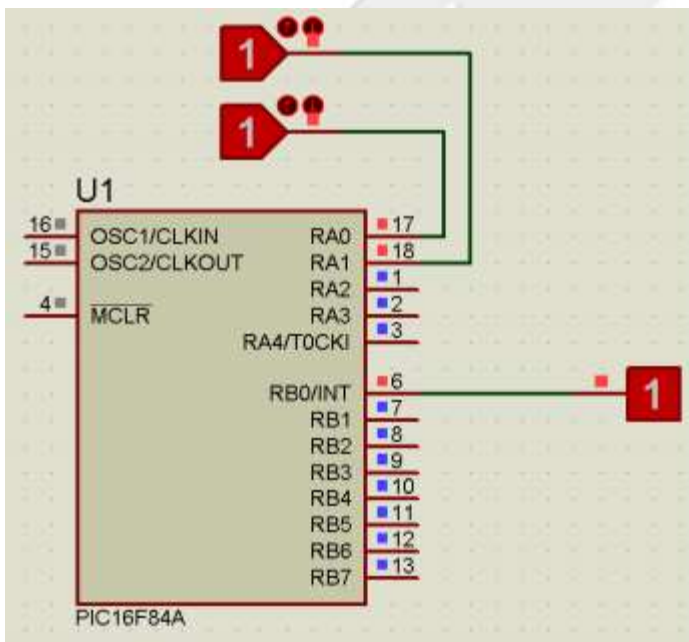
inicio  btfsc   PORTA,0         ;verifica RA0 es 0 o 1
        goto   lab_01
        btfsc   PORTA,1         ;verifica RA1 es 0 o 1
        goto   lab_01
        bcf     PORTB,0         ;apaga la salida
        goto   inicio

lab_01  bsf     PORTB,0         ;activa la salida
        goto   inicio

END
;

```

**Simulación del programa:**

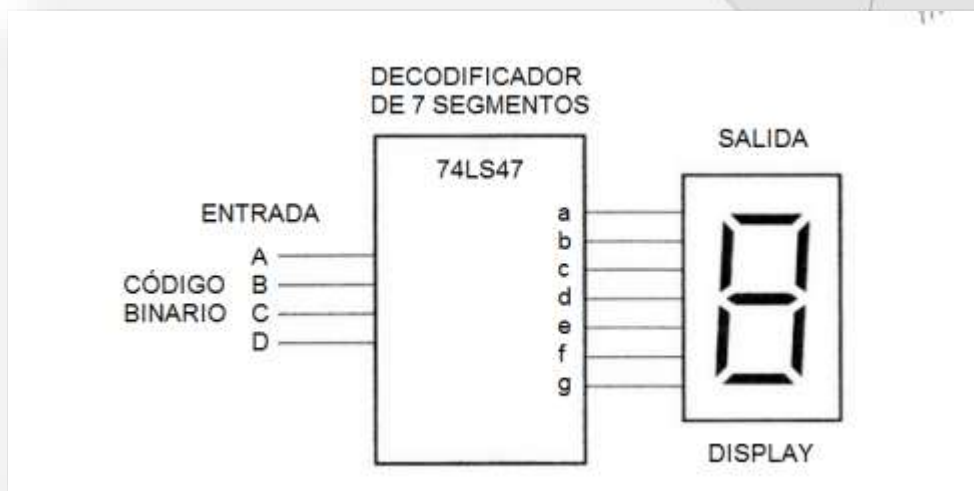


Circuito electrónico implementado en el PROTEUS, en este circuito se realizó la simulación del programa, el cual funcionó de manera normal

### Proyecto 3: Decodificador de tres bits

#### Planteamiento del programa:

El programa consiste en implementar un decodificador de tres bits, aplicando lógica programada, las entradas y salidas se pueden asignar a cualquier puerto y pin, vamos a utilizar el puerto A para las entradas y el puerto B para las salidas, el tiempo de respuesta del decodificador está en el orden de los microsegundos y depende de la frecuencia del cristal. El decodificador recibe un número en formato binario y da como respuesta el equivalente en formato decimal, normalmente se utiliza un display de 7 segmentos para visualizar la respuesta, éste puede ser de ánodo común o cátodo común, existen decodificadores en circuitos integrados como por ejemplo el 74LS47, tal como se muestra en la figura.



*Representación del decodificador de 7 segmentos en circuito integrado 74LS47*

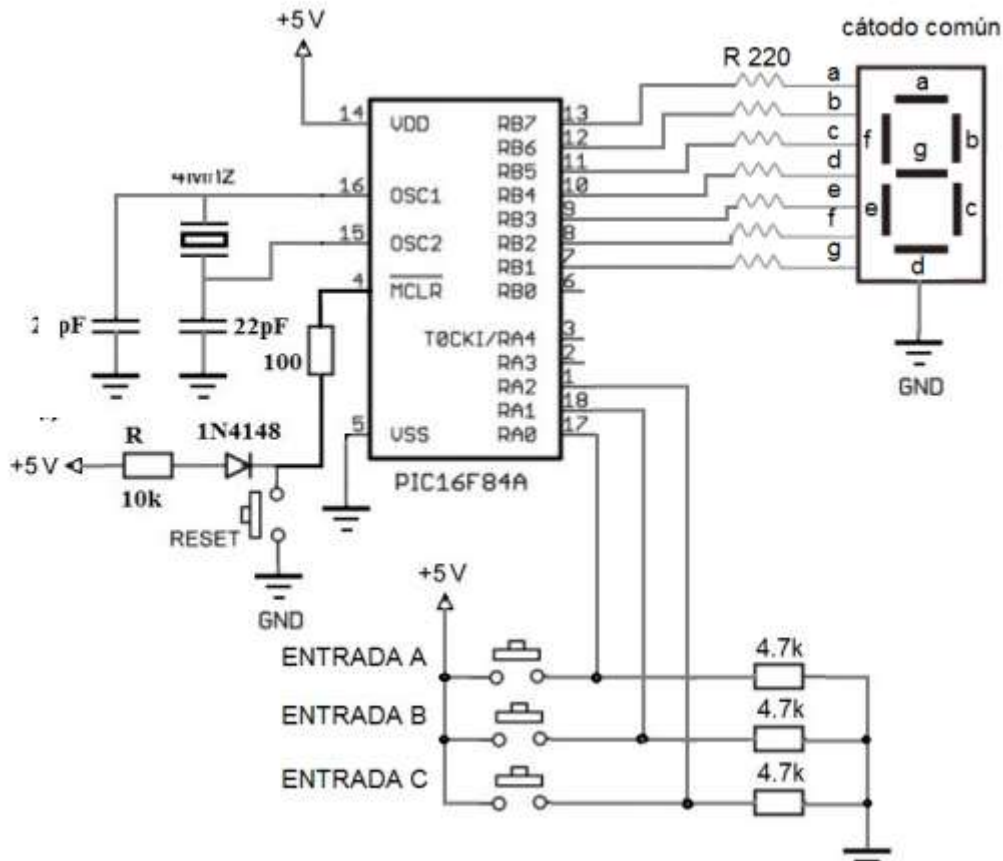
El circuito integrado 74LS47 recibe un código en formato binario y muestra en el display el número equivalente en formato decimal, por ejemplo: si la entrada es 000 en binario, el display muestra 0, si la entrada es 011 en binario, el display muestra 3, con 3 entradas el número máximo es 7, en la tabla 1.3 se muestra las equivalencias entre el formato binario y decimal.

Equivalencias entre el formato binario y decimal

ENTRADAS CÓDIGO BINARIO				SALIDAS DISPLAY							SALIDA EN DECIMAL
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

Se utilizará el puerto A para las entradas, éste debe ser configurado como entrada y los pines asignados son: RA0, RA1 y RA2, entrada 0, entrada 1 y entrada 2 respectivamente, se utilizará el puerto B para las salidas, éste debe ser configurado como salida y los pines asignados son: RB1, RB2, RB3, RB4, RB5, RB6 y RB7, el esquema electrónico se muestra en la figura

### Esquema Electrónico



### Instrucciones a utilizar:

La instrucción addwf:

Suma el valor de los registros w y f, por ejemplo: addwf ESTADO, 1

Si antes de ejecutar la instrucción  $w = 12$  y  $ESTADO = 24$ , al ejecutarse la instrucción se suma el valor de  $w$  y  $ESTADO$ , el resultado es 36, este resultado se guarda en  $ESTADO$ , debido a que el bit destino es 1, finalmente  $ESTADO = 36$ .

La instrucción call: guarda la dirección de retorno en la pila y después llama a la subrutina situada en la dirección cargada en el PC, por ejemplo:

```
    call    sumar
    goto   inicio
sumar  addwf  ESTADO,1
      return
```

La instrucción `call` llama a la subrutina `sumar`, salta a la etiqueta `sumar` y ejecuta todas las instrucciones de la subrutina hasta encontrar la instrucción `return`, que indica retorno de subrutina o fin de subrutina.

La instrucción `retlw`: cumple la misma función que `return`, con la diferencia que `retlw` retorna con un valor cargado en `w`, por ejemplo:

```
    call    sumar
    goto   inicio
sumar  addwf  ESTADO,1
      retlw d'10'
```

Carga 10 en decimal a `w` y luego retorna de la subrutina

### Código del programa:

*Nombre del programa: deco.asm*

*Autor: Génesis Pavique*

*Ensamblador: MPLAB V8.14*

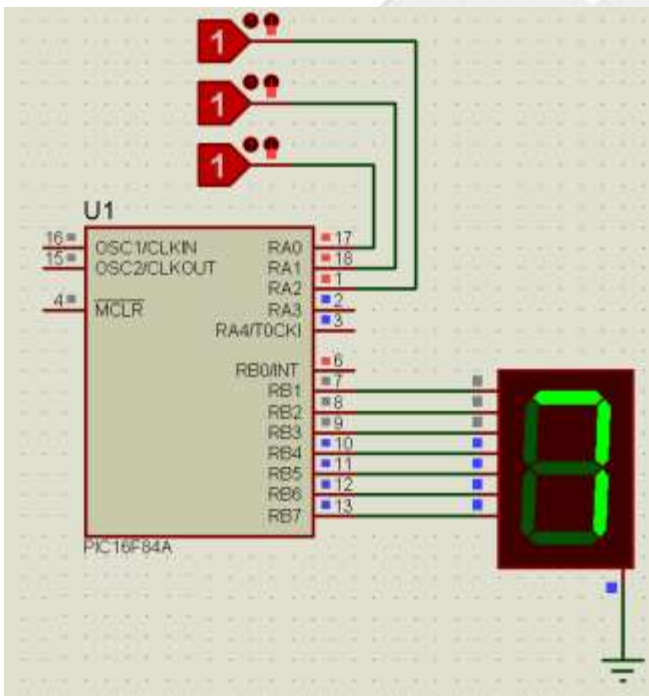
```

;-----
__config H'3FF1'
#include"p16F84.inc"
ORG      0x00
        bsf      STATUS,5
        movlw   0x07
        movwf   TRISA
        clrf    TRISB
        bcf     STATUS,5
inicio  movf    PORTA,0
        andlw   0x07
        call   tabla
        movwf   PORTE
        goto   inicio
;-----

tabla   addwf   PCL,1
        retlw  b'01111111'
        retlw  b'00001101'
        retlw  b'10110111'
        retlw  b'10011111'
        retlw  b'11001101'
        retlw  b'11011011'
        retlw  b'11111011'
        retlw  b'00001111'
;-----
END

```

### Simulación del programa

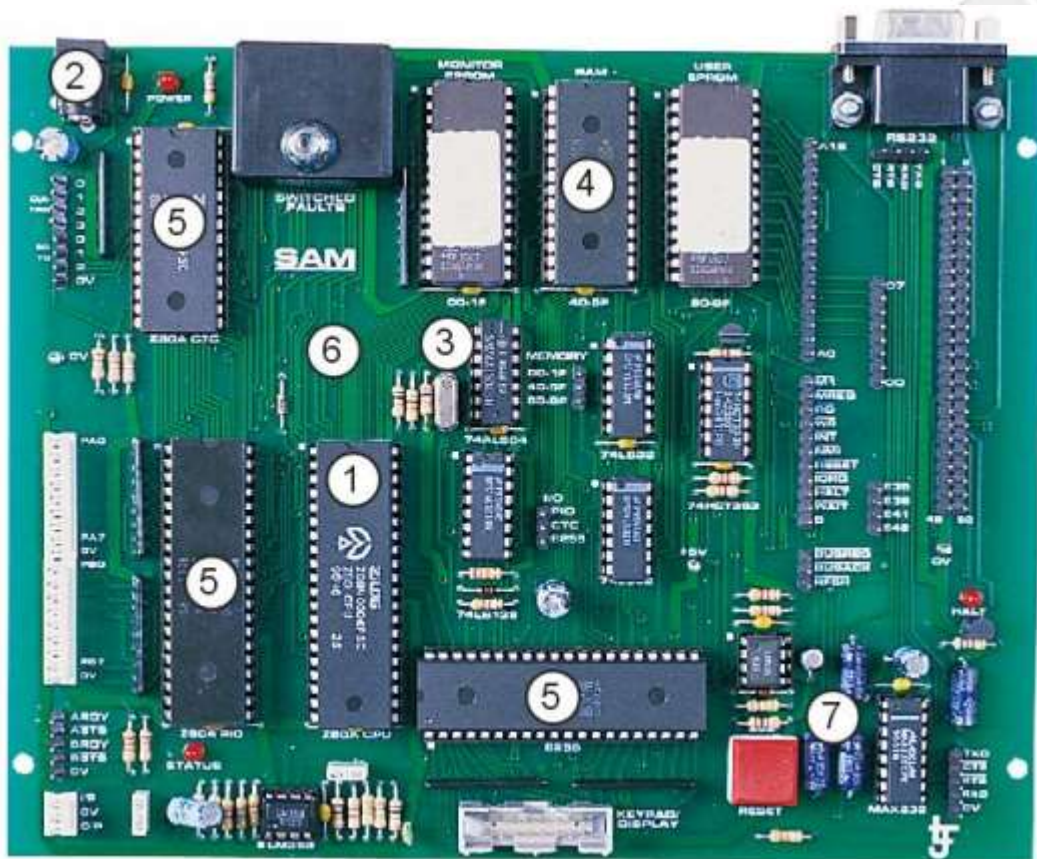


Circuito electrónico implementado en el PROTEUS, en este circuito se realizó la simulación del programa, el cual funcionó de manera normal

## Actividades de Aprendizaje

### Evaluación 1

Este es un circuito de control típico que utiliza un microprocesador. Identifica con el número correcto el componente señalado. Coloca el número que consideres adecuado entre los paréntesis.



- a) Fuente de alimentación ( )
- b) Memoria ( )
- c) Microprocesador ( )
- d) Circuitos auxiliares I/O ( )
- e) Reloj ( ) f) Reset ( )
- g) Buses de comunicación ( )

## Evaluación 2

- 1.- ¿Qué es lo que da flexibilidad a los circuitos digitales de control?
- 2.- Menciona la diferencia principal entre un microprocesador y un microcontrolador:
- 3.- ¿Qué significa ALU, y para qué sirve ese bloque?
- 4.- ¿Cuáles son las dos fuentes de alimentación más comunes para microprocesadores?
- 5.- ¿Para qué sirve la señal de reloj?
- 6.- ¿Para qué sirve la señal de reset?
- 7.- ¿Cuáles son los tres buses principales que se encuentran en cualquier microprocesador?
- 8.- ¿Qué significa “terminales 3-state” y por qué es importante en microprocesadores?
- 9.- ¿Por qué es importante el mapa de memoria?
- 10.- ¿Por qué son necesarias las interrupciones en un microprocesador?

## Evaluación 3

- 1.- ¿Cuál fue el primer microprocesador de 16 bits? ¿Y el de 32 bits? ¿Y el de 64 bits?
- 2.- ¿Cuál fue el primer microprocesador que rompió la barrera de 1MB de RAM?
- 3.- ¿Cuánta memoria puede direccionar teóricamente un microprocesador de 32 bits?
- 4.- ¿A qué se le denomina “memoria protegida” y por qué es importante?
- 5.- ¿Por qué conviene tener mucha memoria instalada en una computadora?
- 6.- ¿Qué tipo de microprocesador usan los Smartphone y las computadoras tipo tablilla?
- 7.- ¿Por qué no es conveniente tratar de programar un procesador de 16 bits o más en lenguaje ensamblador?
- 8.- Entonces, ¿cómo se deben programar?
- 9.- ¿Qué precauciones hay que tomar cuando un procesador se comunica con su memoria, si ésta se encuentra dividida en bloques?
- 10.- El aumento en el número de bits de un microprocesador, ¿afecta su arquitectura interna?.

## Conclusiones

1. El Microprocesador es un circuito electrónico integrado programable para realizar una aplicación embebida. Una solución a un problema de control monitoreo instrumentación que pueda ser implementada en un procesador. Clases de aplicaciones indebidas. Un teléfono celular. Un modem. Un GPS. El control de un horno de microondas. El control de una lavadora. El control de una nevera. Una contestadora automática. Una grabadora digital. Un reproductor mp3. Aplicaciones que requieran procesamiento de información y visualización programación memoria registro control monitoreo accionamiento son susceptibles a ser embebidas.

2. Se pueden programar directamente en el lenguaje ensamblador del micro controlador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C++; en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes interpretados como JAVA. Puesto que los sistemas embebidos se pueden fabricar por decenas de millares o por millones de unidades, una de las principales preocupaciones es reducir los costes. Los sistemas embebidos suelen usar un procesador relativamente pequeño y una memoria pequeña para ello. Los primeros equipos embebidos que se desarrollaron fueron elaborados por IBM en los años 1980. Los programas de sistemas embebidos se enfrentan normalmente a tareas de procesamiento en tiempo real.

3. Porque estudiar procesadores en ingeniería de control. Tanto en ingeniería eléctrica como en ingeniería de control desarrollamos proyectos apropiando tecnología. La apropiación exige coordinar varias tecnologías o sistemas. Esa coordinación precisa de sistemas intermedios llamados circuitos de interfaz. Los procesadores permiten crear de forma rápida y económica eficiente circuitos de interfaz. Hay otros campos de desempeño como la instrumentación y la medición donde se requieren el desarrollo de nuevos aparatos de medición sobre todo cuando es muy complejo medir la variable. Ejemplo: Variables altamente no lineales. espectrometría. En el campo del monitoreo ha evolucionado actualmente a la telemetría y la telemetría simplemente es como su nombre nos da la idea medir una variable de forma remota. adicionalmente podemos requerir a ser un accionamiento remoto. Y en el campo del control uno de los grandes retos del ingeniero de control en su labor de producir tecnología consiste en diseñar nuevos equipos de control para insertar en módulos y gabinetes a nivel de planta que puedan ejecutar algoritmos modernos como controladores predictivos, controladores no lineales y controladores inteligentes.

## Recomendaciones

El presente documento debe ser utilizado exitosamente por los estudiantes de las carreras de Computación, Informática y electrónica del Instituto universitario de tecnología de la información, Iutepi. El antecedente para poder utilizar estas notas es un curso básico de diseño digital en donde se den a conocer las bases de la lógica booleana, la lógica secuencial y del diseño de dispositivos digitales básicos.

Los estudiantes primero deben internalizar Su presencia ha comenzado a cambiar la forma en que percibimos el mundo e incluso a nosotros mismos. Cada vez se hace más difícil pasar por alto el microprocesador como otro simple producto en una larga línea de innovaciones tecnológicas.

Ninguna otra invención en la historia se ha diseminado tan aprisa por todo el mundo o ha tocado tan profundamente tantos aspectos de la existencia humana. Hoy existen casi 15,000 millones de microchips de alguna clase en uso (el equivalente de dos computadoras poderosas para cada hombre, mujer y niño del planeta). De cara a esa realidad, ¿quién puede dudar que el microprocesador no sólo esté transformando los productos que usamos, sino también nuestra forma de vivir y, por último, la forma en que percibimos la realidad?

No obstante que reconocemos la penetración del microprocesador en nuestras vidas, ya estamos creciendo indiferentes a la presencia de esos miles de máquinas diminutas que nos encontramos sin saberlo todos los días. Así que, antes de que se integre de manera demasiado imperceptible en nuestra diaria existencia, es el momento de celebrar al microprocesador y la revolución que ha originado, para apreciar el milagro que es en realidad cada uno de esos chips de silicio diminutos y meditar acerca de su significado para nuestras vidas y las de nuestros descendientes.

## Bibliografía

1. Digital Design and Computer Architecture. D.M. Harris y S.L. Harris. Morgan Kaufman Pub. 2007. ISBN: 0123704979. Ref\_UAM: INF/621.3/HAR.
2. Estructura y diseño de computadores: La interfaz software/hardware. D.A. Patterson y J.L. Hennessy. Ed. Reverte 2011. ISBN: 9788429126204. Ref\_UAM: INF/681.32.3/PAT.
3. Computer Organization And Design: The Hardware/Software Interface. D.A. Patterson y J.L. Hennessy. Morgan Kaufmann. 4ª Ed. 2009. ISBN: 9780123744937. Ref\_UAM: INF/681.3.06/PAT.
4. Fundamentos de diseño lógico y de computadores. M.M.Mano y C.R.Kime. Prentice Hall. 2005. ISBN: 8420543993. Ref\_UAM: INF/621.3/MAN.
5. Organización de computadores. C.Hamacher, Z.Vranesic y S.Zaky. McGraw-Hill. 5ª Ed. 2003. ISBN: 8448139518. Ref\_UAM: INF/681.32.3/HAM.

6. The Student's Guide to VHDL. P. Ashenden. Morgan Kaufman Pub. 1998. ISBN: 1558605207. Ref\_UAM: INF/681.3.062/ASH.
7. Diseño de Sistemas Digitales con VHDL. S.A. Pérez, E. Soto y S. Fernández. Thomson. 2002. ISBN: 8497320816. Ref\_UAM: INF/681.3.062-V/PER.
8. Diseño digital avanzado con VHDL. F. Machado, S. Borromeo y N. Malpica. Serv. Publicaciones URJC. 2009. ISBN: 9788498494198. Ref\_UAM: INF/681.3.062- V/MAC.

