**Análisis de Sistemas Informáticos**

**Introducción a la organización**

**Racionalización administrativos**

A:

De: Eduardo Espinoza,

Tema: Microcontroladores,

Fecha: 14/11/16,

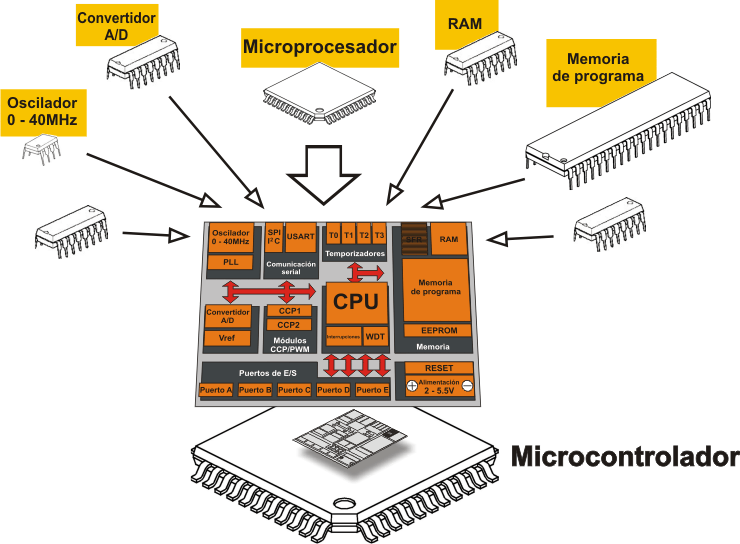
**ELECTRONICA DIGITAL**

* 1. **INTRODUCCIÓN**

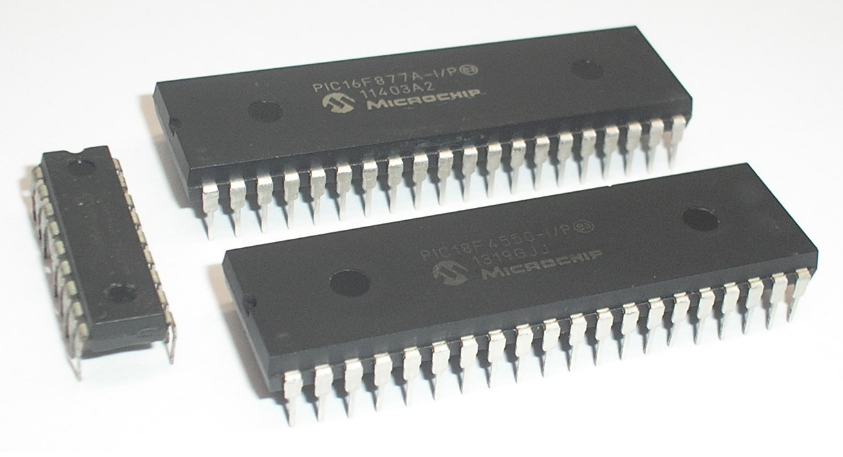
## MICROCONTROLADORES

## MIKROELEKTRONIKA

Los principiantes en electrónica creen que un microcontrolador es igual a un microprocesador. Esto no es cierto. Difieren uno del otro en muchos sentidos. La primera y la más importante diferencia es su funcionalidad. Para utilizar al microprocesador en una aplicación real, se debe de conectar con componentes tales como memoria o componentes buses de transmisión de datos. Aunque el microprocesador se considera una máquina de computación poderosa, no está preparado para la comunicación con los dispositivos periféricos que se le conectan. Para que el microprocesador se comunique con algún periférico, se deben utilizar los circuitos especiales. Así era en el principio y esta práctica sigue vigente en la actualidad.



# **¿Qué es un microcontrolador?**

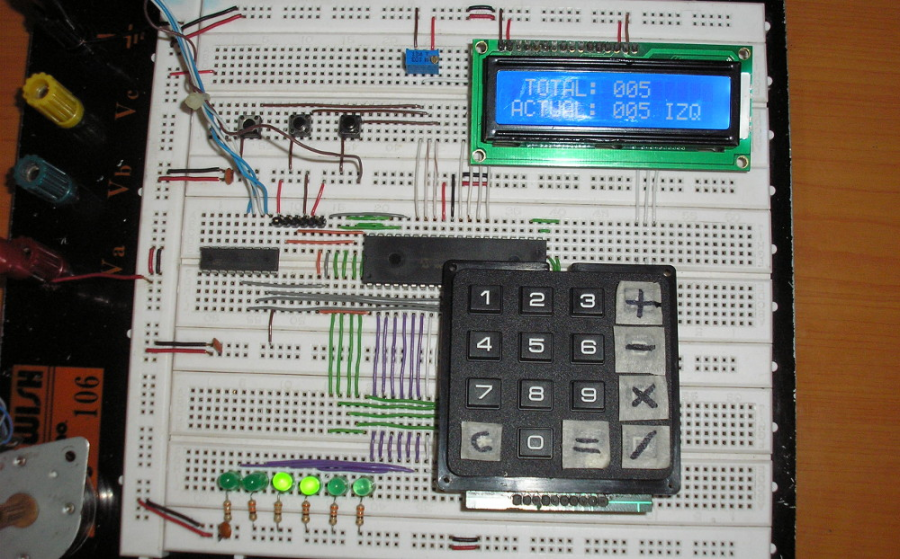


Un microcontrolador es un circuito integrado digital que puede ser usado para muy diversos propósitos debido a que es programable. Está compuesto por una unidad central de proceso (CPU), memorias (ROM y RAM) y líneas de entrada y salida (periféricos).

Como podrás darte cuenta, un microcontrolador tiene los mismos bloques de funcionamiento básicos de una computadora lo que nos permite tratarlo como un pequeño dispositivo de cómputo.

1. **¿QUE PUEDEN HACER LOS MICROCONTROLADORES?**

El dispositivo electrónico capaz de controlar un pequeño submarino, una grúa o un ascensor como el anteriormente mencionado, ahora está incorporado en un sólo chip. Los microcontroladores ofrecen una amplia gama de aplicaciones y sólo algunas se exploran normalmente. Le toca a usted decidir qué quiere que haga el microcontrolador y cargar un programa en él con las instrucciones apropiadas. Antes de encender el dispositivo es recomendable verificar su funcionamiento con ayuda de un simulador. Si todo funciona como es debido, incorpore el microcontrolador en el sistema. Si alguna vez necesita cambiar, mejorar o actualizar el programa, hágalo. ¿Hasta cuándo? Hasta quedar satisfecho. Eso puede realizarse sin ningún problema.



Un microcontrolador puede usarse para muchas aplicaciones algunas de ellas son: manejo de sensores, controladores, juegos, calculadoras, agendas, avisos lumínicos, secuenciador de luces, cerrojos electrónicos, control de motores, relojes, alarmas, robots, entre otros. El límite es la imaginación.

1. **¿Cómo funciona un microcontrolador?**

Como el hardware ya viene integrado en un solo chip, para usar un microcontrolador se debe especificar su funcionamiento por software a través de programas que indiquen las instrucciones que el microcontrolador debe realizar. En una memoria se guardan los programas y un elemento llamado CPU se encarga de procesar paso por paso las instrucciones del programa. Los lenguajes de programación típicos que se usan para este fin son ensamblador y C, pero antes de grabar un programa al microcontrolador hay que compilarlo a hexadecimal que es el formato con el que funciona el microcontrolador.

Para diseñar programas es necesario conocer los bloques funcionales básicos del microcontrolador, estos bloques son:

* **CPU** (Unidad central de proceso)
* **Memoria ROM** (Memoria de solo lectura)
* **Memoria RAM** (Memoria de acceso aleatorio)
* **Líneas de entrada y salida** (Periféricos)

 La CPU posee, de manera independiente, una memoria de acceso rápido para almacenar datos denominada registros, si estos registros son de 8 bits se dice que el microcontrolador es de 8 bits.

1. **¿Dónde consigo lo necesario?**



1. **Aplicación.** Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación

* Procesamiento de datos: puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión. –
  + Entrada Salida: para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un [diagrama](http://www.monografias.com/trabajos14/flujograma/flujograma.shtml) de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este [análisis](http://www.monografias.com/trabajos11/metods/metods.shtml#ANALIT) puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.
  + Consumo: algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en [estado](http://www.monografias.com/trabajos12/elorigest/elorigest.shtml) de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
* • Memoria: para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.
* El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-[código](http://www.monografias.com/trabajos12/eticaplic/eticaplic.shtml), de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.
* • Ancho de palabra: el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).
* • Diseño de la placa: la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.
* Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:
* 8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.
* 8051 (Intel y otros). Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.
* 80186, 80188 y 80386 EX (Intel). Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.
* 68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.
* 683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.
* PIC (MicroChip). Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

**Conclusión**

Por ahora solo nos introducimos en el mundo de los microcontroladores pero aún queda por saber lo más importante: conocer la forma en la que funcionan y elaborar los programas que controlen ese funcionamiento, de eso tratará este tutorial.