

Microcontrolador – qué es y para que sirve

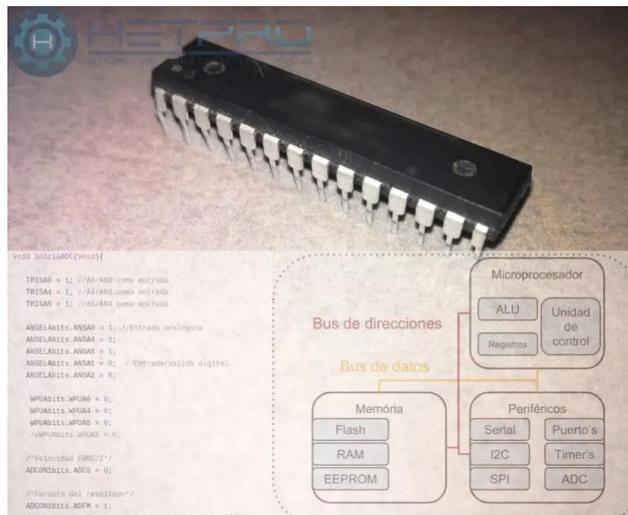
El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información.

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas.

Elementos de un Microcontrolador

Un microcontrolador al menos tendrá:

- Microprocesador.
- Periféricos (unidades de entrada/salida).
- Memoria.



Elementos internos en un microcontrolador

Procesador o Microprocesador.

Un procesador incluye al menos tres elementos, ALU, unidad de control y registros.

ALU. También conocida como Unidad Aritmética y Lógica. Esta unidad está compuesta por los circuitos electrónicos digitales del tipo combinatorios (compuertas, sumadores, multiplicadores), cuya principal función es el realizar operaciones.

Estas operaciones están divididas en tres tipos:

Lógicas. Como las operaciones básicas de las compuertas lógicas, como la suma lógica (OR), multiplicación lógica (AND), diferencia lógica (XOR) y negación (NOT). Una operación lógica sólo puede tener como entradas y como salidas una respuesta lógica (0 o 1). Esto dependiendo de los niveles de voltajes de una señal digital.

Aritméticas. Las operaciones aritméticas son la suma, resta, multiplicación y división. Dependiendo del procesador (8, 16, 32 o 64 bits) será la rapidez con la que se pueden hacer dichas operaciones.

Miscelaneas. En estas operaciones caen todas las demás operaciones como la transferencia de bits (<< >>).

Unidad de control. La unidad de control es el conjunto de sistemas digitales secuenciales (aquellos que tienen memoria) que permiten distribuir la lógica de las señales.

Registros. Los registros son las memorias principales de los procesadores, ya que funcionan a la misma velocidad que el procesador a diferencia de otras memorias un tanto más lentas (como la RAM, FLASH o la CACHE). Los registros están contruidos por Flip-Flops. Los Flip-Flops son circuitos digitales secuenciales.

Periféricos.

Los periféricos son los circuitos digitales que nos permiten una interacción con el mundo «exterior» al microcontrolador. Su función es la de poder habilitar o deshabilitar las salidas digitales, leer sensores analógicos, comunicación con terminales digitales o sacar señales analógicas de una conversión digital.

Puertos de entrada/salida pararelos. Los puertos están relacionados al tamaño del procesador, es decir que un puerto de 8 bits es porque el procesador es de 8 bits. Un procesador de 64 bits, tiene la capacidad de tener un puerto de 64 bits.

Puertos seriales. Nos permiten transformar la información digital pararela (bytes de información) en tramas que se pueden transferir por una o varias líneas de comunicación. Existen por ejemplo: puerto serial, i2c, SPI, USB, CAN, etc.

Periféricos analógicos. Como los que convierten señales analógicas a digitales (ADC) o señales digitales a analógicas (DAC) o comparadores analógicos.

Tipos de Memoria.

La memoria esta dividida en tres:

La memoria para el programa (FLASH), la memoria para los datos o variables del programa (RAM) y la memoria para configuraciones o no volátil (EEPROM).

Diferencia entre un microcontrolador y un microprocesador

La diferencia entre un microcontrolador y un microprocesador son los elementos que lo componen. Por ejemplo, un microcontrolador contiene un microprocesador y un microprocesador al menos incluye: una alu, memoria y una unidad de control. Lo podemos ver de la siguiente forma, un microcontrolador tiene un microprocesador pero un microprocesador no puede contener un microcontrolador (a menos que se programe en una fpga o en silicio).

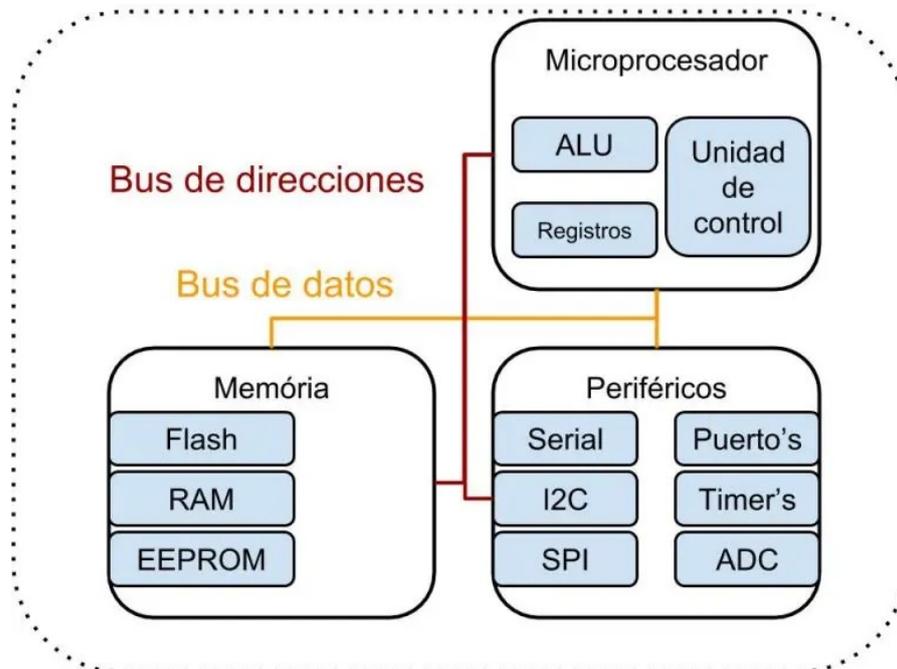


Figura-2. Esquema general de un Microcontrolador. Representa los elementos más representativos de los elementos internos de un Microcontrolador.

¿Qué es la arquitectura de un microcontrolador?

La arquitectura de un microcontrolador se puede referir a dos cosas. Se puede clasificar la arquitectura de un microcontrolador de acuerdo a la organización de sus memorias o al ancho de su bus de memoria y/o datos.

Arquitectura Harvard vs Von Neumann

La primera puede ser la forma en la que esta organizada la memoria. Entonces, de acuerdo a como esta organizada la memoria, podemos hablar de dos tipos de arquitecturas. La Arquitectura Harvard y la arquitectura Von Neumann. Cronológicamente hablando, la primera arquitectura fue la Von Neumann, en esta organización el bus de datos y el bus de memorias son el mismo. Por lo tanto el proceso para ejecutar una instrucción es más lento.

Arquitectura por el tamaño de bits del procesador

La segunda clasificación de la arquitectura de un microcontrolador es por el ancho o tamaño en bits de sus buses de datos y/o memoria. Por ejemplo, una arquitectura de 64bits puede significar que el procesador puede hacer operaciones con operadores o variables de un tamaño de 64 bits e incluso que puede direccionar o guardar hasta 2^{64} localidades de memoria tanto de programa como de

datos. Entonces por lo general, un procesador de más bits puede hacer operaciones matemáticas complejas en un menor tiempo.

Microcontrolador incluye un Microprocesador

Para el Microcontrolador, un Microprocesador es el conjunto de tres sistemas digitales: ALU, unidad de control y registros. Los microprocesadores fueron los padres de los microcontroladores. Uno de los primeros procesadores comerciales que se enseñaban en las universidades era el Z80.

El procesador Z80 requiere de componentes adicional para poder funcionar en alguna aplicación. Por ejemplo, no tiene RAM o FLASH. Estas dos memorias sirven para guardar el contenido de los datos del programa y al programa en si mismo respectivamente.

Para ensamblar un circuito mínimo, se tenia que conectar al Z80: memoria externa y controladores para poder tener algún puerto paralelo. No se diga un puerto serial, se tenia que diseñar mediante electrónica reconfigurable (GALs, Compuertas logicas, Flip-Flops, Multiplexores, etc).

Microcontrolador y sus Periféricos

En el Microcontrolador los Periféricos son módulos electrónicos digitales que se encuentran embebidos o empotrados en el microcontrolador. Permiten controlar distintas funciones de los puertos. Por ejemplo, conexión de puertos de entrada/salida en paralelo. Adicionalmente, los periféricos pueden cumplir con funciones adicionales o secundarias. Dichas funciones se pueden configurar mediante registros de propósito específico.

Periféricos de comunicación

- UART. Unidad de Recepción y Transmisión Serial Asíncrona. Consiste de dos cables llamados RX y Tx que sirven para enviar mensajes binarios con otros sistemas digitales.
- I2C. Puerto de comunicación serial síncrono. Permite enviar y recibir datos mensajes con hasta 127 dispositivos conectados al mismo de bus que consiste en dos cables: SDA y SCL. Serial Data y Serial CLock respectivamente.
- SPI. Serial Peripheral Interface. Interfaz Serial Periferica. Es un puerto de comunicación serial síncrona. Su función es la de enviar información binaria mediante 4 cables de conexión.
- MISO – Master Input Slave Output . Entrada del Maestro y Salida del Esclavo.
- MOSI – Master Output Slave INput. Salida del Maestro Entrada del Esclavo.
- SCK. Serial Clock. Reloj Serial.
- /CS ó /SS. Chip Select ó Slave Select. Selector del Chip o Selector del Esclavo. Permite activar un esclavo.
- Puerto Paralelo. Se controlan por medio de registros. Generalmente se pueden controlar mediante tres registros.
- Registro de dirección para configurar entrada ó salida.
- Registro de estado para la entrada. Nos indica el nivel lógico a la entrada del pin de cada puerto, cuando esté esta configurado como entrada.
- Registro de configuración del estado lógico de salida. Permite escribir 1's y 0's a los pines del puerto, cuando esté esta configurado como salida.

Periféricos de adecuación de señales

- ADC. Convertidor de Señales Analógicas a Digitales. Permiten generar una representación binaria para la magnitud de una señal analógica.
- DAC. Digital to Analog Converter. Lo opuesto a un DAC. También es ideal cuando se quiere generar audio. Una tarjeta de audio no es más que un DAC de muy buena calidad y muy rápido.

Periféricos generales

- Timer/Contador. Son circuitos electrónicos digitales y secuenciales. Es decir que funcionan en flancos de una señal de reloj que oscila a una frecuencia determinada. Su función principal es la de contar eventos, ya sea internos (timer) o externos (contador), a una frecuencia que nos permite por ejemplo, contar el tiempo transcurrido de un proceso.
- Controlador de Interrupciones. Son dispositivos que permiten alterar la secuencia de ejecución del procesador. Esto logra que se pueda interrumpir al procesador para atender el llamado de dispositivos electrónicos de baja velocidad. Por ejemplo, teclados o sensores.
- Controlador de Oscilador. Permite dividir y/o multiplicar a osciladores internos o externos para poder hacer funcionar a la parte secuencial de la electrónica del microcontrolador.

Microcontrolador y sus memorias

La memoria de los microcontroladores puede dividirse en cuatro tipos distintos: para el programa, generalmente es una memoria interna del tipo FLASH. También para las variables de los programas se llama RAM. Los registros de propósito general que utiliza el procesador para guardar los resultados de las operaciones así como los datos que se traen y llevan a la RAM y finalmente la memoria externa que se utiliza para guardar mediciones y/o datos de calibración.

Memoria de programa – FLASH

La memoria de programa es la que se utiliza cuando estamos programando a nuestra aplicación. Cada instrucción del lenguaje ensamblador o de lenguaje C o de un lenguaje de alto nivel, es convertida a instrucciones máquina que requieren de un tamaño particular de bits para ser guardadas. Estos bits o instrucciones se guardan en la FLASH.

Si tenemos poca FLASH, entonces sólo podremos hacer programas pequeños.

Memoria de datos – RAM

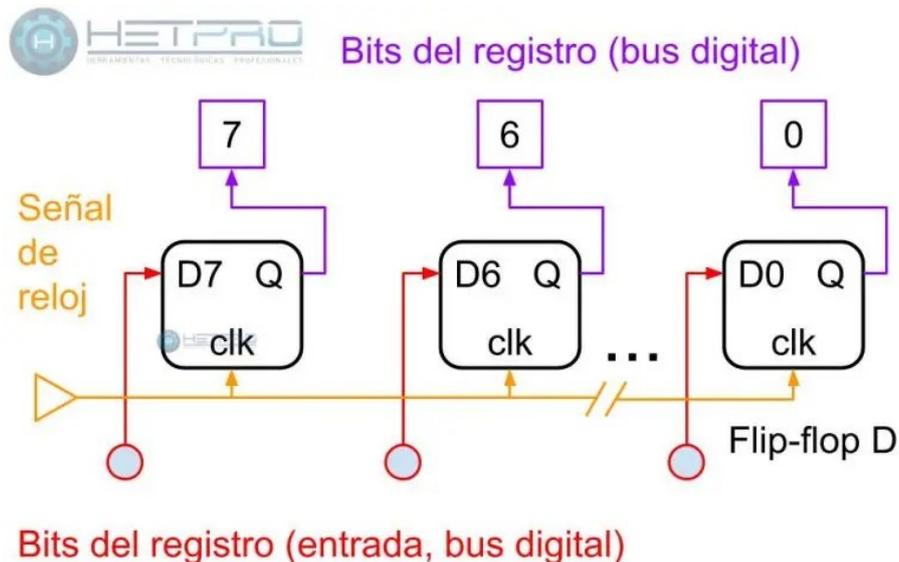
La memoria de datos se utiliza cada vez que agregamos una variable a nuestro programa. Como la variable seguramente se utilizara como entrada para un proceso o calculo, está requiere de estar disponible a una velocidad relativamente rápida. Este tipo de información se guarda en la RAM – Random Access Memory ó Memoria de Acceso Aleatorio por sus siglas en ingles.

Si tenemos poca RAM, nuestra aplicación no podrá tener muchas variables.

Microcontrolador y sus Registros

Los registros son las memorias digitales más rápidas. Se construyen con Flip-Flops y generalmente funcionan a una velocidad cercana a la del procesador. En algunos procesadores, también incluyen un tipo de memoria llamada CACHE. Esta no puede guardar operaciones y sólo es un puente entre

el procesador y la memoria principal. Por ejemplo, se puede ver la estructura del registro 74LS377 el cual puede funcionar como un registro (memoria).



El registro con FF-D de un microcontrolador

Memoria externa

La memoria externa se suele utilizar cuando queremos guardar parte de la información que estamos recolectando y/o procesando. Se suelen usar también para procesos de calibración. Existen dos distintos tipos de memorias no volátiles que pueden ser usadas externamente para los microcontroladores.

- EEPROM – Memoria Electricamente Borrable de Solo Lectura. Ojo, también se pueden programar y escribir.
- FLASH – Como la que tienen internamente los microcontroladores. Pero estas suelen funcionar a una velocidad mucho más lenta a través de protocolos seriales como I2C, Serial o SPI.

Microcontrolador – su historia

Anterior a los microcontroladores, los sistemas reconfigurables se diseñaban para funcionar como autómatas. Los autómatas también eran llamados maquinas de estado.

Para diseñar una maquina de estados, primero se crea una tabla de condiciones lógicas. Esta tabla indica todo el conjunto de las combinaciones lógicas posibles de las entradas y estados actuales de las salidas. El diseño de un automáta involucra a estas dos condiciones.

Cuando se tenían todas las combinaciones posibles, se elegían los estados futuros y se diseñaba el sistema digital. Entonces el proceso es muy lento e involucra una gran cantidad de circuitos digitales secuenciales (flip-flops).

Mientras que las maquinas de estado se volvían más complejas, se comenzaron a crear módulos independientes que funcionaban de forma general para multiples propósitos. Por ejemplo, una ALU – Unidad Aritmética Lógica, es el elemento central en el diseño de un procesador. Sin embargo, este circuito al ser puramente combinacional (sus salidas solo dependen de sus entradas), no es muy útil para ser automatizado. Para esto, se requiere de circuitos secuenciales. Esto dio entrada al diseño de los primeros procesadores.

Como se programa a un microcontrolador

Un microcontrolador se programa en lenguaje ensamblador. Entonces cada microcontrolador, tiene su propio conjunto de instrucciones. Las instrucciones se dividen de acuerdo de las operaciones que realizan en:

Aritméticas.

Lógicas.

Transferencia de bits.

Miscelaneas.

Microcontrolador ejemplo de programa

```
void iniciaADC(void){

    TRISA0 = 1; //A0/AN0 como entrada
    TRISA4 = 1; //A4/AN3 como entrada
    TRISA5 = 1; //A5/AN4 como entrada

    ANSELAbits.ANSA0 = 1; //Entrada analogica
    ANSELAbits.ANSA4 = 1;
    ANSELAbits.ANSA5 = 1;
    ANSELAbits.ANSA1 = 0; //Entrada/salida digital
    ANSELAbits.ANSA2 = 0;

    WPUAbits.WPUA0 = 0;
    WPUAbits.WPUA4 = 0;
    WPUAbits.WPUA5 = 0;
    //WPUAbits.WPUA5 = 0;

    /*Velocidad FOSC/2*/
    ADCON1bits.ADCS = 0;

    /*Formato del resultado*/
    ADCON1bits.ADFM = 1;
    /*Habilitar voltaje de referencia fijo en el ADCON1*/
```

Ejemplo de un programa para un microcontrolador PIC, muestra la configuración de los registros para configurar el ADC.

Por lo general la curva de aprendizaje del lenguaje ensamblador es muy grande. Para el ciclo de desarrollo de las aplicaciones para sistemas embebidos, esto es mucho tiempo. Por lo tanto se utilizan lenguajes de un nivel superior como basic o C/C++. Además estos lenguajes se traducen al lenguaje ensamblador mediante compiladores. Por ejemplo, el compilador C18 permite traducir el código C al lenguaje maquina de los microcontroladores PIC18XXX.

Para poder programar a un microcontrolador se requieren tres cosas:

- Compilador. Traduce lenguaje como el C y C++ al lenguaje ensamblador.
- IDE. Integrated Development Environment. Entorno de Desarrollo Integrado. Es donde mediante el compilador, se crea y compilan los programas.
- Programador. Cada fabricante y arquitectura tiene su propio programador. Existen programadores llamados Universales que pueden programar distintas arquitecturas.
- Programador de un Microcontrolador, llamado USBasp para arquitectura AVR

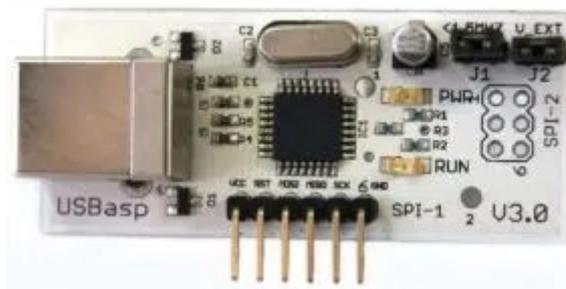


Imagen de un programador USBasp para microcontroladores AVR.

Aplicaciones de los microcontroladores

Los microcontroladores tienen un muchas aplicaciones en los sistemas digitales. Por ejemplo para el diseño de controladores de temperatura automáticos, maquinas dispensadoras, dispositivos biomédicos. En la industria del entretenimiento como juguetes. Incluso en aplicaciones aeroespaciales, sistemas de medición, sistemas de instrumentación. También por ejemplo para el desarrollo y automatización de experimentos científicos. Incluso para automatizar procesos, maquinas tragamonedas, hornos de microondas, lavadoras, controladores de audio, controladores de video. Finalmente como lectores de huellas digitales, cámaras, celulares y así sucesivamente podemos platicar un sin fin de aplicaciones en donde los microcontroladores son útiles. Por ejemplo, Arduino es una de las aplicaciones más usadas de un microcontrolador. Por está razón, el aprender a usar un microcontrolador, es una materia básica en las carreras relacionadas con la electrónica, la robótica, biomedicina, mecatronica y ciencias computacionales. A continuación te presentamos algunos proyectos y tutoriales con microcontrolador:

[Servidor WEB para mostrar temperatura del LM35.c](#)

[Servidor WEB para el control de LEDS.](#)

[Comunicación android con un microcontrolador por bluetooth.](#)

[Lectura de un sensor de temperatura y humedad DHT11 con Arduino.](#)

[Lectura de un sensor de flujo con Arduino.](#)

[Uso de una pantalla OLED con Arduino.](#)