

¿Qué es un Microcontrolador?

- ❖ Un microcontrolador es como una pequeña computadora en un solo chip. Tiene todos los componentes esenciales de una computadora, pero en una escala mucho más reducida.
- ❖ **Componentes Principales:**
- ❖ **Unidad Central de Procesamiento (CPU):** Es el cerebro del microcontrolador. Ejecuta las instrucciones del programa, realiza cálculos y toma decisiones.
- ❖ **Memoria:**
 - **Memoria de Programa (Flash):** Almacena el programa que el microcontrolador debe ejecutar. Este programa es generalmente escrito en un lenguaje de alto nivel (como C) y luego compilado en lenguaje ensamblador o código máquina.
 - **Memoria de Datos (RAM):** Almacena temporalmente datos y variables que el programa utiliza durante su ejecución.
 - **Memoria de Solo Lectura (ROM):** En algunos microcontroladores, se utiliza para almacenar datos de configuración o pequeñas porciones de código que no necesitan ser modificadas.

Periféricos: Son los componentes que permiten al microcontrolador interactuar con el mundo exterior. Incluyen:

- **Puertos de Entrada/Salida (I/O):** Permiten al microcontrolador leer señales de entrada (como botones, sensores) y enviar señales de salida (para controlar LEDs, motores, etc.).
- **Temporizadores/Contadores:** Se utilizan para medir el tiempo, generar pulsos, y realizar otras tareas relacionadas con el tiempo.
- **Convertidores Analógico-Digital (ADC):** Permiten al microcontrolador leer señales analógicas (como voltajes variables) y convertirlas en valores digitales que la CPU pueda procesar.
- **Comunicación Serial:** Interfaces como UART, SPI e I2C permiten al microcontrolador comunicarse con otros dispositivos (sensores, pantallas, otros microcontroladores) a través de cables.
- **Funcionamiento Básico:**
- ❖ **Inicio:** Cuando se enciende el microcontrolador, comienza a ejecutar el programa almacenado en su memoria Flash.
- ❖ **Ciclo de Instrucciones:** La CPU lee una instrucción a la vez de la memoria de programa. Cada instrucción le dice a la CPU qué operación realizar (por ejemplo, leer un valor de un puerto, realizar un cálculo, escribir un valor en un puerto).
- ❖ **Ejecución de Instrucciones:** La CPU ejecuta cada instrucción, utilizando la memoria RAM para almacenar datos intermedios y resultados.
- ❖ **Interacción con el Mundo Exterior:** Los periféricos del microcontrolador se utilizan para leer señales de entrada, procesarlas según las instrucciones del programa, y luego generar señales de salida para controlar dispositivos externos.
- ❖ **Repetición:** El microcontrolador sigue ejecutando el ciclo de instrucciones una y otra vez, a una velocidad muy alta (determinada por su frecuencia de reloj), lo que le permite responder rápidamente a los cambios en su entorno.

Ejemplo: Microcontrolador de semáforo. Su programa hace lo siguiente:

- ❖ Encender la luz verde.
- ❖ Esperar un tiempo determinado (usando un temporizador).
- ❖ Apagar la luz verde y encender la luz amarilla.
- ❖ Esperar un tiempo más corto.
- ❖ Apagar la luz amarilla y encender la luz roja.
- ❖ Esperar un tiempo determinado.
- ❖ Volver al paso 1.
- ❖ Este ciclo se repetirá continuamente, controlando las luces del semáforo en función del tiempo.

PIC16F84A

- ❖ **¿Qué es?** El PIC16F84A es un microcontrolador de 8 bits fabricado por Microchip Technology. Es un pequeño chip de computadora que contiene un procesador, memoria, y puertos de entrada/salida (I/O) que le permiten interactuar con el mundo exterior.
- ❖ **¿Para qué sirve?** Se utiliza para controlar y automatizar una amplia variedad de dispositivos electrónicos, desde electrodomésticos y juguetes hasta sistemas industriales y médicos. Algunas de sus aplicaciones comunes incluyen:
 - Control de luces y motores.
 - Sensores de temperatura, luz, y movimiento.
 - Interfaces de comunicación (USB, RS-232, etc.).
 - Temporizadores y contadores.
 - Sistemas de seguridad y alarma.

2. Ensamblador (Assembly Language):

- ❖ **¿Qué es?** El ensamblador es un lenguaje de programación de bajo nivel que está muy cerca del lenguaje de máquina que entiende el microcontrolador. Cada instrucción en ensamblador corresponde directamente a una operación que el microcontrolador puede realizar.
- ❖ **¿Para qué se usa?** El ensamblador se utiliza para programar microcontroladores como el PIC16F84A. Aunque es más difícil de aprender y usar que lenguajes de alto nivel como C o Python, ofrece varias ventajas:
- ❖ **Control preciso:** Permite un control muy preciso sobre el hardware del microcontrolador, lo que es importante para aplicaciones que requieren un rendimiento optimizado o un acceso directo a los registros y periféricos del microcontrolador.
- ❖ **Eficiencia:** El código ensamblador suele ser más eficiente en términos de tamaño y velocidad de ejecución que el código generado por lenguajes de alto nivel. Esto es crucial en sistemas con recursos limitados, como microcontroladores con poca memoria.
- ❖ **Comprensión profunda:** Aprender ensamblador te brinda una comprensión más profunda de cómo funciona el microcontrolador a nivel de hardware, lo que puede ser útil para solucionar problemas y optimizar el rendimiento.

Ejemplo de Uso del PIC16F84A y Ensamblador:

- ❖ Alarma con sensores para detectar movimiento. Podrías usar un PIC16F84A para leer la señal del sensor, procesarla, y luego controlar la sirena a través de uno de sus pines de salida. El código del ensamblador le diría al PIC16F84A exactamente cómo realizar estas tareas, paso a paso.

¿Por qué usar el PIC16F84A?

- ❖ El PIC16F84A es un microcontrolador muy popular debido a su bajo costo, facilidad de uso, y amplia disponibilidad de herramientas y recursos de desarrollo. Es una excelente opción para proyectos de electrónica y robótica de aficionados, así como para aplicaciones industriales y comerciales a pequeña escala.