



TECNOLÓGICO
NACIONAL DE MÉXICO



SISTEMAS PROGRAMABLES

SEMESTRE ENERO – JUNIO 2024

DOCENTE: ROYCE RODRÍGUEZ

UNIDAD IV – PROGRAMACIÓN

DE MICROCONTROLADORES

EQUIPO 3

ROCÍO VANESA GARDEA HERNÁNDEZ 21550330

HÉCTOR ALEJANDRO RODRIGUEZ BARRÓN 21550353

JOSÉ SEBASTIAN LÓPEZ IBARRA 21550362

ANDRÉS SAÉNZ OLIVAS 21550390

JORGE EDUARDO ESCOBAR BUGARINI 21550317

EDGAR GERARDO DELGADO CERRILLO 21550297

Contenido

Programación de microcontroladores	1
Introducción a los microcontroladores.....	1
Características del lenguaje C	1
Programador (grabador/quemador) de PIC	2
Cuánto tiempo me tomará aprender a programar los PIC?	2
Cuáles son los pasos a seguir para realizar una aplicación real con un PIC?	2
Características de los PIC	3
Conceptos básicos	4
Ejemplos de ejercicios resueltos.	5
Comunicación PIC a PIC	5
Código en mikroC PRO (lenguaje C). Primer código fuente	6
Discador telefónico DTMF con el PIC16F628A	8
Características.....	8
Display POV	9
Código en lenguaje C	9
Lenguajes de programación.....	11
Lenguaje ensamblador	11

Programación de microcontroladores

Introducción a los microcontroladores

Los microcontroladores PIC, desarrollados por Microchip, son chips integrados que pueden ser programados para ejecutar instrucciones automáticamente. Estos dispositivos se programan en un entorno de programación adecuado, como el lenguaje C, y luego el código binario resultante se transfiere a la memoria de programa del microcontrolador. Estos chips constan de una CPU, espacios de memoria para programas y datos, y varios periféricos para interactuar con el entorno externo, todo dentro de un solo chip de silicio.

Los microcontroladores PIC (conocidos simplemente como PIC) son circuitos integrados que pueden ser programados para ejecutar una secuencia de instrucciones automáticamente. El programa se escribe en un computador en un ambiente de programación adecuado (por ejemplo en lenguaje C) y luego se transfiere como código binario a la memoria de programa del microcontrolador para su ejecución en el momento de encendido del dispositivo. Los microcontroladores PIC son fabricados por la empresa Microchip y constan básicamente de una CPU, espacios de memoria para el programa y los datos, y varios periféricos para la interacción con el mundo externo, todo esto encapsulado dentro de un solo chip de silicio.

Características del lenguaje C

El lenguaje C es ampliamente conocido por su versatilidad y se enseña como lenguaje básico de programación en muchas instituciones educativas. Es considerado un estándar de facto en la enseñanza de la programación. Es un lenguaje de alto nivel, potente en comparación con el lenguaje ensamblador. El ambiente de desarrollo mikroC proporciona todas las herramientas y facilidades necesarias para la programación efectiva de microcontroladores PIC.

Programador (grabador/quemador) de PIC

Un programador (grabador o quemador) de PIC es una tarjeta electrónica en la que se inserta un microcontrolador PIC para transferir el código del programa desde el computador del usuario y grabarlo en la memoria interna del microcontrolador. Se conecta al computador a través de puertos como LPT1, USB, etc. El software de control facilita la transferencia del código ejecutable *.hex de una aplicación específica al PIC. Además, un programador también es la persona encargada de escribir un programa en un lenguaje de programación para resolver un problema, y el resultado es una aplicación.

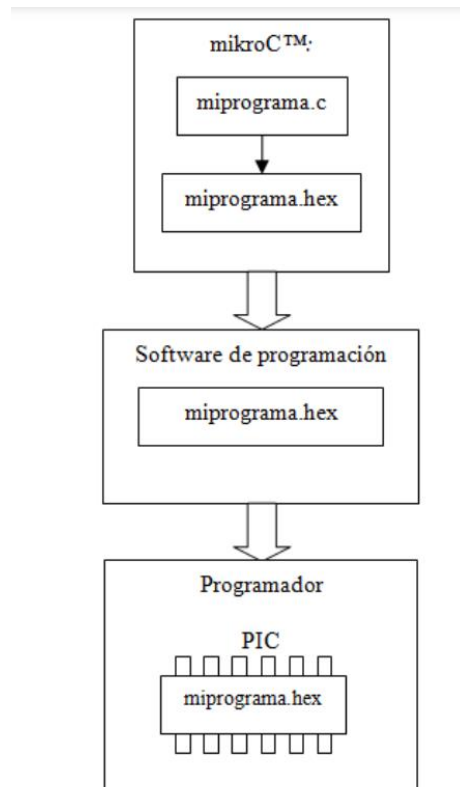
Cuánto tiempo me tomará aprender a programar los PIC?

Por supuesto no hay una respuesta exacta, lo que sí se puede afirmar es que mientras más perseverancia y dedicación se pongan de parte del lector, el tiempo necesario para el aprendizaje se reducirá. Si se dedican unas 2 horas al día el avance será vertiginoso y en unas 4 semanas ya tendrá un dominio aceptable de la programación de PIC en lenguaje C.

Cuáles son los pasos a seguir para realizar una aplicación real con un PIC?

1. Construir o comprar un programador de PIC compatible con los dispositivos a programar y con el sistema operativo disponible.
2. Descargar el software de control (software de programación) del programador e instalarlo: por ejemplo PICkit2 v2.61, etc.
3. Descargar e instalar el ambiente de desarrollo de programación de PIC en lenguaje C (mikroC).
4. Seguir las indicaciones del libro para crear un nuevo proyecto, configurarlo, escribir el código fuente en lenguaje C y compilarlo para obtener el ejecutable *.hex (aplicación).
5. Descargar el fichero ejecutable al PIC por medio del software de programación y del programador conectado a la computadora personal.

6. Retirar el PIC del programador y colocarlo en el circuito de aplicación y, ¡a disfrutar con los maravillosos resultados. La siguiente figura muestra la secuencia de programación:



Características de los PIC

Microchip® (empresa fabricante de los microcontroladores PIC) ofrece una gran variedad de microcontroladores de 8, 16 y 32 bits, de bajo costo, con una arquitectura potente, varias tecnologías de memoria, herramientas de desarrollo fáciles de usar, documentación técnica completa y una red de ventas y distribución global. Estas son las principales ventajas del uso de microcontroladores PIC:

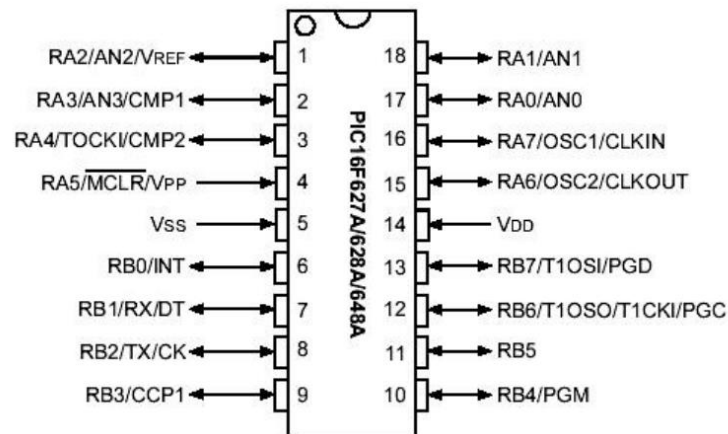
- Fácil migración y actualización de diseños.
- Amplio rango de velocidad de operación para satisfacer las múltiples necesidades de las aplicaciones.
- Dispositivos con memoria de programa desde 384B a 512 KB para cumplir los

requerimientos de código.

- Arquitecturas compatibles para preservar la inversión económica en el desarrollo de códigos.
- Compatibilidad Pin a Pin en múltiples encapsulados, facilitando así el remplazo.
- Menor coste total del sistema con Periféricos Integrados.
- Comunicaciones: SPI, I2C [™], UART, CAN, USB, Ethernet, IrDA®, LIN.
- Control: captura / comparación, contadores, reloj en tiempo real y PWM.
- Módulos Analógicos: Convertidores A / D, comparadores, amplificadores operacionales, Convertidores D / A

Conceptos básicos

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado fabricado de Silicio que puede programarse para realizar tareas (ejecutar instrucciones) de forma automática al conectarlo a una fuente de energía. Las instrucciones se almacenan en un espacio de memoria específicamente destinada a ese propósito. En la siguiente imagen se aprecia el patillaje o pinout del popular PIC16F628A:



Un microcontrolador está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica; incluye en su interior las tres principales unidades funcionales de una computadora:

La estructura básica de un microcontrolador PIC incluye la Unidad Central de

Procesamiento (CPU) para ejecutar instrucciones y operaciones con datos, Memorias como la FLASH para instrucciones del programa, EEPROM para almacenamiento permanente de datos, y RAM para almacenamiento temporal de datos. Los Periféricos de Entrada/Salida permiten la comunicación con el entorno físico mediante señales eléctricas, conectándose a través de las patillas del microcontrolador.

Adicionalmente existe otra terminología que conviene conocer y entender ya que se emplea con mucha frecuencia en el mundo de la programación de microcontroladores:

El proceso de desarrollo de programas para microcontroladores PIC implica el uso de un Compilador, que convierte el código escrito en lenguaje C a código de máquina, y un IDE (Ambiente de Desarrollo Integrado) como mikroC PRO for PIC, que incluye un editor de texto y un compilador. El Programador, dispositivo de hardware, se utiliza para transferir el código ejecutable desde el computador a la memoria de programa del microcontrolador, conectándose mediante el puerto USB.

Los Pines son las patillas del microcontrolador, utilizadas para conectar periféricos. La Sintaxis se refiere al conjunto de reglas para escribir correctamente un lenguaje de programación, siendo la de mikroC PRO para PIC similar a la del lenguaje C de computadoras (ANSI C). Un Programa es una secuencia de instrucciones escritas en un lenguaje de programación para realizar una tarea específica en un microcontrolador, en este caso utilizando la sintaxis de mikroC PRO for PIC.

Ejemplos de ejercicios resueltos.

Comunicación PIC a PIC

La comunicación alámbrica entre microcontroladores PIC es de suma importancia pues permite tener sistemas con múltiples microcontroladores ejecutando distintas tareas en forma coordinada. La transmisión de datos se puede lograr de forma bidireccional usando el módulo USART incorporado o funciones propias del compilador mikroC PRO. Se describe la comunicación entre dos microcontroladores PIC16F628A usando el módulo USART-Software de mikroC.

Consta de dos proyectos, uno para cada PIC:

1. El proyecto PIC a PIC LCD es utilizado para recibir datos y presentarlos en un LCD 2x16.
2. El proyecto PIC a PIC se utiliza para enviar datos.

Código en mikroC PRO (lenguaje C). Primer código fuente

```
//PIC_a_PIC_LCD.c
//Microcontrolador: PIC16F628A
//Oscilador: Interno-4MHz
//Este programa se almacena en uno de los dos microcontroladores (U2) y recibe
continuamente un
//dato (la letra 'r') cada 300 ms por el pin de recepción RB6(12). El pin de transmisión
es //RB7(13). El dato recibido se va presentando en el LCD.
//La velocidad de transmisión de 600 Baudios fue seleccionada experimentalmente
por un
//procedimiento de ensayo y error. El pin RB0(6) se enciende por 1 segundo en el
caso de
//intentar establecer una velocidad de transmisión muy alta o muy baja.
//Declaración de las 12 variables necesarias para la conexión del módulo LCD.
sbit LCD_RS at RA4_bit;
sbit LCD_EN at RA6_bit;
sbit LCD_D4 at RA0_bit;
sbit LCD_D5 at RA1_bit;
sbit LCD_D6 at RA2_bit;
sbit LCD_D7 at RA3_bit;
sbit LCD_RS_Direction at TRISA4_bit;
sbit LCD_EN_Direction at TRISA6_bit;
sbit LCD_D4_Direction at TRISA0_bit;
sbit LCD_D5_Direction at TRISA1_bit;
sbit LCD_D6_Direction at TRISA2_bit;
sbit LCD_D7_Direction at TRISA3_bit;
// Fin de declaración de variables de conexión.

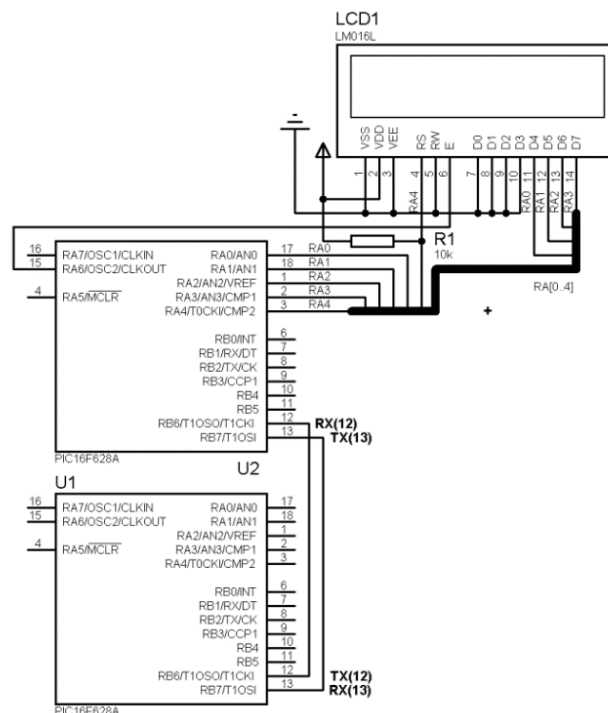
char error, byte_read;
void main(){
```



```

CMCON=0x07; //Pines RA como E/S digital.
PORTB=0x00;
TRISB0_bit=0;
Lcd_Init(); //Inicializa el LCD.
Lcd_Cmd(_LCD_CLEAR); //Borra el display.
Lcd_Cmd(_LCD_CURSOR_OFF); //Apaga el cursor.
error = Soft_UART_Init(&PORTB, 6, 7, 600, 0); //Inicializa el módulo UART-Software
a 600 Baudios.
if (error >0){
    RB0_bit=1;
    Delay_ms(1000);
    RB0_bit=0;
    while(1); //Detiene el programa en caso de error.
}
Delay_ms(100);
while(1) {
    byte_read = Soft_UART_Read(&error); //Recibe un dato y lo guarda en
byte_read.
    if (!error) //Si no hay error.
        Lcd_Chr_CP(byte_read); //Envía el dato recibido hacia el LCD.
    Delay_ms(200);
}
}

```



Discador telefónico DTMF con el PIC16F628A

El ejemplo mostrado intenta describir únicamente el proceso básico de generación de llamadas con el microcontrolador PIC16F628A y el IC W91314A. El discador telefónico DTMF W91314A de Windbond Electronics es un circuito integrado monolítico que proporciona las señales necesarias para marcación por pulsos o tonos. El circuito integrado W91314A tiene una memoria de redial.

Características

- Discador telefónico conmutable entre tonos DTMF ó pulsos.
- 32 dígitos para la memoria de redial.
- Tecla de Pulsos a Tonos (P -> T) para el funcionamiento de llamadas de larga distancia.
- Utiliza un teclado 4x4 o control electrónico de un microcontrolador.
- Fácil manejo con teclas de redial, flash, pausa y P -> T.
- Las funciones de Flash, pausa, P -> T (pulso-a- tonos) se pueden almacenar como un dígito en la memoria.
- Duración mínima de los tonos de salida: 100 ms.
- Pausa mínima entre tonos: 100 ms.
- Reset al encendido incorporado en el chip.
- Utiliza un cristal o resonador cerámico de 3.579545 MHz.
- Envasado en DIP plástico de 18 pines.

Los diferentes discadores en la serie W91310 se muestran en la siguiente tabla:

TYPE NO.	PULSE (ppS)	FLASH (mS)	FLASH PAUSE (mS)	PAUSE (S)
W91312/A	10/20	600/300	600/300	2.0
W91314/A	10/20	73/100	140/100	3.6
W91316	10/20	600	600	3.6

Display POV

El display POV (Persistense Of Vision) opera en base a la persistencia de la visión humana cuando un objeto se mueve rapidamente; la rotación de una columna de LEDs que se encienden en determinada secuencia hace que se forme en la mente del observador una imagen que depende del código programado en un microcontrolador PIC.

El display POV consta de un PIC16F88, un regulador de voltaje ajustable ECG1900 para el PIC (para tener control de la corriente que circula por los LEDs), una columna de 7 LEDs RGB y un motor de CD de 5V, con control de velocidad por variación de voltaje LM317T. Para cada letra el PIC envía secuencialmente 5 grupos de 7 bits, ya sea al puerto A (verde) o al puerto B (rojo). Debido a la rotación de la columna de LEDs en sentido antihorario y al fenómeno de la persistencia de la visión, se va formando en la mente del observador la imagen correspondiente. El PIC se puede programar con diferentes secuencias para presentar una gran variedad de mensajes.

Código en lenguaje C

```
//POV.c void main(){ OSCCON=0x70; //Oscilador interno a 8MHz (TCL=0,5 us).
PORTA=0x00; //Iniciación. PORTB=0x00; ANSEL=0x00; //Pines AN como E/S
digital. TRISA=0x00; //Puerto A como salida.
TRISB=0x00; //Puerto B como salida.
//Delay_us(5000);
PORTB=0x7E;Delay_us(500); //m
PORTB=0x02;Delay_us(900);
PORTB=0x7E;Delay_us(500);
PORTB=0x02;Delay_us(900);
PORTB=0x7C;Delay_us(500);
PORTB=0x00;Delay_us(1500);

PORTA=0xDA;Delay_us(500); //i
PORTA=0x00;Delay_us(1500);

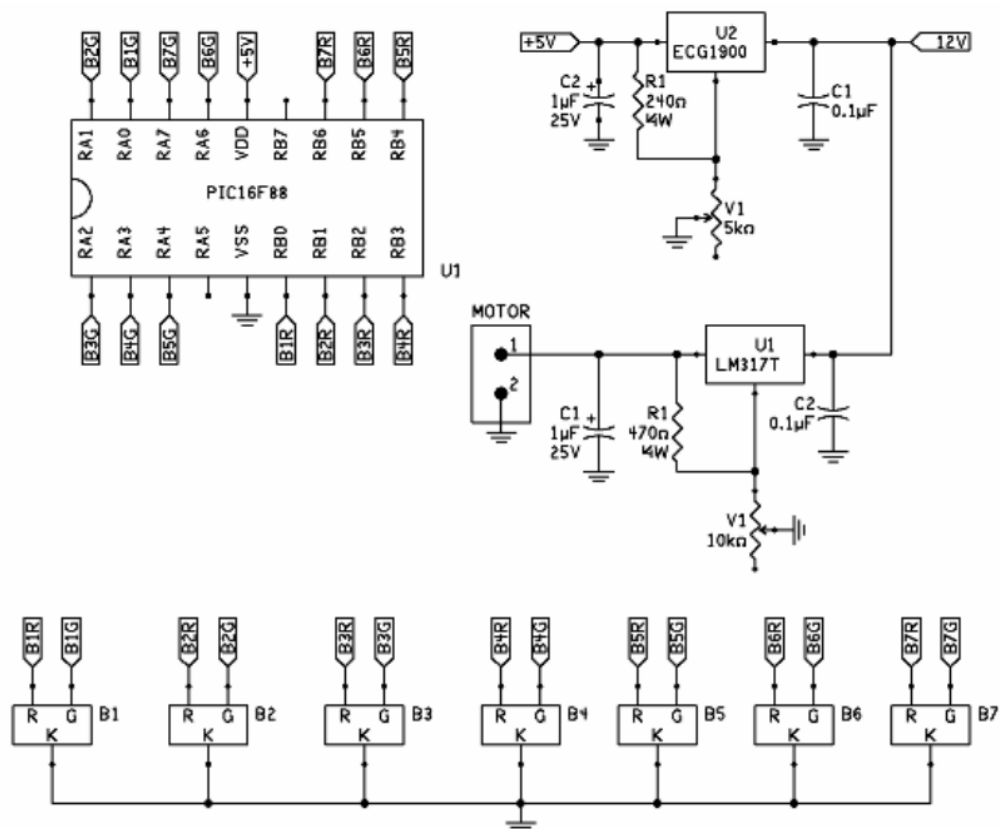
PORTB=0x3C;Delay_us(500); //c
PORTB=0x42;Delay_us(1500);
PORTB=0x24;Delay_us(500);
```

```
PORTB=0x00;Delay_us(1500);
```

```
PORTA=0xDE;Delay_us(500); //r
PORTA=0x04;Delay_us(500);
PORTA=0x02;Delay_us(1000);
PORTA=0x04;Delay_us(500);
PORTA=0x00;Delay_us(1500);
```

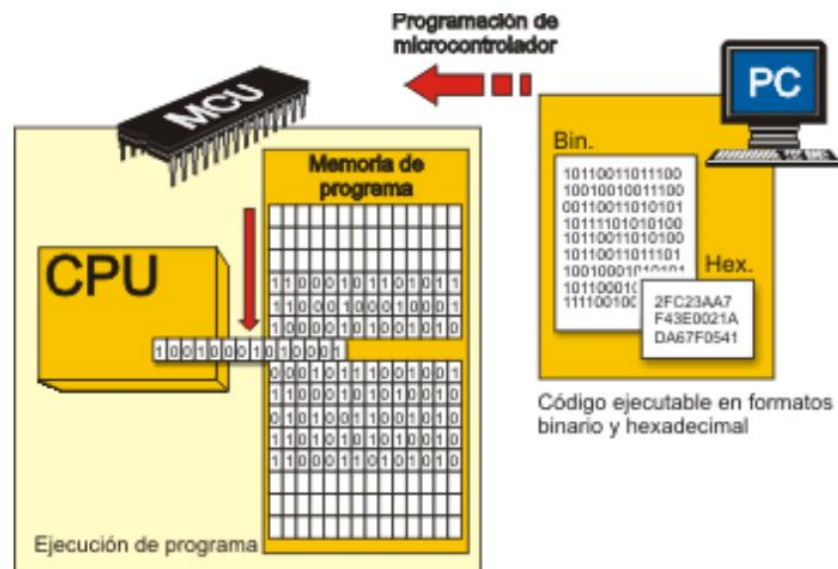
```
PORTB=0x3C;Delay_us(500); //o
PORTB=0x42;Delay_us(1500);
PORTB=0x3C;Delay_us(500);
PORTB=0x00;Delay_us(1500);
```

```
PORTA=0x5E;Delay_us(500); //C
PORTA=0x81;Delay_us(1500);
PORTA=0x42;Delay_us(500);
PORTA=0x00;
}
```



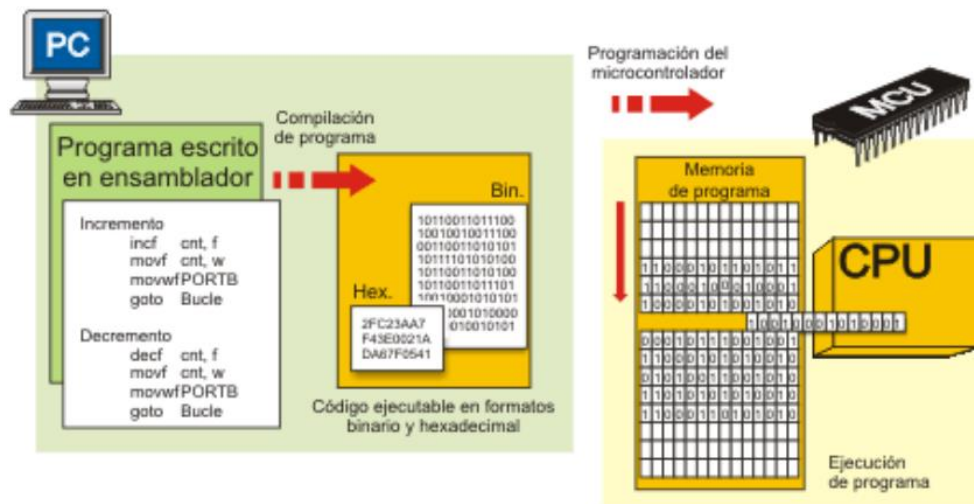
Lenguajes de programación.

El microcontrolador ejecuta programas almacenados en memoria Flash, representados por código ejecutable compuesto de ceros y unos. Estos códigos, dependiendo de la arquitectura del microcontrolador, pueden ser de 12, 14 o 16 bits de anchura. Cada palabra binaria es interpretada como una instrucción por la CPU. Todas estas instrucciones conforman el Conjunto de Instrucciones. Para facilitar la representación, el código ejecutable se convierte a hexadecimal, conocido como código Hex. En microcontroladores PIC de 14 bits, el conjunto de instrucciones consta de 35 instrucciones diferentes.



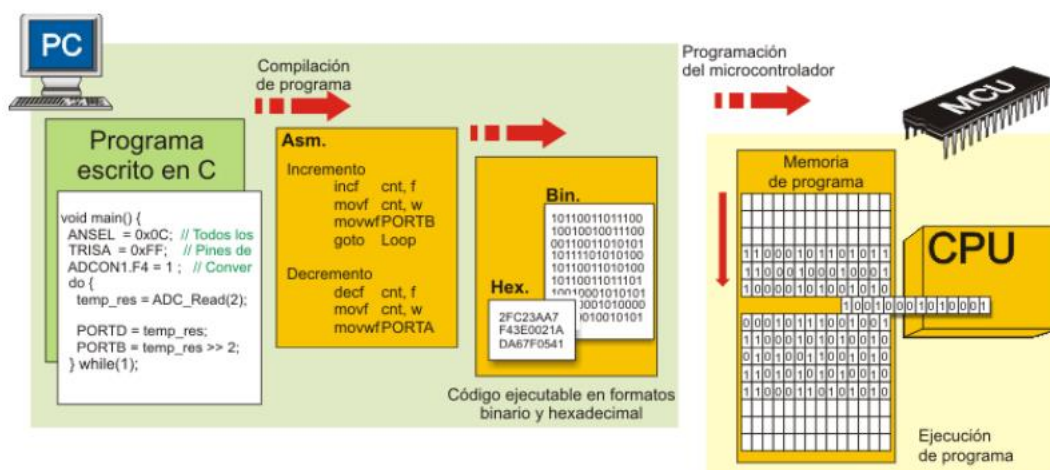
Lenguaje ensamblador

El Lenguaje Ensamblador (ASM) fue creado como una forma más comprensible de escribir código ejecutable, siguiendo una sintaxis básica que facilita la escritura y comprensión del código. Las instrucciones en ensamblador se representan mediante abreviaturas con significado, y cada instrucción corresponde a una localidad de memoria. Un programa ensamblador compila (traduce) estas instrucciones a código máquina (binario).



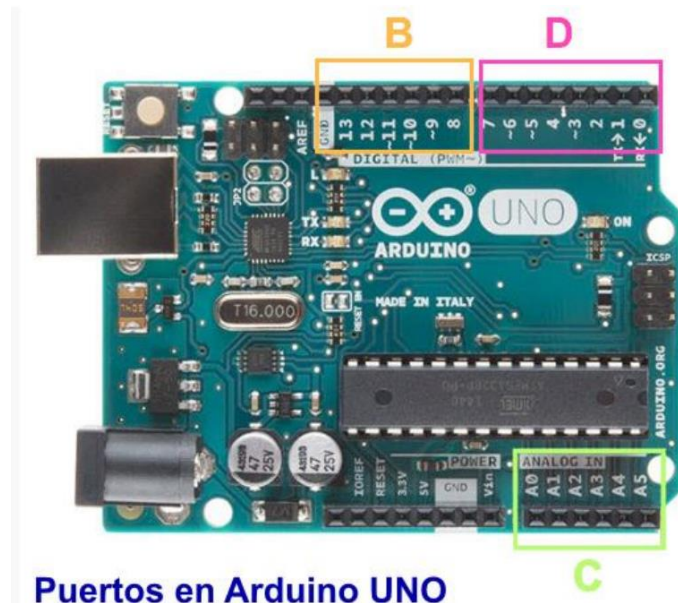
Lenguaje C

El lenguaje C dispone de todas las ventajas de un lenguaje de programación de alto nivel (anteriormente descritas) y le permite realizar algunas operaciones tanto sobre los bytes como sobre los bits (operaciones lógicas, desplazamiento etc.). Las características de C pueden ser muy útiles al programar los microcontroladores. Además, C está estandarizado (el estándar ANSI), es muy portable, así que el mismo código se puede utilizar muchas veces en diferentes proyectos. Lo que lo hace accesible para cualquiera que conozca este lenguaje sin reparar en el propósito de uso del microcontrolador. C es un lenguaje compilado, lo que significa que los archivos fuentes que contienen el código C se traducen a lenguaje máquina por el compilador. Todas estas características hicieron al C uno de los lenguajes de programación más populares.



Programación en Arduino UNO

El PORTB tiene dos líneas ocupadas por el cristal oscilador y el PORTC tiene dos bits no disponibles, uno usado como entrada de reinicio y el otro no cableado hacia el exterior del microcontrolador. Los registros DDRx, PORTx y PINx se utilizan para administrar los puertos de entrada/salida. DDRx determina si un bit es entrada o salida, PORTx controla el nivel del pin (ALTO o BAJO) y define la resistencia de



Puertos en Arduino UNO

polarización pull-up si es una entrada, mientras que PINx permite leer el estado de los pines. Estos registros permiten definir los pines como entrada o salida con una sola instrucción, y también controlar las resistencias pull-up internas para reducir el ruido eléctrico.

PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
PORTC	—	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
DDRC	—	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
PINC	—	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0

