**ARQUITECTURA DE COMPUTADORAS**

**Practica 02. Salida Individual**

**OBJETIVO:**

Comprender la configuración de los puertos del microcontrolador. Configurar varios puertos del PIC como salida. Enviar salida a pines individuales del PIC.

**MATERIALES Y EQUIPO:**

|  |  |
| --- | --- |
| Microcontrolador 16F877A con cristal (16 MHz) y capacitores (22 pF) |  |
| Protoboard (mínimo de 400 puntos) |  |
| 1 Resistencia de 10Kohm |  |
| Cables de conexión (UTP o Dupont) |  |
| Entrenadora Digital (IDL-800) |  |
| Grabadora de Microcontroladores (Master PROG) con su cable USB y el software de grabación. | Resultado de imagen de master prog mercadolibre |

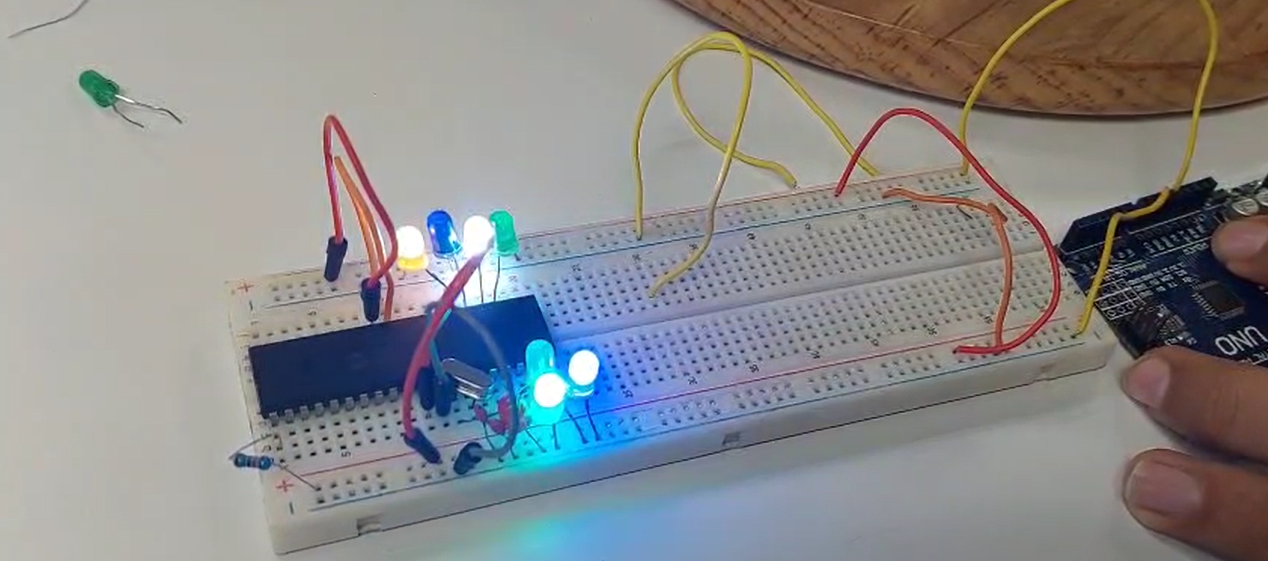
**INTRODUCCIÓN:**

En esta práctica seguimos aprendiendo a manejar los puertos del microcontrolador, pero ahora enviando señales a pines individuales. Los microcontroladores tienen pines que se pueden activar de forma separada y esto es útil para controlar varios dispositivos al mismo tiempo. En este caso usamos el PIC16F877A y trabajamos principalmente con el puerto C, encendiendo y apagando los LEDs uno por uno para entender mejor cómo manejar cada bit del puerto.

**PROCEDIMIENTO:**

1. Realice la conexión básica del microcontrolador.

(Parte 1)



2. Codigo:

/\* ARQUITECTURA DE COMPUTADORAS Ing. Daniel López Piña UAMM-UAT

\* Practica 02: Salida individual

\* Integrantes:

\* Una manera de accesar a los bit individuales es a través del selector "." seguido del identificador F0, F1, ... F7

\* Siendo el F7 el bit mas significativo

\* Se asigna un valor de 0 para LOW y un valor de 1 para HIGH

\* PIC16F877A 16Mhz

\*

\* Conectar todos los pines del puerto C a los leds de la Entrenadora Digital

\*/

void main(){

TRISC=0; //Configura todo el puerto C (RC0, RC1, RC2, RC3, RC4, RC5, RC6 y RC7) como Salida

// TRISC=0x00; //Es el equivalente a la línea anterior, pero asignándole valor Hexadecimales

// TRISC=0b00000000; //Es otro equivalente, pero asignándole valor Binario

PORTC=0; //Apaga todo el puerto C

PORTC.F0=1; // Manda un 1 lógico (HIGH) únicamente al pin RC0

//PORTC=0x01; //Es equivalente a la línea anterior, pero asignándole valor Hexadecimales

//PORTC=0b00000001; //Es otro equivalente, pero asignándole valor Binario

Delay\_ms(5000); //Espera 5 segundos

while(1){ //Inicia un ciclo infinito

PORTC.F0=1;

PORTC.F1=0;

PORTC.F2=1;

PORTC.F3=0;

PORTC.F4=1;

PORTC.F5=0;

PORTC.F6=1;

PORTC.F7=0;

// PORTC=0b01010101; // Equivale a las 8 líneas anteriores

// PORTC=0x55; // Equivalente en Hexadecimal

// PORTC=85; // Equivalente en Decimal

Delay\_ms(1000); // Espera 1 Segundo

PORTC.F0=0;

PORTC.F1=1;

PORTC.F2=0;

PORTC.F3=1;

PORTC.F4=0;

PORTC.F5=1;

PORTC.F6=0;

PORTC.F7=1;

// PORTC=0b10101010; // Equivale a las 8 líneas anteriores

// PORTC=0xAA; // Equivalente en Hexadecimal

// PORTC=170; // Equivalente en Decimal

Delay\_ms(1000); // Espera un segundo

}

}

3. Compilar y grabar el proyecto.

4. Realice otro proyecto y configure los puertos B, C, D y E como salida. Encienda y apague los pines en el siguiente orden:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Secuencia | RB0 | RB1 | RC2 | RC3 | RD4 | RD5 | RE0 | RE1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

En tiempo es de 2 segundos entre secuencia.

Conecte los siguientes pines a los led del entrenador digital: RB0, RB1, RC2, RC3, RD4, RD5, RE0 y RE1.

(Parte 2)

// Practica 02: Salida individual - Proyecto 2

// PIC16F877A @ 16 MHz

void paso1() {

RB0\_bit=1; RB1\_bit=0;

RC2\_bit=1; RC3\_bit=0;

RD4\_bit=1; RD5\_bit=0;

RE0\_bit=1; RE1\_bit=0;

}

void paso2() {

RB0\_bit=1; RB1\_bit=1;

RC2\_bit=0; RC3\_bit=0;

RD4\_bit=1; RD5\_bit=1;

RE0\_bit=0; RE1\_bit=0;

}

void paso3() {

RB0\_bit=0; RB1\_bit=1;

RC2\_bit=1; RC3\_bit=1;

RD4\_bit=0; RD5\_bit=1;

RE0\_bit=1; RE1\_bit=1;

}

void paso4() {

RB0\_bit=1; RB1\_bit=1;

RC2\_bit=1; RC3\_bit=1;

RD4\_bit=0; RD5\_bit=0;

RE0\_bit=0; RE1\_bit=0;

}

void paso5() {

RB0\_bit=0; RB1\_bit=1;

RC2\_bit=0; RC3\_bit=1;

RD4\_bit=0; RD5\_bit=1;

RE0\_bit=0; RE1\_bit=1;

}

void main() {

ADCON1 = 0x07; // PORTA y PORTE como digitales (relevante para RE0/RE1)

TRISB = 0x00; // B como salida

TRISC = 0x00; // C como salida

TRISD = 0x00; // D como salida

TRISE = 0x00; // E como salida

PORTB = 0x00;

PORTC = 0x00;

PORTD = 0x00;

PORTE = 0x00;

while(1){

paso1(); Delay\_ms(2000);

paso2(); Delay\_ms(2000);

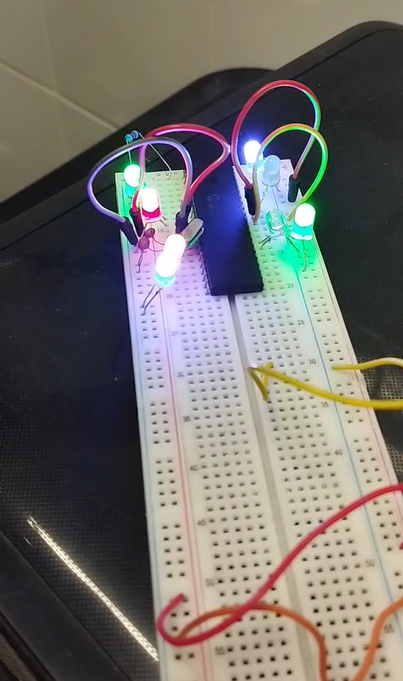
paso3(); Delay\_ms(2000);

paso4(); Delay\_ms(2000);

paso5(); Delay\_ms(2000);

}

}



Conclusión:

Con esta práctica reforzamos el uso de los puertos del microcontrolador, pero de manera más específica, controlando pines individuales. Al observar cómo cambia cada LED, entendimos cómo se manejan los bits de un puerto y cómo darles valores diferentes. Esto es importante porque en proyectos más avanzados necesitaremos controlar varios pines con distintas tareas al mismo tiempo.