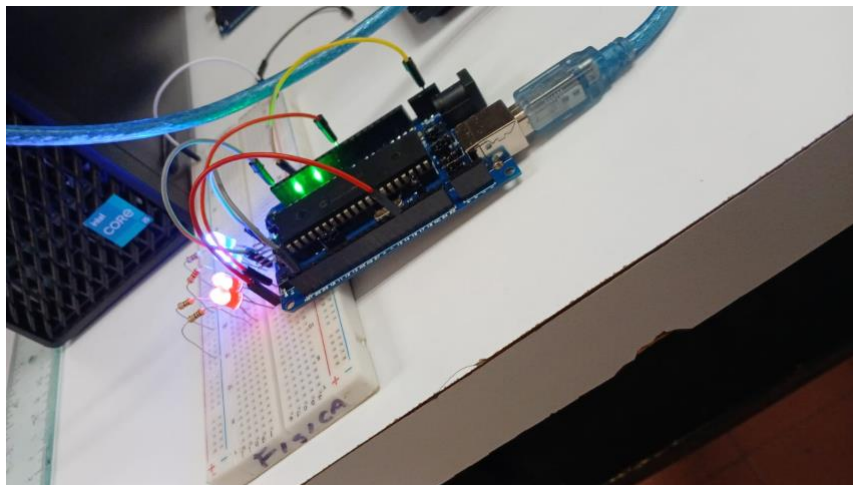


EXAMEN: ENCENDER 4 LEDS CON DIFERENTES PUERTOS DEL PIC16F877A



INTEGRANTES:

MALDONADO VÁZQUEZ MARIA CELESTE
HERNÁNDEZ HERNÁNDEZ MILCA LIZBETH
VERGARA TORRES CHRISTIAN DE JESUS
ENRIQUEZ HERNANDEZ GALDINO
CORPUS RENTERIA ANTONIO
VARGAS JIMENEZ JESSICA JANETH
VÁZQUEZ HERNÁNDEZ EMILY EDITH

DR. DANIEL LOPEZ PIÑA

MATERIA: PROGRAMACIÓN DE MICROCONTROLADORES

UNIDAD ACADÉMICA MULTIDISCIPLINARIA MANTE

ISC

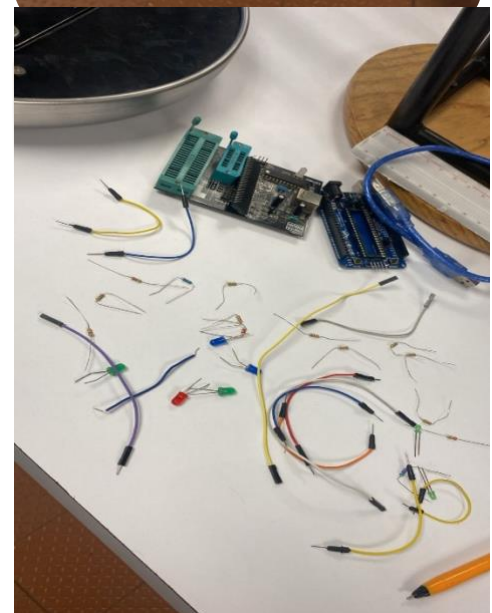
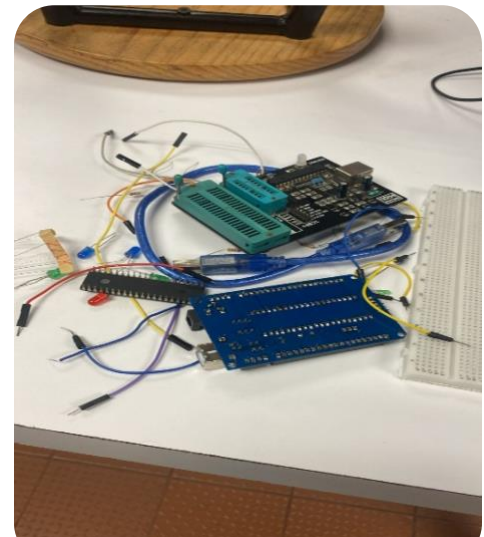
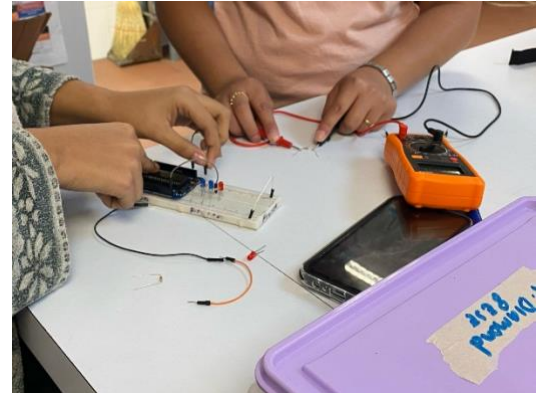
8EJF

INTRODUCCIÓN

El proyecto consiste en encender y apagar cuatro LEDs utilizando diferentes puertos del microprocesador PIC16F877A. Mediante la programación del PIC, se controlará el estado de cada LED de forma independiente, demostrando el manejo de salidas digitales. Este proyecto es ideal para comprender el funcionamiento básico del microcontrolador, la configuración de sus registros y la interacción con dispositivos externos.

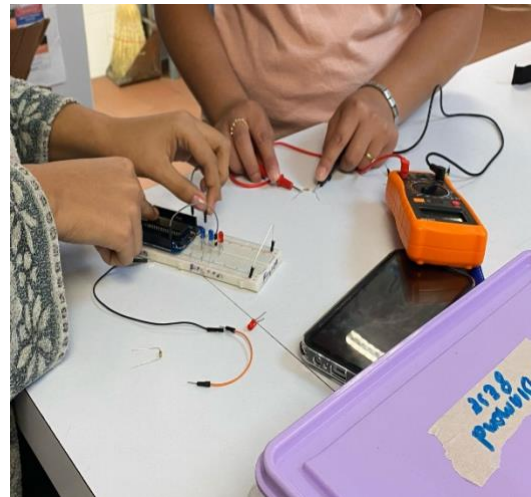
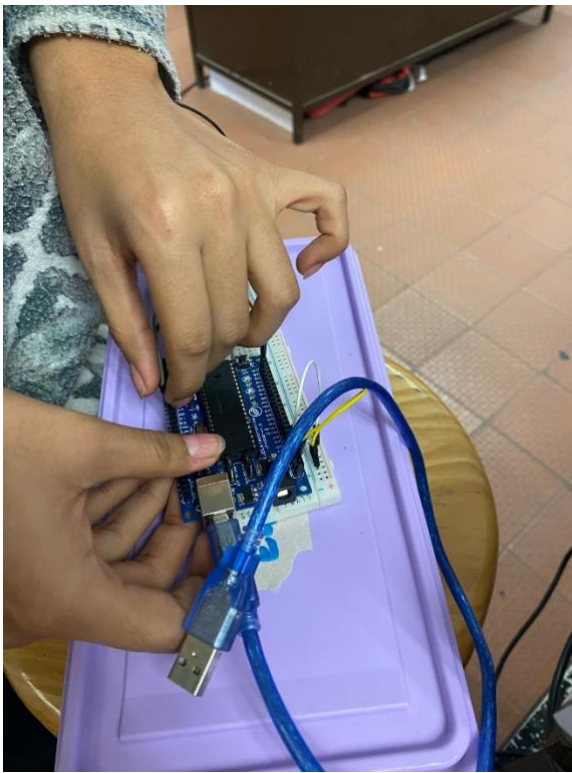
MATERIALES:

- MICROPROCESADOR PIC16F877a
- 4 LEDS
- 4 RESISTENCIAS (220 OHMIOS)
- CABLES DE CONEXION
- TARJETA ENTRENADORA
- PROTOBOARD
- MULTIMETRO
- TESTER
- SOFTWARE DE DESARROLLO MIKRO C PRO



DESARROLLO DEL PROYECTO

- Puerto A0
- Puerto E0
- Puerto C0
- Puerto B0
- PIC Conectado a 3 voltios Y GND
- Los ánodos del led conectados a los puertos del pic y los cátodos de los led conectados a resistencias de 220 ohms, Y luego a tierra



DESARROLLO DEL CÓDIGO

```
Lab_F:\OneDrive\Escritorio\Nueva carpeta\PracticaPrueba2.mcspi - NOT REGISTERED
File Edit View Tools Help
Start Page PracticePrueba2.c
//EQUIPO DIAMOND
- Vasquez Hernandez Emily Edith
- Corpus Renteria Antonio
- Enriquez Hernandez Galdino
- Hernandez Hernandez Milca Lizbeth
- Vergara Torres Crithian Jesús
- Maldonado Vázquez María Celeste
- Vargas Jimenez Jessica Janeth
- */
10
- void main() {
-     // Configura los puertos A, B, C, D y E como salida
-     TRISA = 0x00; // Puerto A como salida
-     TRISB = 0x00; // Puerto B como salida
-     TRISC = 0x00; // Puerto C como salida
-     TRISE = 0x00; // Puerto E como salida
-
-     while(1) {
-         PORTA = 0x01; // Enciende el LED en RA0
-         PORTB = 0x01; // Enciende el LED en RB0
-         PORTC = 0x01; // Enciende el LED en RC0
-         PORTE = 0x01; // Enciende el LED en RE0
-
-         Delay_ms(1000); // Espera 1 segundo
-
-         PORTA = 0x00; // Apaga el LED en RA0
-         PORTB = 0x00; // Apaga el LED en RB0
-         PORTC = 0x00; // Apaga el LED en RC0
-         PORTE = 0x00; // Apaga el LED en RE0
-
-         Delay_ms(1000); // Espera 1 segundo
-     }
- }
33
✓ Hints
Message Text
All files Compiled in 47 ms
```

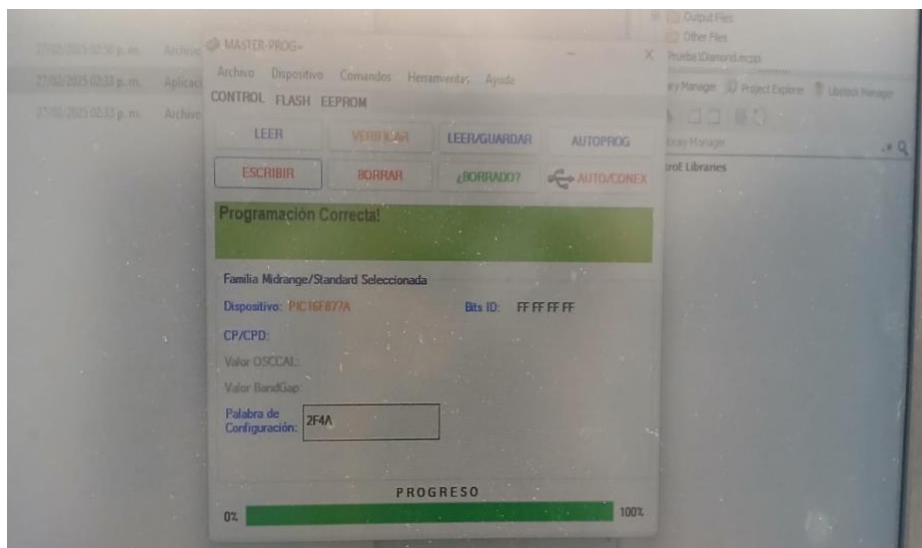
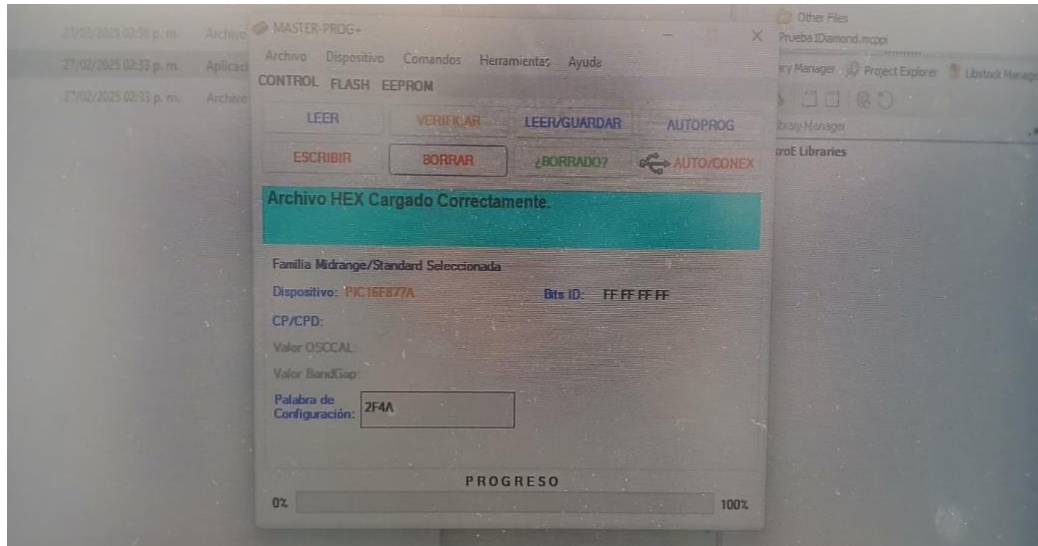
Resumen del funcionamiento del código:

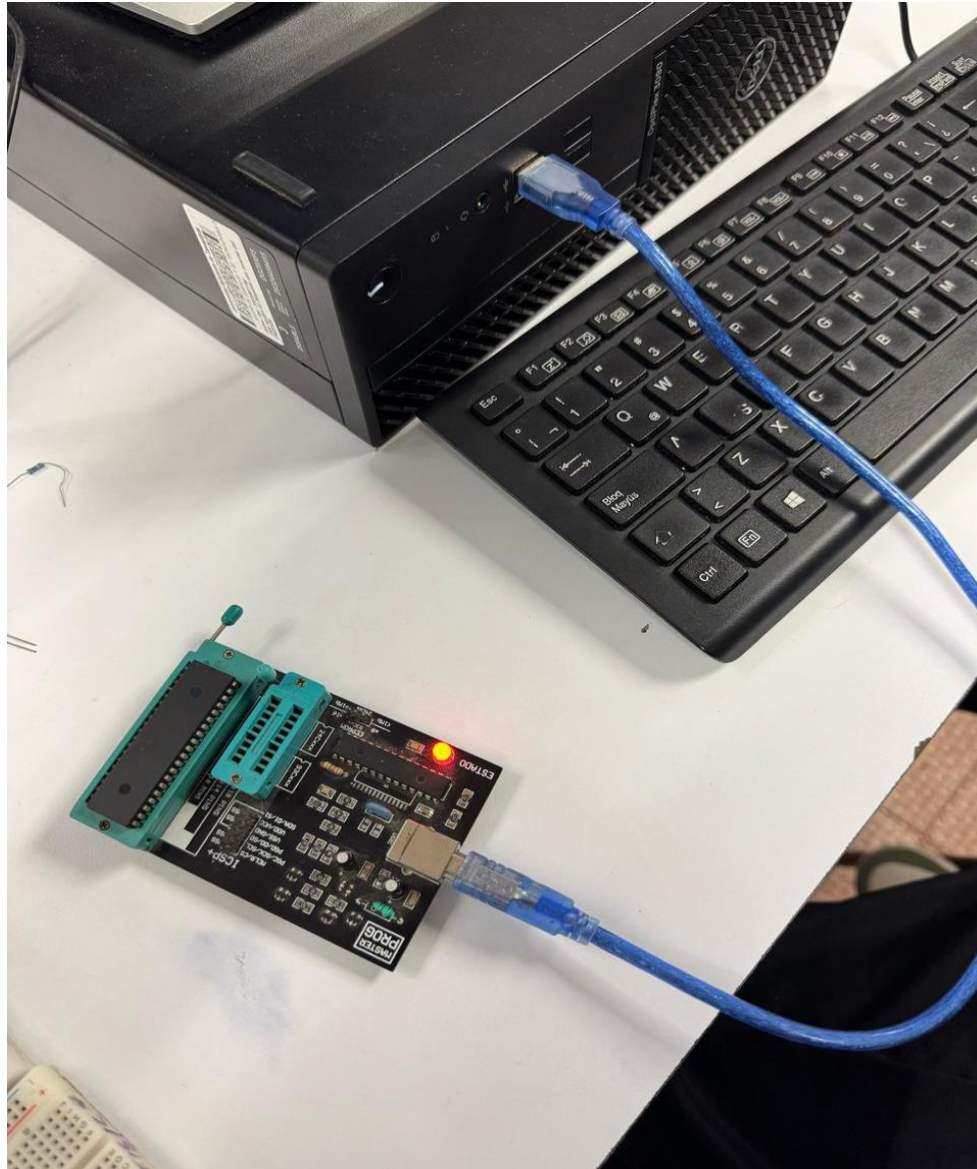
- El código está diseñado para microcontroladores PIC (como el PIC16F877A) y tiene la siguiente función principal:
- Configura los puertos A, B, C y E como salidas digitales. Esto permite controlar los pines de estos puertos para encender o apagar dispositivos como LEDs.
- **Enciende los LEDs** conectados a los pines **RA0, RB0, RC0 y RE0** al asignar un valor de 0x01 a cada puerto (enciende el primer bit de cada puerto).
- **Espera 1 segundo** utilizando la función `Delay_ms(1000);`, que introduce una pausa de **1000 milisegundos** (1 segundo).
- **Apaga los LEDs** asignando 0x00 a los mismos puertos, lo que apaga los pines de salida.
- Vuelve a esperar 1 segundo antes de repetir el proceso.
- Se ejecuta en un bucle infinito (`while(1)`), lo que significa que el programa no termina y el ciclo de encender y apagar los LEDs se repite indefinidamente.

Efecto visible:

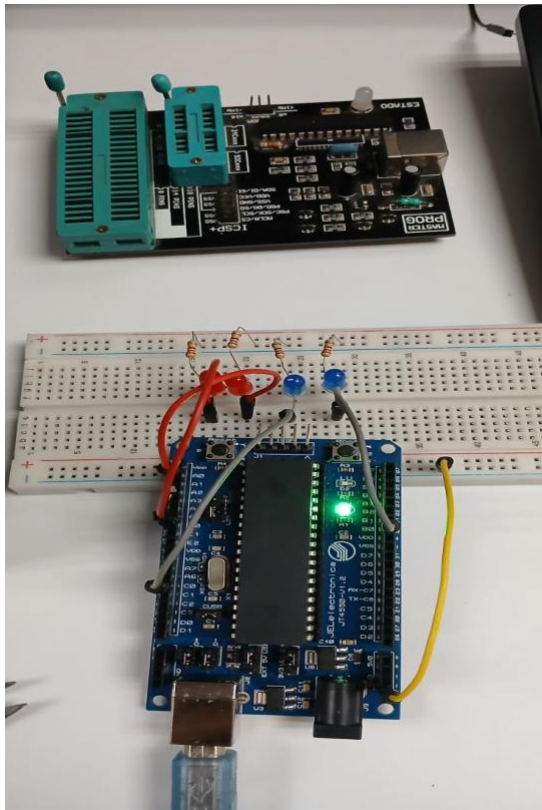
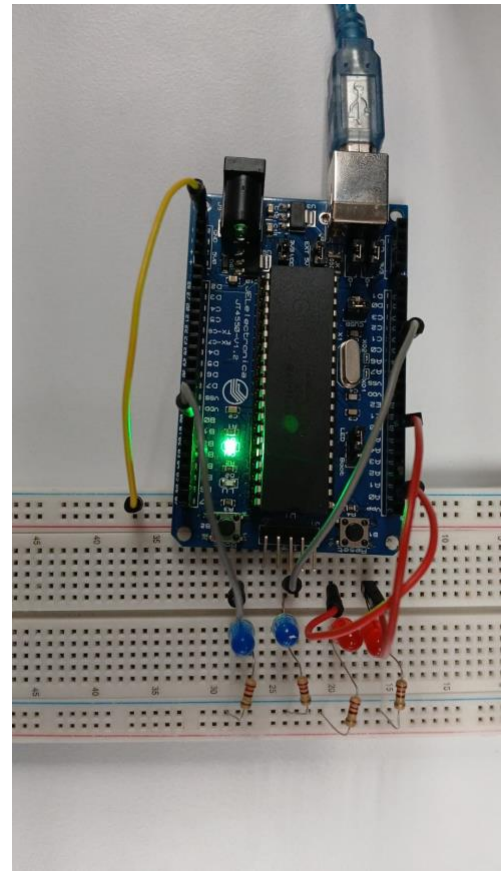
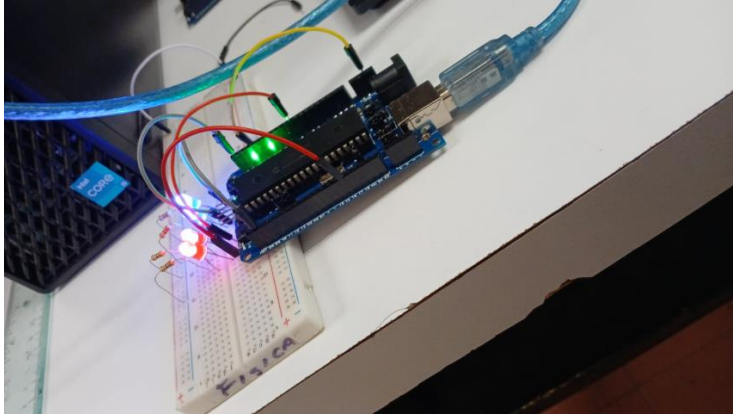
Los **LEDs** conectados a **RA0, RB0, RC0 y RE0** parpadean con un intervalo de **1 segundo** (1 segundo encendidos, 1 segundo apagados) de manera continua.

GRABADO DE MICROPROCESADOR





RESULTADOS



CONCLUSIÓN

El proyecto de encender cuatro LEDs con el microprocesador PIC16F877A permitió comprender el manejo de salidas digitales y la configuración de diferentes puertos. A través de la programación, se logró controlar cada LED de manera independiente, lo que facilita la base para desarrollar aplicaciones más complejas. Este ejercicio es fundamental para reforzar conocimientos en programación de microcontroladores, manipulación de registros y control de dispositivos electrónicos.