



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica y Electrónica

Carrera de Electrónica y Automatización

SISTEMAS BASADOS EN MCU

Práctica 3.3

**PROGRAMACIÓN DE GPIO BÁSICOS EN LENGUAJE C
PARA EL PIC16F877**

Autor:

Iza Tipanluisa Alex Paul

Docente:

Ing. Amparo Meythaler

NRC: 4891

1) OBJETIVOS

- Identificar la forma de trabajo de paquetes que utilizan lenguaje C para programar el PIC16F877.
- Realizar una aplicación de GPIO básicos en lenguaje C con el PIC16F877.

2) MARCO TEORICO

LENGUAJE C

Al programar en Lenguaje C, u otro de alto nivel, el programador ya no tiene que conocer el conjunto de instrucciones o características del hardware del microcontrolador utilizado. Ya no es posible conocer exactamente cómo se ejecuta cada sentencia.

C es un lenguaje bastante conciso y en ocasiones desconcertante. Considerado ampliamente como un lenguaje de alto nivel, posee muchas características importantes, tales como: programación estructurada, un método definido para llamada a funciones y para paso de parámetros, potentes estructuras de control, etc.

El lenguaje C 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.

3) OBJETIVOS

- Identificar la forma en la que están elaborados los teclados matriciales.
- Identificar la forma de implementar y programar un teclado matricial con el PIC16F877.

4) EQUIPOS Y MATERIALES

- PC con el paquete CCS
- PC con el paquete PROTEUS.

5) ACTIVIDADES

1) Trabajo Preparatorio:

- a) Realice el diagrama de flujo, la codificación en lenguaje C y la implementación correspondientes de un programa que presente un contador para 8 leds (inicio y fin

el que Ud. desee) siempre que está cerrado un interruptor, caso contrario los leds estarán apagados.

Entre datos del contador debe estar un retardo de 0,2 segundos.

b) Verifique el funcionamiento del MPLAB IDE y del PROTEUS.

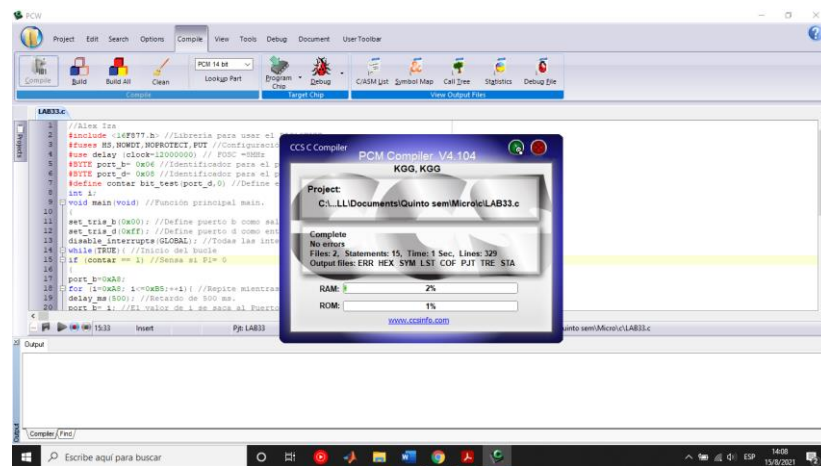


Ilustración 1, Compilación del programa

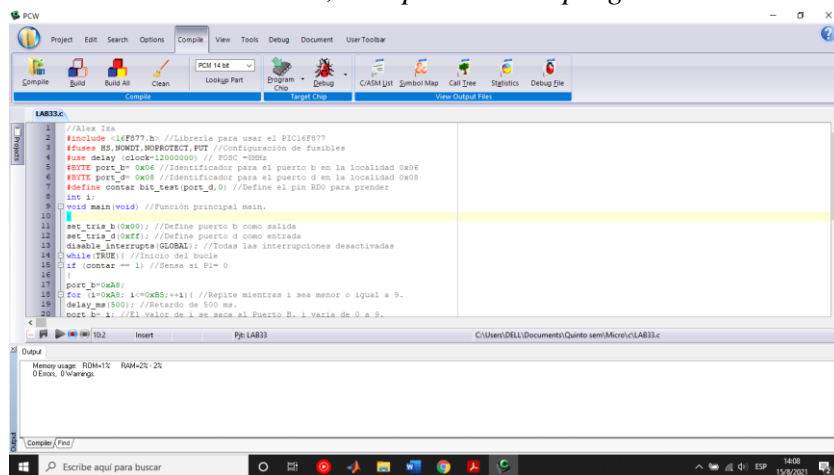


Ilustración 2, Compilación exitosa, 0 errores

6) RESULTADOS

- Explique los errores cometidos en el ejercicio realizado (si los tuvo) y la forma de corregirlos.

Tuve el error de ingresar los valores en hexadecimal para sacar por el puerto; lo solucioné colocando port_b=0xA8;

Explique para qué sirven las primeras instrucciones colocadas en el programa realizado.

Sirven para configurar los puertos; fusibles y la librería del PIC que se va usar.

7) DISEÑO

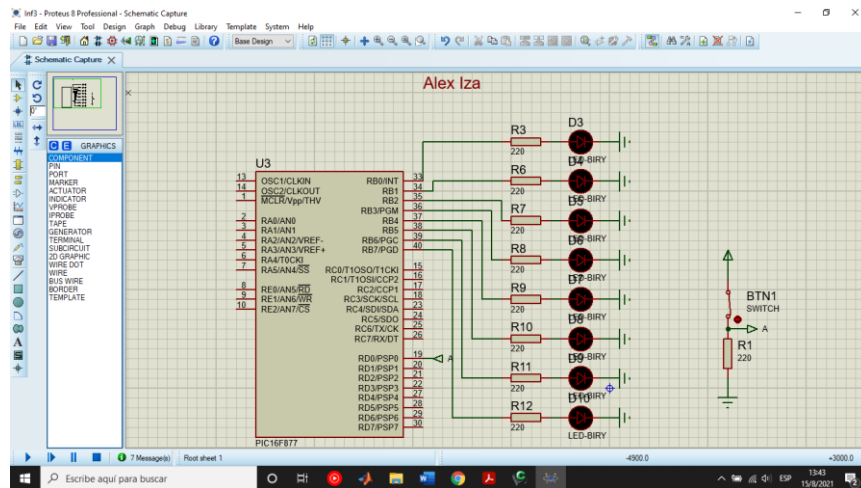


Ilustración 3, Implementación del circuito

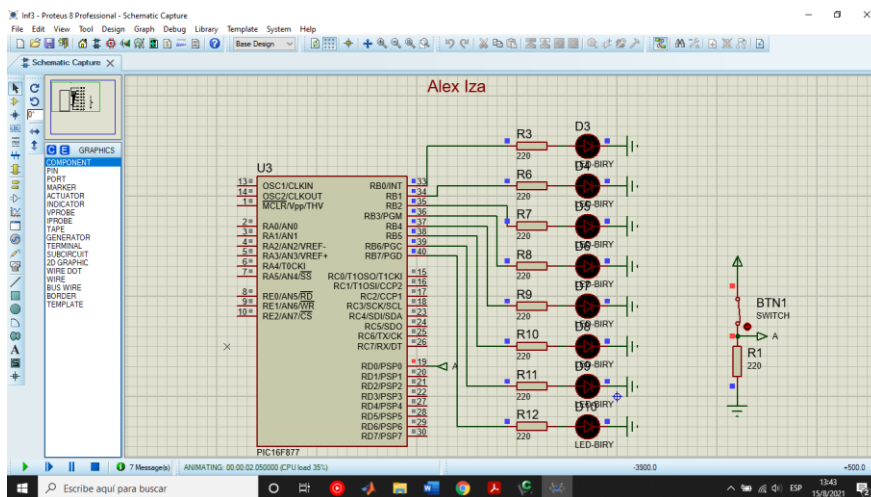


Ilustración 4, Switch cerrado no cuenta

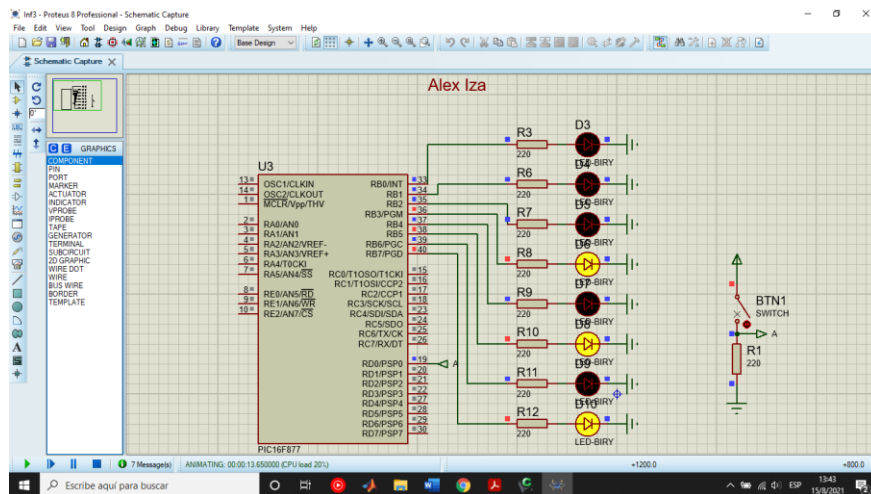


Ilustración 5, Switch abierto cuenta

8) DIAGRAMA DE FLUJO

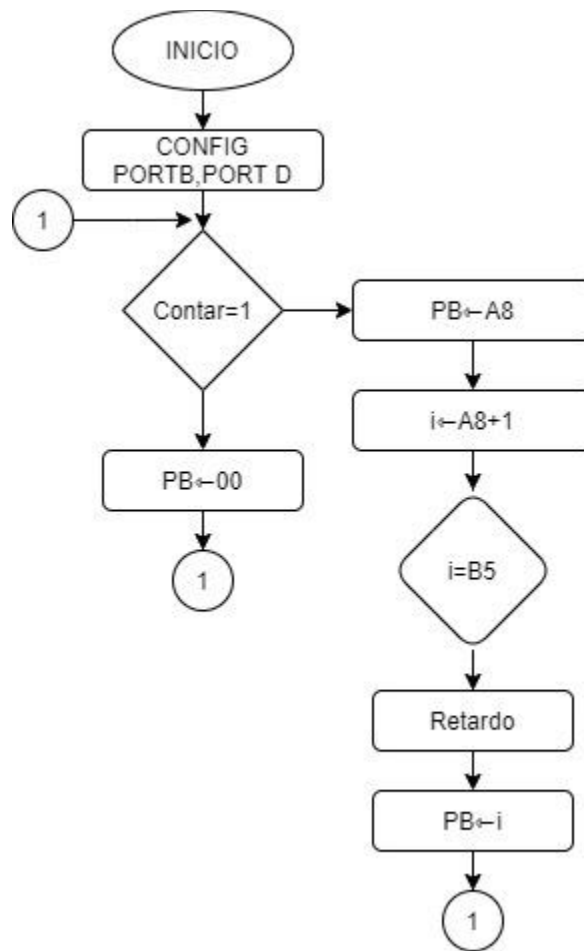


Diagrama 1, Programa

9) PROGRAMA

```
//Alex Iza

#include <16F877.h> //Librería para usar el PIC16F877

#fuses HS,NOWDT,NOPROTECT,PUT //Configuración de fusibles

#use delay (clock=12000000) // FOSC =8MHz

#BYTE port_b= 0x06 //Identificador para el puerto b en la localidad 0x06
#BYTE port_d= 0x08 //Identificador para el puerto d en la localidad 0x08

#define prender bit_test(contar_d,0) //Define el pin RD0 para contar

int i;

void main(void) //Función principal main.
{
    set_tris_b(0x00); //Define puerto b como salida
```

```

set_tris_d(0xff); //Define puerto d como entrada

disable_interrupts(GLOBAL); //Todas las interrupciones desactivadas

while(TRUE){ //Inicio del bucle

if (contar == 1) //Sensa si P1= 0

{

port_b=0xA8; El valor de A8 se saca al Puerto B

for (i=0xA8; i<=0xB5;++i){ //Repite mientras i sea menor o igual a B5.

delay_ms(500); //Retardo de 500 ms.

port_b= i; //El valor de i se saca al Puerto B. i varia de A8 a B5.

}

}

if (contar==0) //Sensa si el el interruptor = 0

{

port_b= 00; El valor de 00 i se saca al Puerto B

}

} //Fin de bucle infinito

} //Fin del main

```

10) CONCLUSIONES

- Se ha concluido que la pregunta del switch en el programa depende de la conexión que se realiza, si esta aterrizado a tierra o a Vcc.
- Se ha concluido que los dato que se desean sacar por los puertos; en el programa se pueden digitar como decimales o hexadecimales.
- Se ha concluido que se debe tener en cuenta los bits con los que se va a analizar en el PUERTO D ya que aquí pueden ingresar ceros o unos dependiendo la conexión de los pulsadores

11) RECOMENDACIONES

- Se recomienda que revisar la configuración de los puertos ya sean de entrada o de salida.
- Se recomienda verificar el funcionamiento del programa con la configuración de los pulsadores si son conectados a tierra o a Vcc, ya que cada uno de estos tiene una programación diferente

- Se recomienda revisar exhaustivamente los comandos correctos a usar en la programación CCS; este tiene una diferente programación, pero gracias al diagrama de flujo este tiene el mismo proceso .

12) BIBLIOGRAFIA

Meythaler, A. (2021). Sistemas basados en MCU. Ecuador: UFA ESPE