



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 2.3

PROGRAMACIÓN DE LOS GPIOs DEL PIC16F877

Autor:

**Iza Tipanluisa
Alex Paul**

Docente:

Ing. Amparo Meythaler

NRC: 4891

1) OBJETIVOS

- Consolidar el manejo del paquete PROTEUS para simular programas realizados con el Microcontrolador PIC16F877.
- Realizar ejercicios de programación utilizando GPIOs, leds y pulsantes.

2) MARCO TEORICO

GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General)

Es un pin genérico en un circuito integrado programable, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario el tiempo de ejecución.

Los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. La idea es que a veces, para el diseño de un sistema completo que utiliza el circuito integrado programable podría ser útil contar con un grupo de líneas digitales de control adicionales.

Un puerto GPIO es un grupo de pines GPIO (normalmente 8 pines) dispuestos en un grupo, y se tratan como un único puerto.

3) EQUIPOS Y MATERIALES

- PC con los paquetes MPLAB IDE
- PC con los paquetes PROTEUS.

4) ACTIVIDADES

1. Trabajo Preparatorio:

Realizar el diagrama de flujo, la codificación e implementación correspondiente de un programa que, para prender un led se requiere el pulso en secuencia de dos botones colocados en el Microcontrolador PIC16F877; para que se apague el led es suficiente el accionar de cualquiera de los dos pulsantes.

a) Trabajo en el paquete MPLAB IDE.

- Digite el programa. (Ponga al inicio como un comentario su nombre).
- Compile el ejercicio hasta que obtenga 0 errores.

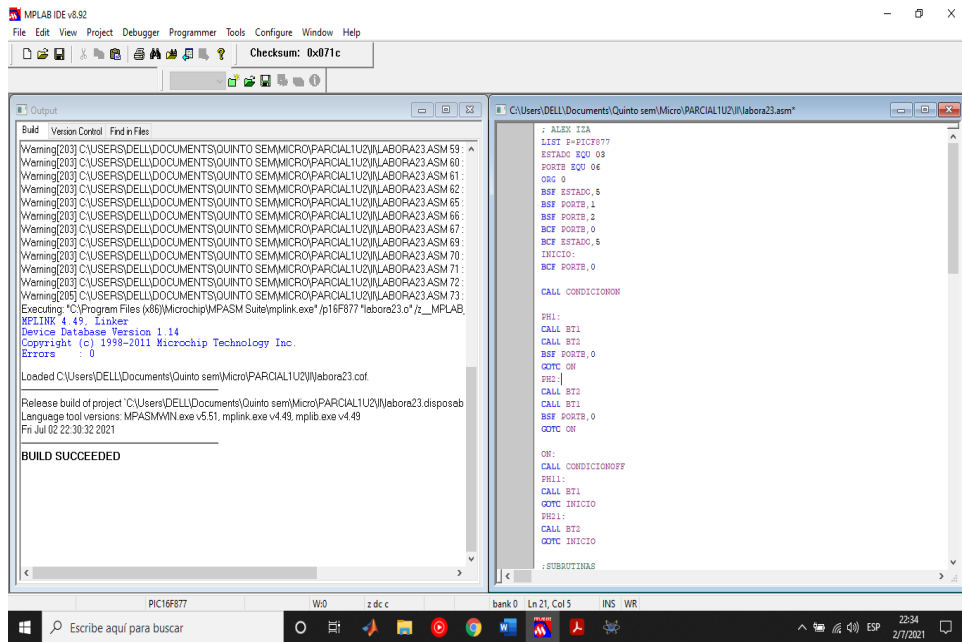


Ilustración 1

2) Trabajo en el paquete PROTEUS.

- Realice el diagrama esquemático. (Ponga en la hoja su nombre).
- Cargue el programa compilado en el microcontrolador.
- Corra el diseño y verifique el funcionamiento.

NOTA: INCLUYA 3 CAPTURAS DE LA SIMULACIÓN

Si NO funciona el ejercicio, debe revisar el programa realizado en el MPLAB, corregirlo, compilarlo y correr nuevamente en el paquete PROTEUS.

5) RESULTADOS

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

Cuando simulé tuve problemas ya que el led se encendía solo cuando el botón colocado en el PORTB,1 era presionado primero de lo contrario no; lo solucione colocando dos condiciones la primera preguntando que si el botón colocado en el PORTB,1 fue presionado encienda el led si este no se cumplía pregunto si botón colocado en el PORTB,2 fue presionado encienda el led; lo mismo paso cuando quise apagarlo, solo se apagaba con un botón con el otro se mantenía encendido, lo solucione de la misma manera preguntando si fue o no presionado uno de los botones entonces realice la acción de apagado.

- Explique cómo se corrige el problema del Rebote en los pulsadores.

Se puede corregir de dos maneras la primera por software que sería colocando como condición que la acción debe realizarse después de que se haya realizado el pulso; y la segunda sería por hardware, esto colocando un pulsador en paralelo al botón.

6) DISEÑO

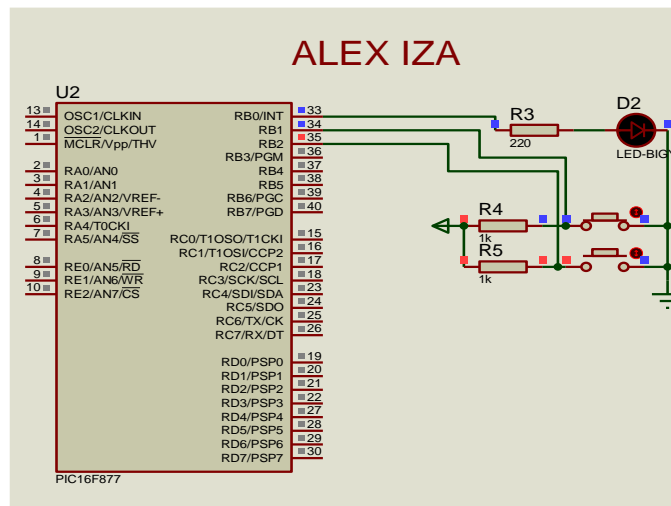


Ilustración 2

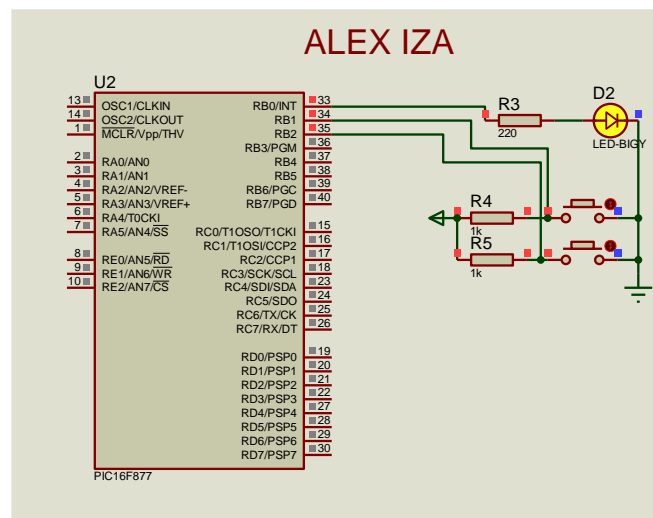


Ilustración 3

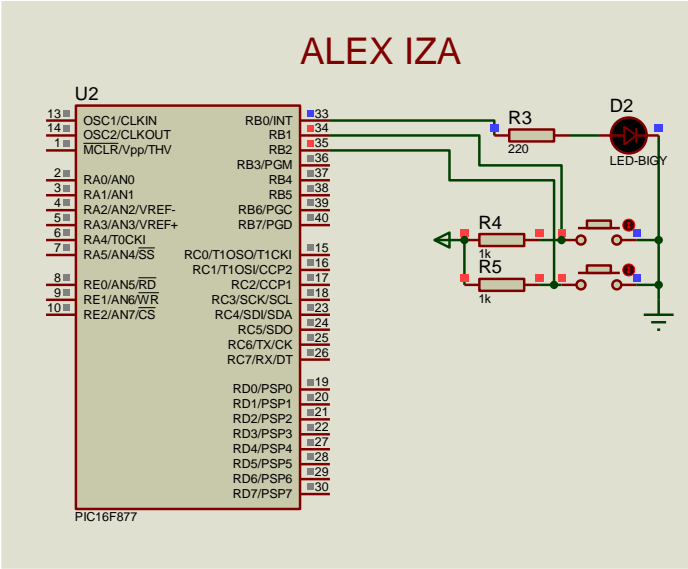


Ilustración 4

7) DIAGRAMA DE FLUJO

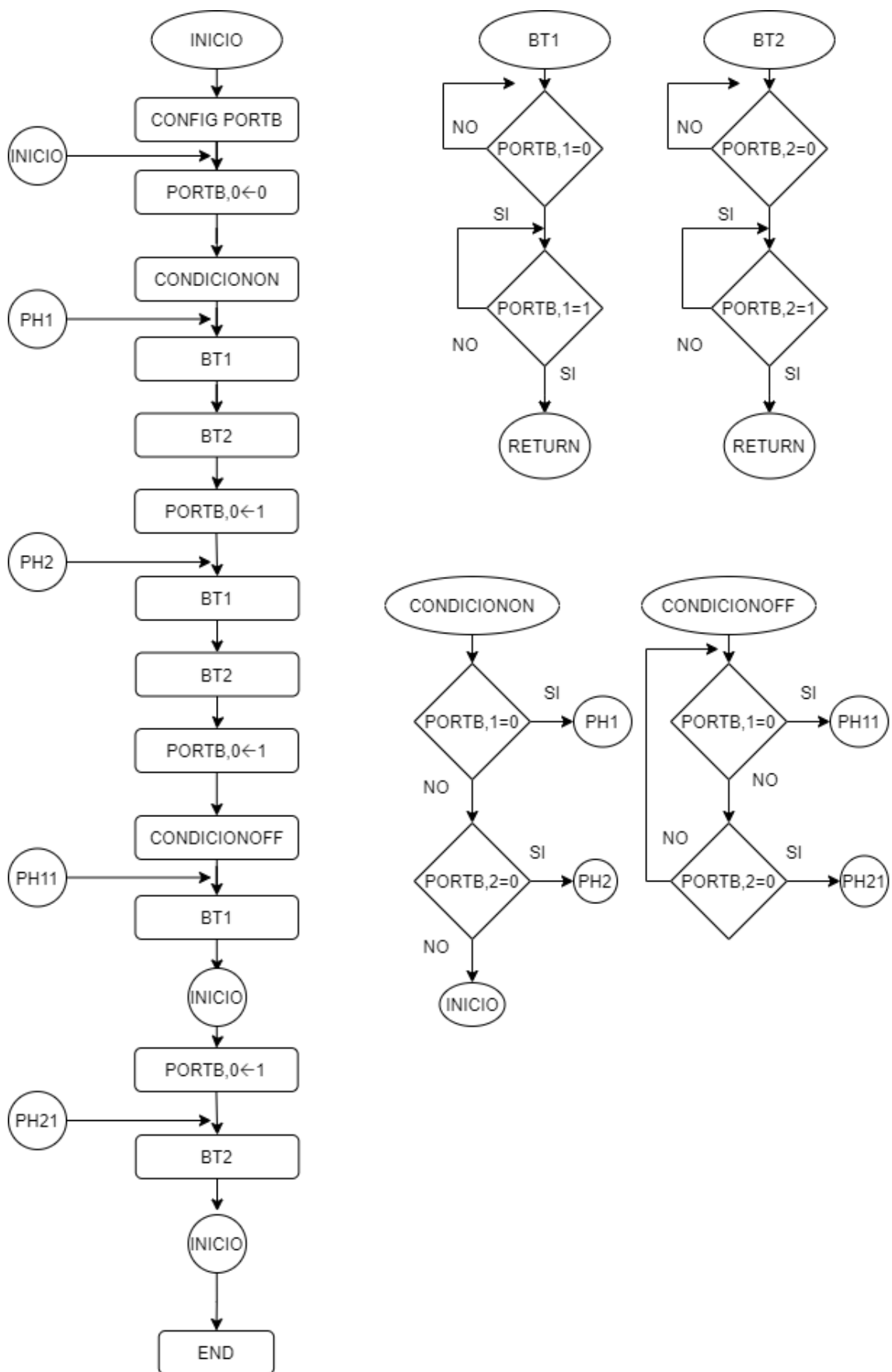


Diagrama 1

8) PROGRAMA

```
;ALEX IZA
LIST P=PICF877
ESTADO EQU 03
PORTB EQU 06
ORG 0
BSF ESTADO,5
BSF PORTB,1
BSF PORTB,2
BCF PORTB,0
BCF ESTADO,5
INICIO:
BCF PORTB,0

CALL CONDICIONON

PH1:
CALL BT1
CALL BT2
BSF PORTB,0
GOTO ON
PH2:
CALL BT2
CALL BT1
BSF PORTB,0
GOTO ON

ON:
CALL CONDICIONOFF
```

PH11:

CALL BT1

GOTO INICIO

PH21:

CALL BT2

GOTO INICIO

;SUBROUTINAS

BT1:

PREGA:

BTFSC PORTB,1

GOTO PREGA

PREGB:

BTFSS PORTB,1

GOTO PREGB

RETURN

BT2:

PREGG

BTFSC PORTB,2

GOTO PREGG

PREGH

BTFSS PORTB,2

GOTO PREGH

RETURN

CONDICIONON:

BTFSC PORTB,1

GOTO COND

GOTO PH1

COND:

BTFSC PORTB,2

GOTO INICIO

GOTO PH2

RETURN

CONDICIONOFF:

BTFSC PORTB,1

GOTO COND1

GOTO PH11

COND1:

BTFSC PORTB,2

GOTO CONDICIONOFF

GOTO PH21

RETURN

END

9) CONCLUSIONES

- Se ha concluido que la función del antirrebote es necesario usarlo cuando tenemos pulsadores ya que estos tienen el defecto de generar en nuestra señal de entrada, algo similar a un ruido.
- Se ha concluido que en la generación de un diagrama de flujo se debe ser claro y conciso para la solución del problema.
- Se ha concluido que en la entrada y la salida del Pic, se deben analizar correctamente que estado lógico estamos ingresando y que estado lógico debemos sacar para en este caso encender el led este variara su forma de trabajo dependiendo de su conexión como lógica positiva o lógica negativa

10) RECOMENDACIONES

- Se recomienda que si es posible implementar subrutinas en el programa hacerlo para que el programa ocupe menos memoria y sea más rápida.

- Se recomienda usar en las etiquetas cuyos nombres sean fáciles de diferenciar y entender cuál es el propósito en el programa.
- Se recomienda revisar y estudiar los comandos aprendidos para así evitar problemas en la compilación del programa

11) BIBLIOGRAFIA

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