**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**



**LABORATORIO DE MICROCONTROLADORES**

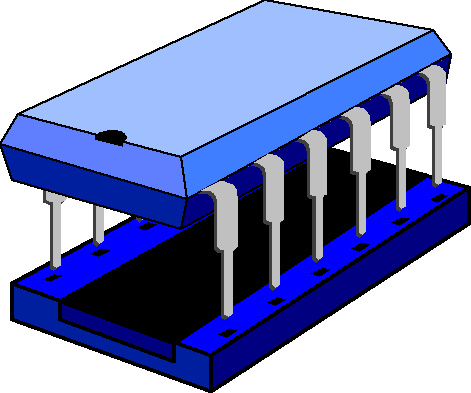
**PROYECTO # 2**

**DETECTOR DE CLAVES DE 5 DIGITOS**

**ALUMNO:**

FREDDY ANTONIO MAQUILON FALCONI

**PARALELO:**

 7

**GRUPO#3**

**PROFESOR:**

ING. [CARLOS ENRIQUE VALDIVIESO ARMENDARIZ](javascript:mailto('cvaldivi@espol.edu.ec'))

**FECHA DE ENTREGA:**

14/07/2011

**I TERMINO 2011 – 2012**

1. ***Especificaciones Técnicas del Proyecto.-***

El proyecto se trata de un detector de claves de 5 dígitos.

Los 5 primeros pines del PORTB son de ingreso de la clave o sea representan los números del 1 al 5, el sexto pin es para evaluar es decir uno ingresa los 5 digito, el sexto botón es para cerrar la caja fuerte siempre que se la haya abierto correctamente con la clave que es 12345 y está clave puede ser modificada y el ultimo pin es el reset, este botón sirve para que cuando el usuario se equivoque se resetee el sistema.

El usuario tiene 3 oportunidades de adivinar la clave, luego de eso el programa se mantiene mostrando error a través de los pines del PORTC, hasta que se presione el botón de reset.

1. ***Diagrama de Bloques.-***

**PIC**

**16F887**

**Bloque de Visualización**

**(LED’S) PORT C**

**Bloque de Ingreso de Clave**

**PORT B**

**Programación**

**Bloque de Visualización**

**(DISPLAY) PORT A**

**Bloque de Visualización**

**(DISPLAY) PORT D**

1. ***Diagrama de Flujo de Programa principal.-***

**Crear variables**

**Inicio**

**Seteo puertos**

**Guardar Clave**

**Main**

0

**INGRESO=5**

**Preguntar**

1

**Evaluar**

***Diagrama de Flujo funcional de las Subrutinas.***

**Preguntar**

0

**Presiono Tecla=1**

**Devuelvo 0**

**RETURN**

**Devuelvo valor presionado**

1

**RETURN**

**Verifica Clave**

**EVALUAR**

**Mal Clave++**

**1° Digito ok**

**Mal Clave++**

**2° Digito ok**

**3° Digito ok**

**Mal Clave++**

**4° Digito ok**

**Mal Clave++**

**4° Digito ok**

**Mal Clave++**

**ABRIR**

**CERRAR**

**RETURN**

1. ***Descripción del algoritmo o estrategia utilizada.***

Lo primero que realizamos es crear variables que utilizaremos en la programación, utilizamos

T1SAVE, T2SAVE, T3SAVE, T4SAVE, T5SAVE: Que son variables en las cuales almacenaremos la clave de la caja fuerte.

CONTADOR1, CONTADOR2, CONTADOR3, CONTADOR3: Se almacenaran valores que nos servirán para hacer los respectivos retardos.

T1, T2, T3, T4, T5: Son las variables en donde se almacenara los bits ingresados por el puerto B.

INGRESO: Este recibirá bits a través del pin 6 del puerto B y este comprobara el ingreso de la clave.

MAL: Aquí se almacenara la cantidad de veces que ingreso la clave incorrectamente y si este número es tres le pedirá al usuario que presiones reset y que lo vuelva intentar.

Al iniciar el programa se setearan todos las variables y todos los puertos y se configuraran como entrada al puerto B, y como salida al puerto A, puerto B, puerto C.

Luego se procede a ingresar los valores de la clave almacenándolos en T1SAVE, T2SAVE, T3SAVE, T4SAVE, T5SAVE, respectivamente.

Ahora en el main primero comprobamos que se haya ingresado los 5 números respectivos de la clave realizando la operación SUBWF y como W tiene almacenado 5 si se ha ingresado los 5 números esta operación nos da cero y por lo tanto Z=1, como al comienzo no se ha ingresado clave Z=0, y por lo tanto voy a la etiqueta PREG, en etiqueta se procederá a ingresar los valores a través de los pin del puerto B.

Después procedemos a llamar a TECLA en la cual se comprueba el ingreso de la clave si no se ha ingresado clave se mantiene en lazo KEY1, hasta que se proceda a ingresar cualquier número este esperara 20ms al estar ingresando el número ósea mantener la botonera presionada, al ingresar el numero este almacenara el numero ingresado en W y se procede a ir a la etiqueta CONVERSION\_7SEG este devolverá el numero respectivo que se haya ingresado y lo mostritos en un display que está conectado a los pines del PUERTO D. Lo mismo se procede a realizar en cada dato ingresado este proceso lo tenemos que hacer 5 veces para poder ingresar los 5 dígitos de la clave.

Luego de ingresar los 5 dígitos, se vuelve al main y en este se vuelve a comprobar si se ha ingresado los 5 Dígitos, como si ingresaron los 5 Dígitos se procede a ir a la etiqueta EVAL, en el cual ponemos cero al pin cero del puerto C, y preguntamos por el pin 5 del puerto B que es cuando se ha aplasto la botonera que indica ingresar clave, se mantiene en un lazo mientras no se haya puesto cero a ingresar, cuando este es cero vamos a la etiqueta EVALUAR en la cual compararemos los valores ingresados con los valores que hayan sido almacenados al comienzo en las variables T#SAVE, si toda la clave esta correcta se mostrara en el puerto C que la clave esta correcta enciendo los 4 ultimos leds, o sea que se le envía al puerto C 11110000.

Como es correcta la clave seguimos a la etiqueta CERRAR y nos mantendremos en hasta que se presione la botonera del pin 7 del puerto B que indica que se ha procedido a cerrar la caja fuerte en ese instante se borra todo lo del puerto c y se vuelve el programa al main.

Si la clave es incorrecta se procederá ir a la etiqueta MAL\_CLAVE y se encenderán los 4 últimos leds que están conectados a los pines del Puerto C, o sea que se enviara 00001111 al puerto C, luego este cuenta tres veces si es que no coinciden las claves en tres ocasiones y cada vez que va contando lo va mostrando en un display que está conectado en los pines del puerto A.

Entonces si está mal la clave ingresada y si es la primera vez que ingreso, se procede a ir a la etiqueta PIDE\_RESET y se mantendrá en esta etiqueta por medio de un lazo hasta que se ingrese cero al pin 7 del puerto B que indica que se ha ingresado reset y mientras tanto los ledas estarán encendido apagado, encendido, apagado…, o sea se envía 10101010 a los pines del puerto C, esto lo hará solo dos veces a la tercera vez ya no muestra nada en el puerto C y se procede a regresar al main lo que indica que ha intentado tres veces y se ha equivocado.

1. ***Listado del programa fuente***

; ESCUELA SUPERIOR POLITECNICA DEL LITORAL

; AUTOR: FREDDY MAQUILON FALCONI

;

;DETECTOR DE CLAVES DE 5 DIGITOS

;DIRECTIVAS

LIST p=16F887 ;Tipo de microcontrolador

INCLUDE P16F887.INC ;Define los SFRs y bits del

;P16F887

\_\_CONFIG \_CP\_OFF&\_CPD\_OFF&\_LVP\_OFF&\_WDT\_OFF&\_XT\_OSC

;Setea parámetros de configuración

errorlevel -302 ;Deshabilita mensajes de

;advertencia por cambio

;bancos

T1SAVE EQU 0X20

T2SAVE EQU 0X21

T3SAVE EQU 0X22

T4SAVE EQU 0X23

T5SAVE EQU 0X24

CONTADOR1 EQU 0X25

CONTADOR2 EQU 0X26

CONTADOR3 EQU 0X27

T1 EQU 0X28

T2 EQU 0X29

T3 EQU 0X2A

T4 EQU 0X2B

T5 EQU 0X2C

INGRESO EQU 0X2D

MAL EQU 0X2E

OK EQU 0X2F

;INICIO DEL PROGRAMA ---------------------------

ORG 0

GOTO INICIO

INICIO

BSF STATUS,5

BCF STATUS,6

BSF OSCCON,6

BSF OSCCON,5

BCF OSCCON,4

BSF OSCCON,0

MOVLW B'00000000'

MOVWF TRISC

MOVLW B'00000000'

MOVWF TRISD

MOVLW B'00000000'

MOVWF TRISA

MOVLW B'000'

MOVWF TRISE

MOVLW B'00000000'

MOVWF TRISB

BSF STATUS,6

CLRF ANSEL

CLRF ANSELH

BCF STATUS,6

BCF STATUS,5

CLRF PORTA

CLRF PORTC

CLRF PORTD

CLRF PORTB

CLRF PORTE

CLRF T1

CLRF T2

CLRF T3

CLRF T4

CLRF T5

;DECLARO LOS VALORES DE LA CLAVE

MOVLW d'1'

MOVWF T1SAVE

MOVLW d'2'

MOVWF T2SAVE

MOVLW d'3'

MOVWF T3SAVE

MOVLW d'4'

MOVWF T4SAVE

MOVLW d'5'

MOVWF T5SAVE

GOTO MAIN

MAIN

CLRF PORTB

MOVLW d'5'

SUBWF INGRESO,W

BTFSS STATUS,Z

GOTO PREG

GOTO EVAL

PREG

BSF PORTC,0

KEY1

CALL TECLA

MOVWF T1

MOVLW 0X00

SUBWF T1,W

BTFSC STATUS,Z

GOTO KEY1

MOVFW T1

CALL CONVERSION\_7SEG

MOVWF PORTD

CALL RETARDO\_1S

CLRF PORTD

INCF INGRESO

GOTO KEY2

KEY2

CALL TECLA

MOVWF T2

MOVLW 0X00

SUBWF T2,W

BTFSC STATUS,Z

GOTO KEY2

MOVFW T2

CALL CONVERSION\_7SEG

MOVWF PORTD

CALL RETARDO\_1S

CLRF PORTD

INCF INGRESO

CALL CONVERSION\_7SEG

GOTO KEY3

KEY3

CALL TECLA

MOVWF T3

MOVLW 0X00

SUBWF T3,W

BTFSC STATUS,Z

GOTO KEY3

MOVFW T3

CALL CONVERSION\_7SEG

MOVWF PORTD

CALL RETARDO\_1S

CLRF PORTD

INCF INGRESO

GOTO KEY4

KEY4

CALL TECLA

MOVWF T4

MOVLW 0X00

SUBWF T4,W

BTFSC STATUS,Z

GOTO KEY4

MOVFW T4

CALL CONVERSION\_7SEG

MOVWF PORTD

CALL RETARDO\_1S

CLRF PORTD

INCF INGRESO

GOTO KEY5

KEY5

CALL TECLA

MOVWF T5

MOVLW 0X00

SUBWF T5,W

BTFSC STATUS,Z

GOTO KEY5

MOVFW T5

CALL CONVERSION\_7SEG

MOVWF PORTD

CALL RETARDO\_1S

CLRF PORTD

INCF INGRESO

GOTO MAIN

TECLA

BTFSS PORTB,0

GOTO TESTB1

BTFSS PORTB,1

GOTO TESTB2

BTFSS PORTB,2

GOTO TESTB3

BTFSS PORTB,3

GOTO TESTB4

BTFSS PORTB,4

GOTO TESTB5

RETLW d'0'

TESTB1

CALL PAUSA\_20MS

BTFSS PORTB,0

RETLW d'1'

GOTO TESTB1

TESTB2

CALL PAUSA\_20MS

BTFSC PORTB,1

RETLW d'2'

GOTO TESTB2

TESTB3

CALL PAUSA\_20MS

BTFSC PORTB,2

RETLW d'3'

GOTO TESTB3

TESTB4

CALL PAUSA\_20MS

BTFSC PORTB,3

RETLW d'4'

GOTO TESTB4

TESTB5

CALL PAUSA\_20MS

BTFSC PORTB,4

RETLW d'5'

GOTO TESTB5

EVAL

BCF PORTC,0

CERRO

BTFSS PORTB,5

GOTO EVALUAR

GOTO CERRO

EVALUAR

CLRF INGRESO

MOVFW T1

SUBWF T1SAVE,W

BTFSS STATUS,Z

GOTO MAL\_CLAVE

MOVFW T2

SUBWF T2SAVE,W

BTFSS STATUS,Z

GOTO MAL\_CLAVE

MOVFW T3

SUBWF T3SAVE,W

BTFSS STATUS,Z

GOTO MAL\_CLAVE

MOVFW T4

SUBWF T4SAVE,W

BTFSS STATUS,Z

GOTO MAL\_CLAVE

MOVFW T5

SUBWF T5SAVE,W

BTFSS STATUS,Z

GOTO MAL\_CLAVE

GOTO CORRECTO

MAL\_CLAVE

MOVLW B'00001111'

MOVWF PORTC

INCF MAL

MOVFW MAL

CALL CONVERSION\_7SEG

MOVWF PORTA

MOVLW d'3'

SUBWF MAL,W

BTFSC STATUS,Z

GOTO PIDE\_RESET

CALL RETARDO\_1S

CLRF PORTC

GOTO MAIN

CORRECTO

CLRF PORTA

CLRF MAL

MOVLW B'11110000'

MOVWF PORTC

CIERRA

BTFSC PORTB,6

GOTO CIERRA

CLRF PORTC

CLRF OK

GOTO MAIN

PIDE\_RESET

MOVLW B'10101010'

MOVWF PORTC

BTFSC PORTB,7

GOTO PIDE\_RESET

CLRF PORTC

CLRF PORTD

CLRF MAL

CLRF PORTA

GOTO MAIN

CONVERSION\_7SEG

ADDWF PCL,F ; PCL + W -> W

; El PCL se incrementa con el

; valor de W proporcionando un

; salto

RETLW 0x3F ; Retorna con el código del 0

RETLW 0x06 ; Retorna con el código del 1

RETLW 0x5B ; Retorna con el código del 2

RETLW 0x4F ; Retorna con el código del 3

RETLW 0x66 ; Retorna con el código del 4

RETLW 0x6D ; Retorna con el código del 5

RETLW 0x7D ; Retorna con el código del 6

RETLW 0x07 ; Retorna con el código del 7

RETLW 0x7F ; Retorna con el código del 8

RETLW 0x67 ; Retorna con el código del 9

PAUSA\_20MS

MOVLW 0X0

MOVWF CONTADOR1

MOVLW 0X1A

MOVWF CONTADOR2

RETARDO1

DECFSZ CONTADOR1,F

GOTO RETARDO1

DECFSZ CONTADOR2,F

GOTO RETARDO1

RETURN

RETARDO\_1S

MOVLW 0X0

MOVWF CONTADOR1

MOVLW 0X0

MOVWF CONTADOR2

MOVLW 0X05

MOVWF CONTADOR3

RETARDO2

DECFSZ CONTADOR1,F

GOTO RETARDO2

DECFSZ CONTADOR2,F

GOTO RETARDO2

DECFSZ CONTADOR3,F

GOTO RETARDO2

RETURN

END

1. ***Circuito en Proteus***

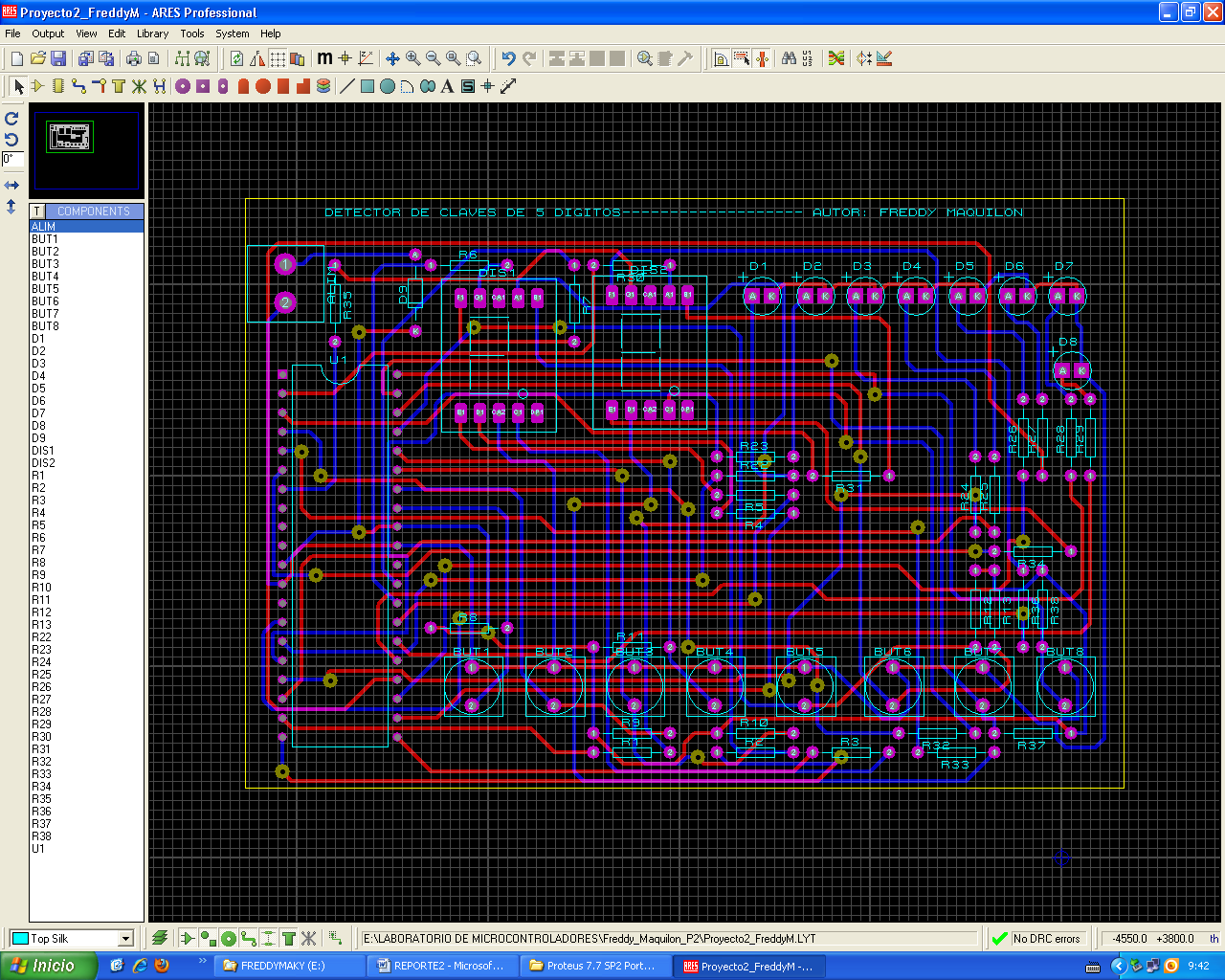
******

Simulacion de Proyecto Detector de Digitos

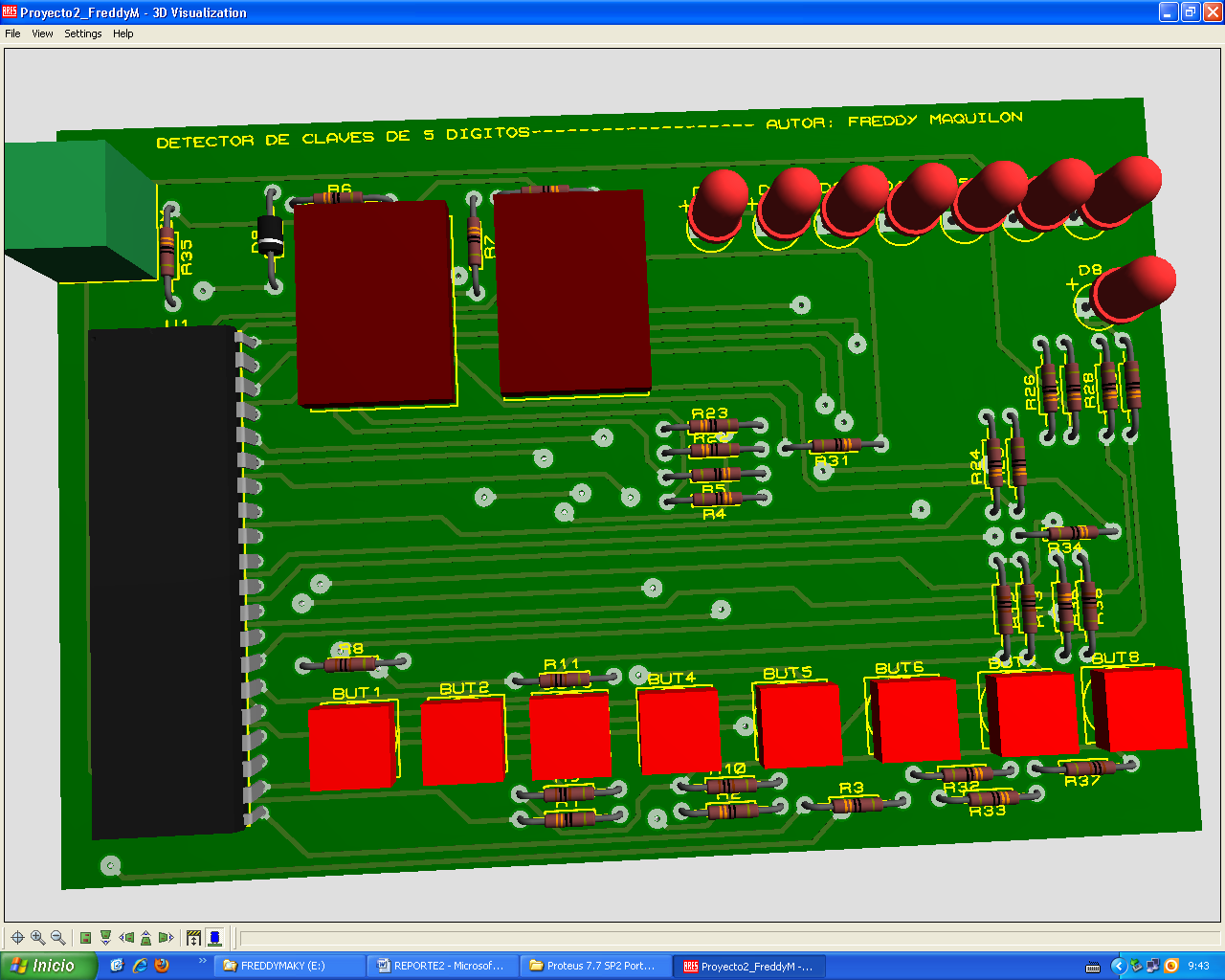
******

Simulacion de Proyecto: Simulacion Funcionando lista para el ingreso de clave

1. ***Circuito en ARES***

******

Placa del Circuito en ARES

******

Placa del Circuito en ARES en 3D

1. ***Conclusiones***

Al realizar este proyecto se aprendieron muchas cosas sobre el PIC16F877A, y pudimos aprovechar las funciones que tiene el mismo, como notamos en este proyecto la velocidad con que realiza la operación matemática y lo muestra en los leds en la simulación es inmediata, también mostramos los números respectivos en el display de inmediatamente para nosotros pero en realidad en el programa hay un retardo de 20ms también al observar el programa nos damos cuenta que realiza algunas operaciones básicas para poder sumar y mostrar el resultado.

Se pudo comprobar que el encendido de los displays en los microcnotraldores es tan fácil solo se tiene que enviar el numero en hexadecimal a la puerta especificada y este mostrara la información en binario a sus pines para que el display reciba la señal de altos y cero para poder activar el numero especificado

También hemos utilizado las operaciones aprendidas en clases y pudimos comprender el funcionamiento que nos fueron útil en el desarrollo de este proyecto, estas funciones que son muy básicas nos permitieron sumar y mover valores de una variable a otra, es por eso que podemos decir que un programa que aparentemente es complicado se lo puede realizar en el microcontrolador con operaciones básicas.

Para ser la primera vez que uso un pic, ha sido un trabajo duro en el que he dedicado mucho esfuerzo y por encima de los resultados me ha servido para irme afianzando en lo que se refiere a implementación de circuitos controlados por pic’s. Aunque solo es simulada sin duda alguna es un gran paso el cual me servirá para mi desarrollo como profesional.

1. ***Recomendaciones***

Es muy importante mencionar que los datos ingresados no solamente se pueden realizar la comparación haciendo la resta y comprobando el estado del bit Z, si no como hemos aprendido en clases tenemos muchas más funciones para poder realizar la comparación.

Al momento de trabajar con un nuevo pic debemos asegurarnos de conocer los registros correctos para habilitar los puertos como entradas o salidas digitales ya que ese fue un inconveniente en el presente trabajo, al mismo tiempo que debemos deshabilitar el wathdog ya que este se desbordaba continuamente.