|  |  |
| --- | --- |
|  | **TECNOLÓGICO NACIONAL DE MÉXICO**  **TECNOLÓGICO DE ESTUDIOS SUPERIORES DE JILOTEPEC** |

**DIVISIÓN DE INGENIERÍA EN SISTEMA COMPUTACIONALES**

**ARQUITECTURA DE COMPUTADORAS**

**MANUAL SEMAFORO**

**IMANOL GONZÁLEZ HERNÁNDEZ**

**GRUPO 3101**

**LIC. JUAN ALBERTO MARTINEZ ZAMORA**

**JILOTEPEC, MÉXICO 17 DE ENERO DEL 2020**

índice

[Simulador (proteus) 3](#_Toc29637302)

[Código (MPLAB) 7](#_Toc29637303)

[Entablación de la simulación en físico. 16](#_Toc29637304)

# Simulador (proteus)

Esta es la simulación del semáforo en el programa de proteus, cada 5 leds son, los 2 primeros son los rojo y verde del paso peatonal, los 3 de la derecha son los de los automóviles, que son los leds color, verde, rojo, y ámbar, cada led esta conectado a tierra y el lado positivo a una resistencia de 375 Homs, todos los leds están conectados a tierra por un solo cable, un circuito que no es necesario pero aun así lo pusimos es el de la izquierda que contiene una conexión a el PIC y una a tierra, dentro de el se conectan una resistencia de 33k, una de 40k y una de 10k, un transistor y un objeto que en proteus se ubica como 1N4728A, este circuito es para que se haga un ciclo infinito de repeticiones del semáforo, también se coloca un circuito de un cristal oscilador de 4000 MHz, dos capacitores cerámicos de 22 p que irán conectados a positivo.

Las especificaciones para conectar son las siguientes:

* Se hacen los 4 semáforos pero solo se van a conectar 2 a el PIC.
* Realmente si se conectan a positivo los circuitos y a tierra los leds o viceversa funcionara el circuito.

Leds semáforo uno:

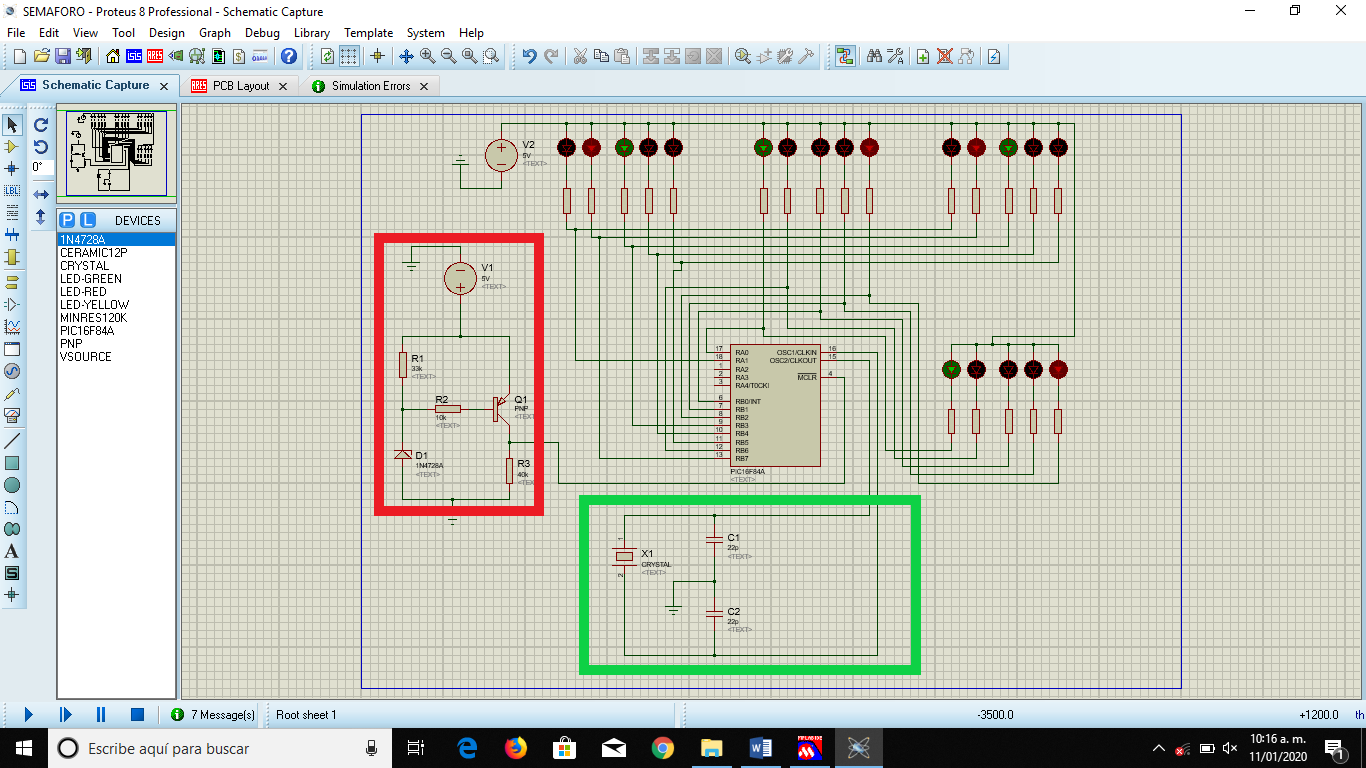
* Rojo para peatones: RB7
* Verde para peatones: RA1
* Verde para autos: RB3
* Rojo para autos:RB5
* Ámbar para autos:RB4

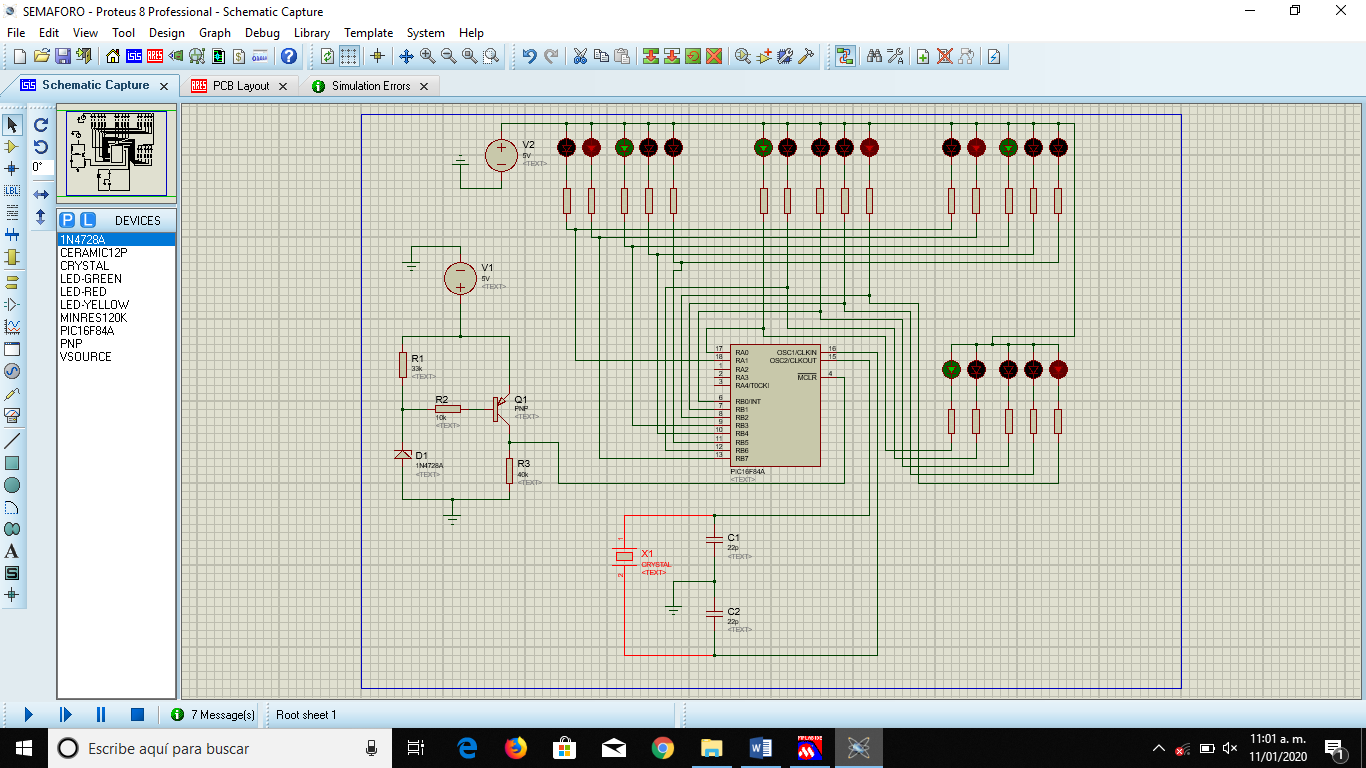
Leds semáforo dos:

* Verde para peatones: RA0
* Rojo para peatones: RB6
* Verde para autos:RB0
* Rojo para autos:RB2:
* Ámbar para autos: RB1

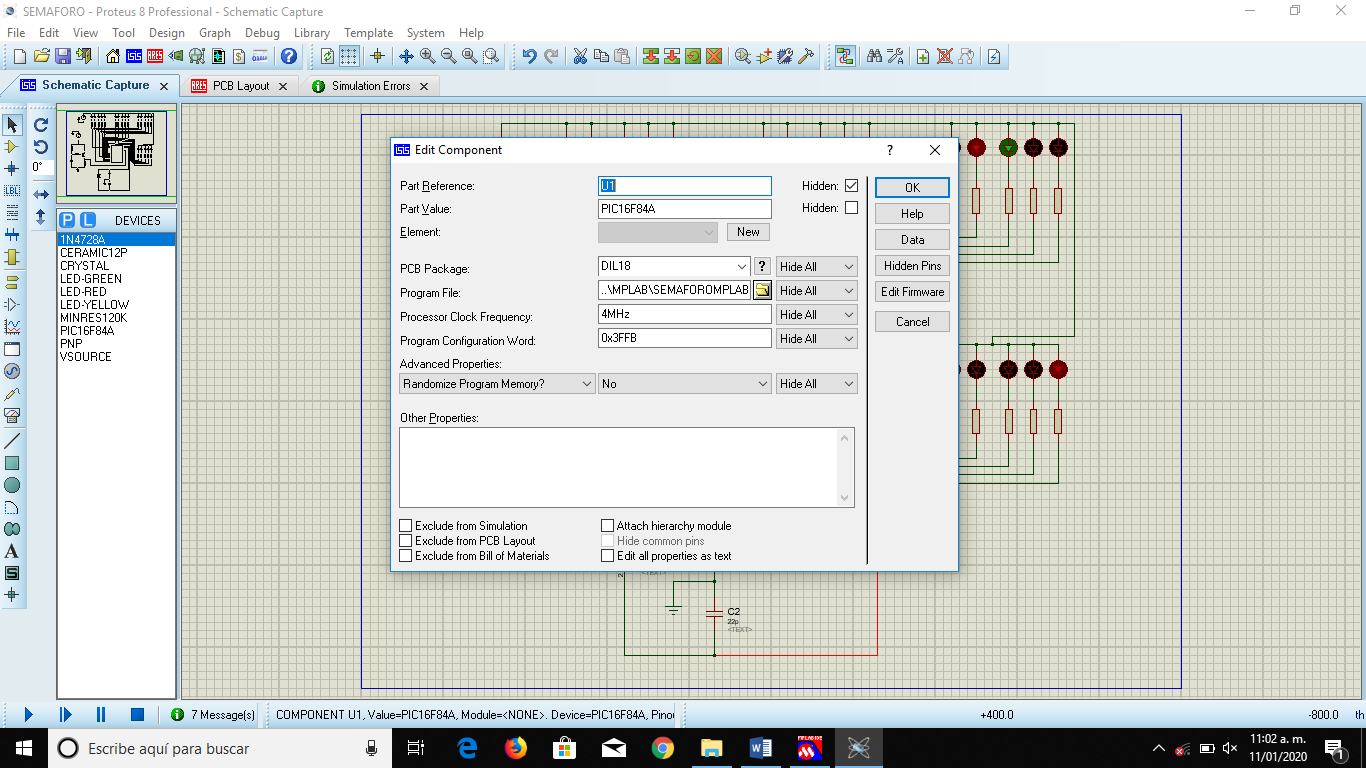
Circuitos:

* El de la izquierda o el marcado de rojo se conecta a la salida MVLR que es la (4) y a tierra.
* El de abajo o el de verde, se conecta a la (15 y 16), y a tierra

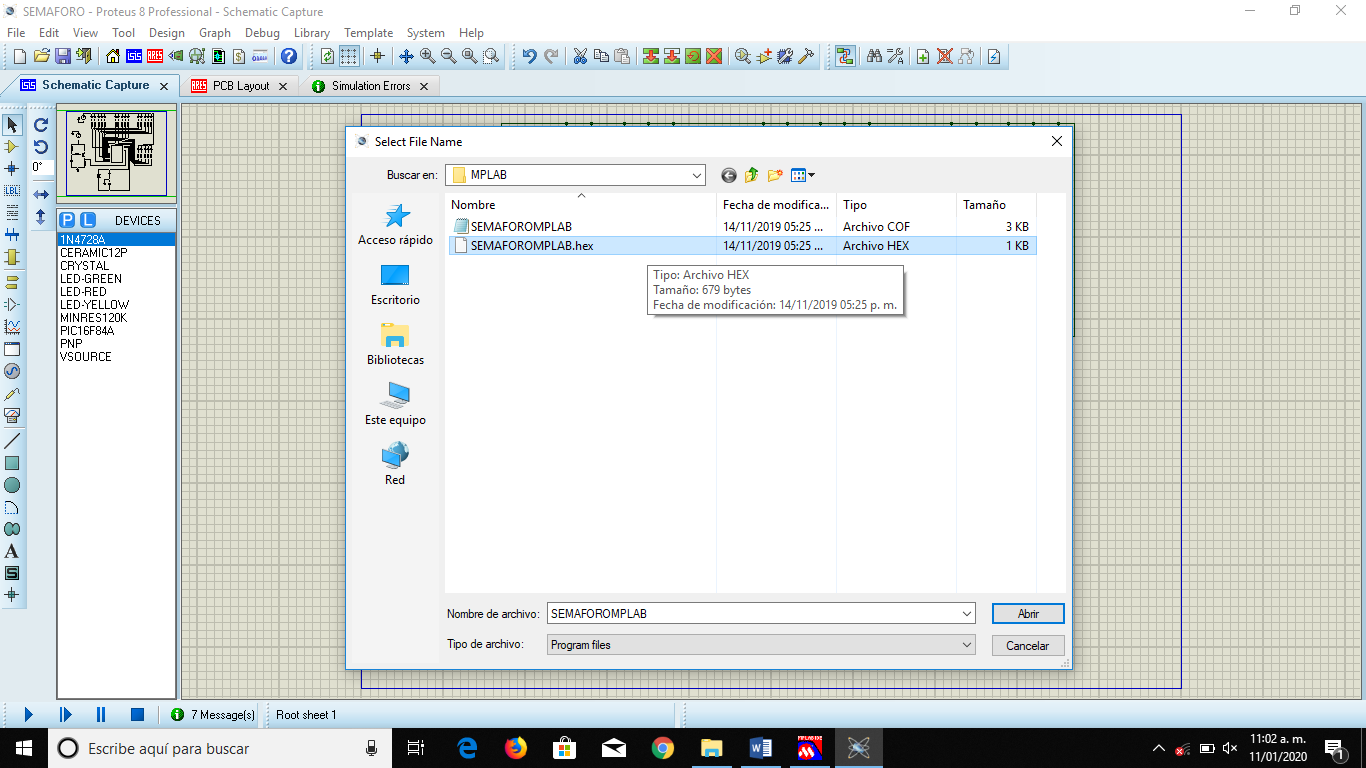




Para ejecutar la simulación solo le damos doble clic en el PIC

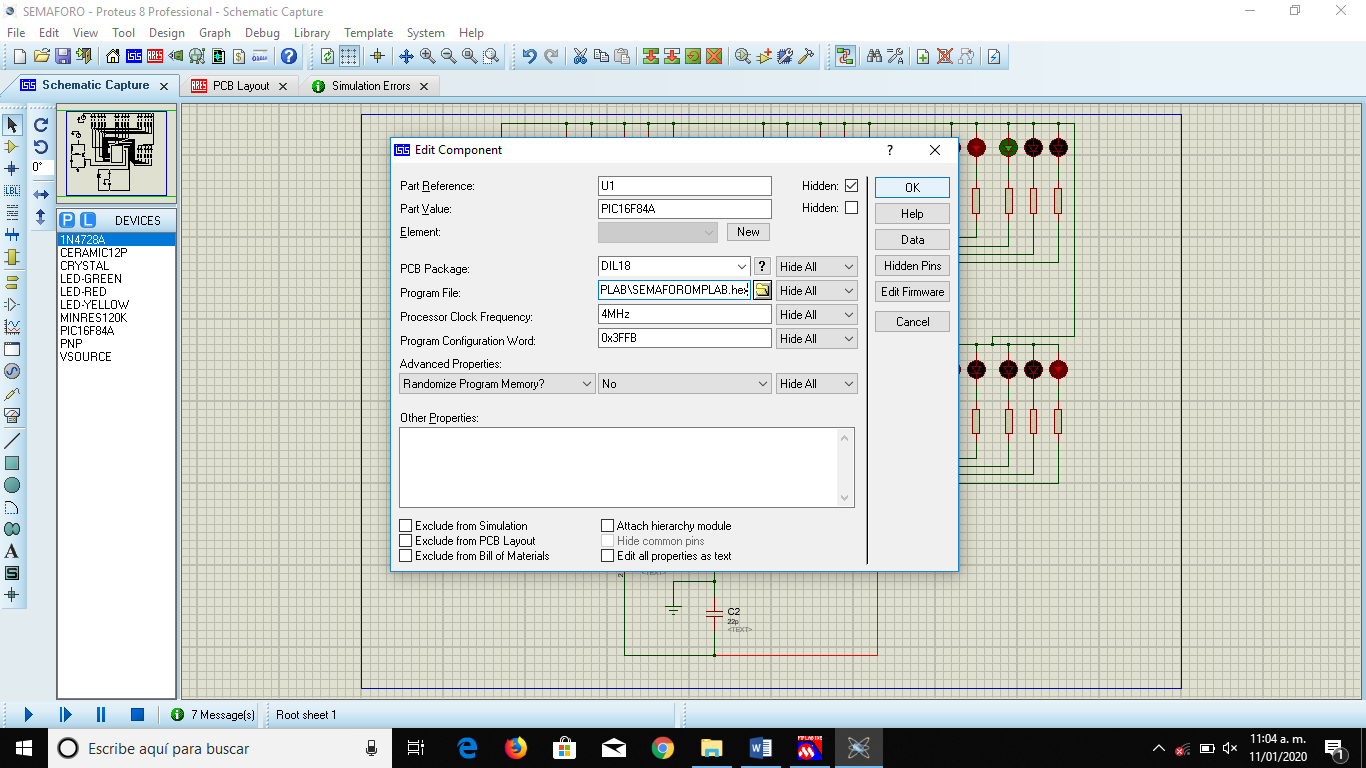


Le damos clic en el icono de el folder delante de program file.

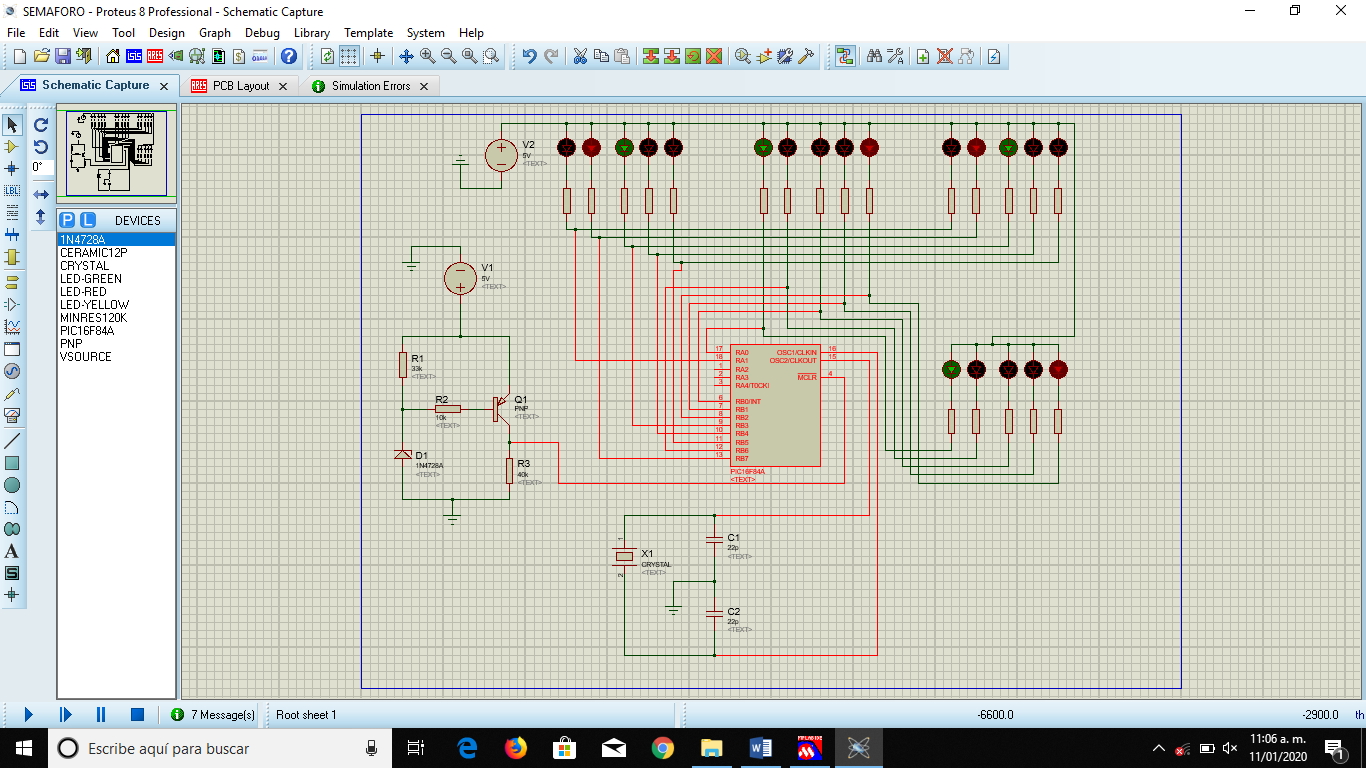


Y seleccionamos el archivo con terminal (.hex), que hayamos hecho desde un editor de código ensamblador.

Le damos clic en el archivo y después en abrir.



Después le damos clic en “OK”.



Le damos clic en correr para ejecutar la simulación, en el icono de play.

Si nos marca errores en la parte de abajo se aparecerán.

# Código (MPLAB)

\_\_CONFIG \_CP\_OFF & \_PWRTE\_OFF & \_WDT\_ON & \_XT\_OSC

;la primer linea sirve para que se pueda reescribir el codigo despues en un PIC

LIST P=P16F84A, R=DEC

#INCLUDE <P16F84A.INC> ;BIBLIOTECA

ORG 0

GOTO PRINCIPAL ;SALTO AL PROGRAMA PRINCIPAL AL RESETEAR EL PIC

ORG 4

GOTO IRQ ;SALTO A LA RUTINA DE INTERRUPCIÓN

;DEFINIR CONSTATNTE

CUENTA EQU 16 ;VALOR DE CUENTA DE TMR0(256-240)

RP2 EQU 7 ;BITS DE LOS SEMAFOROS DE LOS PEATONES

RP1 EQU 6

VP2 EQU 1

VP1 EQU 0

;DECLARACIÓN DE LAS VARIABLES Y LAS POCISIONES DE MEMORIA QUE USARAN

CONTADOR EQU 0X10

RETARDO EQU 0X11

;PROGRAMA PRICNIPAL

;

;INCIALIZAR EL PIC16F84A, REGISTROS DE CONTROL, PUERTOS E INTERRUPCIONES

;EL PROGRAMA LANZARA EL TEMPORIZADOR PARA QUE SE ACTIVADA LA

;Y SE QUEDARA EN ESPERA DE UN CICLO INFINITO

;

PRINCIPAL:

;SE INICIARA EL PIC Y PROGRAMAR EL REGISTRO OPTION

;BIT7 RPBU=1 RESISTENCIAS DEL PULL-UP INTERNAS DEL PORTB DESABILITADAS

;LAS CUALES NO SERA NECESARIO YA QUE SE DESABILITAN AL USAR EL PORTB COMO

;SALIDA.

;BIT6 INTEDG=0 DA LO MISMO PORQUE NO SE USARA INT

;BIT5 TOCS=0 ES EL TEMPORIZADOR DEL RELOJ INTERNO

;BIT5 TOSE=0 DA IGUAL

;BIT3 PSA=0 Y PRE-ESCALAR EL TMR0

;BIT2-0= 111 AJUSTA E L PRE-ESCALAR DEL TMR0 A 256

MOVLW B'10000111' ;ES EL VALOR DE LA PALBRA PARA OPTION SE MODIFICA LOS BITS 2-0

BSF STATUS,RP0 ;ES EL BANCO DE MEMORIA 1

MOVWF OPTION\_REG ;ESCRIBE LA PALABRA EN EL REGISTRO OPTION

BCF STATUS,RP0 ;ES EL BANCO 0

;PROGRAMACION DE LOS PUERTOS

;EL PORTB=SALIDAS=>TOD A 0

;ESTRUCTURA DEL PORTB: 7 6 5 4 3 2 1 0

; RP2 RP1 R2 A2 V2 R1 A1 V1

; \_\_^\_\_ \_\_^\_\_

; S2 S1

BSF STATUS,RP0 ;BANCO 1

CLRF TRISB ;BORRA TODOS LOS BITS DEL PORTB

CLRF TRISA ;HACE LO MISMO QUE ARRIBA PARA TRISA

BCF STATUS,RP0 ;BANCO 0

;INICILISAR EL CONTADOR Y EL 'PUERTO

CLRF CONTADOR ;PONER EL CONTADOR EN 0

MOVLW B'10011110' ;S2 ROJO, S1 VERDE. PEATONES, ROJO PARA EL S1

MOVWF PORTB ;LO SACA POR EL PUERTO

BCF PORTA, VP2 ;PEATONES, VERDE PARA EL S2

BSF PORTA, VP1 ;PEATONES,VERDE PARA EL S1

;PROGRAMAR EL TEMPORIZADOR

MOVLW CUENTA

MOVWF TMR0 ;LANZA LA CUENTA INICIAL DEL TEMPORAZADOR

;PROGRAMACION DEL INTERRUPCIONES

;BIT7 GIE=1 PARA HABILITAR LAS INTERRUPCIONES

;BIT5 TOIE=1 PARA LA HABILITACION DEL TEMPORIZADOR

;RESTO A 0 PARA DESABILITAR LAS DEMAS INTERRUPCIONES Y BORRAR BANDERAS

MOVLW B'10100000'

MOVWF INTCON ;PROGRAMA LAS INTERRUPCIONES

ESPERA: ;BUCLE INFINITO QUE BORRA CONTINUMANETE EL PERRO GUARDIAN

CLRWDT ;EL TRANAJO LO HACE LA IRQ.SOLO SI HAY ALGUN PROBLEMA, SE

GOTO ESPERA ;REINIARA EL MICRO

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;RUTINA DE ATENCION A LA INTERRUPCION DEL TEMPORIZADOR

;

;LA INTERRUPCION SE PRODUCIRA CADA 0'05S.

;CONTAREMOS 5 LLAMADAS

;SE INCREMENTARA UN CONTADOR DE SEGUNDOS PARA CONTROLAR EL TEMPORIZADOR DEL SEMAFORO, Y

;SE MODIFICA EL ESTADO DE LOS SEMAFOROS, ESCRIBIENDO EN EL PUERTO SEGU EL TIEMPO PASADO

;

;NO SE COMPRUEBA CUAL ES LA FUENTE DE INTERRUPCION, PORQUE SOLO ESTA HABILITADA

;LA INTERRUPCION DEL TEMPORIZADOR.

IRQ:

;EMPEZAMOS PROGRAMANDO EL TEMPORIZADOR, PARA QUE VAYA CONTANDO YA, Y NO INTRODUCIR

;UN RETARDO EN LA TEPORIZACION DEBIDO AL PROCESO DE VARIAS LINEAS EN EL CODIGO

MOVLW CUENTA

MOVWF TMR0 ;REPROGRAMACION DEL TIMER

INCF RETARDO

MOVLW 5

SUBWF RETARDO, W ;COMPARA RETARDO CON 5

BTFSS STATUS, Z ;SI NO ES 5

GOTO FINSWITCH ;SALTA AL FINAL DE RUTINA DE INTERRUPCION

CLRF RETARDO

;ESTE CODIGO SE EJECUTA CADA 0'5S. EL CONTADOR CUENTA MEDIOS SEGUNDOS

INCF CONTADOR, F ;CONTADOR=CONTADOR+

;IMPLEMENTACION DE LA SENTENCIA SWITCH-CASE

CASO20: ;CONTADOR<20

MOVLW 40

SUBWF CONTADOR, W ;COMPARA CONTADOR CON 40

BTFSC STATUS, C ;¿ACARREO?

GOTO FINCASO20 ;HO HAY ACARREO0=>CONTADOR=>40

;CONTADOR<40 (NO HAY ACARREO)

MOVLW B'10011110' ;COCHES ;S2 ROJO, SI VERD,PEATONES RP2 OFF,RP1 ON

MOVWF PORTB ;LO ESCRIBE EN EL PUERTO

BCF PORTA, VP2 ;PEATONES, VERDE PARA S2

BSF PORTA, VP1 ;PEATONES, VERDE PARA S1

GOTO FINSWITCH

FINCASO20:

CASO25: ;CONTADOR<25

MOVLW 50

SUBWF CONTADOR, W ;COMPARA SI EL CONTADOR ES 50

BTFSC STATUS, C ;OYE GUTIERREZ HAY ACARREO (CONDICION)

GOTO FINCASO25 ;NO HAY ACARREO =>CONTADOR>=50

;CONTADOR < QUE 25 (HO HAY ACARREO)

MOVLW B'10011110' ;AUTOS RS2, VS1. RP2 OFF, RP1 ON

MOVWF PORTB ;LO MANDAMOS AL PUERTO DE SALIDA

BTFSC CONTADOR, 0 ;SI CONTADOR ES PAR ENTONCES ES = A 1 SEGUNDO

GOTO FINSWITCH

MOVLW B'10' ;MASCARA XOR PARA CONMUTAR VP2 SIN MODIFICAR EL RESTO (1 SEGUNDO)

XORWF PORTA, F ;CONMUTA EL VP2 (CADA SEGUNDO)

GOTO FINSWITCH

FINCASO25:

CASO30: ;25>=CONTADOR<30

MOVLW 60

SUBWF CONTADOR, W ;COMPARA EL CONTADOR CON 60

BTFSC STATUS, C ;OYE GUTIERREZ ES MAYOR QUE 60???

GOTO FINCASO30 ;NO HAY ACARREO => POR LO TANTO EL CONTADOR>=60

;CONTADOR < 30 (NO HAY ACARREO)

MOVLW B'00000010' ;MASCARA DEL XOR PARA CONMUTAR AS1 CADA 0.5S ENTRE 25 Y 30 SEGUNDOS

XORWF PORTB, F

BSF PORTB, 0 ;APAGA EL VS1

BTFSC CONTADOR, 0 ;SI EL CONTADOR ES PARA ENTONCES

GOTO FINSWITCH

MOVLW B'10' ;MASCARA PARA EL VP2

XORWF PORTA, F ;CONMUTA VP2

GOTO FINSWITCH

FINCASO30:

CASO50: ;30>=CONTADOR<50 ENTONCES

MOVLW 100

SUBWF CONTADOR, W ;COMPARA EL VALOR DEL CONTADOR CON 100

BTFSC STATUS, C ;¿HAY ACARREO?

GOTO FINCASO50 ;HO HAY ACARREO => CONTADOR >=100

;CONTADOR<50 CUANDO NO HAY ACARREO

MOVLW B'01110011' ;VS2, RS1, RP2 ON, RP1 OFF.

MOVWF PORTB ;LO ENVIAMOS AL PUERTO B

BSF PORTA, VP2 ;VP1

BCF PORTA, VP1 ;VP1

GOTO FINSWITCH

FINCASO50:

CASO55: ;30>=CONTADOR<55

MOVLW 110

SUBWF CONTADOR, W ;COMPARA EL CONTADOR CON 110

BTFSC STATUS, C ;¿HAY ACARREO?

GOTO FINCASO55

;CONTADOR <55 HO HAY ACARREO

MOVLW B'01110011' ;VS2,RS1,RP2 ON, RP1 OFF

MOVWF PORTB ;LO ESCRIBE EN EL PUERTO

BTFSS CONTADOR, 0 ;SI EL CONTADOR ES PAR => 1 SEGUNDO

GOTO FINSWITCH

MOVLW B'01' ;ENMASCARA XOR PARA CONMUTAR VP1 SI MODIFICAR EL

XORWF PORTA, F ;CONMUTA VP1 CADA 2 UNIDADES 1 SEGUNDO

GOTO FINSWITCH

FINCASO55:

CASO60: ;55<=CONTADOR>60

MOVLW 120

SUBWF CONTADOR, W ;COMPARA CONTADOR CON 120

BTFSC STATUS, C ;¿HAY ACARREO?

GOTO FINCASO60 ;HO HAY ACARREO => CONTADOR >=120

;CONTADOR < 60 NO HAY ACARREO

MOVLW B'00010000' ;ENMASCARA XOR PARA CONMUTAR AS2 SIN MODIFICAR EL RESTO

BTFSS CONTADOR, 0 ;SI EL CONTADOR ES PAR CADA SEGUNDO

BSF W, VP1 ;ACTIVA LA MASCARA TAMBIEN PARA VP1

XORWF PORTB, F ;CONMUTA EN AS2, CADA 0.5 SEGUNDOS DURANTE EL TIEMPO ENTRE 55 Y 60

BSF PORTB, 3 ;APAGA VS2

BTFSS CONTADOR, 0 ;SI EL CONTADOR ES PARA CADA SEGUNDO

GOTO FINSWITCH

MOVLW B'01' ;MASCARA PARA VP1

XORWF PORTA, F ;CONMUTA VP1

GOTO FINSWITCH

FINCASO60:

OTHERWISE: ;LLEGO A 60

MOVLW B'10011110' ;AUTOS RS2,VS1, RP2 OFF, RP1 ON.

MOVWF PORTB ;LO ENVIAMOS AL PUERTO

BCF PORTA, VP2 ;VP2

BSF PORTA, VP1 ;VP1

CLRF CONTADOR ;EMPIEZA TODO DE NUEVO

FINSWITCH: ;FIN DE TODA LA SENTENCIA SWTICH-CASE

MOVLW B'10100000'

MOVWF INTCON

RETFIE ;RETORNO DE LA INTERRUPCION

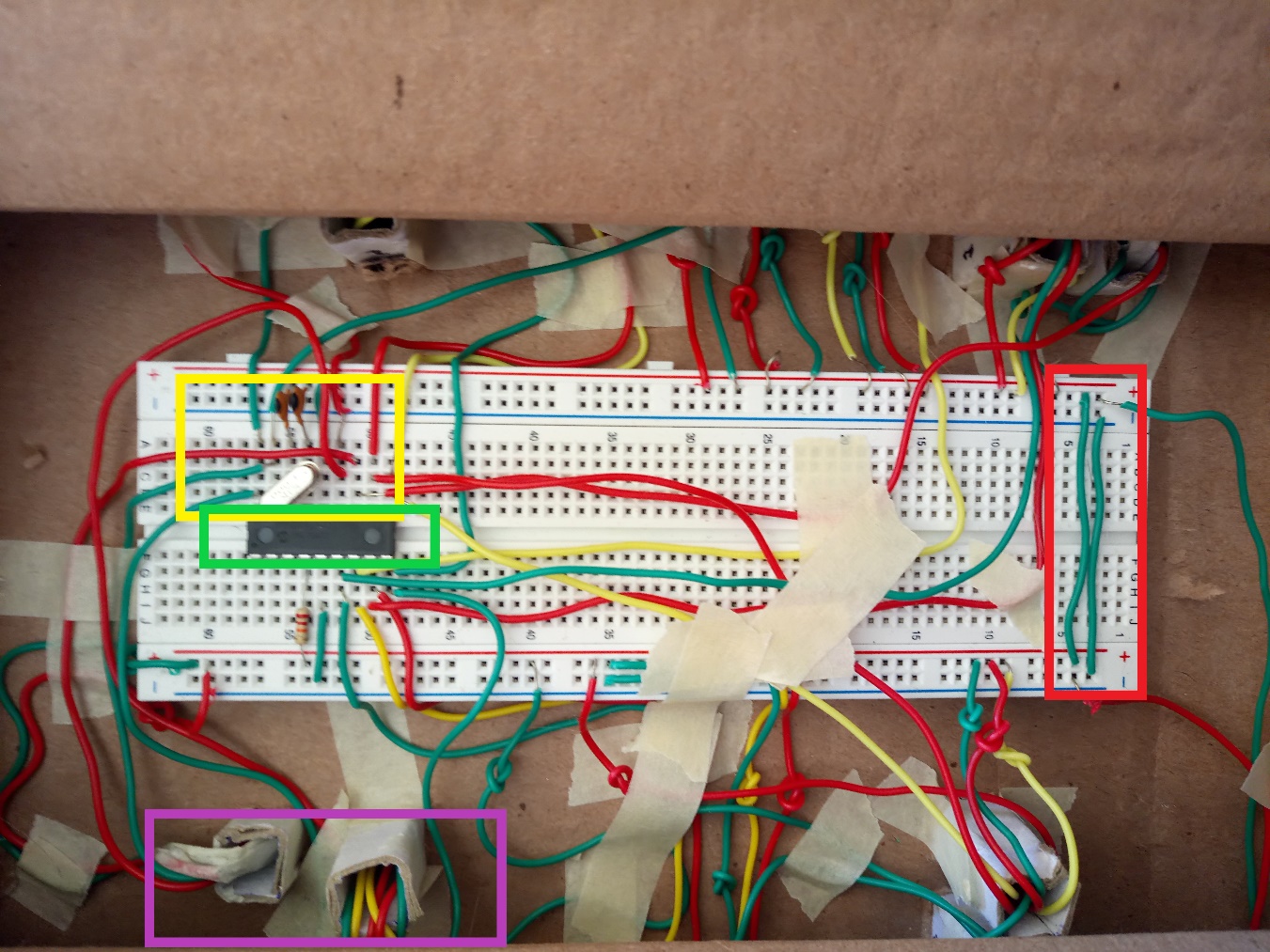
END

Ese es el código en lenguaje ensamblador, el cual hace que las instrucciones de prender los leds de cada salida esten bien, y que todo se ejecute de manera correcta.

Cuando ya se compilo y esta todo bine, se hace un archivo .ex ese se encuentra donde esta guardado el proyeto.

En el editor de código ensmblador se hace un proyecto que contenga las especificaciones, que es para un PIC 16F84A, el cual utilizamos, después de eso agregamos un nuevo archivo y lo guardamos como (.asm), que es la terminal del lenguaje ensamblador, escribimos el código y lo compilamos, dando clic en debuguer luego en selectool y en la opción de MPLAB SIM. Después le damos clic en el botón de correr código y ya nos debería de esar corriendo el código, en otra pantalla nos va a aparecer los errores que tenga el código, solo nos dira los errores del código no nos dira si va a hacer lo que queramos que haga.

# Entablación de la simulación en físico.

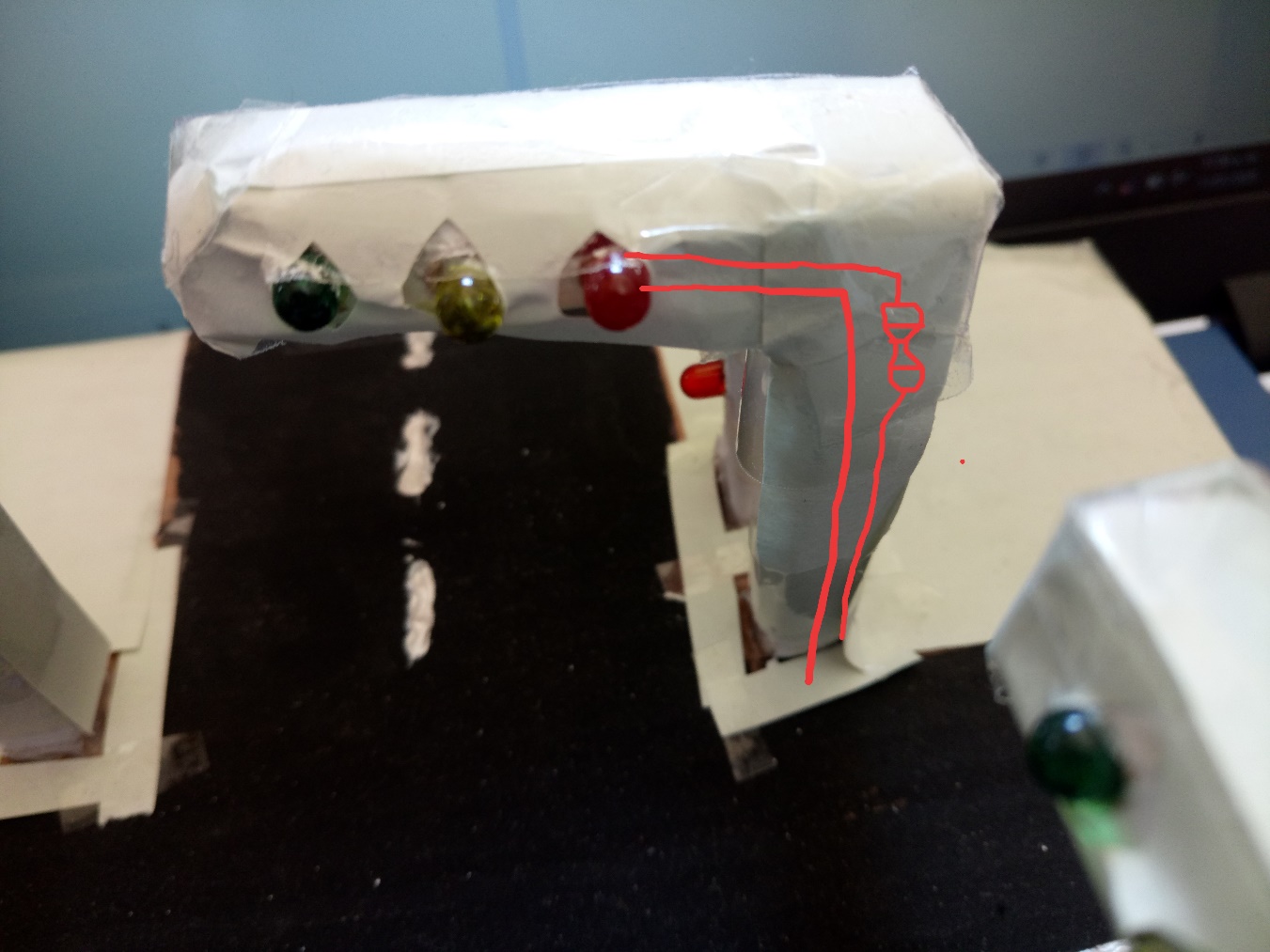


Yo conecte los cables de esa forma, en la parte verde está el PIC.

En la parte roja están los puentes para dar corriente a las dos partes de la tabla proto para poder conectar ahí los cables a corriente o a tierra.

En la parte morada se ve de donde salen los cables.

En la parte amarilla se ve un circuito, y en la salida 4 esta una resistencia en lugar de el circuito ya que también funciona solo con una resistencia.



En esta imagen se muestra cono es que los leds tienen soldada la resistencia de 375 Homs.