



Nombre de la práctica	Semáforo			No.	2
Asignatura:	Arquitectura de computadoras	Carrera:	Ingeniería en sistemas computacionales	Duración de la práctica (Hrs)	Muchas horas

NOMBRE DEL ALUMNO: Jose Armando Blas Arce  
GRUPO:3012

### I. Competencia(s) específica(s):

### II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): aula

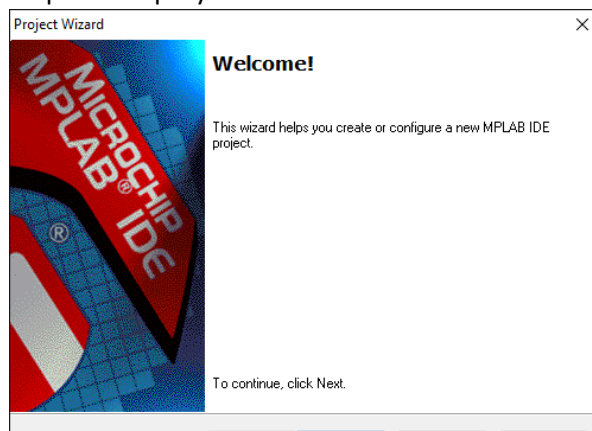
### III. Material empleado:

- Tabla PROTOBOARD
- Cables para PROTOBOARD
- PIC16F14A
- Capacitores cerámicos de 22 pf
- Cristal oscilador de 5 o 4 MHz
- Resistencia de 375Ω
- Laptop con el software de PROTEUS y MPLAB
- Un cargador (que sirva, pero no se use)
- Programador de PIC's
- 4 Leds jumbo rojos
- 4 Leds jumbo verdes
- 4 Leds jumbo ámbar
- 4 Leds rojos
- 4 Leds verde

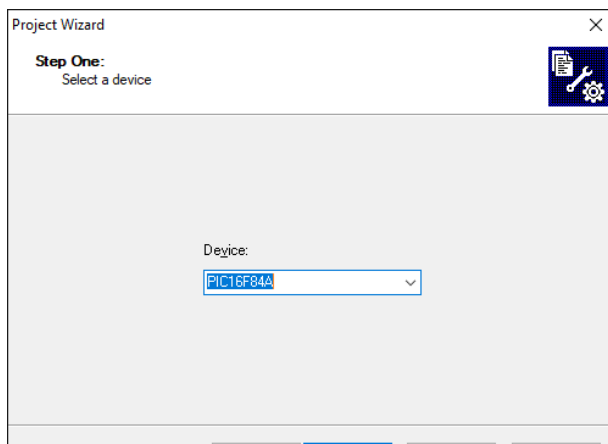
### IV. Desarrollo de la práctica:

#### Creación de código en MPLAB

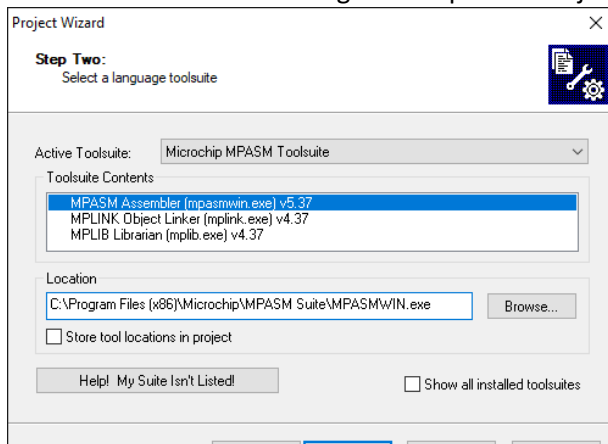
1. Paso1: Ingresar a MPLAB, una vez dentro vamos a la pestaña de proyect, la seleccionamos colocamos la opción de project Wizard



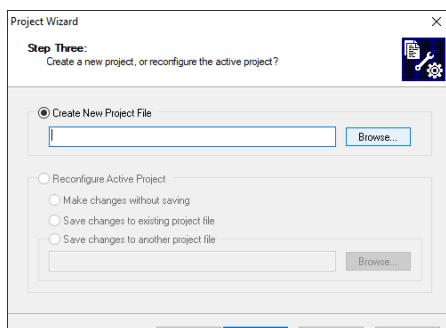
2. le damos siguiente y ahora seleccionaremos el nombre del PIC en nuestro caso sería el PIC16F84A



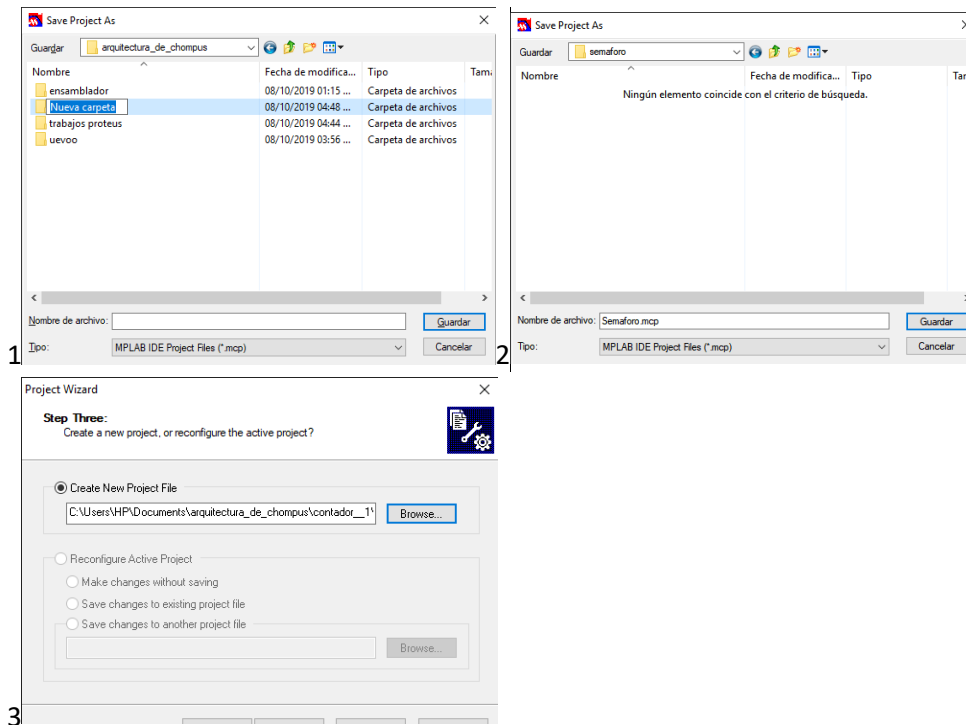
3. Una vez seleccionado la siguiente opción la dejamos por defecto



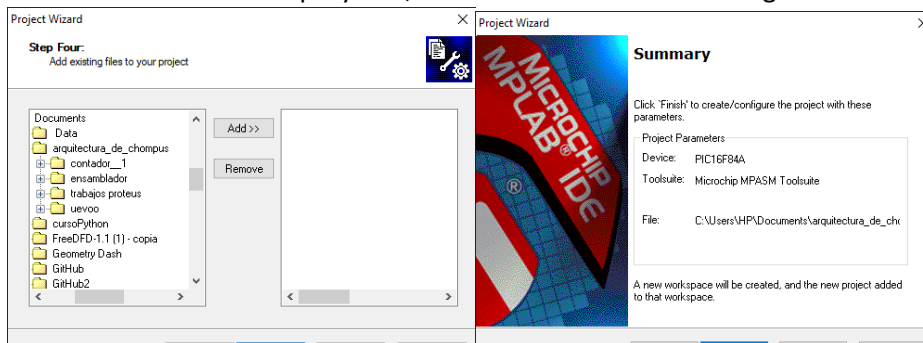
4. La siguiente opción es donde se va a encontrar la carpeta que tendrá dentro el código, para eso le damos la opción de BROSWE



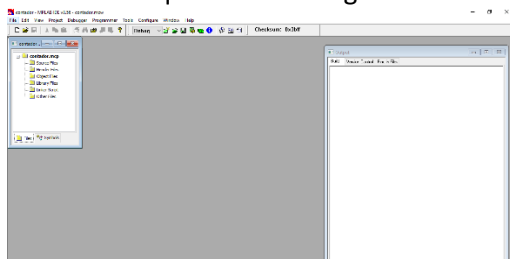
5. Creamos una carpeta con el nombre que queramos y luego creamos el nombre del proyecto que tendrá una extensión .mcp. Una vez seleccionada la ruta le oprimimos enter y seguimos



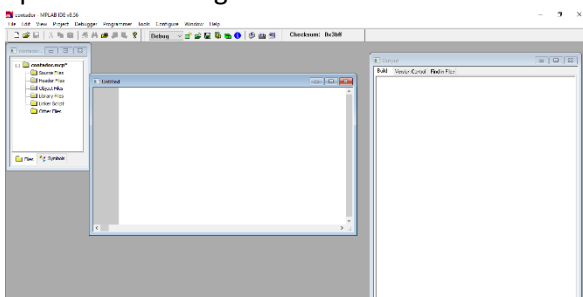
6. Luego aparecerá la siguiente pantalla (esta aparece por si tenemos un documento ya creado y lo quisiéramos agregar). Como solo es el inicio le damos next y nos aparecerá una carpeta marcando que ya se finalizó la creación del proyecto, ahora solo falta hacer el código



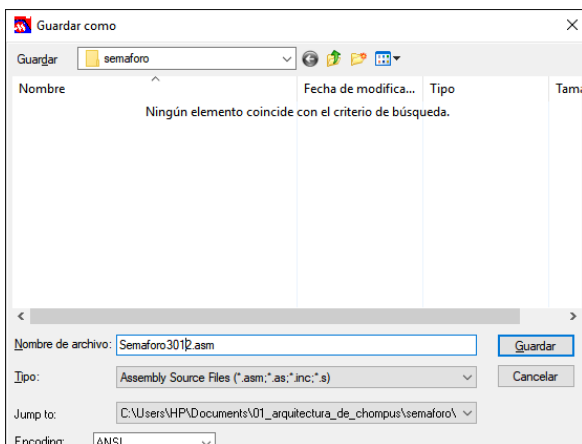
7. Ahora nos quedara de la siguiente forma



8. Ahora nos vamos a la pestaña de files y le damos new file (lo que hará es crear un nuevo documento) y aparecerá de la siguiente forma



9. Una vez creado nos iremos a la opción de file y a la opción de save all, al seleccionarla tendremos que buscar la carpeta de donde creamos el proyecto y colocar el nombre del documento con una extensión .ASM



10. Le damos aceptar y ahí es donde empezaremos a hacer el código  
11. El código para el semáforo es el siguiente

```

LIST P=PIC16F84A, R=DEC
#include <P16F84A.INC>
__CONFIG _CP_OFF & _FWRITE_OFF & _WDT_ON & _XT_OSC; 0x3FFD
ORG
GOTO PRINCIPAL ;SALTO AL PROGRAMA PRINCIPAL AL RESETEARLO

ORG
GOTO IRQ ;SALTO A LA RUTINA DE INTERRUPCION

;DEFINIR LAS CONSTANTES
CUENTA EQU 16 ;LLEVA LA CUENTA DEL TMRO (256-240)
RP2 EQU 7 ;SON LOS BITS DE LOS SEMAFOROS DE PEATONES
RP1 EQU 6
VP2 EQU 1
VP1 EQU 0
;DECLARAR VARIABLES
CONTADOR EQU 0x10
RETARDO EQU 0x11

;PROGRAMA PRINCIPAL
;SE INICIALIZA PIC Y SE PROGRAMA EL REGISTRO DE CONTROL, LOS PUERTOS E INTERRUPCIONES
;EL PROGRAMA LANZARA EL TEMPORIZADOR PARA ACTIVAR LAS INTERRUPCIONES
;PARA QUEDAR EN ESPERA DE UN CICLO INFINITO

PRINCIPAL:
;INICIALIZAR EL PIC Y PROGRAMAR EL OPTION_REG
;BIT7 RPBUS1 RESISTENCIAS DEL PULL-UP INTERNAS DEL PORTE DESHABILITADAS
;NO ES NECESARIO PORQUE SE DESHABILITAN AL USARLO COMO SALIDA
;BIT6 INTEDG=0 NO SE USARA PORQUE NO UTILIZAREMOS INT
;BIT5 TOCS=0 TEMPORIZADOR DEL RELOJ INTERNO DEL PIC
;BIT4 TOSE=0 DA IGUAL
;BIT3 PSA=0 PRE-ESCALER DEL TMRO
;BIT2-0=111 AJUSTA EL PRE-ESCALER DEL TMRO A 256
MOVLW B'10000111' ;PALABRA PARA OPTION Y SE TENDRA QUE MODIFICAR
BSF STATUS,RP0 ;BANCO1
MOVWF OPTION_REG ;SE ESCRIBE LA PALABRA W EN EL REGISTRO OPTION

```



```
BCF      STATUS,RP0      ;BANCO 0
;PROGRAMACION DE LOS PUERTOS
;PORTB=SAIDAS ==> TRISB IGUALA TODO A CERO
;ESTRUCTURA DEL PORTB:  7  6  5  4  3  2  1  0
;                        RP2 RP1 R2  A2 V2 R1 A1 V1
;                        ~~~~~
;                        S2
;
BSF      STATUS,RP0      ;BANCO 1
CLRFB   TRISB            ;BORRA LOS BITS DE SALIDA DEL PORTB
CLRFB   TRISA            ;BORRA LOS BITS DE SALIDA DEL PORTA
BCF      STATUS,RP0      ;BANCO 0

;INICIALIZAR EL CONTADOR Y EL PUERTO
CLRFB   CONTADOR         ;CONTADOR=0
MOVLW   B'01100001'      ;S2 ROJO, S1 VERDE. PEATONES, ROJO EN
MOVFB   PORTB            ;LO EXTRAER DEL PORTB
BCF      PORTA,VP2        ;S2 VERDE PARA LOS PEATONES
BSF      PORTA,VP1        ;S1 VERDE PARA LOS PEATONES
;PROGRAMAR TEMPORIZADOR
MOVLW   CUENTA
MOVWF   TMR0             ;INICIALIZA EL TEMPORIZADOR E INICIA

;PROGRAMACION DE INTERRUPTONES
;BIT7 GIE=1 PARA HABILITAR LAS INTERRUPTONES
;BITS TOE=1 PARA HABILITAR LAS INTERRUPTONES DEL TEMPORIZADOR
;EL RESTO A 0 PARA DESHABILITAR EL RESTO DE LAS INTERRUPTONES Y LIMPIAR LAS BANDERAS
MOVLW   B'10100000'
MOVWF   INTCON           ;PROGRAMA LAS INTERRUPTONES
ESPERA:                ;CICLO INFINITO QUE BORRA CONTINUAMENTE
CLRWDIT                ;LA CAMBIA LA HACE LA IRQ, SOLO SI HUBIERA UN PROBLEMA
GOTO    ESPERA           ;SE REINICIA EL PIC16F84A

;ROUTINA DE INTERRUPTON DEL TEMPORIZADOR
;LAS INTERRUPTONES SE GENERARAN CADA SEGUNDOS
;SE CONTARAN UN TOTAL DE 5 LLAMADAS
;SE INCREMENTARÁ EL CONTADOR DE SEGUNDOS PARA CONTROLAR LA TEMPORIZACION DEL SEMAFORO
; Y MODIFICAR SU ESTADO A TRAVES DEL TIEMPO

;NO ES NECESARIO COMPROBAR LA FUENTE DE LA INTERRUPTON PORQUE SOLO SE HABILITARA EN EL TEMPORIZADOR
IRQ
;SE EMPEZARA REPROGRAMANDO EL TEMPORIZADOR PARA QUE INICIE EL CONTEO Y ASI NO INTRODUCIR UN RETARDO EN EL TEMPORIZADOR

;SE IMPLEMENTA LA SENTENCIA SWITCH-CASE
CASO20:                ;CONTADOR<20
MOVLW   B'00010000'
MOVWF   TMR0            ;REPROGRAMACION DEL TIMER
INCF    RETARDO
MOVLW   005
SUBWF   RETARDO,W       ;COMPARA EL RETARDO CON 5
BTFS    STATUS,Z        ;SI NO ES 5 ENTONCES
GOTO    FINSWITCH       ;SALTA AL FINAL DE LA ROUTINA DE INTERRUPTON
CLRF    RETARDO         ;LIMPIA RETARDO
;EL SIGUIENTE CODIGO SE EJECUTARA CADA 0.5 SEGUNDOS (UNIDAD)
INCF    CONTADOR,F      ;ACUMULADOR (CONTADOR=CONTADOR+1)
MOVLW   40
SUBWF   CONTADOR,W      ;COMPARA CONTADOR CON 40
BTFS    STATUS,C        ;OYE RAMIREZ HAY ACARREO???
GOTO    FINCASO20       ;NO HAY ACARREO ==> CONTADOR<=40
;CONTADOR<40 (NO HAY ACARREO)
MOVLW   B'01100001'     ;AUTOS: RS2, VS1, PEATONES: RP2 OFF, RP1 ON
MOVWF   PORTB           ;ENVIAMOS LA SALIDA AL PORTB
BCF      PORTA,VP2       ;VP2
BSF      PORTA,VP1       ;VP1
GOTO    FINSWITCH
FINCASO20:
CASO25:                ;CONTADOR<25
MOVLW   50
SUBWF   CONTADOR,W      ;COMPARA EL CONTADOR CON 50
BTFS    STATUS,C        ;HAY ACARREO???
GOTO    FINCASO25       ;NO HAY ACARREO ==> CONTADOR<=50
;CONTADOR<25 NO HAY ACARREO
MOVLW   B'01100001'     ;AUTOS: RS2, VS1 PEATONES: RP2 OFF,RP1 ON
MOVWF   PORTB           ;LO ENVIAMOS AL PORTB
BTFS    CONTADOR,0      ;SI CONTADOR ES PAR 2 UNIDADES=1 SEGUNDO
GOTO    FINSWITCH
MOVLW   B'00000010'     ;MASCARA XOR PARA CONMUTAR VP2 SIN MODIFICAR EL RESTO
XORWF   PORTA,F         ;CONMUTA VP2 CADA 2 UNIDADES
MOVLW   B'01100000'     ;d'096' " "
MOVWF   PORTB           ; PORTB - TRISB
GOTO    FINSWITCH
FINCASO25:
CASO30:                ;25<=CONTADOR<30
MOVLW   60
SUBWF   CONTADOR,W      ;COMPARA CONTADOR CON 60
BTFS    STATUS,C        ;HAY ACARREO?????
GOTO    FINCASO30       ;NO HAY ACARREO==>60 CONTADOR<30 NO HAY ACARREO
MOVLW   B'01100010'     ;MASCARA XOR PARA CONMUTAR AS1 SIN MODIFICAR EL RESTO
MOVWF   PORTB           ;CONMUTA AS1 CADA 0.5 SEGUNDOS DURANTE UN LAPSO DE 25 A 30 SEGUNDOS
BTFS    CONTADOR,0      ;APAGA VS1
GOTO    FINSWITCH
MOVLW   B'00000010'     ;MASCARA PARA VP2
XORWF   PORTA,F         ;CONMUTA VP2
GOTO    FINSWITCH
FINCASO30:
```



```

CASO50:                                ;30<=CONTADOR<50
MOVLM      100
SUBWF      CONTADOR,W                ;COMPARA EL CONTADOR CON 100
BTFS      STATUS,C                    ;SI HAY ACARREO???
GOTO       FINCASO50
;CONTADOR <50 NO HAY ACARREO
MOVLM      B'10001100'                ;AUTOS VS2,RS1 PEATONES:RP2 OFF,RP1 ON
MOVWF      PORTB                      ;LO ENVIAMOS AL PUERTO
BSF        PORTA,VP2
BCF        PORTA,VP1
GOTO       FINSWITCH
FINCASO50:

CASO55:                                ;30<=CONTADOR<55
MOVLM      110
SUBWF      CONTADOR,W                ;COMPARA CONTADOR CON 110
BTFS      STATUS,C                    ;HAY ACARREO???
GOTO       FINCASO55
;CONTADOR<55 NO HAY ACARREO
MOVLM      B'10001100'                ;AUTOS: VS2,RS1 PEATONES: RP2 ON,RP1 OFF
MOVWF      PORTB                      ;LO ENVIAMOS AL PORTB
BTFS      CONTADOR,0                  ;CUANDO EL CONTADOR SEA PAR ENTONCES ES 1 SEGUNDO
GOTO       FINSWITCH
MOVLM      B'00000001'                ;MASCARA XOR PARA CONMUTAR VP1 SIN MODIFICAR EL RESTO
XORWF      PORTA,F                    ;CONMUTAR VP1 CADA 1 SEGUNDO
MOVLM      B'10000100'                ; b'10000100' d'132'
MOVWF      PORTB                      ; PORTB - TRISB
GOTO       FINSWITCH
FINCASO55:

CASO60:                                ;55<=CONTADOR<60
MOVLM      120
SUBWF      CONTADOR,W                ;COMPARA EL CONTADOR CON 120
BTFS      STATUS,C                    ;HAY ACARREO???
GOTO       FINCASO60
;NO HAY ACARREO=>CONTADOR>=120 CONTADOR<60 NO HAY ACARREO
MOVLM      B'10010100'                ;MASCARA XOR PARA CONMUTAR AS2 SIN MODIFICAR EL RESTO
BTFS      CONTADOR,0                  ;SI EL CONTADOR ES PAR ENTONCES LO HARA CADA SEGUNDO
;CONMUTA AS2 CADA 0.5 SEGUNDOS DURANTE EL TIEMPO DE 55 Y 60 SEGUNDOS
BSF        INDE,0                     ;APAGA VS2
BTFS      CONTADOR,0                  ;SI EL CONTADOR ES PA ENTONCES LO HARA CADA SEGUNDO
GOTO       FINSWITCH
MOVLM      B'00000001'                ;MASCARA PARA EL VP1
GOTO       FINSWITCH
FINCASO60:

OTHERWISE:
MOVLM      B'10011110'                ;LEGO A 60
MOVWF      PORTB                      ;AUTOS RS2, VS1 PEATONES RP2 OFF, RP1 ON
BCF        PORTA,VP2                  ;LO ASIGNA AL PUERTO B
BSF        PORTA,VP1
CLRF      CONTADOR                   ;EMPIEZA EL CICLO NUEVAMENTE
FINSWITCH
MOVLM      B'10100000'                ;FIN DE LA SENTENCIA SWITCH-CASE
MOVWF      INTCON
RETFIE
;RETORNO DE INTERRUPCION
END

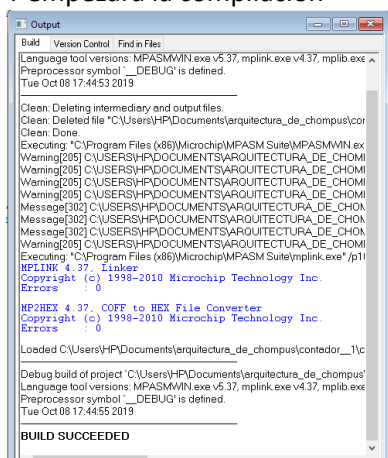
```

12. Una vez terminado el código continuaremos con lo que es la compilación. Para eso solamente le daremos al

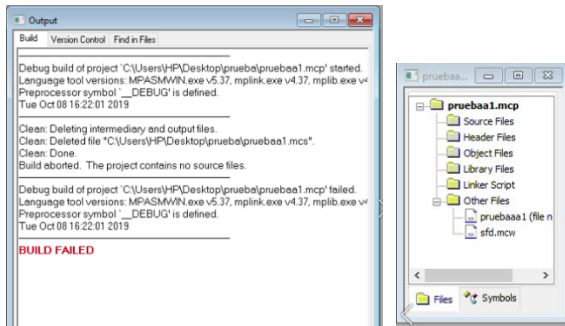
iconito



13. Y empezara la compilación



14. Si nos marca un error como este es porque la ubicación del proyecto no coincide con la del documento .ASM o puede ser también que haya ingresado un documento .ASM a otra carpeta que no sea la de SOURCE FILES que está dentro del programa



15. Una vez que el código se haya compilado correctamente iremos a la carpeta donde está el proyecto creado y veremos que dentro del él estará un archivo con una extensión .HEX

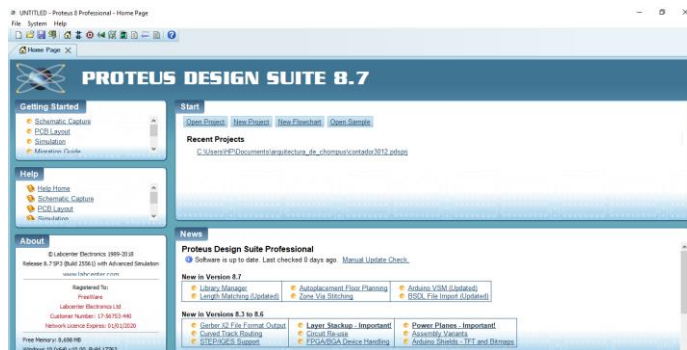
Documentos > arquitectura\_de\_chompus > contador\_1

Nombre	Fecha de modifica...	Tipo	Tamaño
contador.cof	08/10/2019 05:44 ...	Archivo COF	2 KB
contador.hex	08/10/2019 05:44 ...	Archivo HEX	1 KB
contador.map	08/10/2019 05:44 ...	Archivo MAP	3 KB
contador.mcp	08/10/2019 05:44 ...	Microchip MPLAB...	1 KB
contador.mcw	08/10/2019 04:49 ...	Microchip MPLAB...	32 KB
contador_Armando.asm	08/10/2019 05:36 ...	Archivo ASM	3 KB
contador_Armando.err	08/10/2019 05:44 ...	Archivo ERR	2 KB
contador_Armando.lst	08/10/2019 05:44 ...	Archivo LST	11 KB
contador_Armando.o	08/10/2019 05:44 ...	Archivo O	2 KB

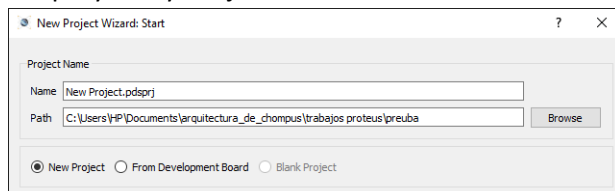
## Simulación de PROTEUS

PROTEUS es una aplicación que simula el comportamiento de los componentes electrónicos para evitar un fallo o una mala conexión, en este caso lo usaremos para simular un contador para el PIC16F84A

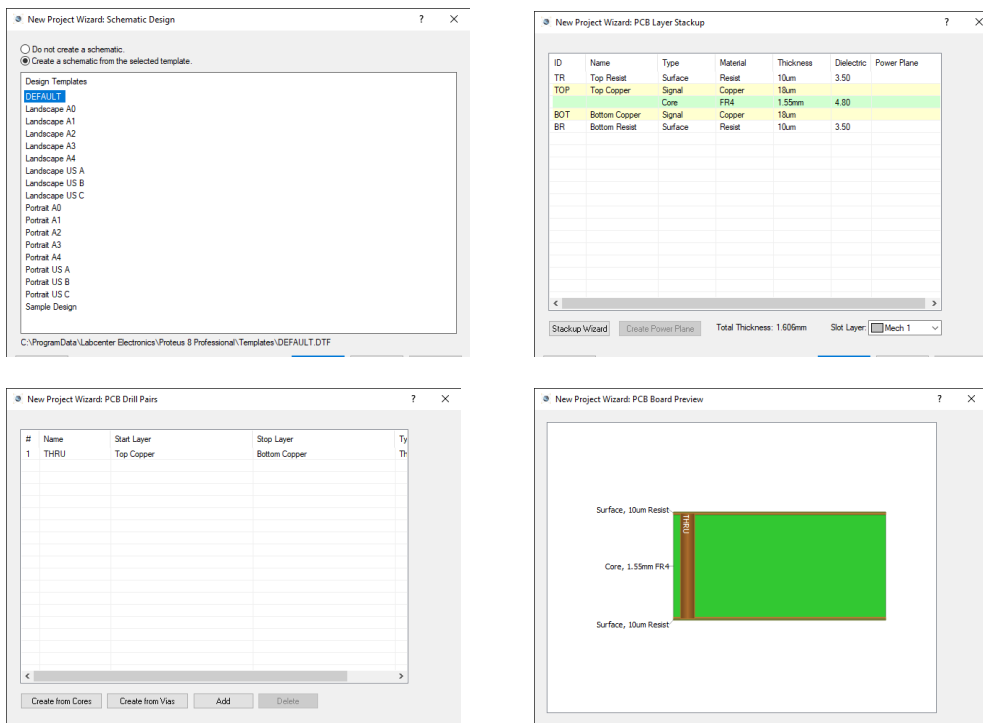
1. Comenzamos abriendo PROTEUS



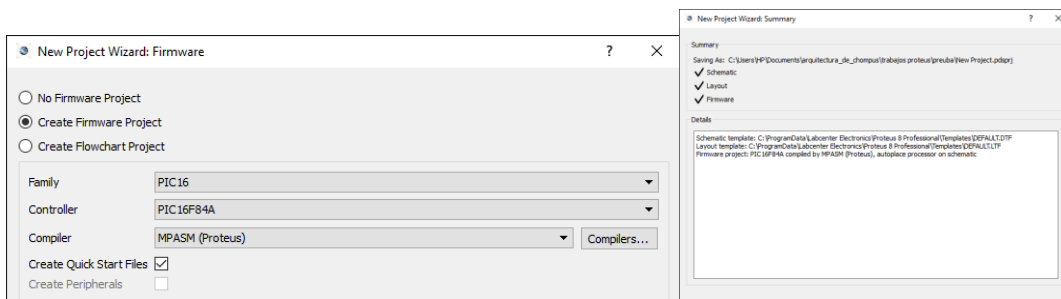
2. Seleccionamos la opción de FILE y NEW PROJECT, aparecerá la siguiente pantalla donde colocaremos el nombre del proyecto y abajo la dirección de donde se encontrará ubicado y continuaremos



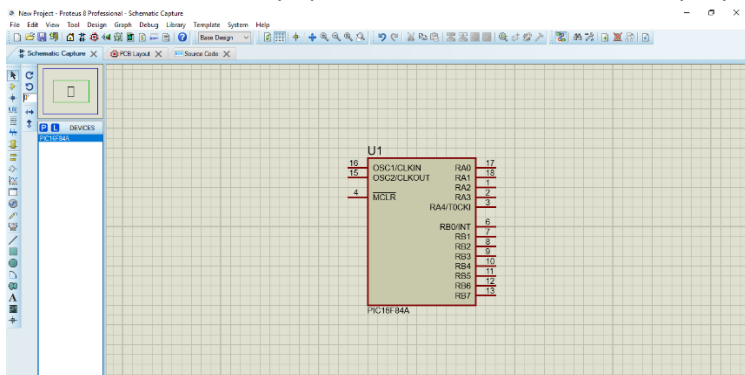
### 3. Lo siguiente lo daremos por defecto



### 4. Una vez que nos aparezca la siguiente ventana por lo que crearemos un nuevo firmware de la familia PIC16 y el controlador PIC16F84A y le daremos next y aparecerá que se ha creado exitosamente el proyecto

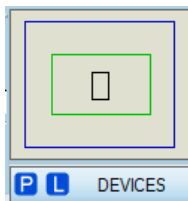


### 5. Le damos a continuar y aparecerá la base donde está el proyecto

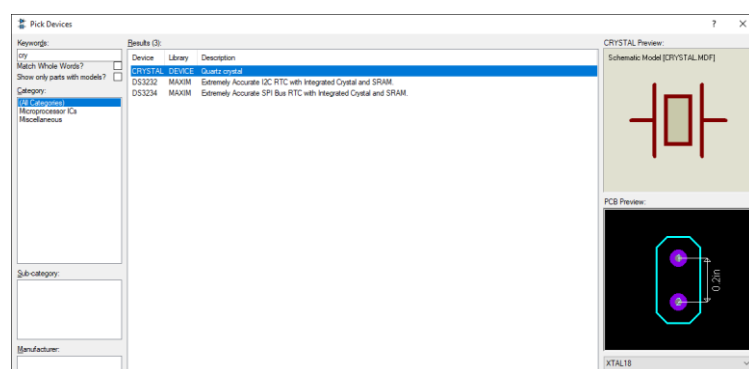




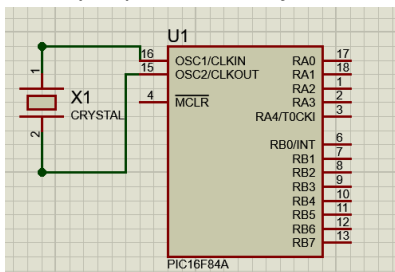
## 6. Seleccionamos la letra P ubicada debajo del mapa del pic



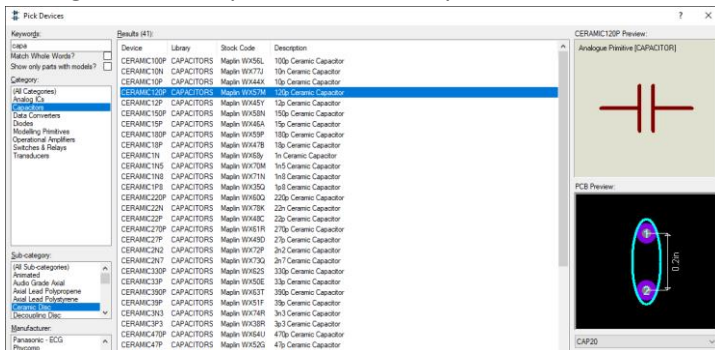
## 7. Ahora lo que sigue es poner los componentes para es iremos de nuevo abajo del mapa y seleccionaremos la letra P y buscaremos **CRYSTAL\_MFD** (esto lo que seria es el cristal oscilador que en nuestro caso sería de 4 o 5 Mhz).



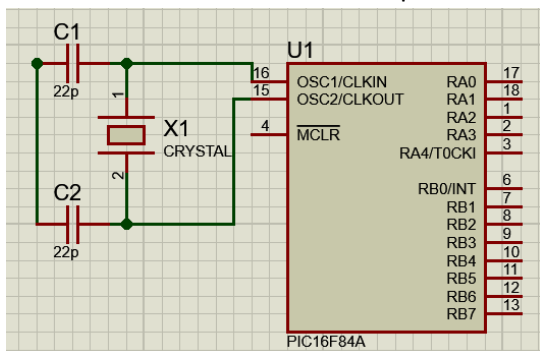
## 8. Lo colocamos en la simulación y el primer punto de arriba del cristal oscilador lo uniremos con la entrada del pic **OSC1** y el punto de abajo a la entrada **OSC2**



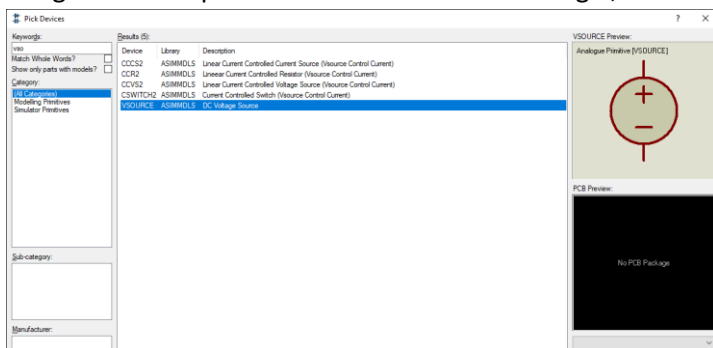
## 9. Los siguientes componentes son 2 capacitores cerámicos de 22 pf



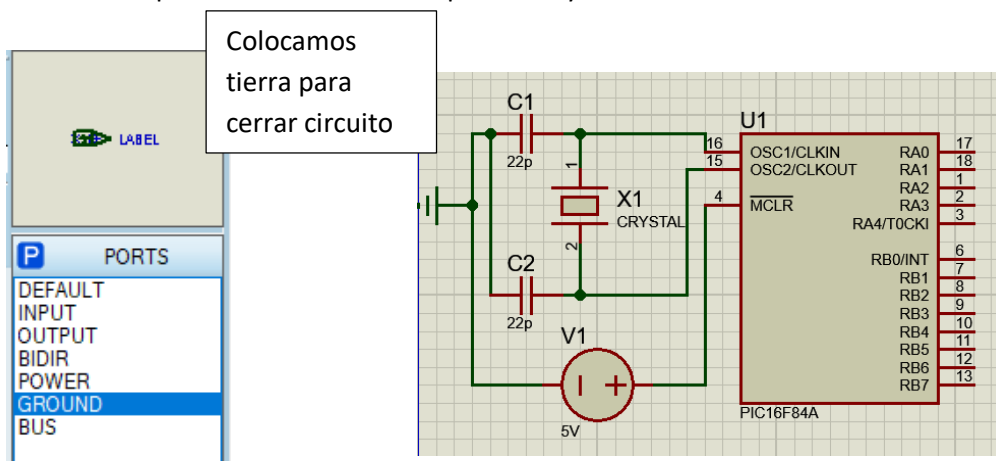
10. Los condensadores cerámicos se pondrán de la siguiente manera



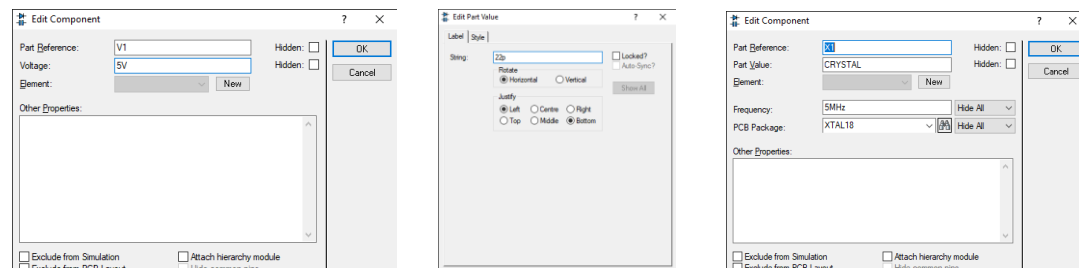
11. El siguiente componente seria la fuente de energía, este lo buscaremos como VSOURCE DC voltaje



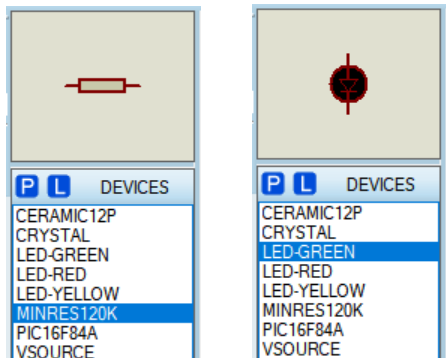
12. Ahora solo queda acomodar los componentes y conectarlos mediante líneas



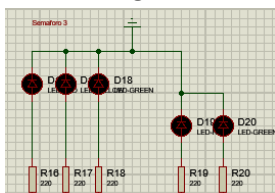
13. Ahora solo queda cambiar los valores de los componentes, para eso le damos doble clic y le cambiamos los valores a:



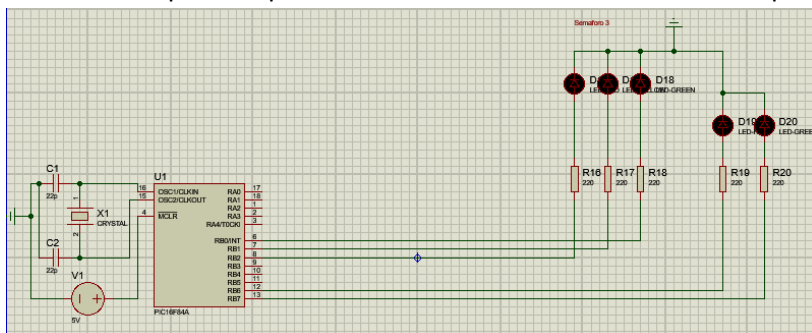
14. Los siguientes materiales son 8 leds rojos, 8 leds verdes y 4 leds ámbar



15. Ahora lo siguiente es hacer el primer semáforo, conectando primero los leds con una resistencia

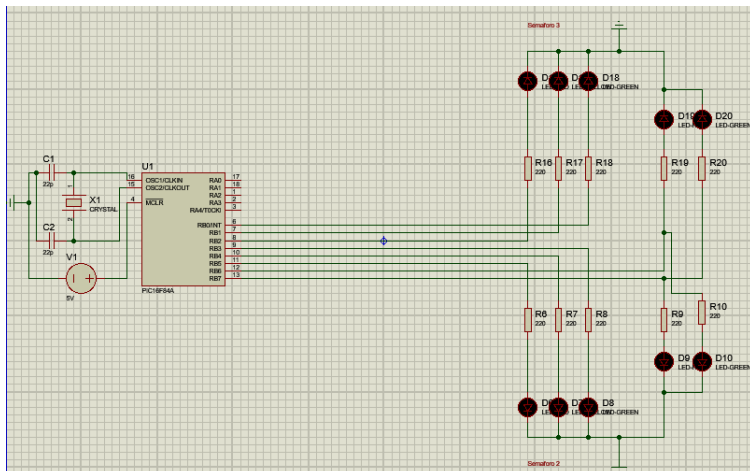


16. Ahora conectaremos el led verde con el puerto RB0, el led ámbar con el puerto RB1 y el led rojo con el puerto RB2. Los leds para los peatones irán conectados del led verde al puerto RB7 y led rojo al puerto RB6



Todos los leds  
llevan resistencias

17. Continuaremos con el siguiente semáforo que empezara con el led verde que ira al puerto RB3, el led ámbar va conectado al puerto RB4 y el led rojo al puerto RB5. De igual forma los leds de los peatones irán conectados a los puertos RB6 y RB7, pero donde iría el led verde debe estar el led rojo y en el led rojo estará el led verde

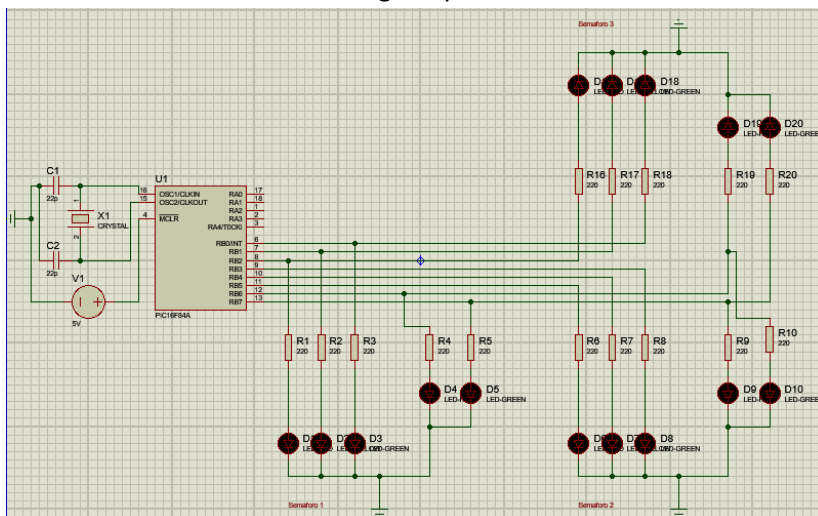


Los puertos RB0, RB1 y RB2 son los  
usados para los semáforos 1 y 3

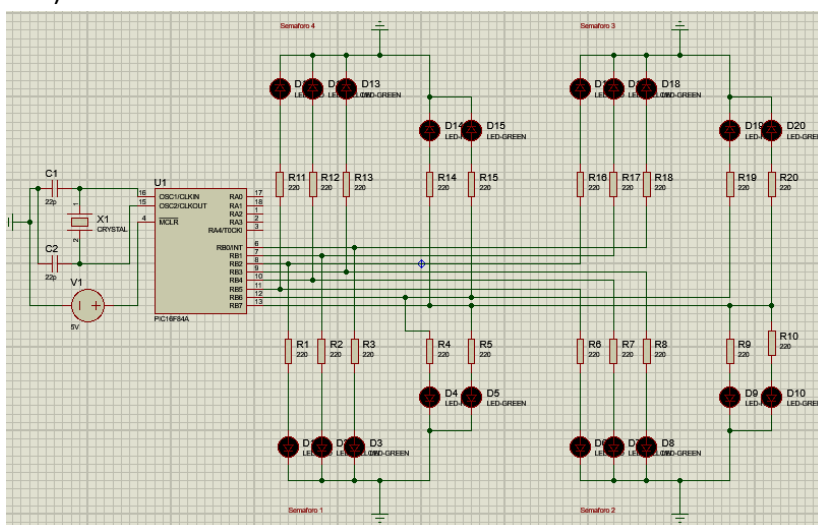
Los puertos RB3, RB4 y RB5 son  
usados para los semáforos 2 y 4

Los puertos RB6 y RB7  
son los usados para los  
semáforos

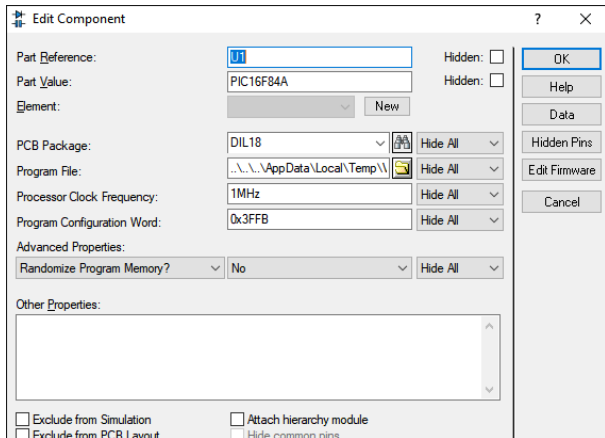
18. La conexión del semáforo 3 es igual que la conexión del semáforo 1 (con los puertos RB0, RB1 y RB2)



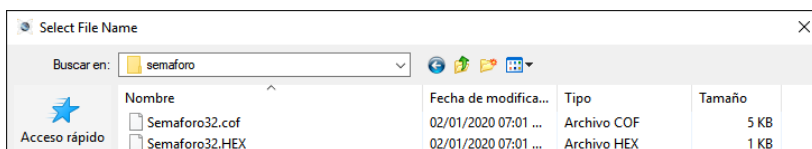
19. Por último, la conexión del 4° semáforo que será la misma que la del semáforo 2 (con los puertos RB3, RB4 y RB5)



16. Ya terminado estole daremos doble clic al PIC16f84A y nos aparecerá la siguiente ventana

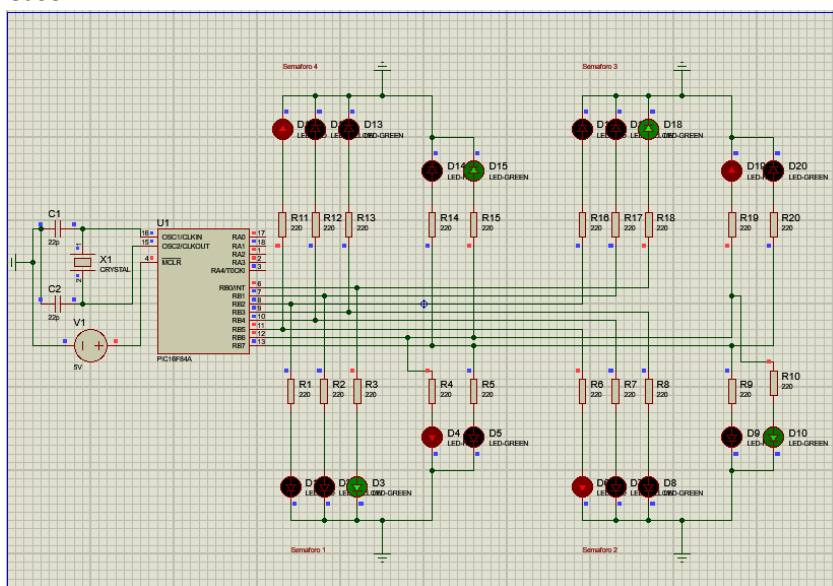


17. Una vez realizado eso seleccionaremos la carpeta que tiene y buscaremos la carpeta con el archivo .HEX que se creó anteriormente

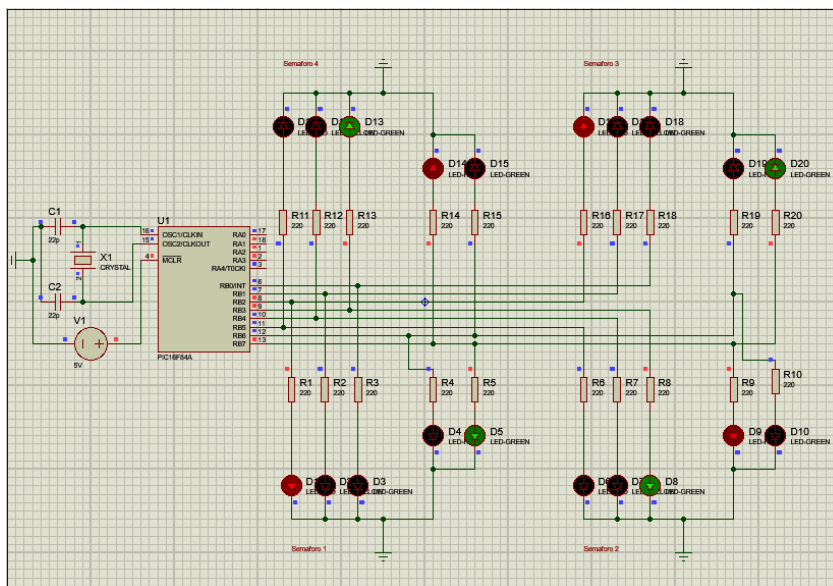


18. Y le daremos play con los botones ubicados en la parte inferior izquierda, empezaremos a notar como es que empieza a correr el programa

Caso 1:



Caso 2:



## Programación del pic16F84A con el programador de pic's

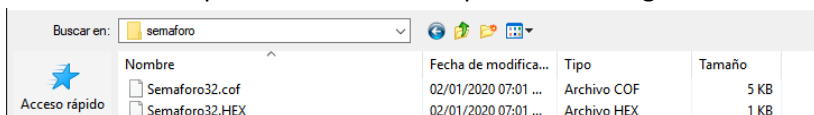
1. Para programar el pic necesitamos tener instalado el software necesario para hacer la programación del pic (la instalación del software no es tan compleja como las demás instalaciones) que en nuestro caso vendría junto con el programador de pic's.
2. Primero abriremos el software del programador de pic's
3. Luego conectamos el pic dentro del programador
4. Luego en el software aparecerá el botón AUTO/CONEX, lo presionamos y aparecerá que se ha encontrado el PIC(en dado caso que no lo encuentre lo que haría es que pusiera el pic en forma contraria a la que estaba).



5. Una vez que ya haya localizado al PIC36F84A seleccionamos la opción LEER para que comience a leer lo que ingresemos en el PIC



6. Ya que este leyendo, lo que seguiría es irnos a la pestaña de archivos / abrir HEX, le damos clic y buscamos el formato .hex que viene cuando compilamos el código en MPLAB



7. Ya realizado eso aparecerá un mensaje diciendo que se agregó correctamente



8. Ahora lo que seguiría es irnos a la opción escribir y darle clic, en automático empezara a programar el PIC y al último aparecerá que se ha programado de forma correcta



9. Ahora faltaría verificarlo con la opción verificar y empezara a ver que se haya programado de forma correcta

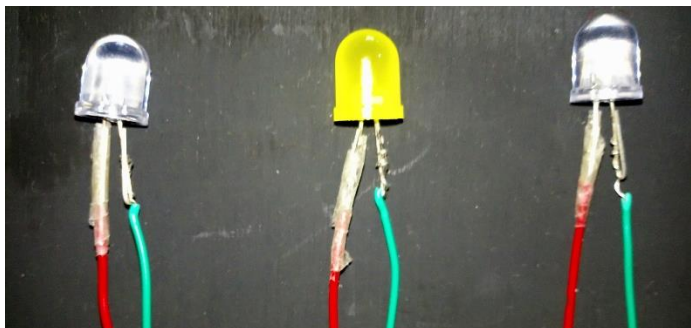


10. Lo último que faltaría es desconectar el PIC y ya estaría programado para su uso

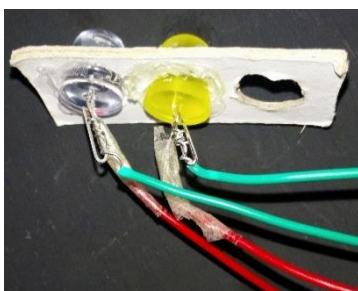
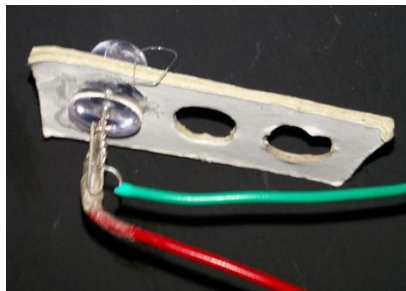


## Implementación de los materiales en la tabla PROTOBOARD

1. Para iniciar en esta práctica debemos tener claro que necesitamos todos los materiales pedidos al principio
2. Comenzamos con los leds, a cada uno de los leds le conectaremos un cable rojo para la corriente positiva (es la punta más grande) y un cable verde para la corriente negativa (la punta pequeña)



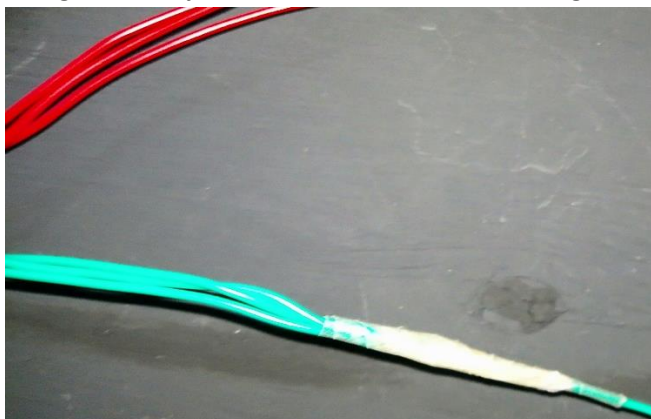
3. Lo siguiente es hacer cajas chiquitas con lo que queramos con tres hoyos donde quepan los tres leds



4. Ahora los meteremos los leds dentro de la caja

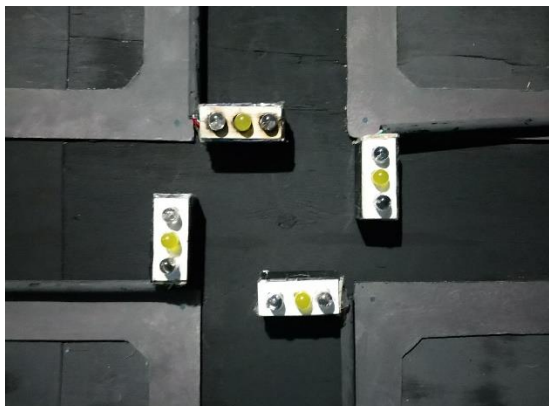


5. Lo siguiente es juntar los cables de corriente negativa (los cables verdes) y hacer un solo cable





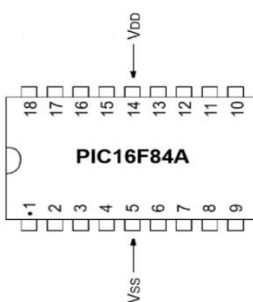
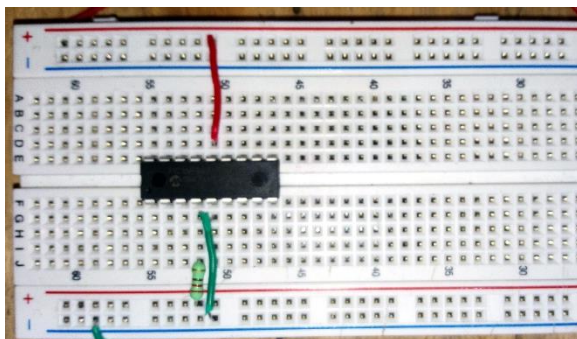
6. Una vez que ya tengamos el primer semáforo hacemos lo mismo con los otros 3 que faltan



7. Continuamos colocando 2 semáforos en cada esquina

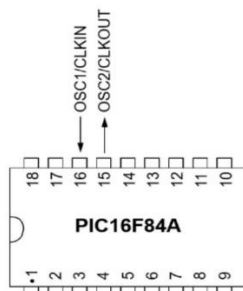
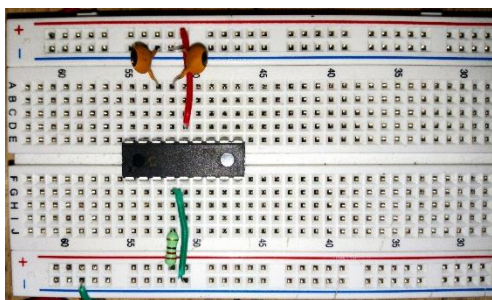


8. Una vez colocados, tendremos que pegar la tabla protoboard atrás de la tabla

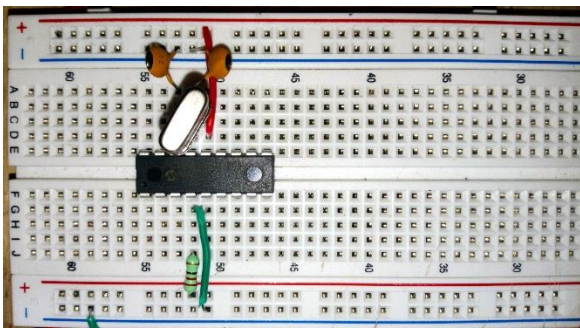


Alimentaremos al pic  
conectando el cable rojo que  
va desde el puerto VDD a la  
corriente negativa y  
conectando el cable verde  
que va del puerto VSS a  
corriente negativa

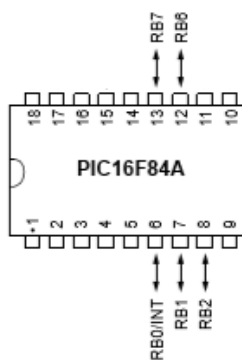
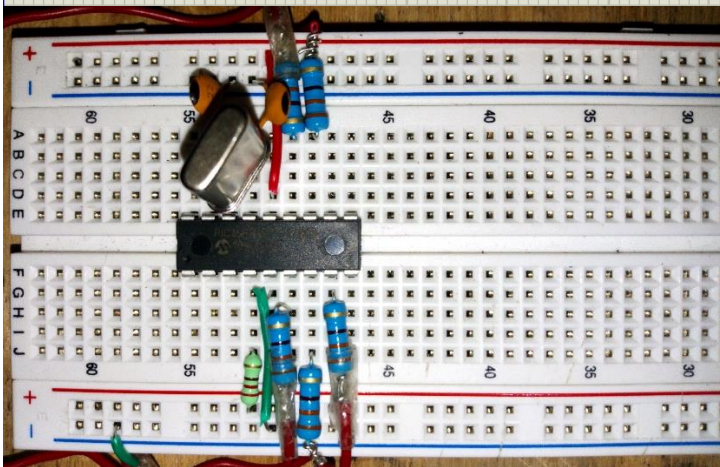
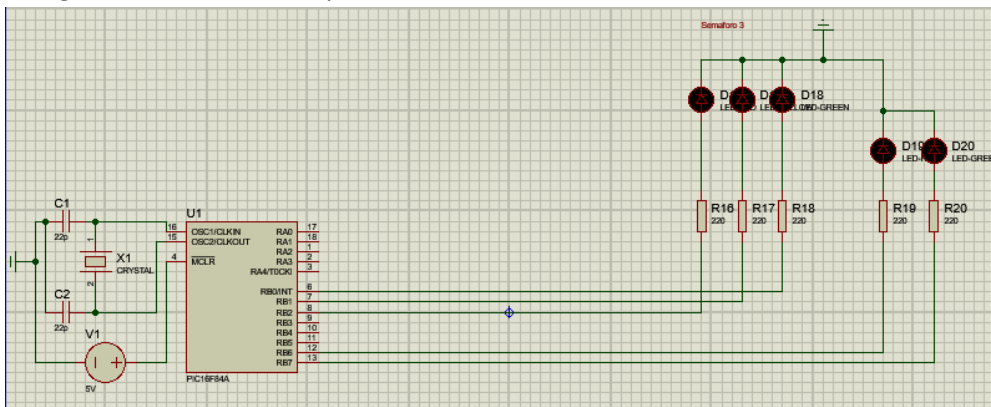
9. Lo siguiente es conectar los condensadores cerámicos de 22pf en los puertos OSC1 y OSC2



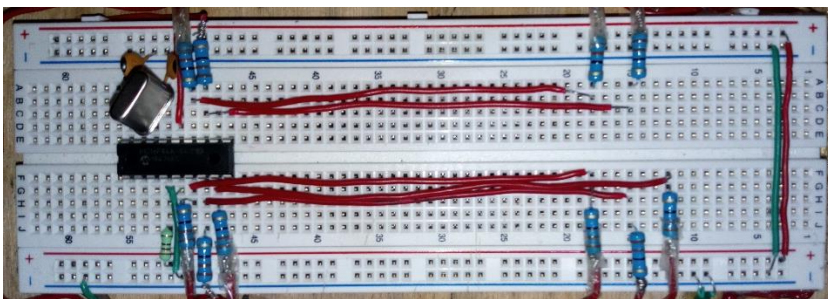
10. De igual forma conectaremos también el cristal oscilador como los condensadores



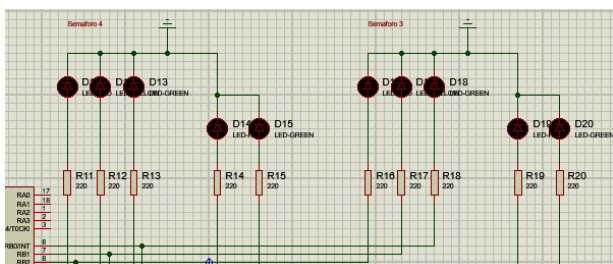
11. Lo siguiente es conectar el primer semáforo como en la simulación



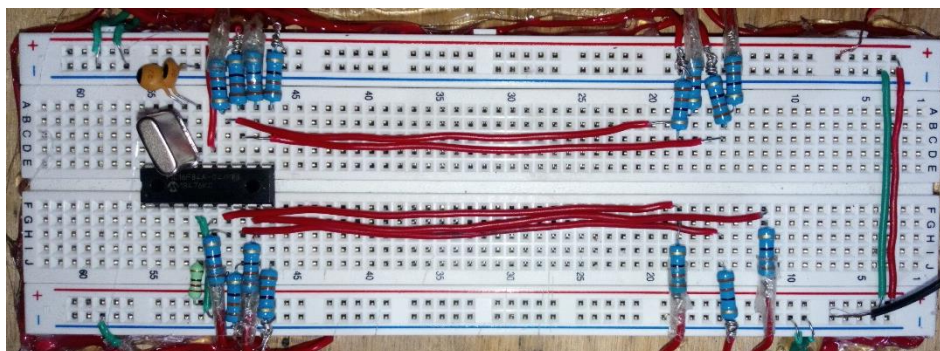
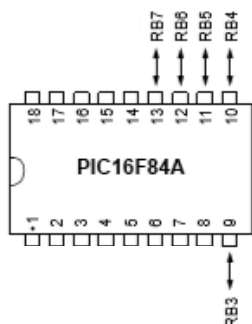
12. Ahora conectaremos el segundo semáforo con forme a la simulación.



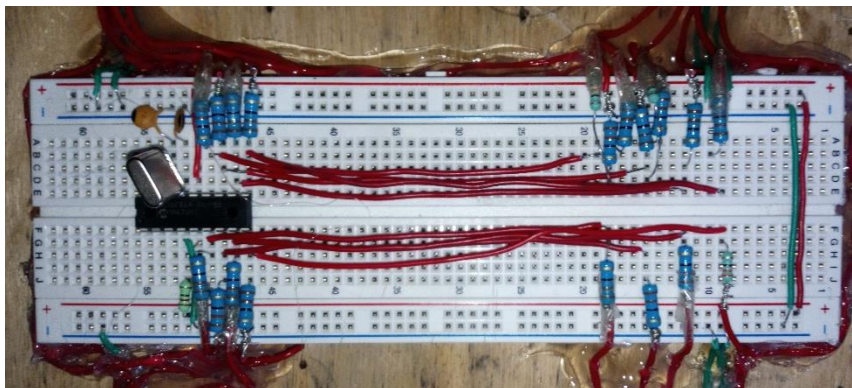




13. Continuaremos con el tercer semáforo y colocándolos en la maqueta que ira de la siguiente forma

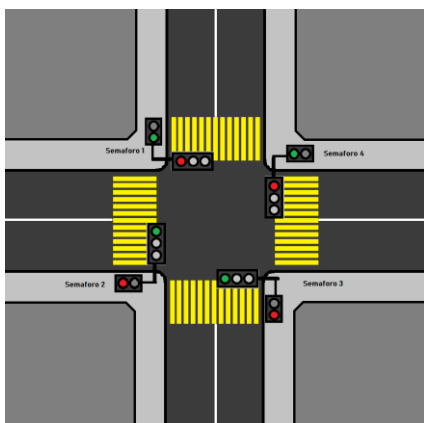


14. Por último, el 4° semáforo que tendrá las mismas líneas que el semáforo 3

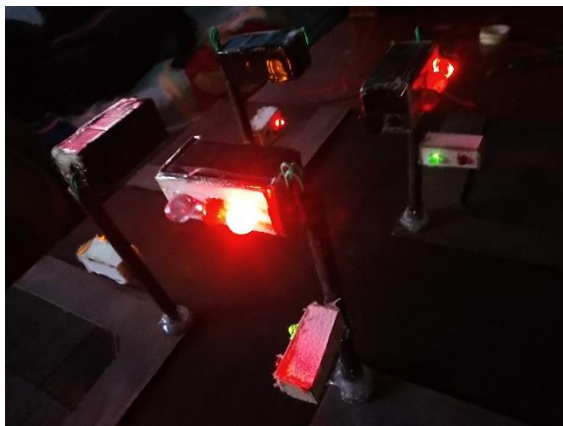
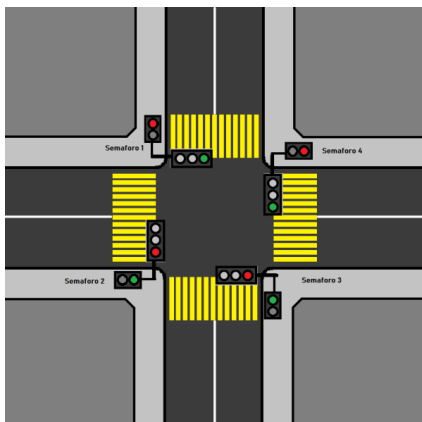


15. Al último conectamos a corriente y checamos

Caso 1:



## Caso 2:



## Vista aérea



## V. Conclusiones:

En esta práctica lo que fue el código para programar el **pic** fue algo complicado de entender al menos en opinión personal, pero con forme a vamos avanzando y haciendo el proyecto comprendemos para que sirven algunas líneas del código y para qué sirven. Este proyecto es algo largo y complicado de hacer ya que debemos estar pendientes de que no se nos quemen ningún led ni algo similar y aparte que la elaboración de cada uno de los semáforos requiere de una gran concentración y precisión para que ningún semáforo quede mal o quede feo.