

Nombre de la práctica	Proyecto Semáforo			No.	4
Asignatura:	Arquitectura de Computadoras	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	20 horas

NOMBRE DEL ALUMNO:

Tenyo Emanuel Petrearce Pineda

GRUPO: 3012

I. Competencia(s) específica(s):

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

Aula de clases y laboratorio independiente

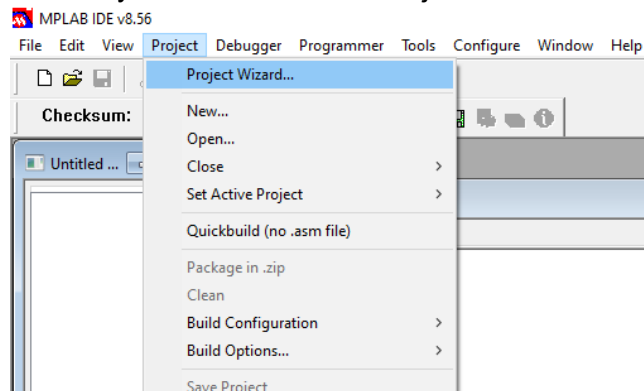
III. Material empleado:

MPLAB
Proteus
PIC16F84A
Tabla Protoboard
Cable para Protoboard
Resistencias de 220 Ohms
Leds Jumbo
Cautín y Soldadura
Tubo de Metal
Cartón

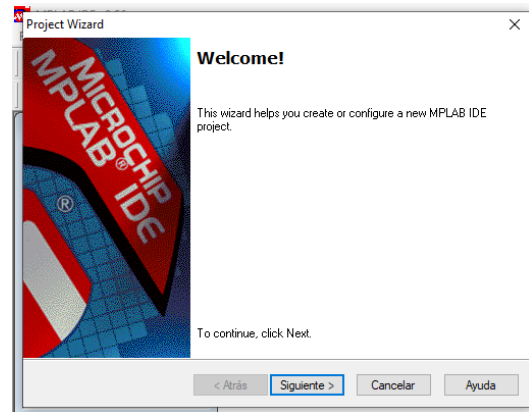
IV. Desarrollo de la práctica:

En este proyecto se inició con el código en el lenguaje ensamblador y fue desarrollado en MPLAB

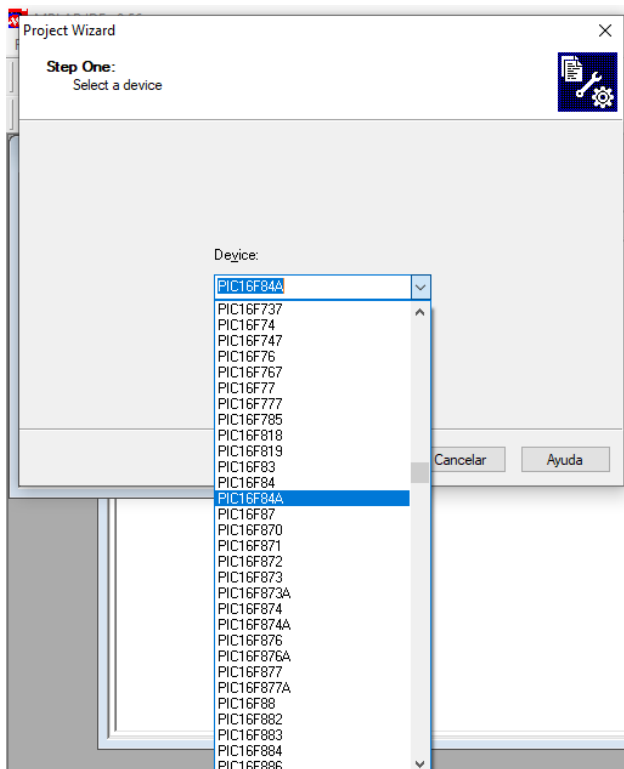
En la pestaña "Project" se deslizará una ventana y seleccionaremos "Project "Wizard"



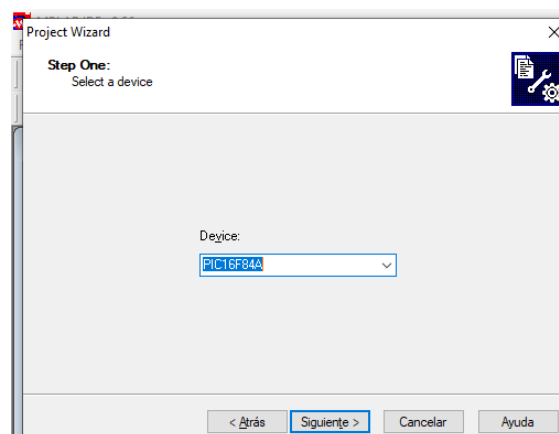
Aparecera una ventana donde daremos siguiente



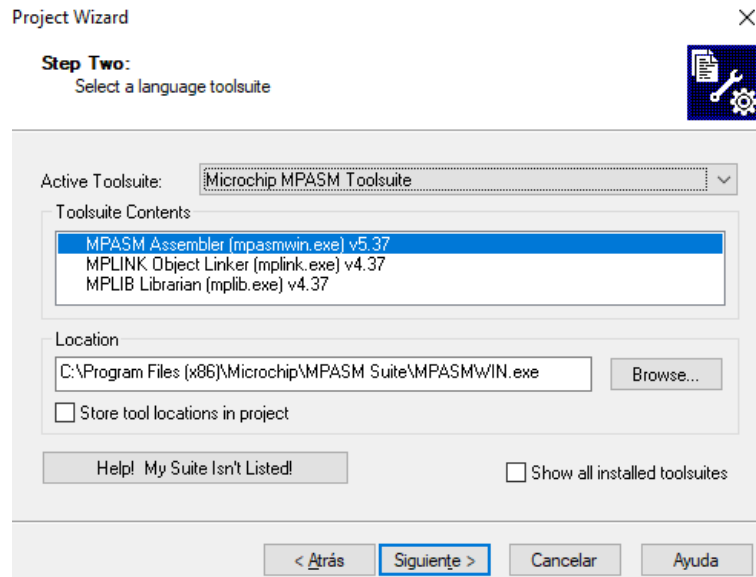
Seleccionaremos nuestro PIC correspondiente, en este caso ocupamos el PIC16F84A



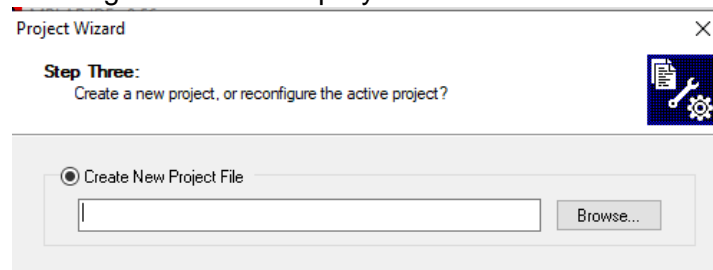
Pulsamos siguiente



En esta ventana daremos
siguiente sin cambiar nada

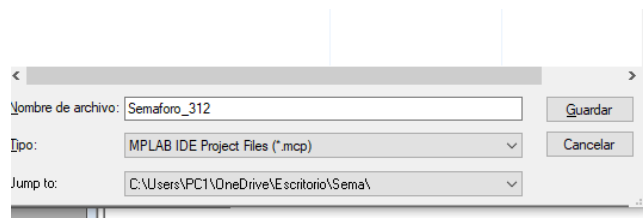


Seleccionaremos donde guardar nuestro proyecto

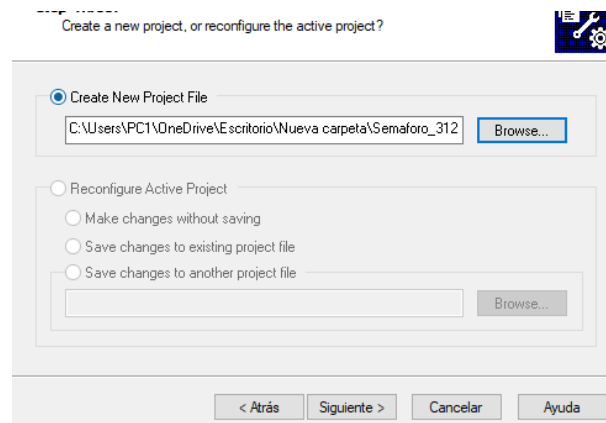


Escribimos el nombre de nuestro
proyecto
Ejemplo:
Semaforo_312

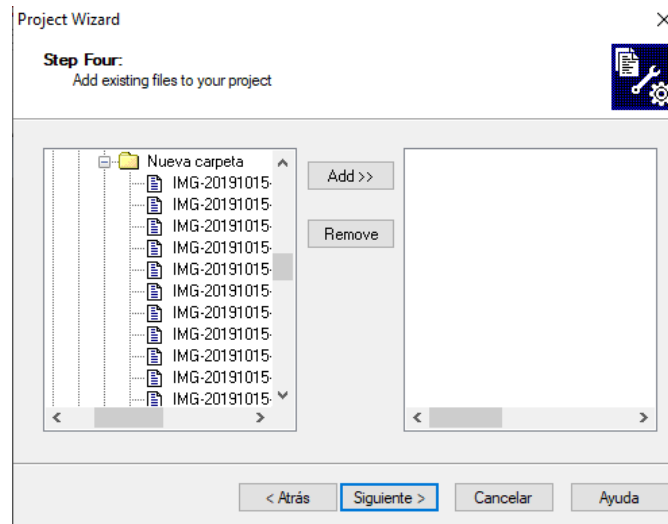
Dar click en guardar



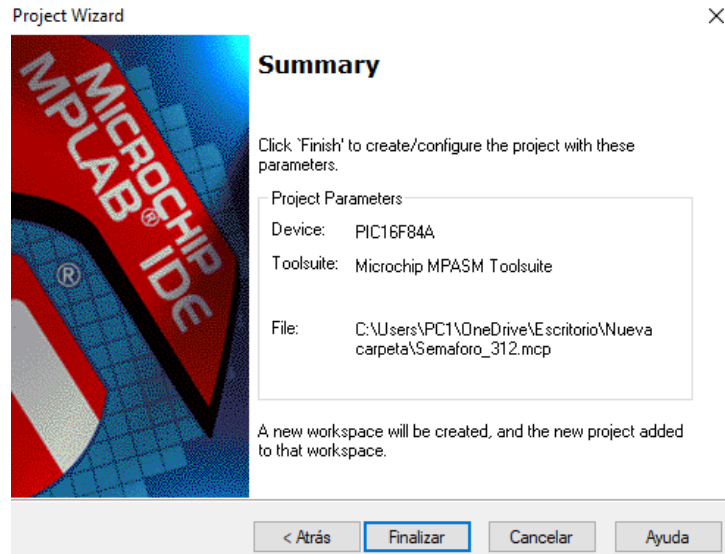
Se guardará la dirección que
especificamos y daremos en siguiente



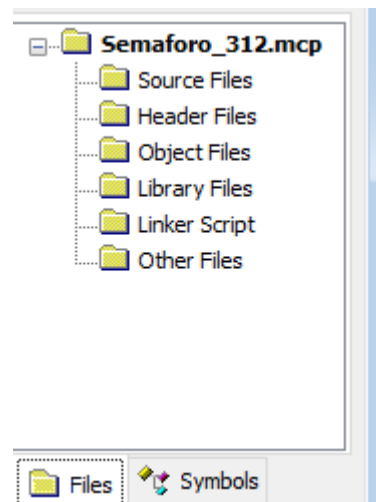
En la siguiente ventana daremos siguiente



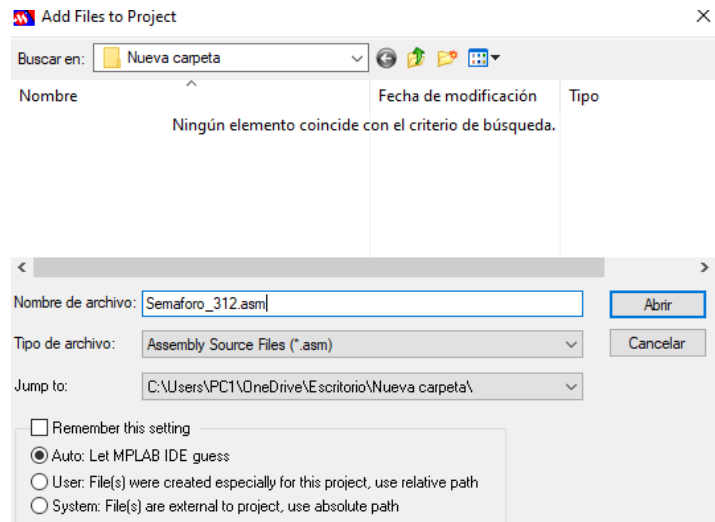
Daremos en Finalizar



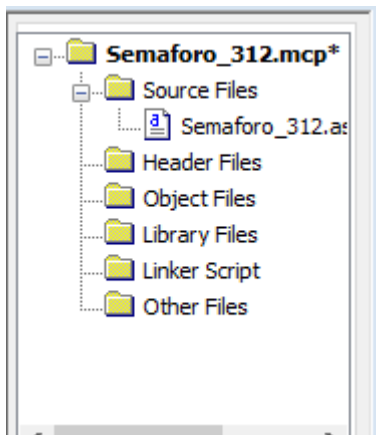
Damos clic derecho en “Source Files” y seleccionamos “Add Files”



Agregaremos un archivo con extensión asm y damos abrir



Se nos abra creado un archivo en “Source Files”



Damos doble clic para abrirlo y escribimos el siguiente código

```
;ZONA DE DATOS
PROCESSOR 16F84A
#include <P16F84A.INC>
__CONFIG _CP_OFF & _PWRTE_OFF & _WDT_ON & _XT_OSC ; 0x3FFD

; RAM-Variable
LRAM_0x10 EQU 0x10
LRAM_0x11 EQU 0x11

; Program

ORG 0x0000

; RESET-VECTOR
GOTO LADR_0x0005

ORG 0x0004|

; INTERRUPT-VECTOR
GOTO LADR 0x0018

LADR_0x0005
    MOVLW 0x87                ; b'10000111' d'135'
    BSF STATUS,RP0           ; Register-Bank(0/1)-Select
    MOVWF TMR0               ; TMR0 - OPTION_REG
    BCF STATUS,RP0           ; Register-Bank(0/1)-Select
    BSF STATUS,RP0           ; Register-Bank(0/1)-Select
    CLRF PORTB               ; PORTB - TRISB
    CLRF PORTA               ; PORTA - TRISA
    BCF STATUS,RP0           ; Register-Bank(0/1)-Select
    CLRF LRAM_0x10
    MOVLW 0x61               ; b'01100001' d'97' "a"
    MOVWF PORTB              ; PORTB - TRISB
    BCF PORTA,1              ; PORTA - TRISA
    BSF PORTA,0              ; PORTA - TRISA
    MOVLW 0x10               ; b'00010000' d'16'
    MOVWF TMR0               ; TMR0 - OPTION_REG
    MOVLW 0xA0               ; b'10100000' d'160'
    MOVWF INTCON
```

```
LADR_0x0016
  CLRWD
  GOTO LADR_0x0016

LADR_0x0018
  MOVLW 0x10          ; b'00010000' d'016'
  MOVWF TMR0          ; TMR0 - OPTION_REG
  INCF LRAM_0x11,F
  MOVLW 0x05          ; b'00000101' d'005'
  SUBWF LRAM_0x11,W
  BTFSS STATUS,Z
  GOTO LADR_0x0069
  CLRF LRAM_0x11
  INCF LRAM_0x10,F
  MOVLW 0x28          ; b'00101000' d'040' "("
  SUBWF LRAM_0x10,W
  BTFSC STATUS,C
  GOTO LADR_0x002A
  MOVLW 0x61          ; b'01100001' d'097' "a"
  MOVWF PORTB          ; PORTB - TRISB
  BCF PORTA,1          ; PORTA - TRISA
  BSF PORTA,0          ; PORTA - TRISA
  GOTO LADR_0x0069

LADR_0x002A
  MOVLW 0x32          ; b'00110010' d'050' "2"
  SUBWF LRAM_0x10,W
  BTFSC STATUS,C
  GOTO LADR_0x0037
  MOVLW 0x61          ; b'01100001' d'097' "a"
  MOVWF PORTB          ; PORTB - TRISB
  BTFSC LRAM_0x10,0
  GOTO LADR_0x0069
  MOVLW 0x02          ; b'00000010' d'002'
  XORWF PORTA,F        ; PORTA - TRISA
  MOVLW 0x60          ; b'01100000' d'096' "`"
  MOVWF PORTB          ; PORTB - TRISB
  GOTO LADR_0x0069
```



```

LADR_0x0037
    MOVLW 0x3C          ; b'00111100' d'060' "<"
    SUBWF LRAM_0x10,W
    BTFSC STATUS,C
    GOTO LADR_0x0042
    MOVLW 0x62          ; b'01100010' d'098' "b"
    MOVWF PORTB         ; PORTB - TRISB
    BTFSC LRAM_0x10,0
    GOTO LADR_0x0069
    MOVLW 0x02          ; b'00000010' d'002'
    XORWF PORTA,F       ; PORTA - TRISA
    GOTO LADR_0x0069

LADR_0x0042
    MOVLW 0x64          ; b'01100100' d'100' "d"
    SUBWF LRAM_0x10,W
    BTFSC STATUS,C
    GOTO LADR_0x004B
    MOVLW 0x8C          ; b'10001100' d'140'
    MOVWF PORTB         ; PORTB - TRISB
    BSF PORTA,1         ; PORTA - TRISA
    BCF PORTA,0         ; PORTA - TRISA
    GOTO LADR_0x0069

LADR_0x004B
    MOVLW 0x6E          ; b'01101110' d'110' "n"
    SUBWF LRAM_0x10,W
    BTFSC STATUS,C
    GOTO LADR_0x0058
    MOVLW 0x8C          ; b'10001100' d'140'
    MOVWF PORTB         ; PORTB - TRISB
    BTFSS LRAM_0x10,0
    GOTO LADR_0x0069
    MOVLW 0x01          ; b'00000001' d'001'
    XORWF PORTA,F       ; PORTA - TRISA
    MOVLW 0x84          ; b'10000100' d'132'
    MOVWF PORTB         ; PORTB - TRISB
    GOTO LADR_0x0069

```



```
LADR_0x0058
    MOVLW 0x78          ; b'01111000' d'120' "x"
    SUBWF LRAM_0x10,W
    BTFSC STATUS,C
    GOTO LADR_0x0064
    MOVLW 0x94          ; b'10010100' d'148'
    MOVWF PORTB         ; PORTB - TRISB
    BTFSS LRAM_0x10,0
    BSF INDF,0
    BTFSC LRAM_0x10,0
    GOTO LADR_0x0069
    MOVLW 0x01          ; b'00000001' d'001'
    GOTO LADR_0x0069

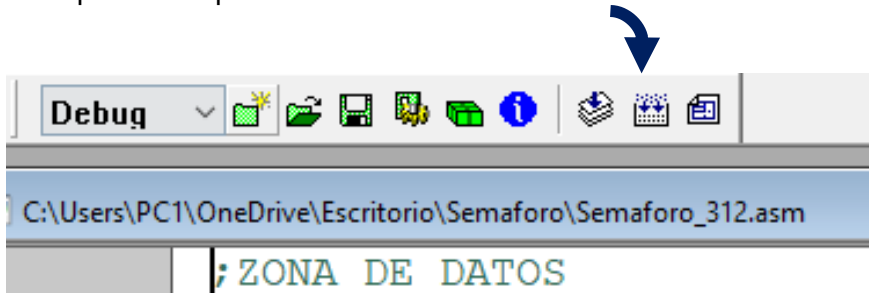
LADR_0x0064
    MOVLW 0x61          ; b'01100001' d'097' "a"
    MOVWF PORTB         ; PORTB - TRISB
    BCF PORTA,1         ; PORTA - TRISA
    BSF PORTA,0         ; PORTA - TRISA
    CLRF LRAM_0x10

LADR_0x0069
    MOVLW 0xA0          ; b'10100000' d'160'
    MOVWF INTCON
    RETFIE

End
```

El código estaría acabado

Para poder compilarlo damos clic en este botón



Mandara un mensaje donde se compilo correctamente y abra generado un archivo .hex

```

Output
Build Version Control Find in Files
Debug build of project 'C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semafaro.mcp' started.
Language tool versions: MPASMWIN.exe v5.37, mmlink.exe v4.37, mplib.exe v4.37
Preprocessor symbol: '_DEBUG' is defined.
Sat Jan 11 10:53:42 2020

Clean: Deleting intermediary and output files.
Clean: Deleted file 'C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semaforo_312.o'.
Clean: Deleted file 'C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semaforo_312.enr'.
Clean: Deleted file 'C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semaforo_312.lst'.
Clean: Done.
Executing: 'C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe' /q /p16F84A "Semaforo_312.asm" /I"Semaforo_312.lst" /e"Semaforo_312.enr" /o"Semaforo_312.o" /d__DEBUG=1
Executing: 'C:\Program Files (x86)\Microchip\MPASM Suite\mmlink.exe' /p16F84A "Semaforo_312.o" /u__DEBUG /z__MPLAB_BUILD=1 /z__MPLAB_DEBUG=1 /o"Semaforo.cof" /M"Semaforo.map" /W
MPLINK 4.37. Linker
Copyright (c) 1998-2010 Microchip Technology Inc.
Errors : 0

MP2HEX 4.37. COFF to HEX File Converter
Copyright (c) 1998-2010 Microchip Technology Inc.
Errors : 0

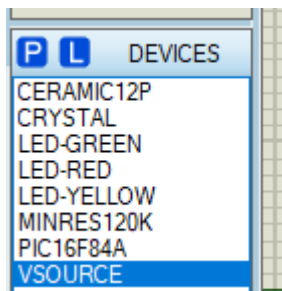
Loaded C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semaforo.cof.

Debug build of project 'C:\Users\PC1\OneDrive\Escritorio\Semafaro\Semaforo.mcp' succeeded.
Language tool versions: MPASMWIN.exe v5.37, mmlink.exe v4.37, mplib.exe v4.37
Preprocessor symbol: '_DEBUG' is defined.
Sat Jan 11 10:53:47 2020

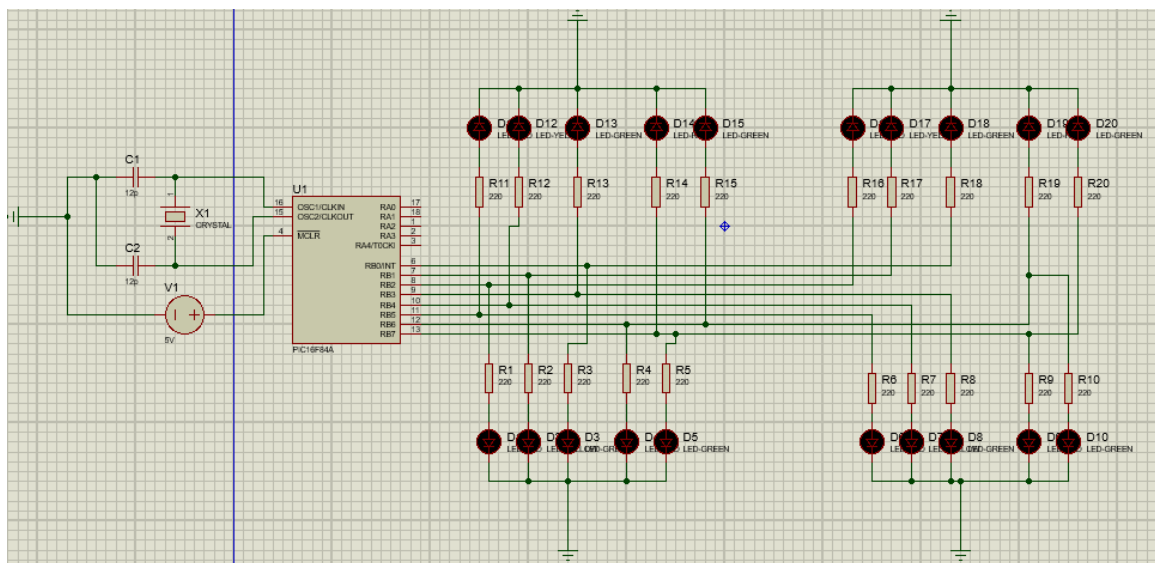
BUILD SUCCEEDED
  
```

Pasamos a Proteus para representa nuestro circuito

Primero agregamos algunas cosas que vamos a necesitar, así que le damos al boto “P” los buscamos y lo agregamos

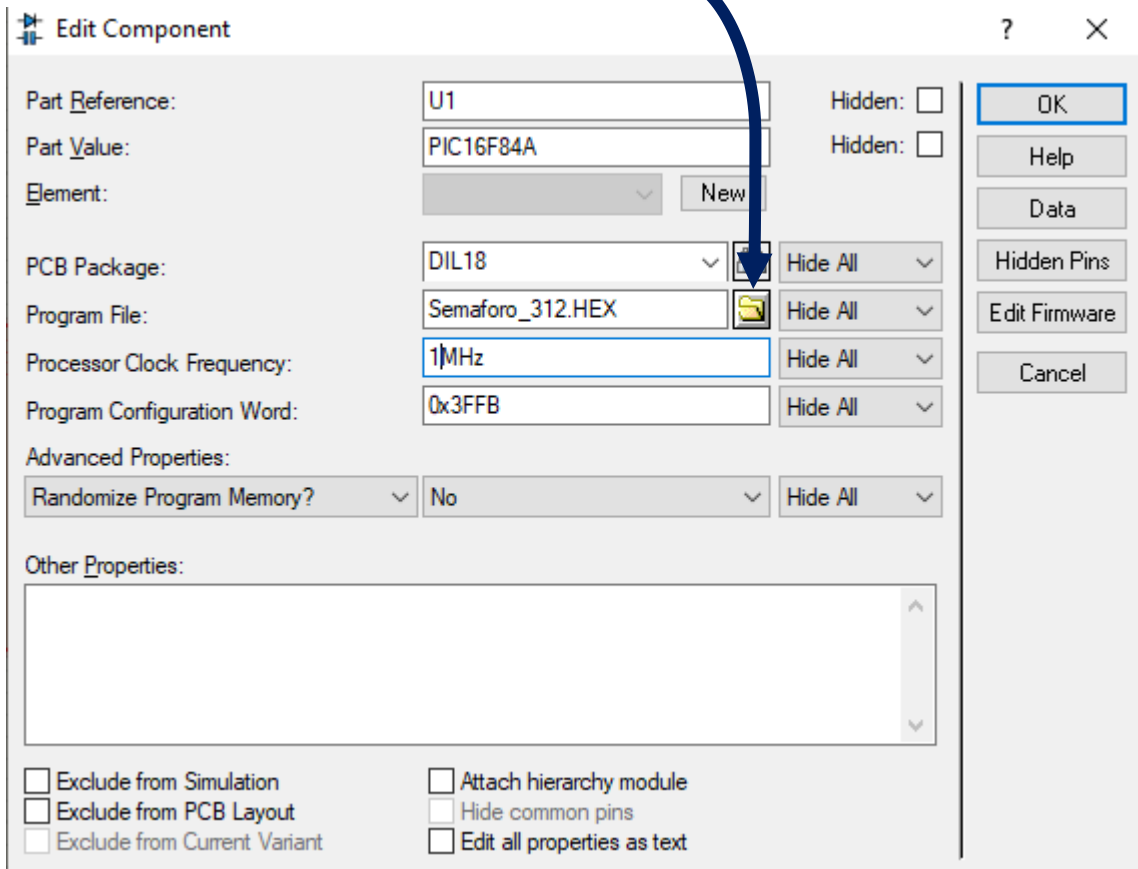


Teniendo esto, pasamos a realizar el circuito de la siguiente manera



En las resistencias hice un cambio; son de 120K las cambié a 220

Damos doble clic en nuestro PIC y agregamos el archivo .hex que generamos anteriormente y damos ok

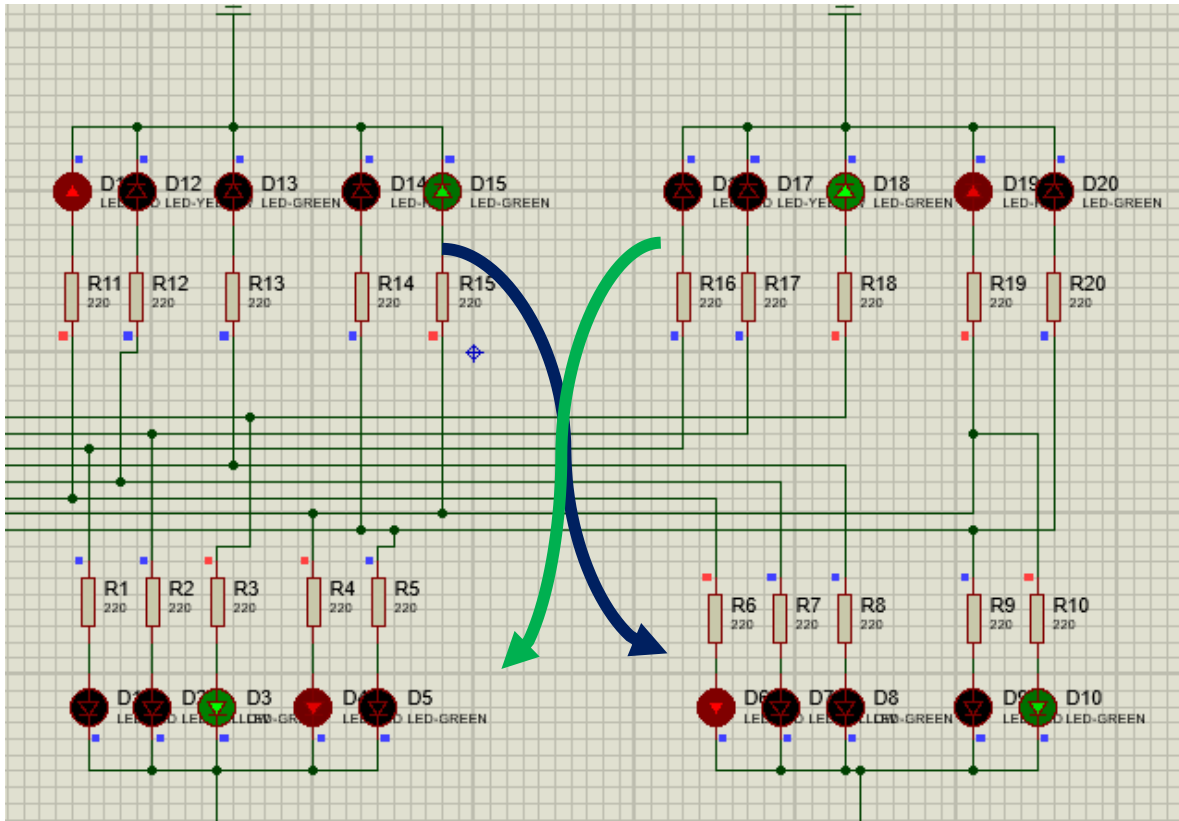


En la esquina inferior la damos a botón de play para verificar que funciona correctamente

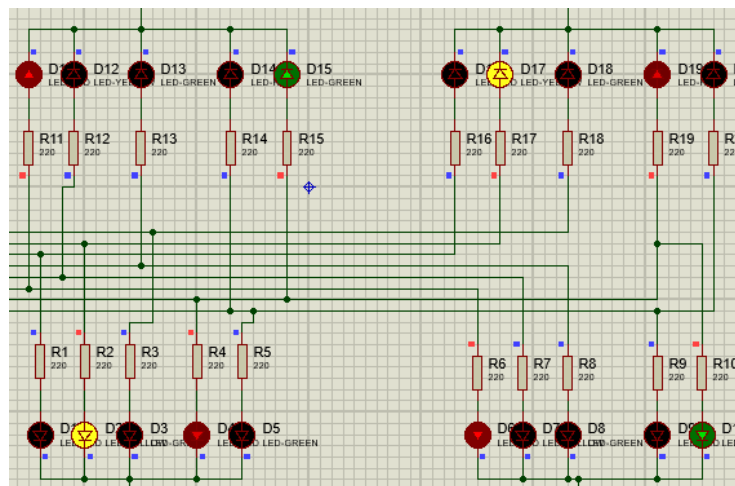


Se abran encendido algunos leds y tomamos como referencia los primeros dos de arriba, donde el primero es el Semáforo 1 y el segundo el Semáforo 2, y los de abajo el primero el Semáforo 4 y el segundo el Semáforo 3.

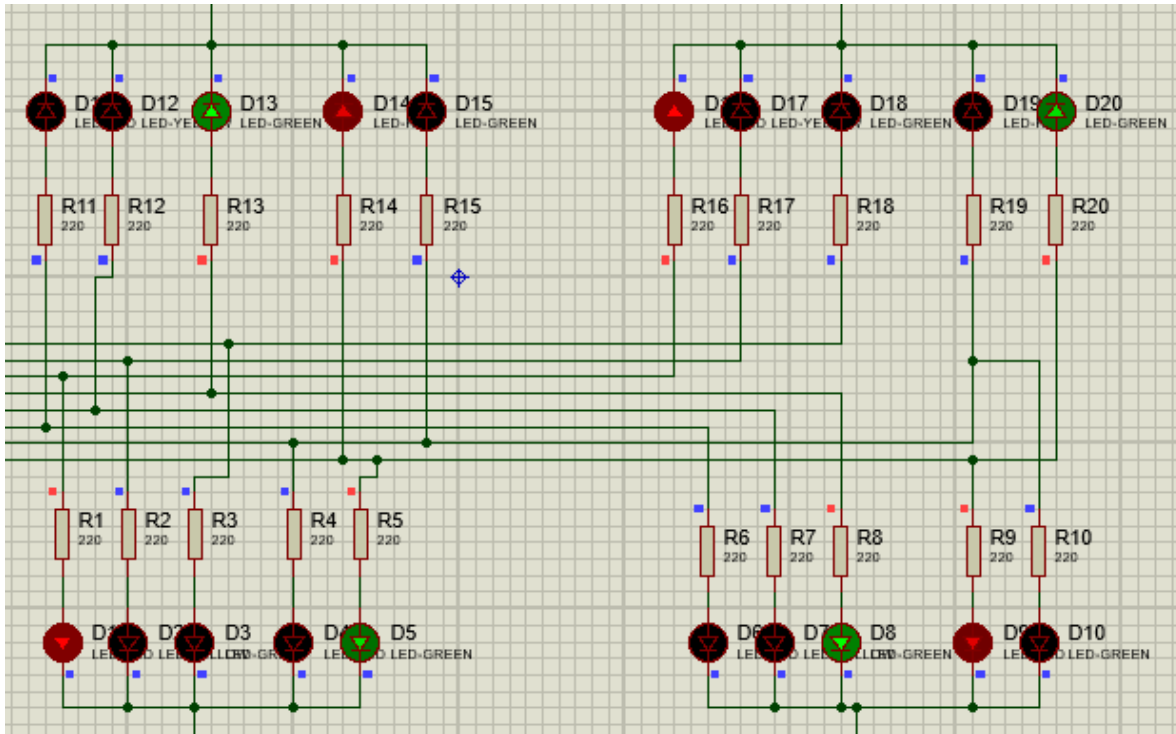
Deben de ir coordinados en “X”



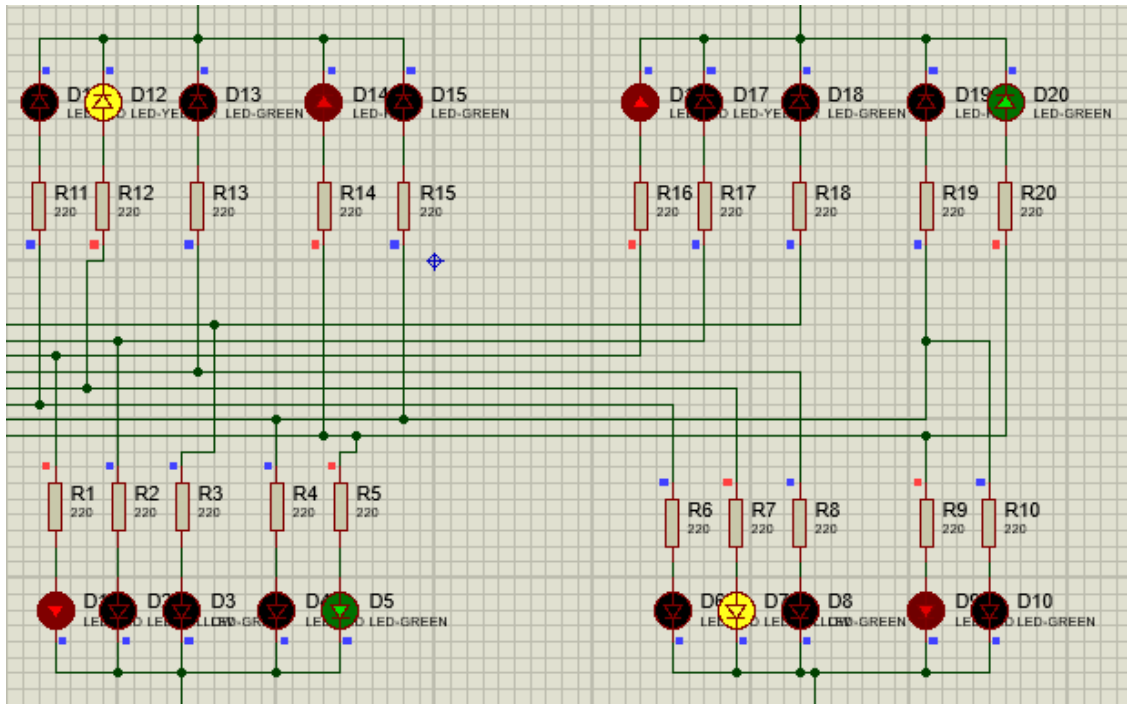
Después de un rato vemos que el verde parpadea y cambia a Amarillo



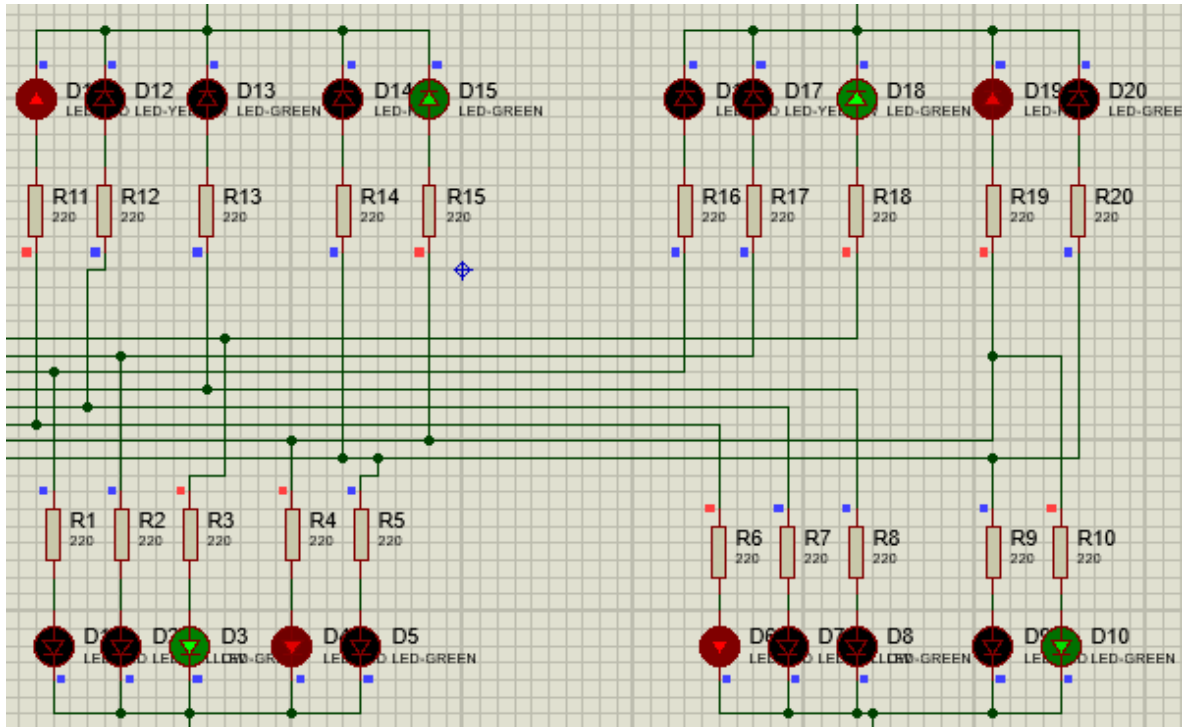
Y en seguida cambian todos



Nuevamente después de un momento parpadean los verdes y cambian a Amarillo



Después cambian todos y quedan como la primera vez.



Lo dejamos varios minutos y hacia el mismo proceso, así que concluimos que funciona correctamente.

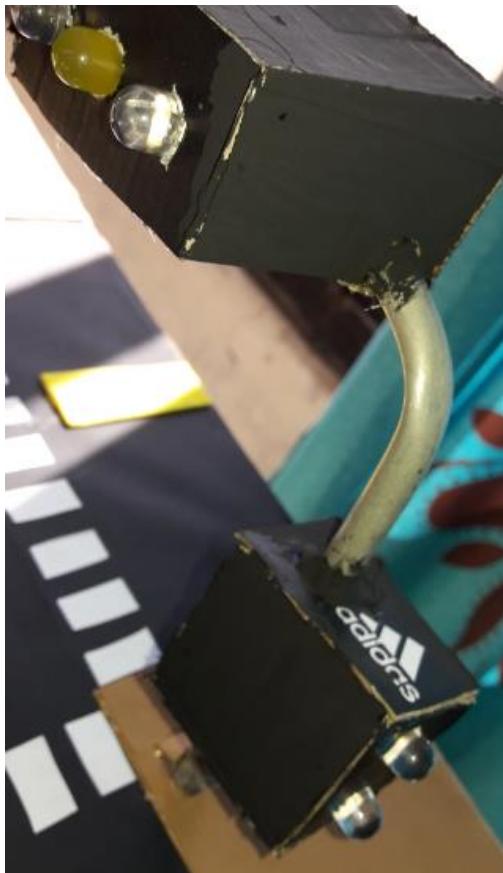
Pasamos a realizar la maqueta

Así que soldamos los leds con las resistencias de 220 y al cable

Verdes	8
Rojos	8
Amarillos	4



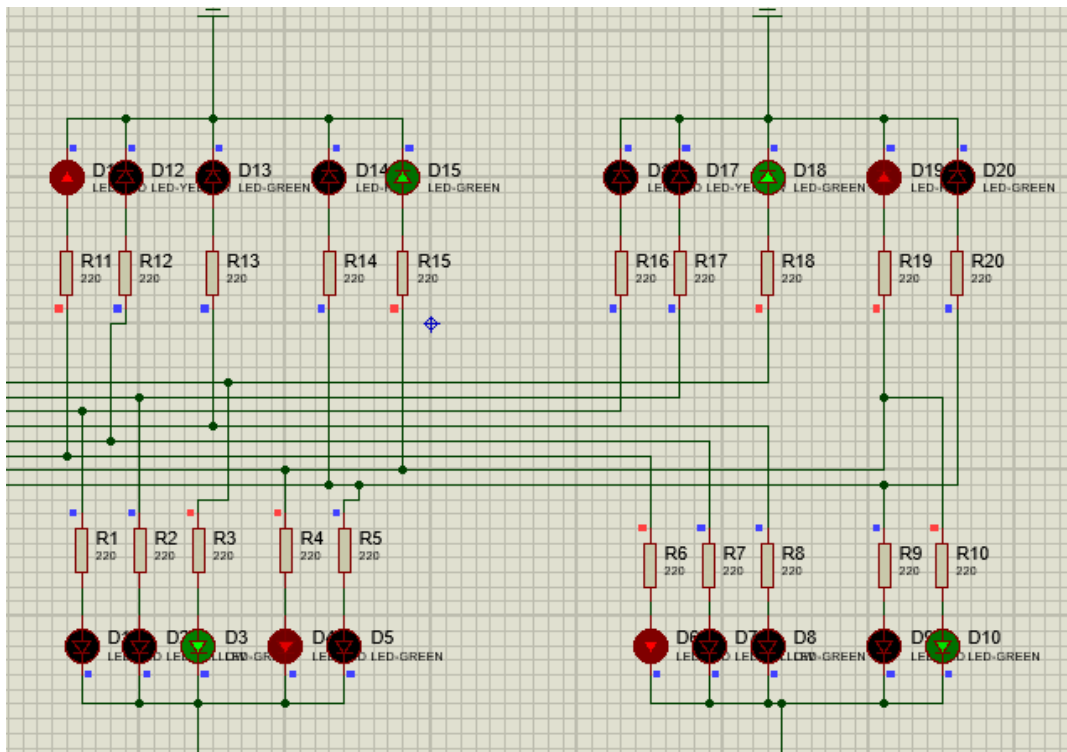
Después metemos el cable por un tubo de metal dejando los leds afuera y le construimos una casita pagándola con silicón y después de esta la pintamos, y esto lo hacemos con los 4 Semáforos



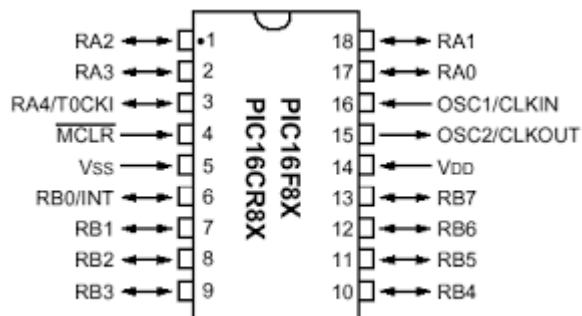
Agujaramos una tabla de madera para poder meter el tubo y los pegamos, después colocamos hojas negras para simular la carretera y fomi para decorar

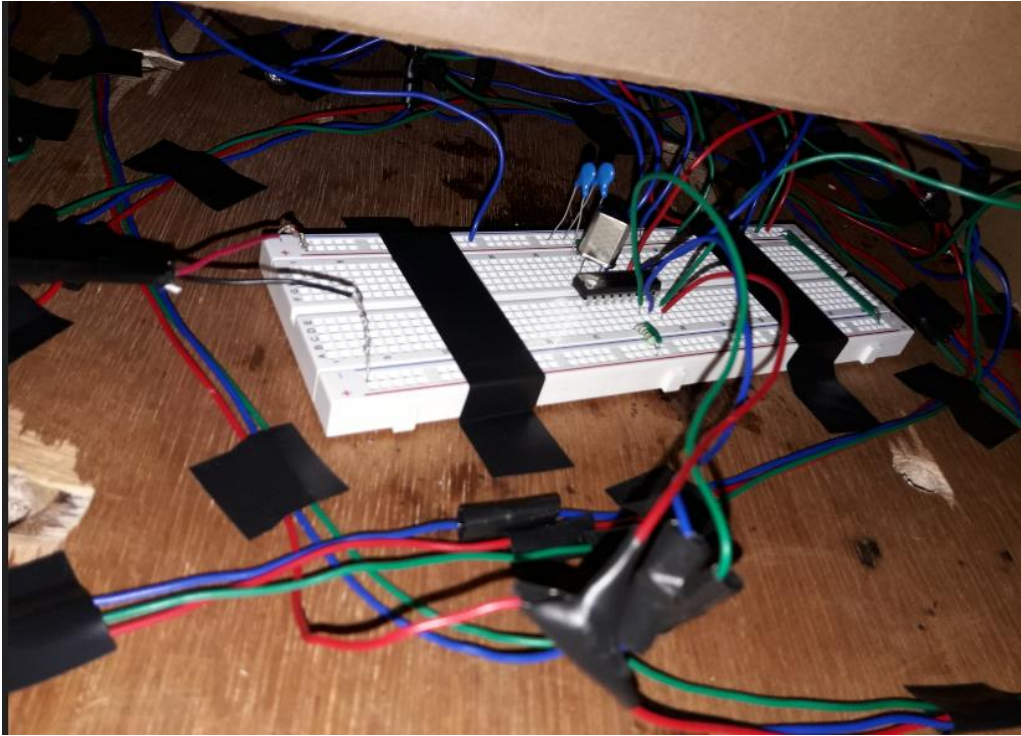


Con los cables que están en la parte de abajo los soldamos como las uniones que muestra Proteus



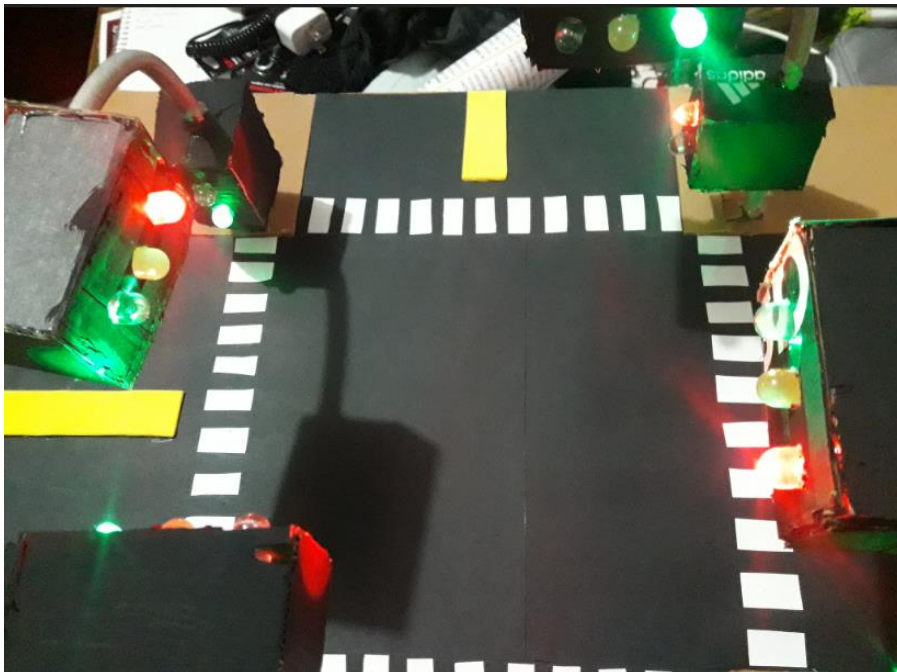
Después de esto colocamos nuestro PIC en la Protoboard y vamos colocando los cables correspondientes en cada punto. Para esto me guie con el Data Sheet del propio PIC y el Proteus

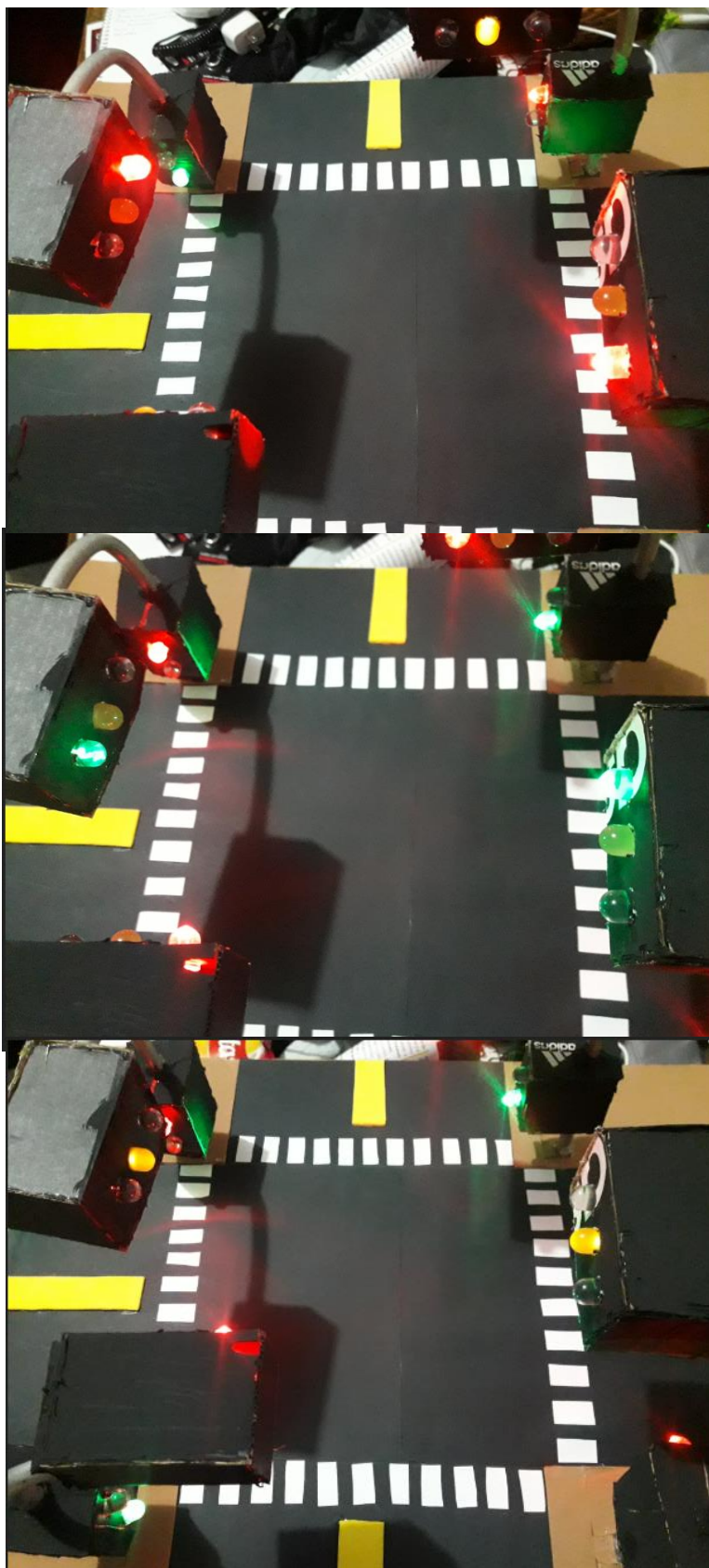




Por último, esta parte la cubrimos con una caja para que al momento de voltearla y ponerla en una superficie no se desacomode y para cubrir los cables, solo dejando el que va a conectarlo a corriente

Después lo probamos y comprobamos que funciona







Al momento de dar el cambio, el amarillo no se aprecia demasiado porque estos LEDS son distintos a los demás, ¡¡¡pero Sí Prenden!!!

Conclusión

El semáforo es algo laborioso, pero es interesante ya que aprendes demasiado, empezando por el lenguaje que se ocupó (ensamblador), para mí es un lenguaje complicado de entender, pero es muy eficiente. Además, hacer uso correcto del PIC para que funcione y poder realizar las tareas o acciones que debe de ejecutar.