

Nombre de la práctica	Semáforo			No.	2
Asignatura:	Arquitectura de computadoras	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	8 horas

NOMBRE DEL ALUMNO: Daniel García García

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

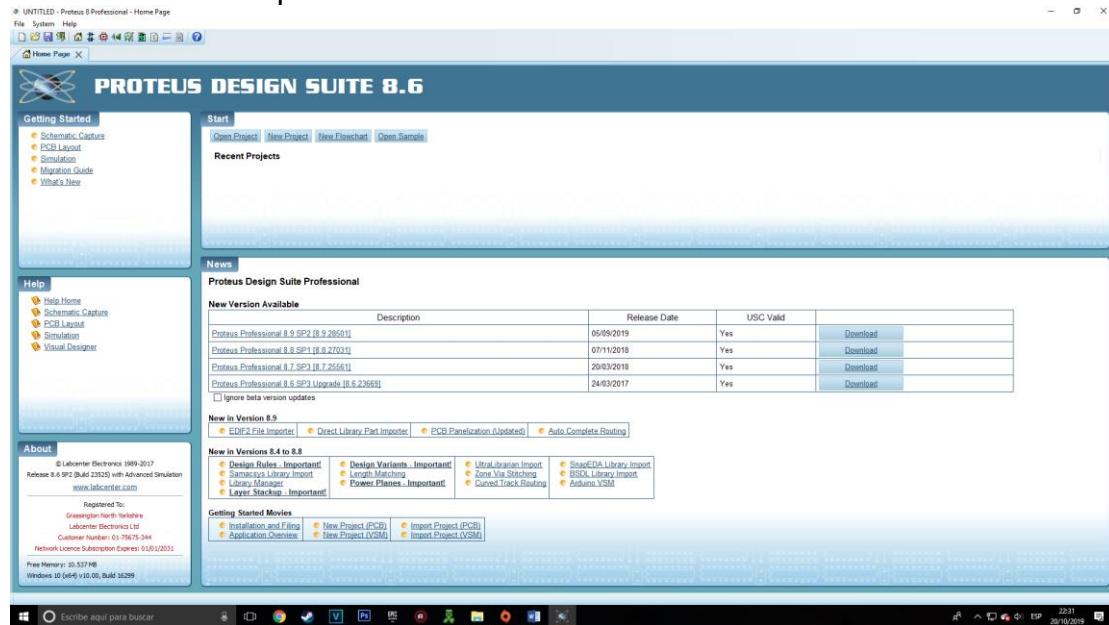
III. Material empleado: Laptop, protoboard, cables, PIC16f84A, cátodo común de 7 segmentos, 2 capacitores cerámicos de 22 picofaradios, resistencia 5 volts, cristal oscilador 5 picofaradios, luces led de colores verde, ámbar y rojo.

IV. Desarrollo de la práctica:

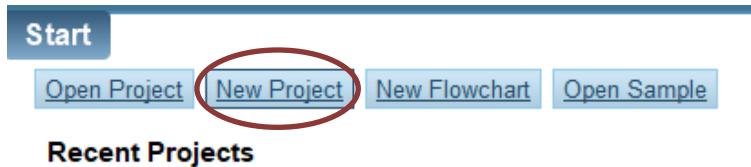
1.- ¿Qué vamos a hacer?

Primero que nada, vamos a programar en **Proteus** el programa que procederemos a incorporar en nuestro PIC, además que nos ayudara a estructurar el sistema que incorporaremos en nuestra protoboard, para esto debemos tener instalado proteus en su versión 8.2 o superior.

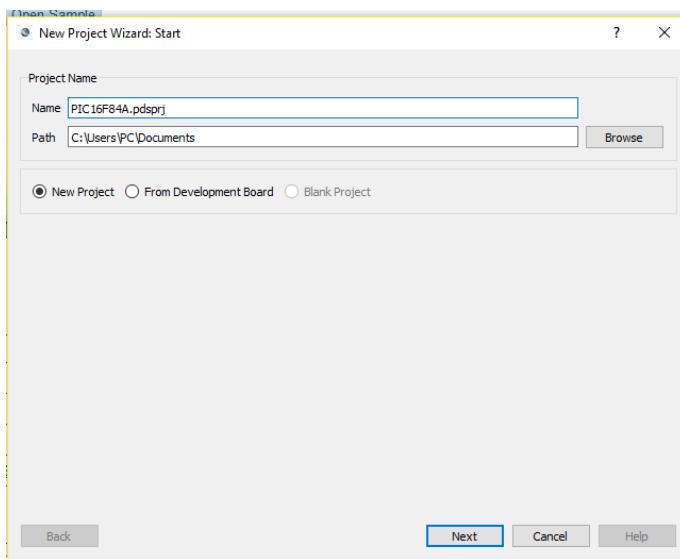
Paso 1.- Abrimos proteus



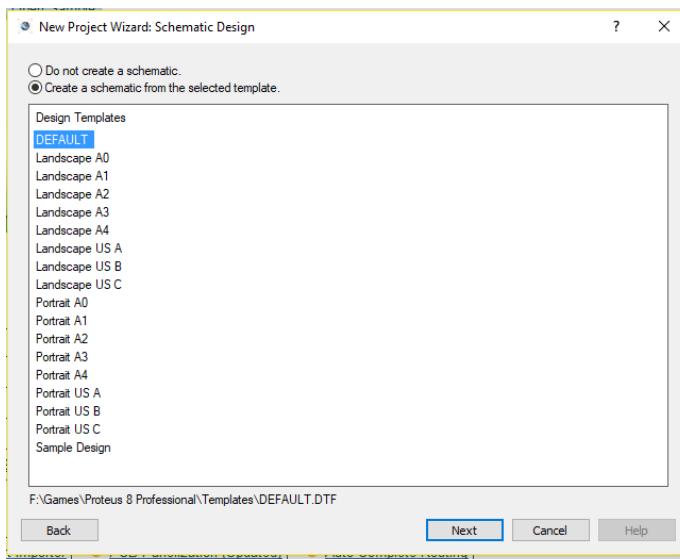
Paso 2.- Damos click en New Project



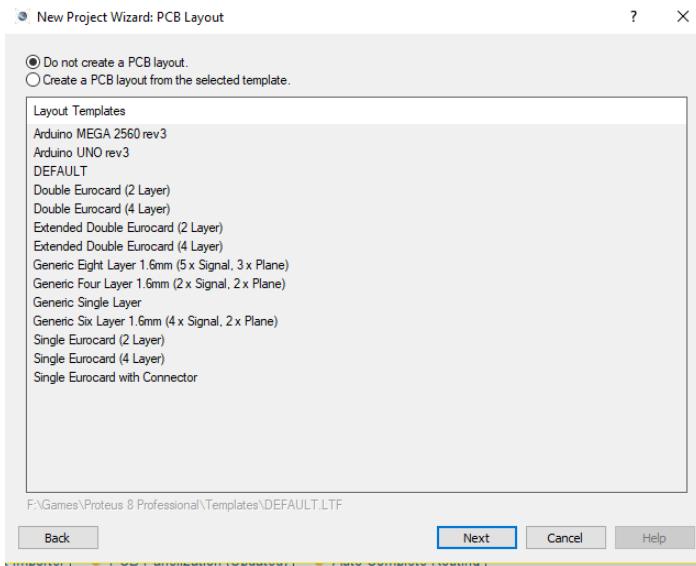
Paso 3.- Configuramos el nombre del archivo como PIC16F84A y damos clic en next



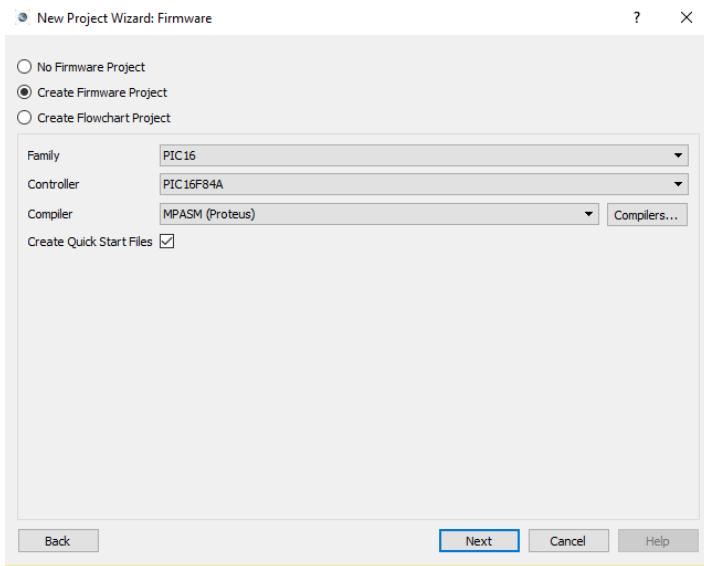
Paso 4.- En esta opción lo dejamos en DEFAULT y damos next



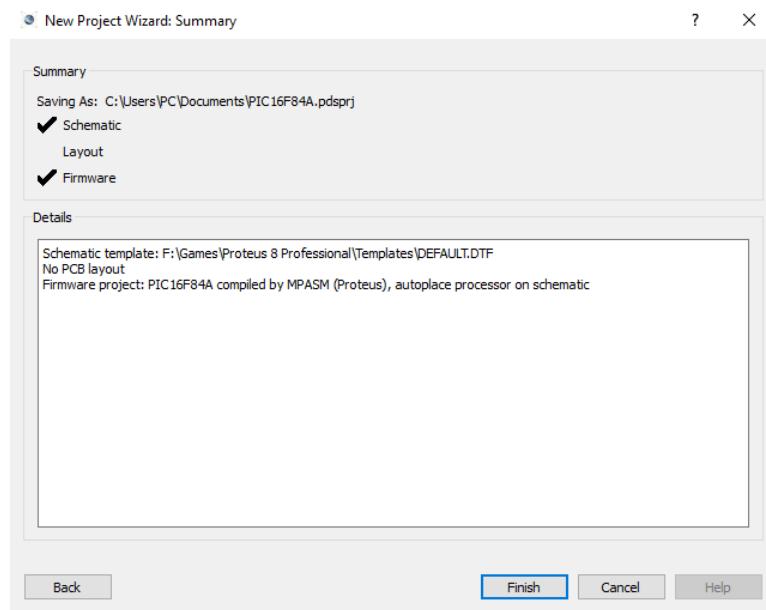
Paso 5.- Lo dejamos en Do not y damos next



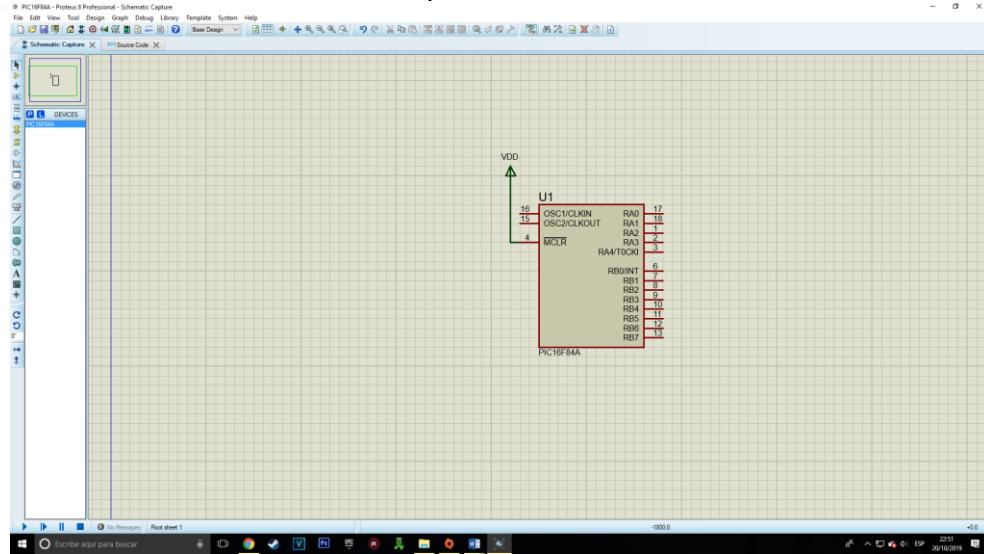
Paso 6.- Damos clic en create firmware project y seleccionamos las siguientes especificaciones:



Paso 7.- Luego de eso nos mostrara una pantalla con lo que hemos seleccionado hasta el momento, damos clic en finalizar.



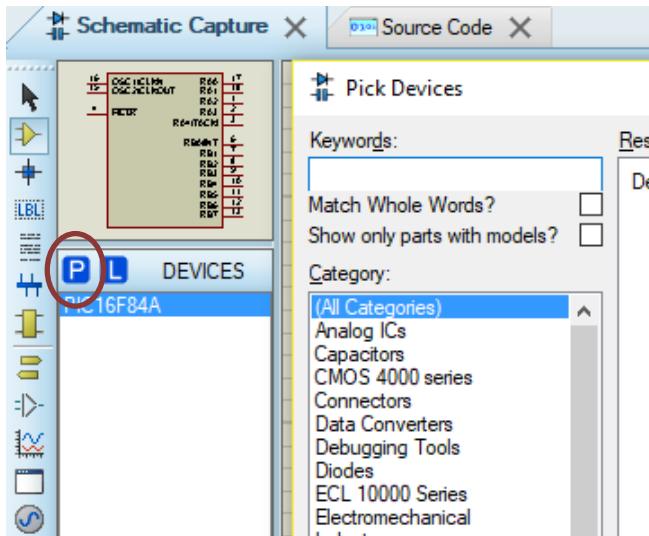
Paso 8.- Tendremos esto en pantalla



Paso 9.- Comenzaremos a añadir todos los circuitos para que nuestro PIC funcione

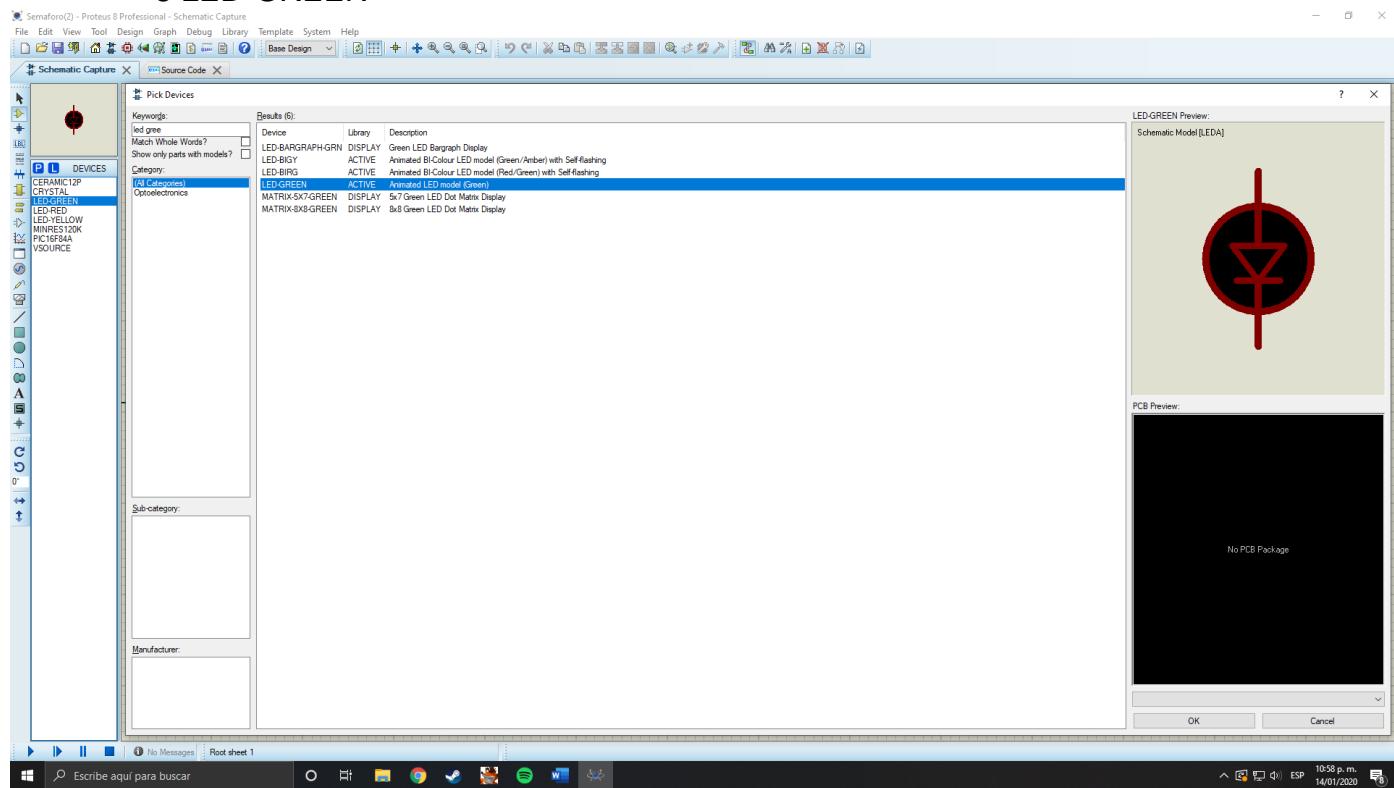


9.1.-Para esto damos clic en la letra P



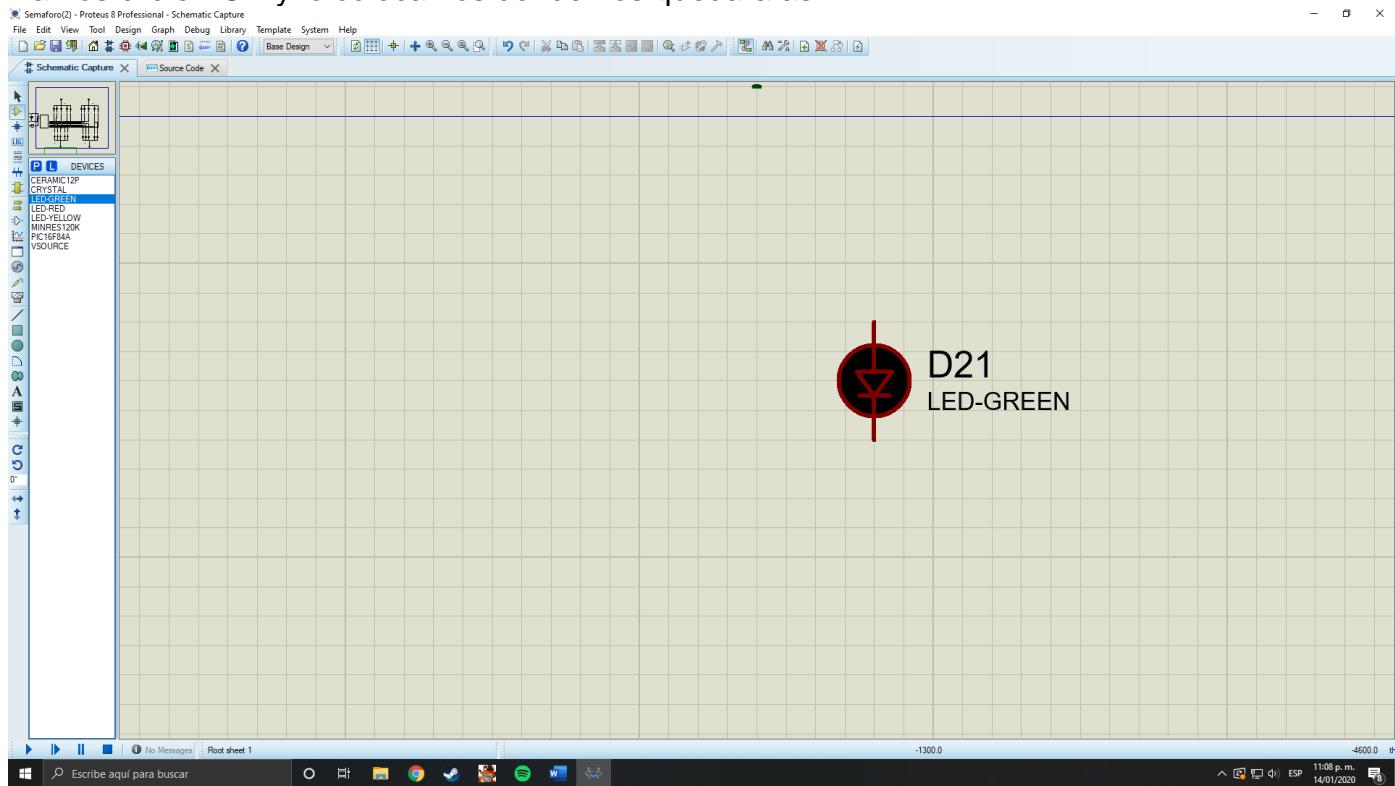
Y en el menú Keywords buscaremos los siguientes circuitos y los añadiremos.

• 8 LED-GREEN



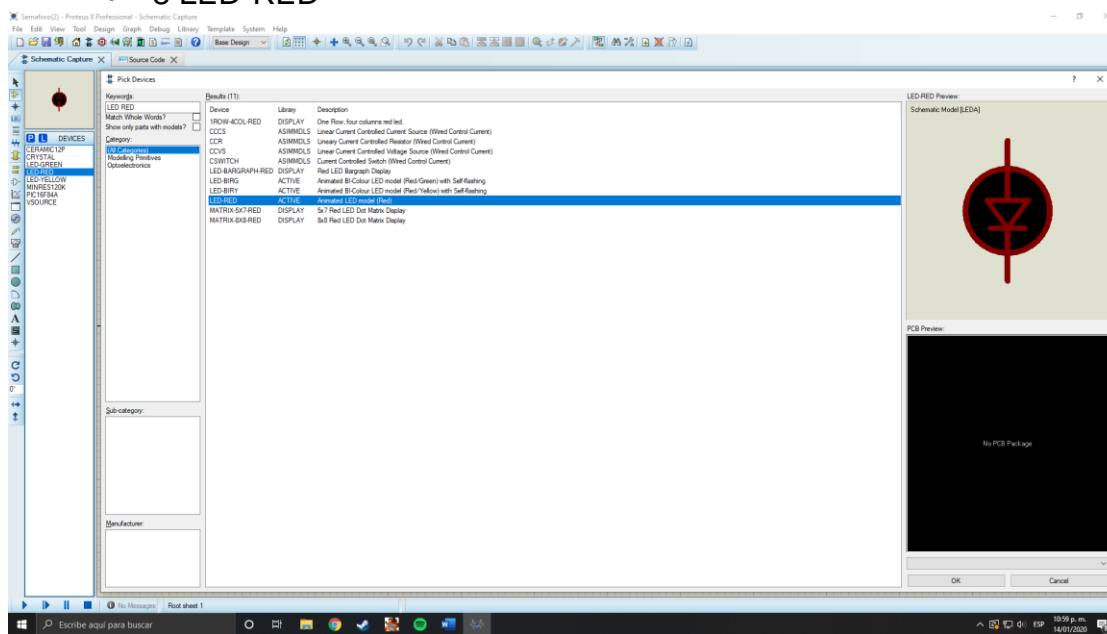


Damos clic en **OK** y lo colocamos donde nos quedara así:



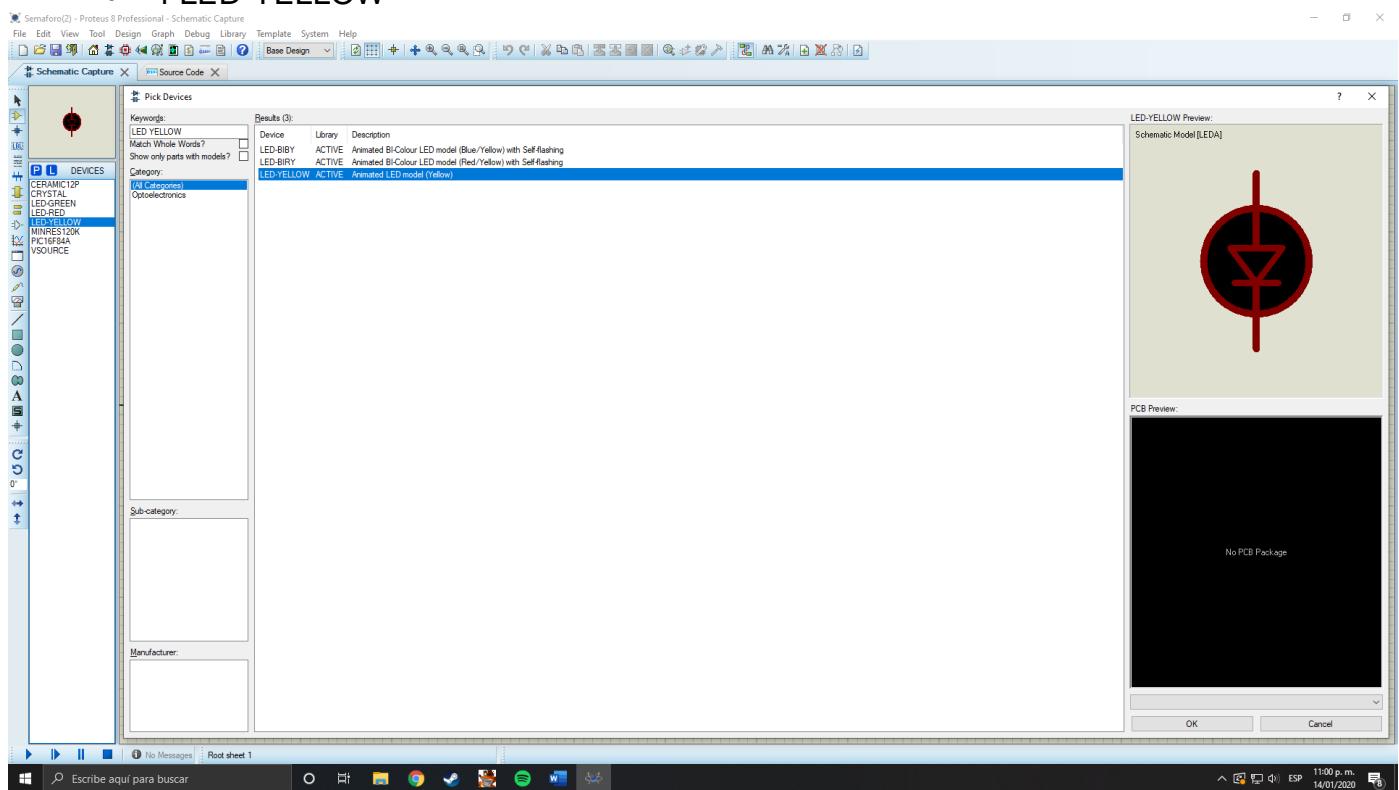
Hacemos lo mismo con los siguientes circuitos:

- 8 LED-RED



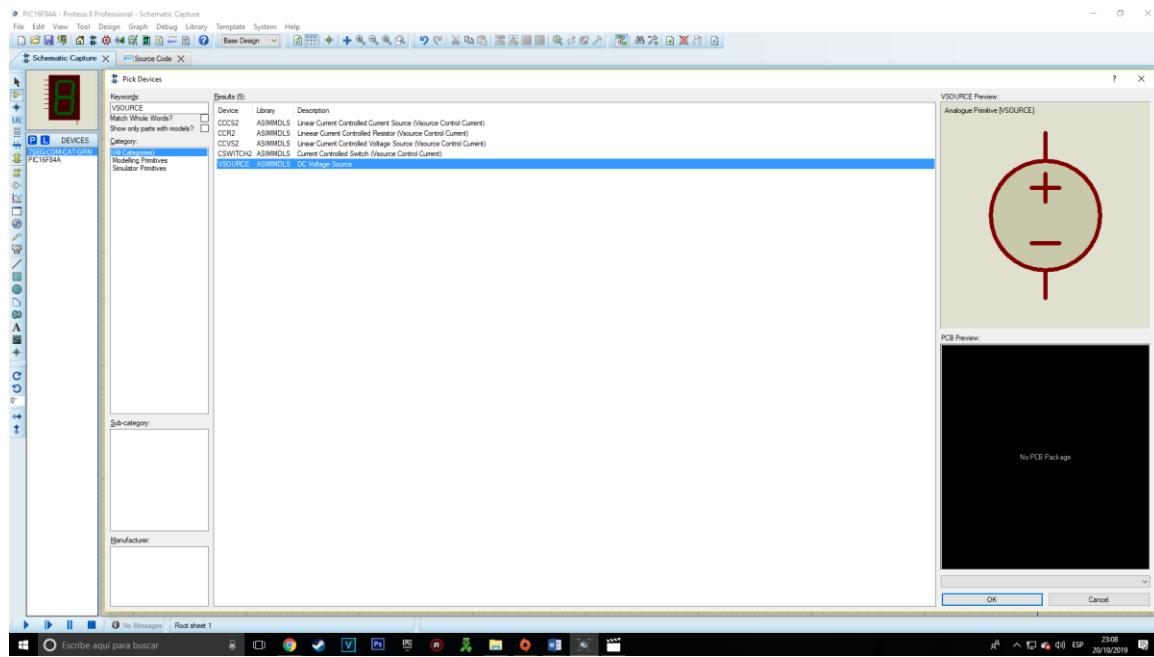


• 4 LED-YELLOW

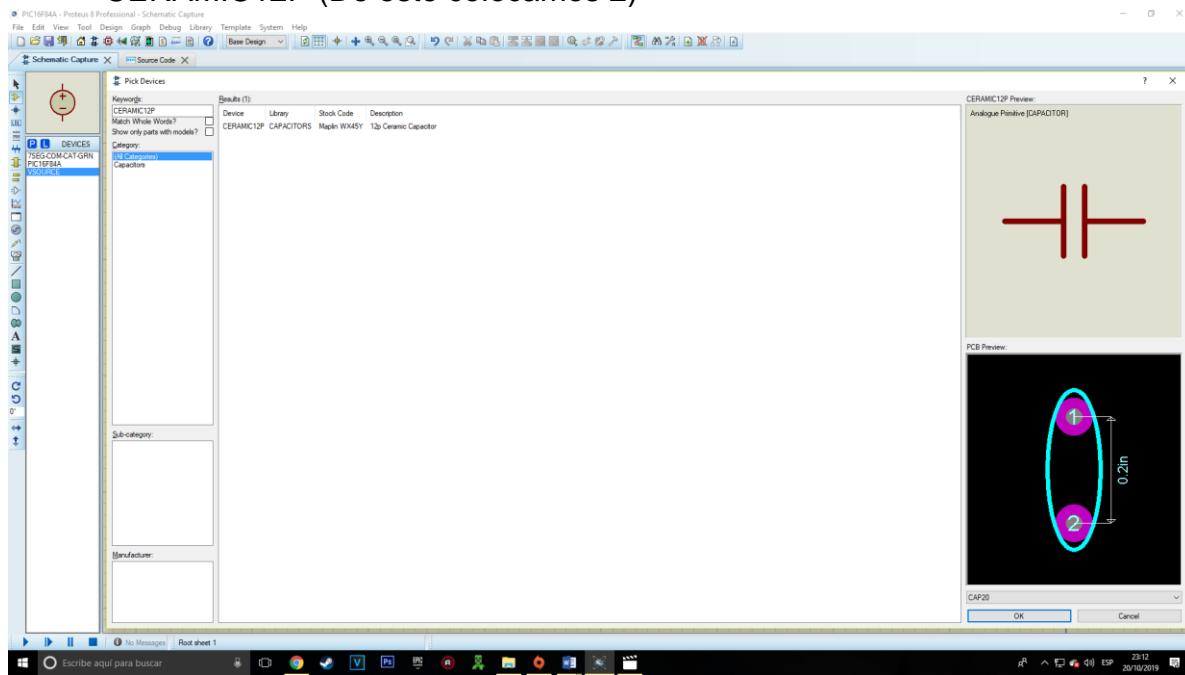




• VSOURCE

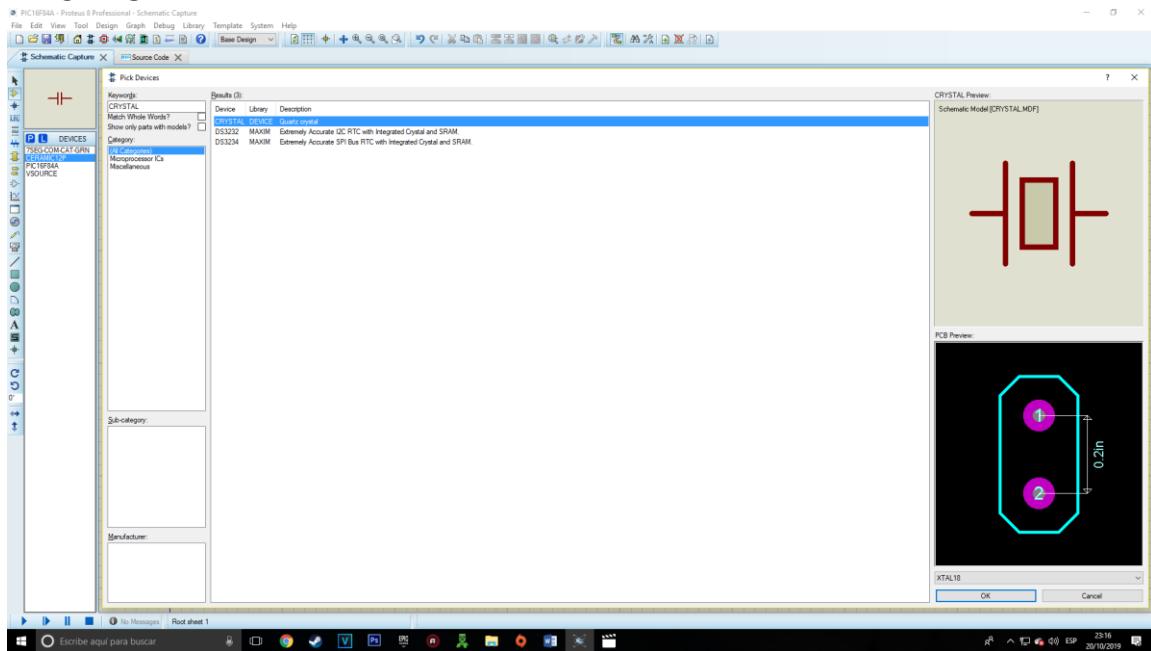


• CERAMIC12P (De este colocamos 2)

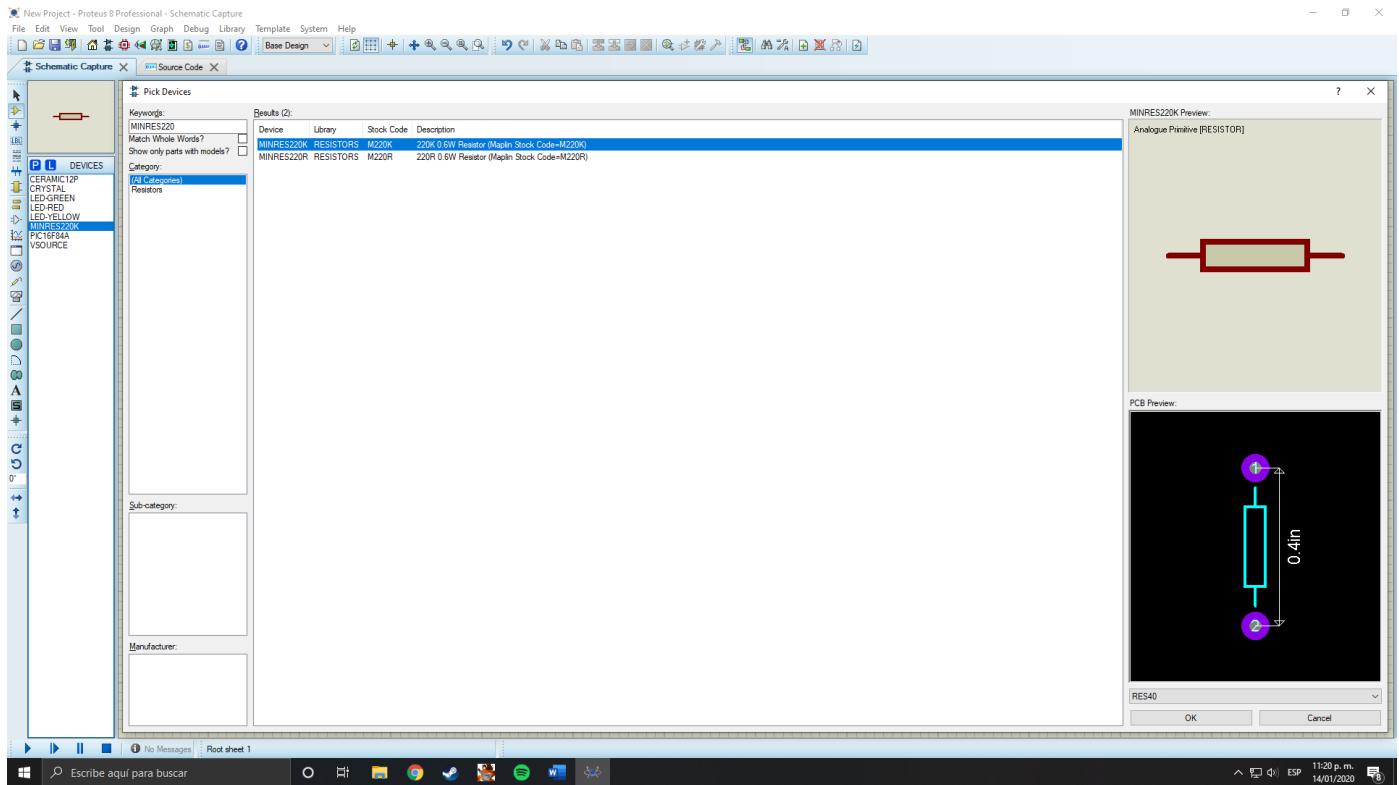




● CRYSTAL

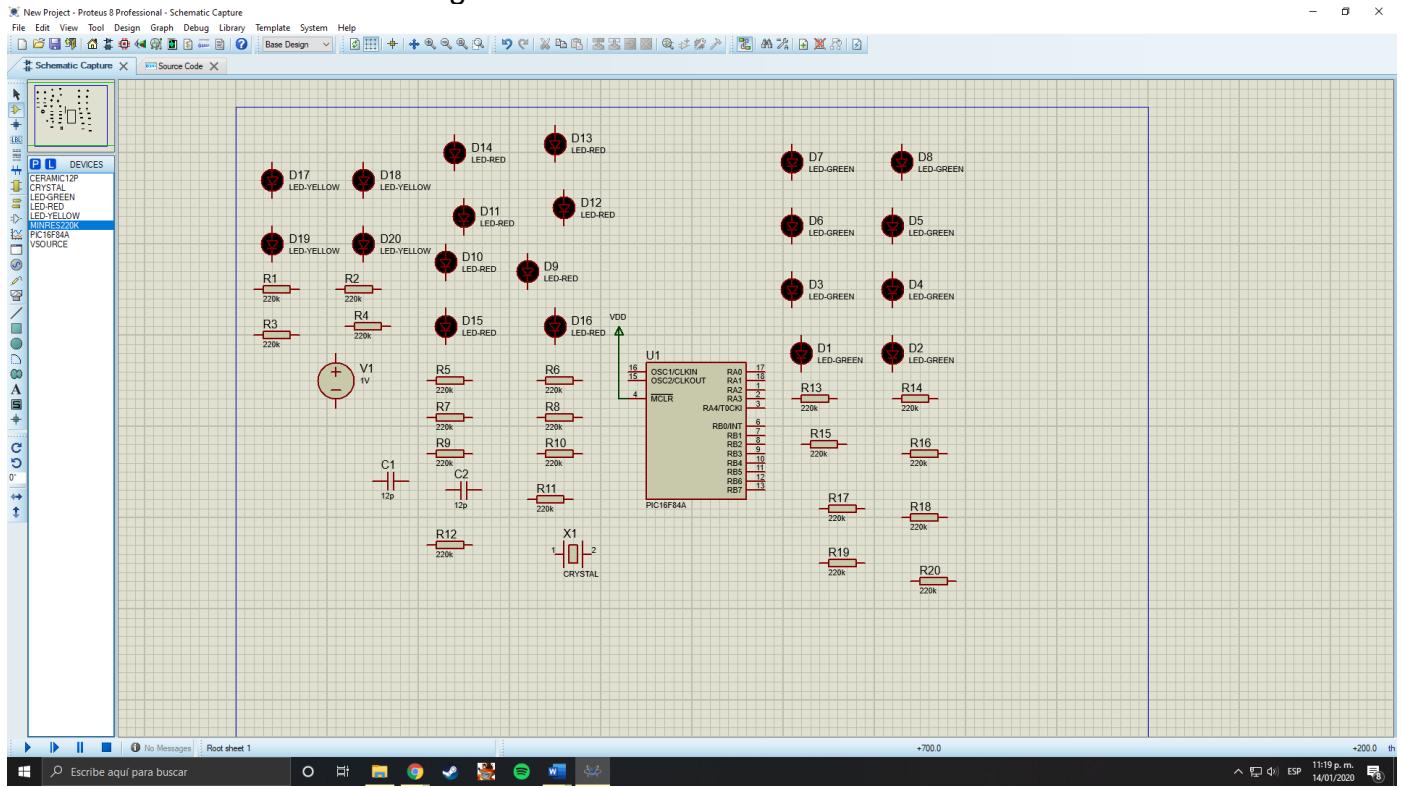


● 20 MINRES220K





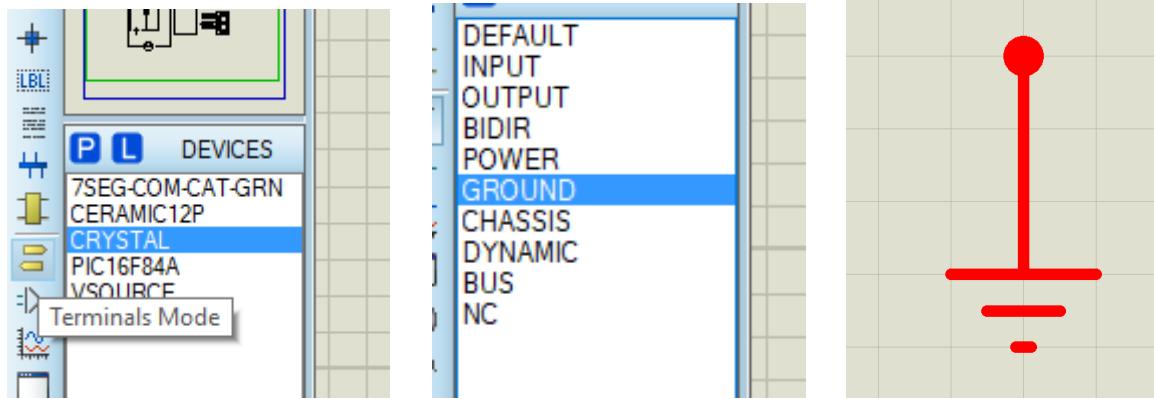
10.- Hasta ahora tendríamos algo así:



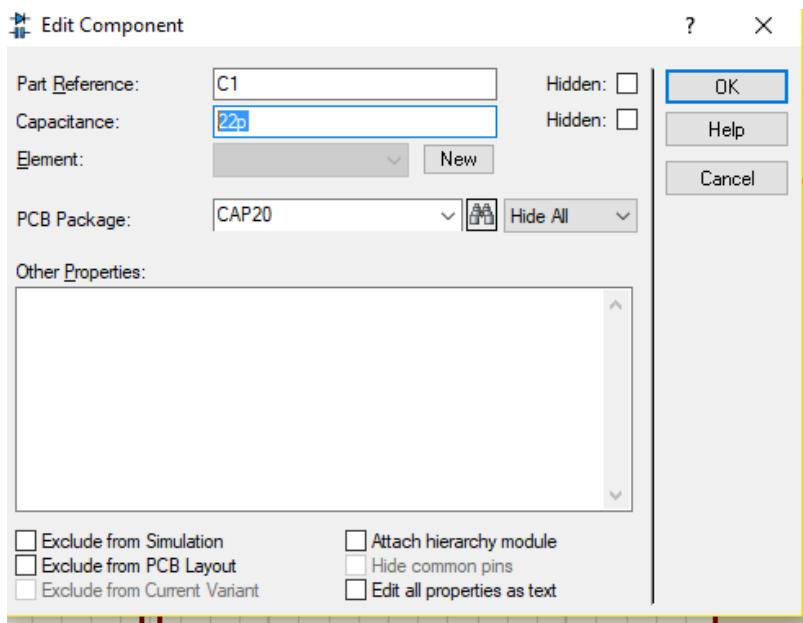
11.- Lo único que faltaría es la **tierra**, para colocarla hacemos lo siguiente:

1.-Click en Terminals Mode

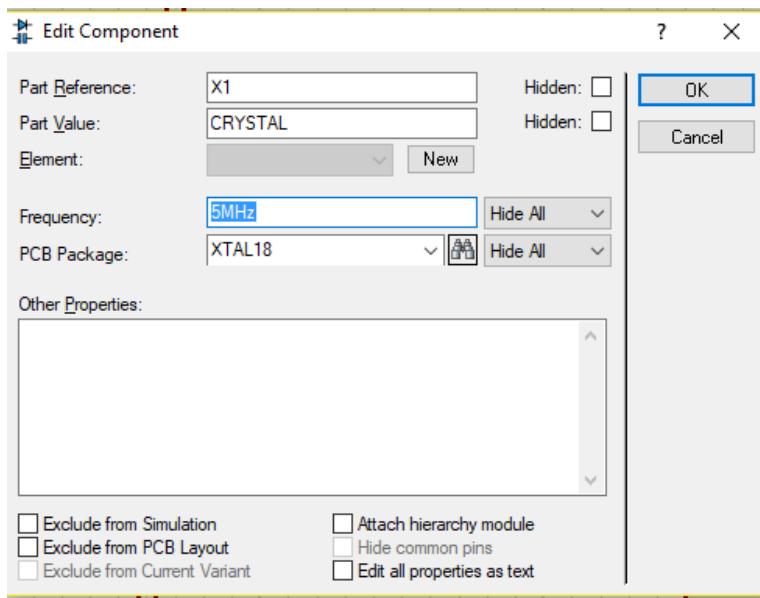
2.- Click en GROUND



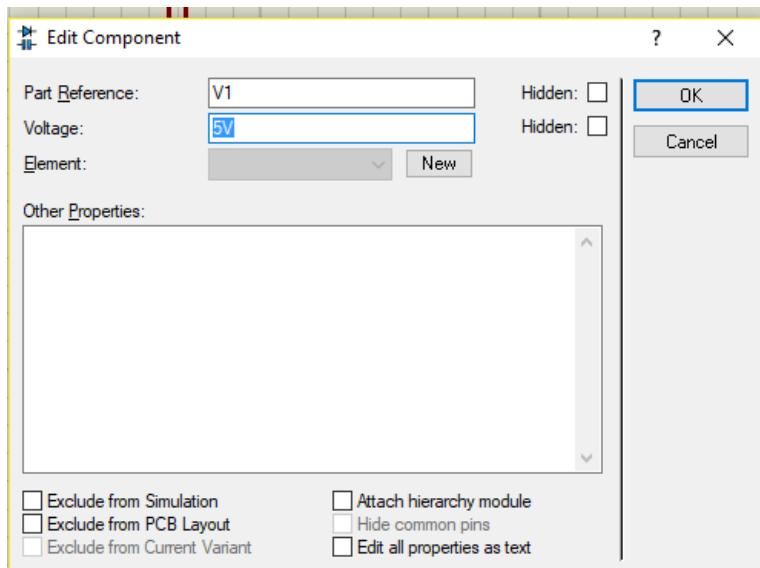
12.- Tendremos que cambiar los valores internos de nuestros CERAMIC12P, nuestro CRYSTAL y VSOURCE. Para los primeros damos doble clic y modificamos su capacidad de 12 picofaradios (12p) a 22 picofaradios (22p):



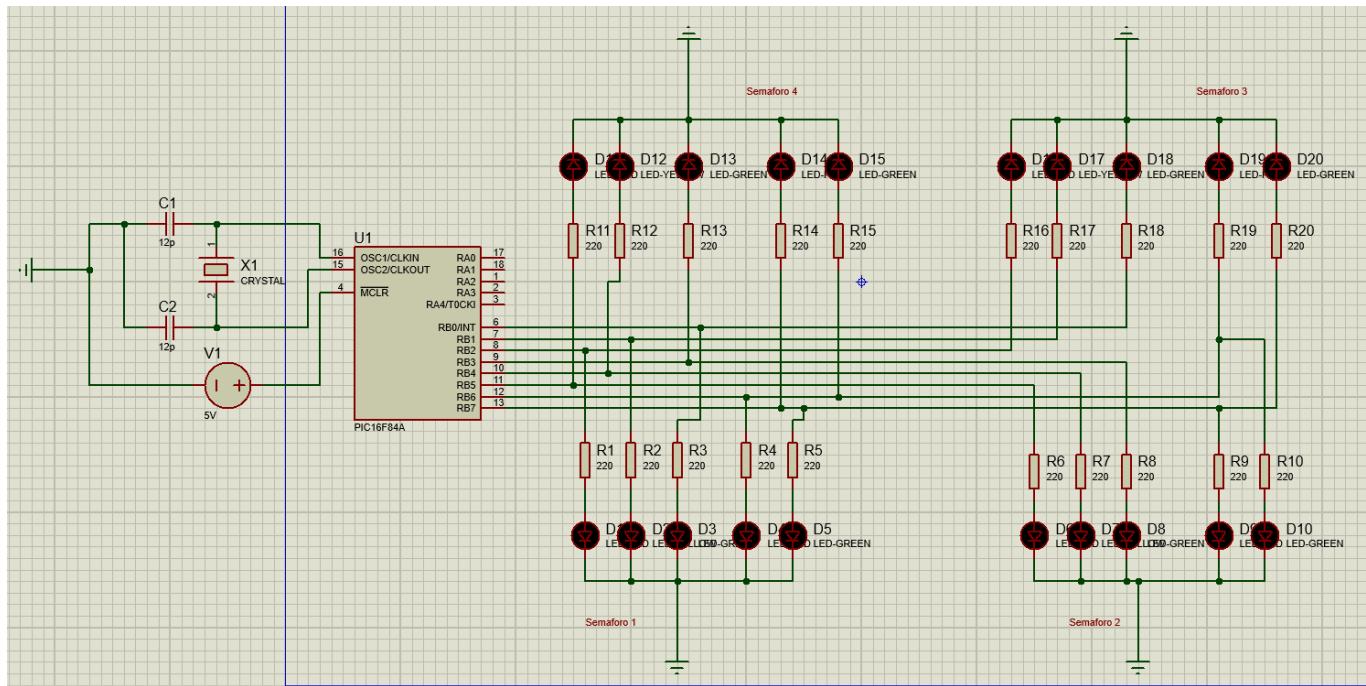
Del segundo modificamos su frecuencia de 1 Mega Hertz (1MHZ) a 5 Mega Hertz (5MHZ)



Del tercero modificamos la capacidad de voltaje de 1 a 5



13.- Una vez hecho esto, conectamos cada uno de los componentes de la siguiente manera, procura que te queden igual



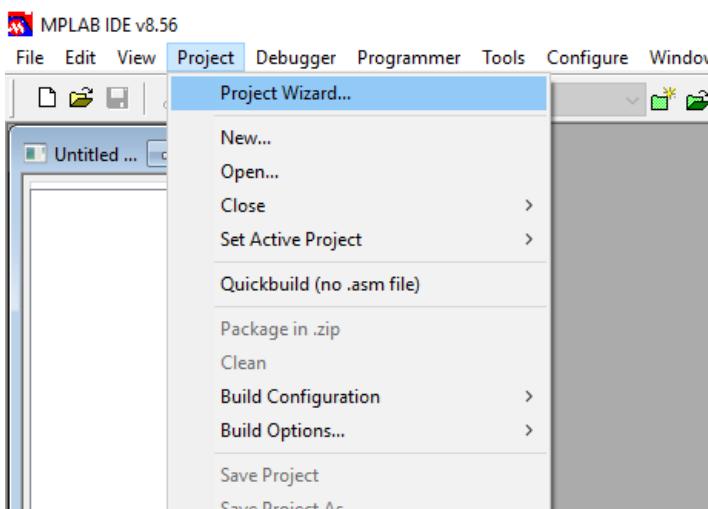
¡Y así es como queda nuestro diseño de hardware!

Para el código:

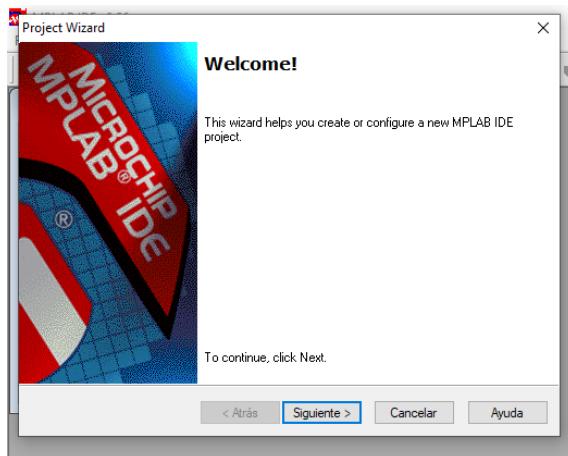
1.- Abrimos Mplab



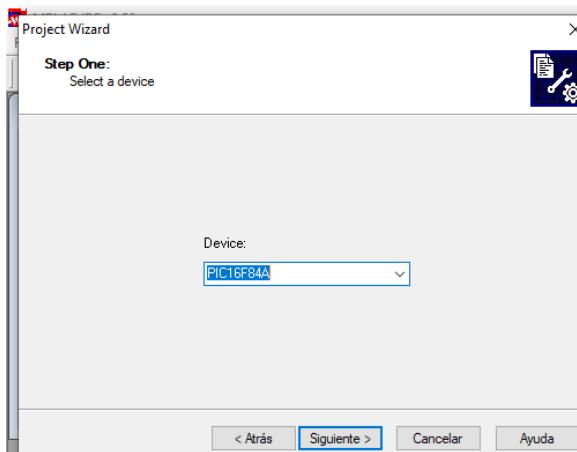
2.- Damos clic en Project, y ahí en Project Whizard



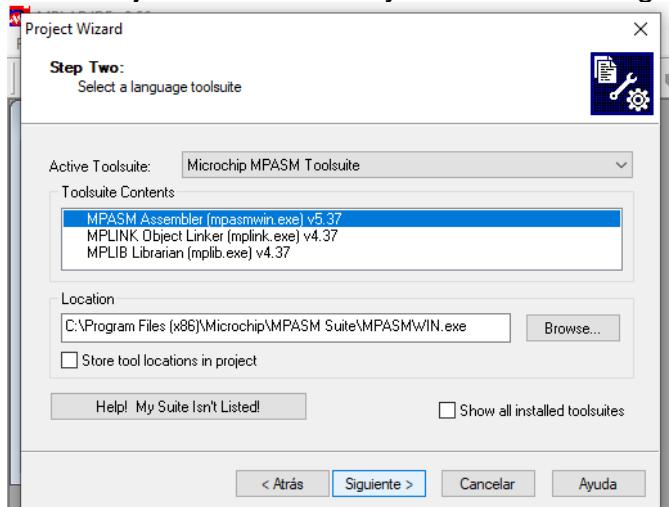
3., Aparecerá una ventana, le damos siguiente



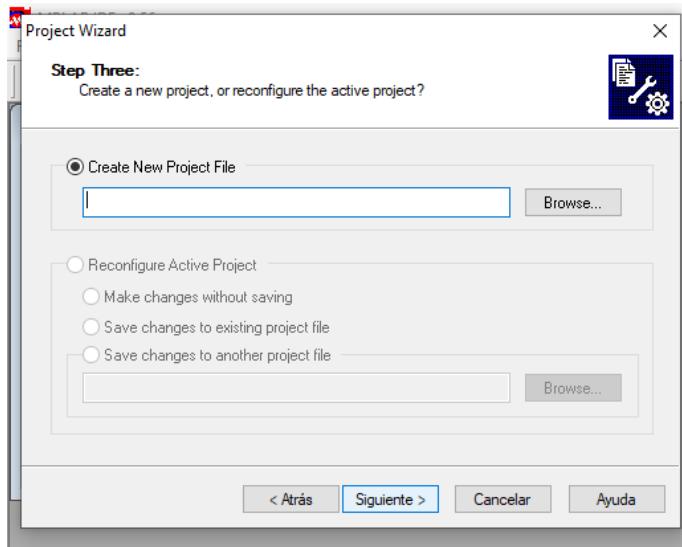
4.- En device seleccionamos PIC16F84A



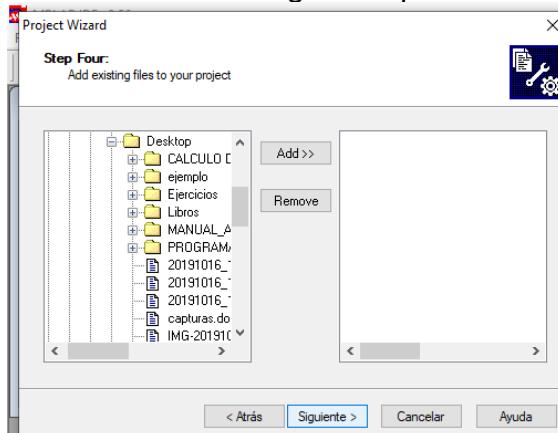
5.- Lo dejamos como esta y damos clic en siguiente



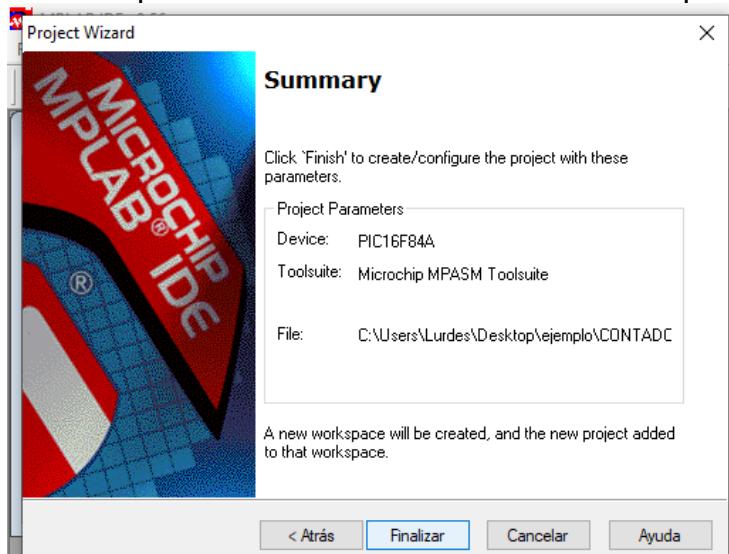
6.- Seleccionamos donde se guardara el archivo y lo nombramos como Contador



7.- Damos clic en siguiente para confirmar la ruta



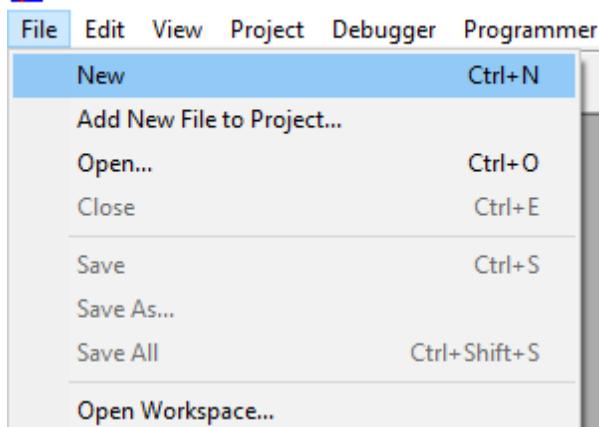
8.- Nos aparecerá una ventana con nuestras especificaciones, damos clic en finalizar





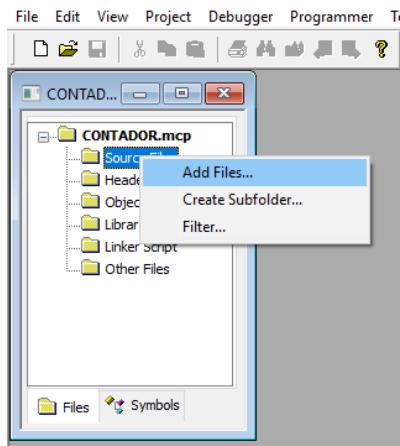
9.- Ahora damos clic en File/New (Saldrá una nueva ventana)

CONTADOR - MPLAB IDE v8.56 - CONTADOR.mcw



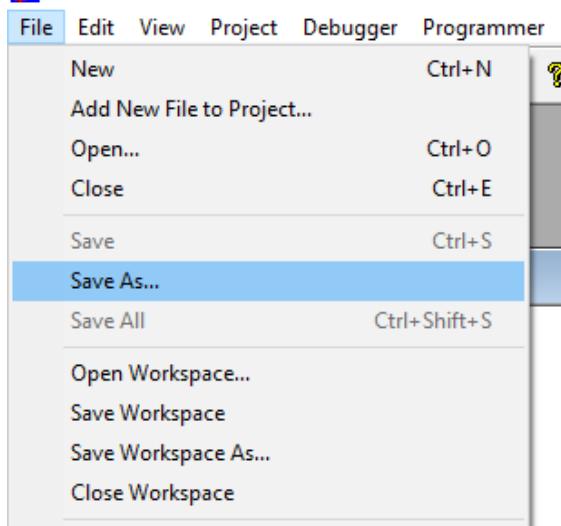
10.- Damos clic en Source Files/ Add Files

CONTADOR - MPLAB IDE v8.56 - CONTADOR.mcw



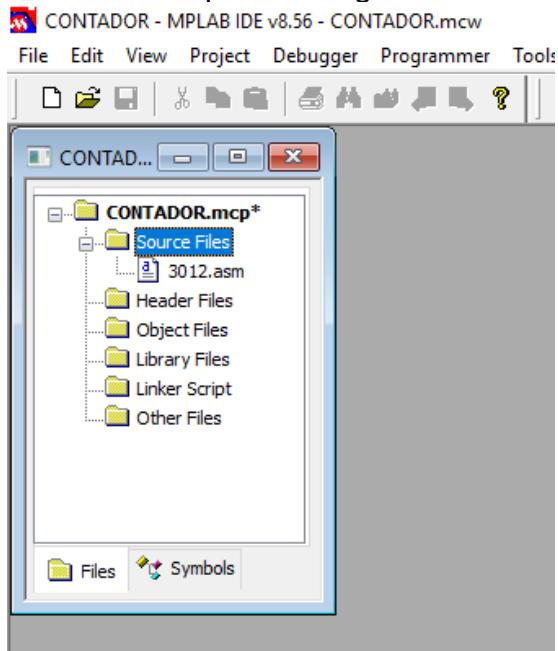
11.- Para guardarlo damos clic en file/ Save As

CONTADOR - MPLAB IDE v8.56





12.- Debería quedarte algo así al final



Aquí programaras el código de que es lo que quieres que tu protoboard realice, en este caso es el siguiente:



```
----- ;ZONA DE DATOS
----- processor 16F84A
----- #include <16F84A.INC>
----- __config _CP_OFF & _PWRTE_OFF & _WDT_ON & _XT_OSC ; 0X3FFD
-----
----- ; RAM-Variable
----- LRAM_Ox10 equ 0x10
----- LRAM_Ox11 equ 0x11
-----
----- ; Program
----- Org 0x0000
-----
----- ; Reset-Vector
0000 GOTO LADR_0x0005
-----
----- Org 0x0004
-----
----- ; Interrupt-Vector
0004 GOTO LADR_0x0018
----- LADR_0x0005
0005 MOVlw 0x87 ; b'10000111' d'135'
0006 BSF STATUS,RPO ; !Bank Register-Bank(0/1)-Select
0007 MOVWF TMRO ; !Bank!! TMRO - OPTION_REG
0008 BCF STATUS,RPO ; !Bank Register-Bank(0/1)-Select
0009 BSF STATUS,RPO ; !Bank Register-Bank(0/1)-Select
000A CLRF PORTB ; !Bank!! PORTB - TRISB
000B CLRF PORTA ; !Bank!! PORTA - TRISA
000C BCF STATUS,RPO ; !Bank Register-Bank(0/1)-Select
000D CLRF LRAM_0x10
000E MOVlw 0x61 ; b'01100001' d'097' "a"
000F MOVWF PORTB ; !Bank!! PORTB - TRISB
0010 BCF PORTA,1 ; !Bank!! PORTA - TRISA
0011 BSF PORTA,0 ; !Bank!! PORTA - TRISA
0012 MOVlw 0x10 ; b'00010000' d'016'
0013 MOVWF TMRO ; !Bank!! TMRO - OPTION_REG
0014 MOVlw 0xA0 ; b'10100000' d'160'
0015 MOVWF INTCON
----- LADR_0x0016
0016 CLRWDT
0017 GOTO LADR_0x0016
----- LADR_0x0018
0018 MOVlw 0x10 ; b'00010000' d'016'
0019 MOVWF TMRO ; !Bank!! TMRO - OPTION_REG
001A INCF LRAM_0x11,F
001B MOVlw 0x05 ; b'000000101' d'005'
001C SUBWF LRAM_0x10,W
001D BTFSS STATUS,Z
001E GOTO LADR_0x0069
001F CLRF LRAM_0x11
0020 INCF LRAM_0x10,F
0021 MOVlw 0x28 ; b'00101000' d'040' "("
0022 SUBWF LRAM_0x10,W
0023 BTFSC STATUS,C
0024 GOTO LADR_0x002A
0025 MOVlw 0x61 ; b'01100001' d'097' "a"
0026 MOVWF PORTB ; !Bank!! PORTB - TRISB
0027 BCF PORTA,1 ; !Bank!! PORTA - TRISA
0028 BSF PORTA,0 ; !Bank!! PORTA - TRISA
0029 GOTO LADR_0x0069
----- LADR_0x002A
002A MOVlw 0x32 ; b'00110010' d'050' "2"
002B SUBWF LRAM_0x10,W
002C BTFSC STATUS,C
002D GOTO LADR_0x0037
```



MANUAL DE PRÁCTICAS



```
002D      GOTO LADR_0x0037
002E      MOVLW 0x61          ; b'01100001' d'097' "a"
002F      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0030      BTFSC LRAM_0x10,0
0031      GOTO LADR_0x0069
0032      MOVLW 0x02          ; b'00000010' d'002'
0033      XORWF PORTA,F      ; !!Bank!! PORTA - TRISA
0034      MOVLW 0x60          ; b'01100000' d'096' ""
0035      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0036      GOTO LADR_0x0069
----- LADR_0x0037
0037      MOVLW 0x3C          ; b'00111100' d'060' "<"
0038      SUBWF LRAM_0x10,W
0039      BTFSC STATUS,C
003A      GOTO LADR_0x0042
003B      MOVLW 0x62          ; b'01100010' d'098' "b"
003C      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
003D      BTFSC LRAM_0x10,0
003E      GOTO LADR_0x0069
003F      MOVLW 0x02          ; b'00000010' d'002'
0040      XORWF PORTA,F      ; !!Bank!! PORTA - TRISA
0041      GOTO LADR_0x0069
----- LADR_0x0042
0042      MOVLW 0x64          ; b'01100100' d'100' "d"
0043      SUBWF LRAM_0x10,W
0044      BTFSC STATUS,C
0045      GOTO LADR_0x004B
0046      MOVLW 0x8C          ; b'10001100' d'140'
0047      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0048      BSF PORTA,1         ; !!Bank!! PORTA - TRISA
0049      BCF PORTA,0         ; !!Bank!! PORTA - TRISA
004A      GOTO LADR_0x0069
----- LADR_0x004B
004B      MOVLW 0x6E          ; b'01101110' d'110' "n"
004C      SUBWF LRAM_0x10,W
004D      BTFSC STATUS,C
004E      GOTO LADR_0x0058
004F      MOVLW 0x8C          ; b'10001100' d'140'
0050      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0051      BTFSS LRAM_0x10,0
0052      GOTO LADR_0x0069
0053      MOVLW 0x01          ; b'00000001' d'001'
0054      XORWF PORTA,F      ; !!Bank!! PORTA - TRISA
0055      MOVLW 0x84          ; b'10000100' d'132'
0056      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0057      GOTO LADR_0x0069
----- LADR_0x0058
0058      MOVLW 0x78          ; b'01111000' d'120' "x"
0059      SUBWF LRAM_0x10,W
005A      BTFSC STATUS,C
005B      GOTO LADR_0x0064
005C      MOVLW 0x94          ; b'10010100' d'148'
005D      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
005E      BTFSS LRAM_0x10,0
005F      BSF IND,0
0060      BTFSC LRAM_0x10,0
0061      GOTO LADR_0x0069
0062      MOVLW 0x01          ; b'00000001' d'001'
0063      GOTO LADR_0x0069
----- LADR_0x0064
0064      MOVLW 0x61          ; b'01100001' d'097' "a"
0065      MOVWF PORTB        ; !!Bank!! PORTB - TRISB
0066      BCF PORTA,1         ; !!Bank!! PORTA - TRISA
0067      BSF PORTA,0         ; !!Bank!! PORTA - TRISA
0068      CLRF LRAM_0x10
----- LADR_0x0069
0069      MOVLW 0xA0          ; b'10100000' d'160'
006A      MOVWF INTCON
006B      RETFIE
----- End
0068      CLRF LRAM_0x10
----- LADR_0x0069
0069      MOVLW 0xA0          ; b'10100000' d'160'
006A      MOVWF INTCON
006B      RETFIE
----- End
```



Lo compilamos dando click en este icono:



Comenzará la compilación

```
Build | Version Control | Find in Files
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mpilib.exe
Preprocessor symbol '_DEBUG' is defined.
Tue Oct 08 17:44:53 2019

Clean: Deleting intermediary and output files.
Clean: Deleted file 'C:\Users\HP\Documents\arquitectura_de_chompus\cor
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe"
Warning[205] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Warning[205] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Warning[205] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Warning[205] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Message[302] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Message[302] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Message[302] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Warning[205] C:\USERS\HP\DOCUMENTS\ARQUITECTURA_DE_CHOMI
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p1
MLINK 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\HP\Documents\arquitectura_de_chompus\contador_1.c

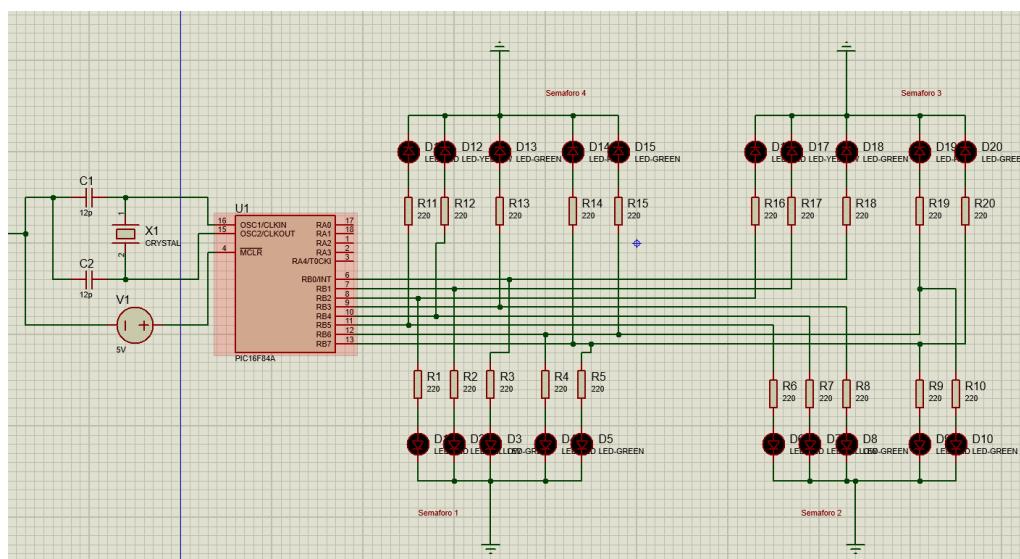
Debug build of project 'C:\Users\HP\Documents\arquitectura_de_chompus'
Language tool versions: MPASMWIN.exe v5.37, mplink.exe v4.37, mpilib.exe
Preprocessor symbol '_DEBUG' is defined.
Tue Oct 08 17:44:55 2019

BUILD SUCCEEDED
```

Cuando termine lo buscaremos en su carpeta, lo reconocerá ya que es un tipo de documento .HEX

Semaforo_3012.asm	02/12/2019 11:32 a. m.	Archivo ASM	0 KB
Semaforo_3012.cof	02/12/2019 11:32 a. m.	Archivo COF	5 KB
Semaforo_3012.err	02/12/2019 11:32 a. m.	Archivo ERR	0 KB
Semaforo_3012.HEX	02/12/2019 11:32 a. m.	Archivo HEX	1 KB
Semaforo_3012.lst	02/12/2019 11:32 a. m.	Archivo LST	15 KB
Semaforo_3012.map	02/12/2019 11:32 a. m.	Archivo MAP	4 KB

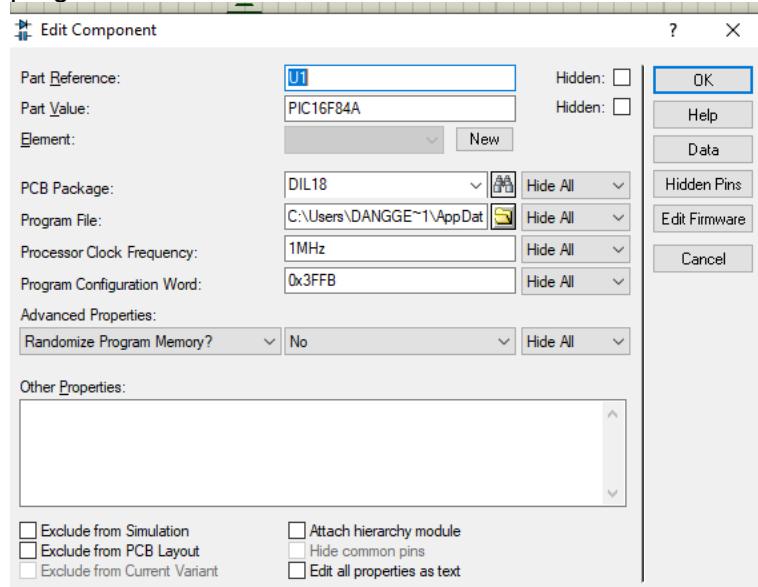
Para probar el código, abrimos Proteus con el diseño de nuestro hardware que ya teníamos.



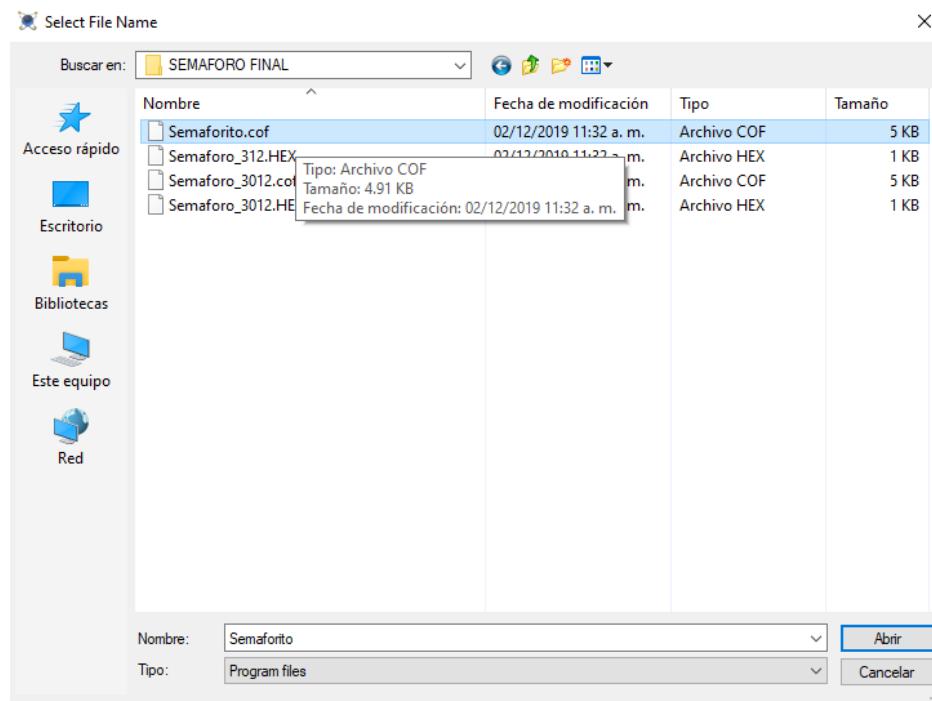


Damos doble clic en el pic

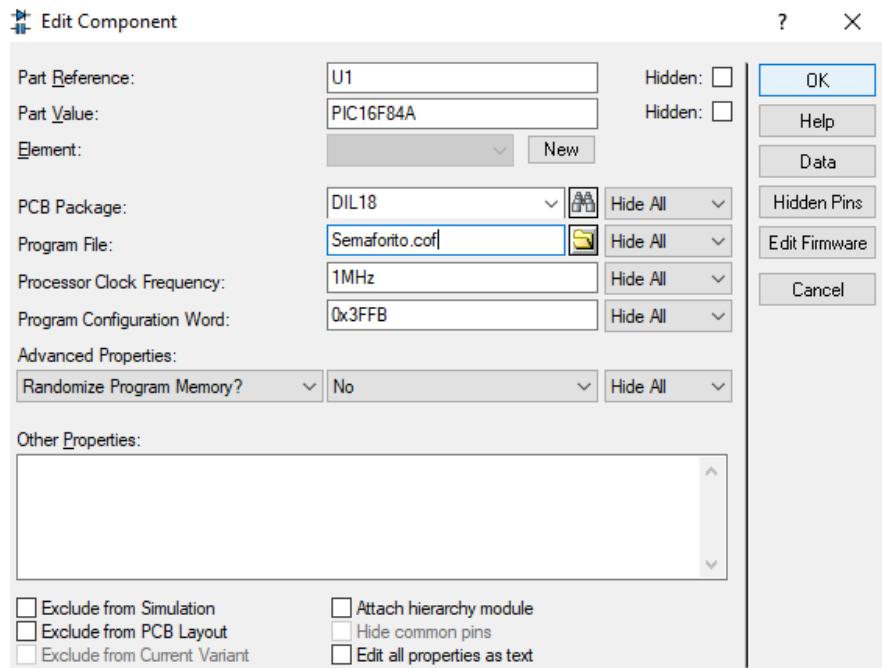
En esta ventana seleccionaremos el archivo de nuestro código compilado, para eso damos clic en program file.



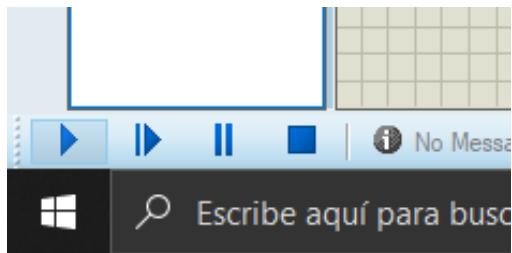
Seleccionamos la ruta de nuestro archivo y damos clic en abrir:



En la siguiente ventana damos clic en OK



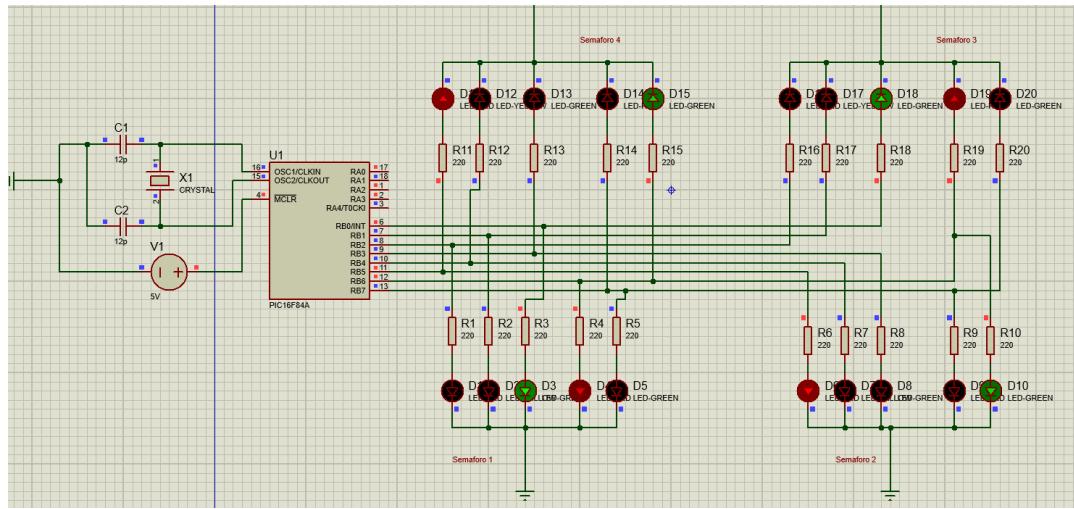
Para correr la simulación deberemos dar clic en el siguiente ícono que se encuentra en la esquina inferior izquierda:



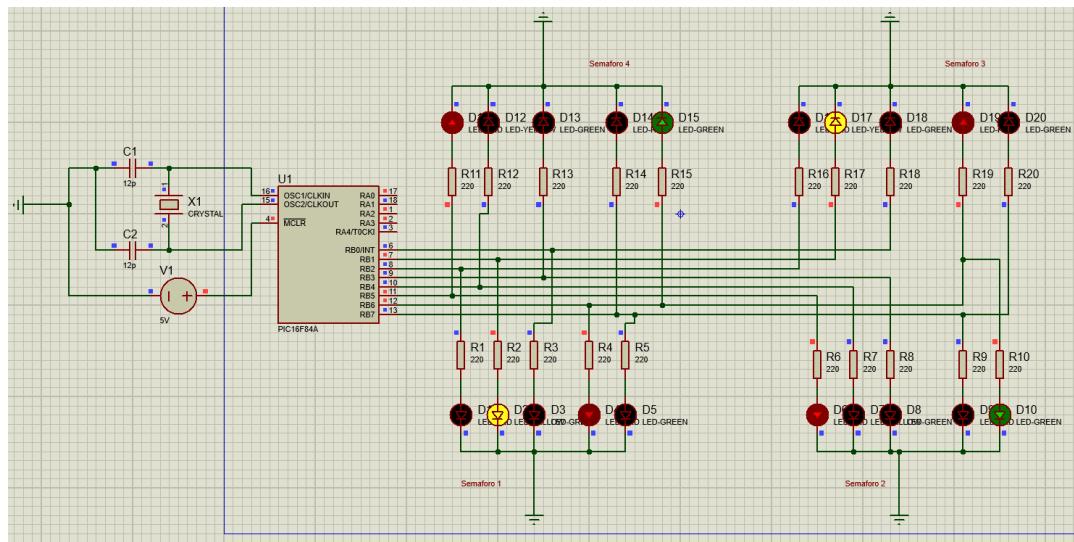


MANUAL DE PRÁCTICAS

Pruebas:



El semáforo 1 y 3 de peatones se encuentra en rojo, por lo tanto, su semáforo para autos esta en verde, el semáforo 2 y 4 de peatones se encuentra en verde, por lo tanto, su semáforo de autos está en rojo.

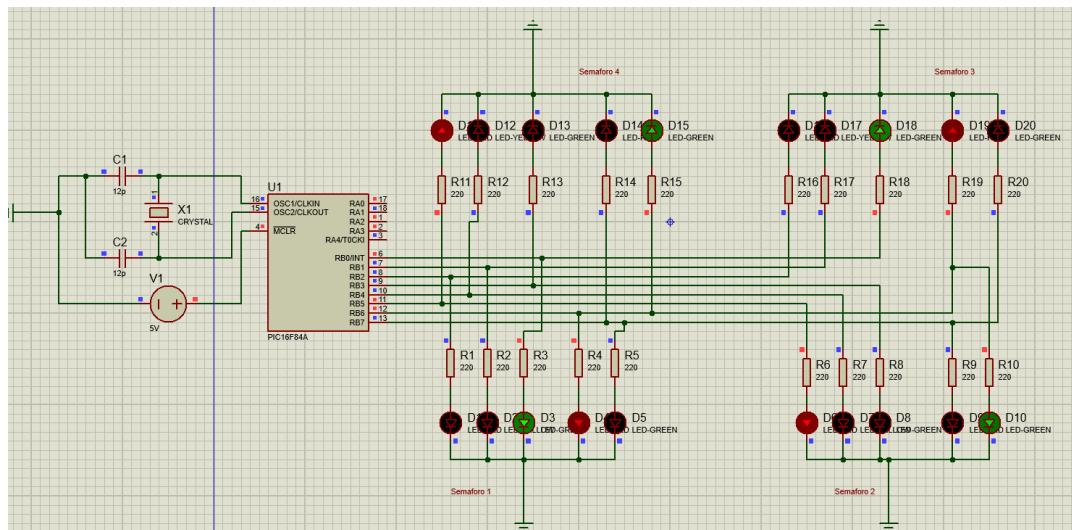


Al cabo de 20 segundos los semáforos 1 y 3 pasaran a color amarillo por 5 segundos después de eso cambiaran a rojo, además su semáforo de peatones pasara a encenderse la luz verde.

Los semáforos 2 y 4 de autos pasaran a cambiar a color verde y sus semáforos de peatones pasaran a color rojo.



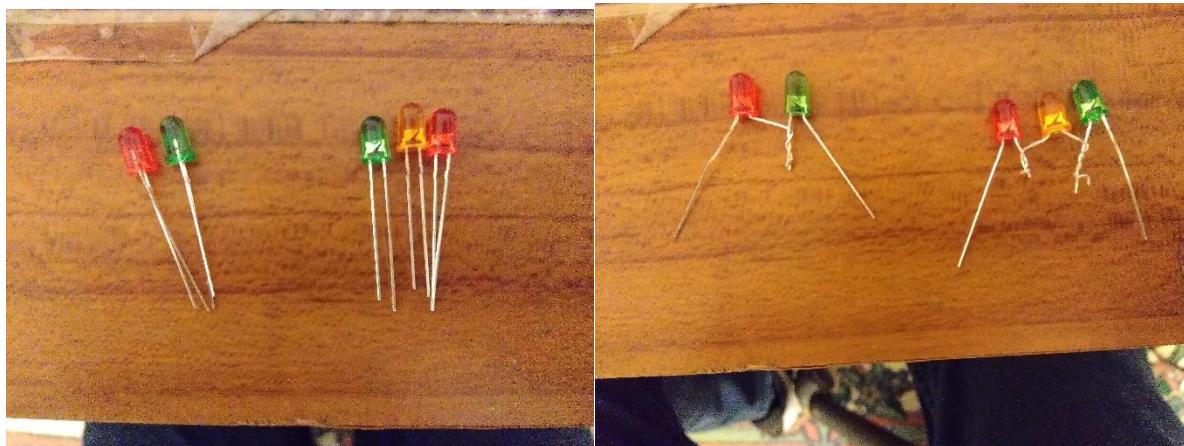
MANUAL DE PRÁCTICAS



Esto se repetirá de manera infinita hasta que pausemos el código, lo que quiere decir que nuestra simulación es correcta y podemos proceder a ensamblar nuestro semáforo.

Ensamblaje

- 1.- Preparamos nuestros semáforos de peatones y autos con leds, conectando sus tierras simultáneamente y dejando sus salidas a positivo libre:

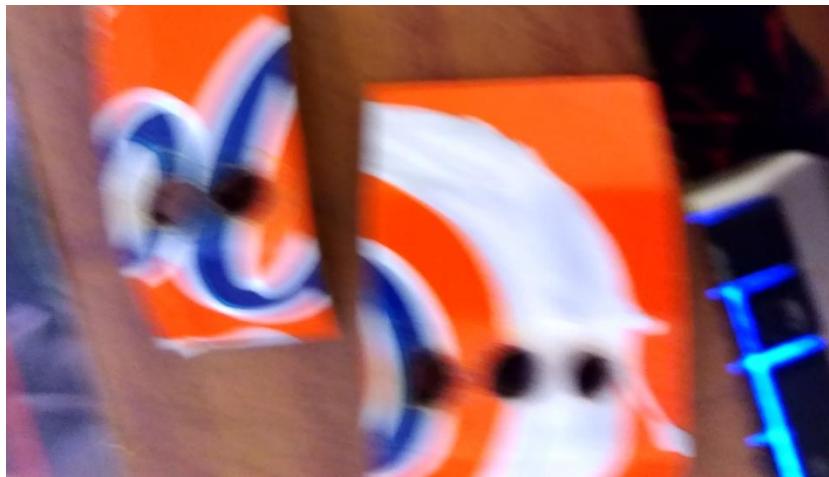


- 2.- Para la estructura de los semáforos cortaremos cartón de las siguientes medidas

- 4 de 6cm*3.5cm
- 4 de 6cm*2cm
- 16 cuadritos de 1.5cm*1.5cm



Con ayuda del cautín haremos los huecos para cada semáforo (2 para el de peatones y 3 para el de autos)





Colocamos los leds de manera que entren en los huecos



Después procederemos a pegar y cellar el semáforo, le pegamos una tapa arriba:



Para la tapa de abajo es necesario hacerle un hueco con el cautín en el centro para que los cables pasen.

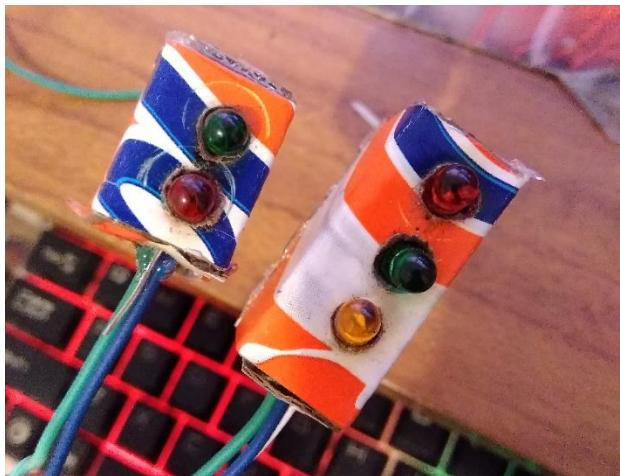




MANUAL DE PRÁCTICAS



Para finalizar pegaremos un popote por debajo que servira como soporte para nuestros semáforos:

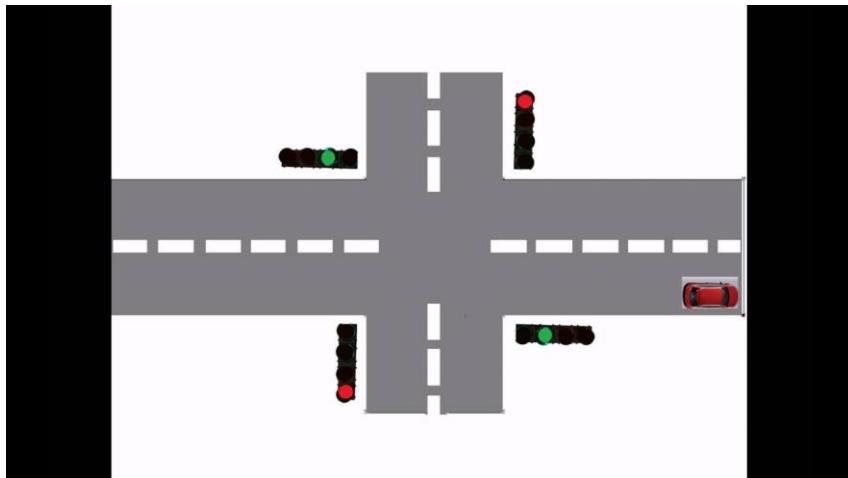


Haremos esto otras 3 veces, es decir al terminar tendremos 4 semáforos de peatones y 4 de autos.

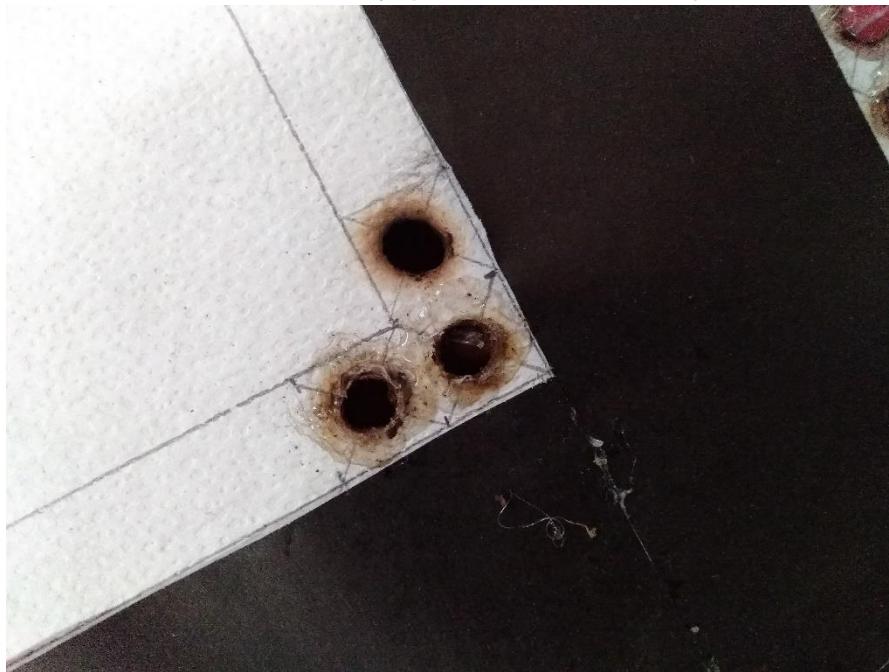


Maqueta:

En la maqueta primero procedí a realizar una intersección de 2 calles así:



Para colocar mis semáforos y que estos estuvieran fijos hice huecos en mi maqueta con alluda del cautín:

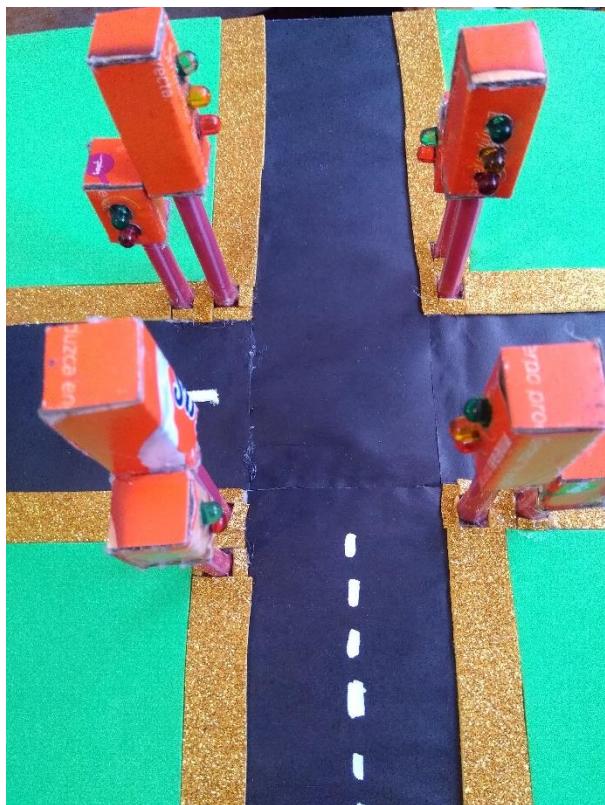


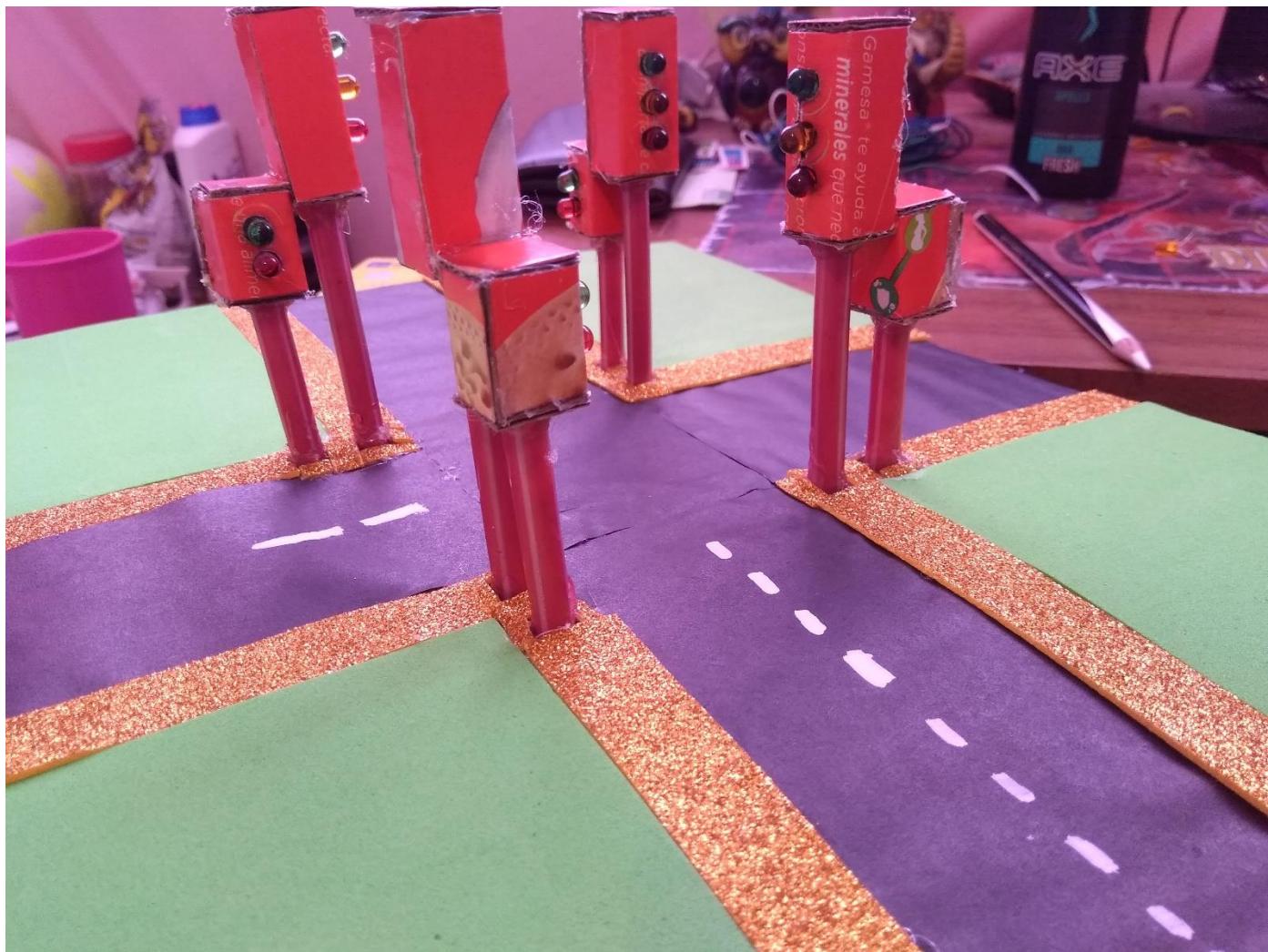
Dando como resultado esto:



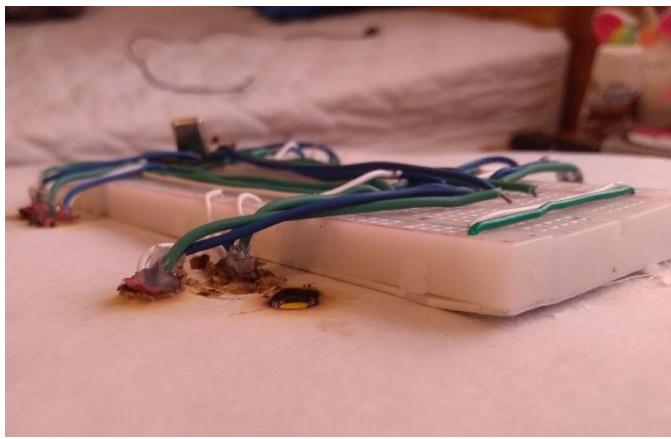
GOBIERNO DEL
ESTADO DE MÉXICO

MANUAL DE PRÁCTICAS





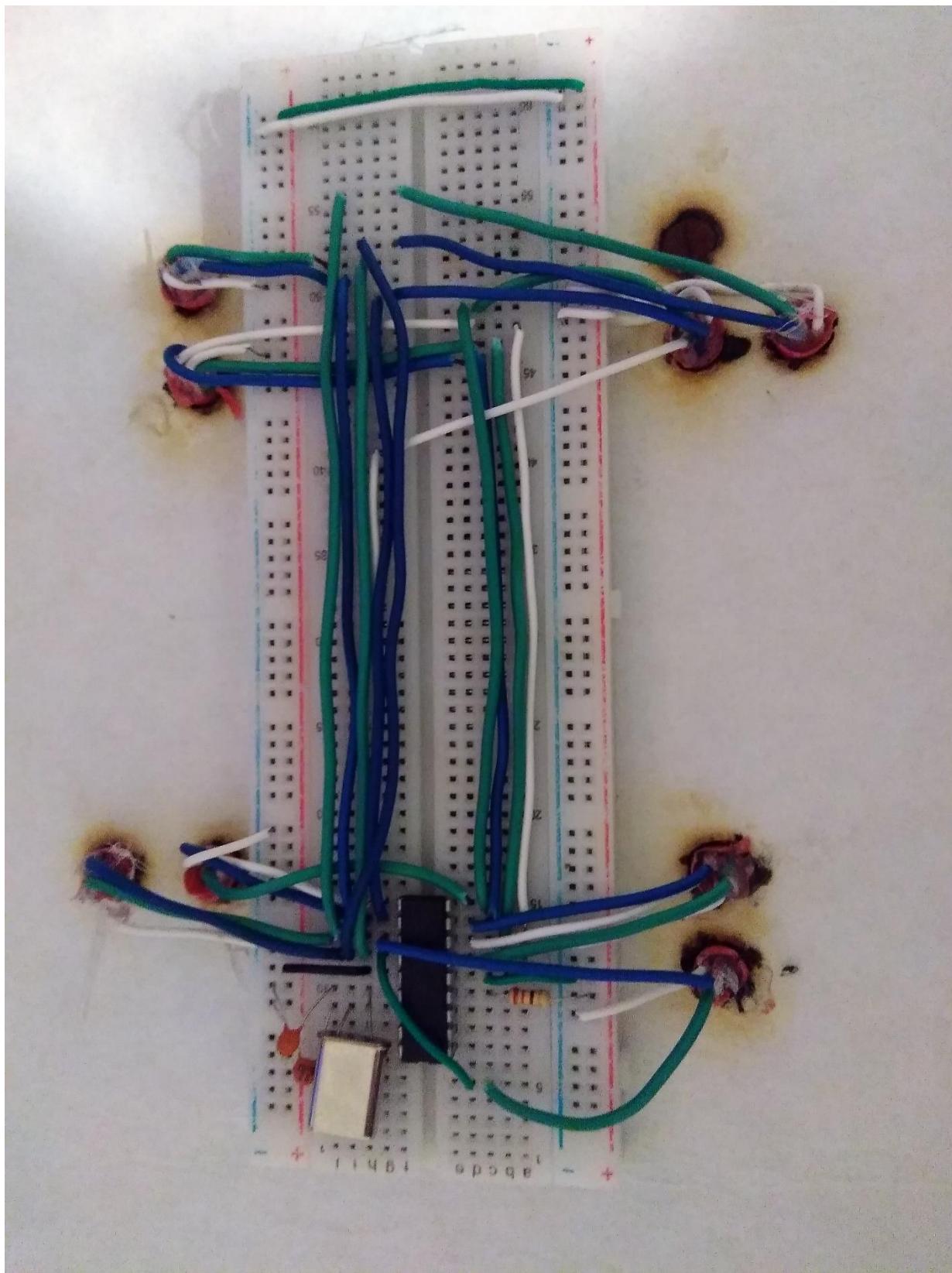
Los cables salen por debajo así que la protoboard se encuentra ahí, solo faltaría conectarlos dando como resultado esto:





GOBIERNO DEL
ESTADO DE MÉXICO

MANUAL DE PRÁCTICAS



Una vez tengas tu código compilado, es hora de programarlo en el PIC.

1.-Colocamos El PIC en el programador, la computadora lo detectara automáticamente



2.- Damos clic en leer y seleccionamos nuestro archivo programado



3.- Lo seleccionamos para pasarlo

Semáforo_3012.en	02/12/2019 11:32 a. m.	ARCHIVO ENK	0 KB
Semaforo_3012.HEX	02/12/2019 11:32 a. m.	Archivo HEX	1 KB
Semaforo_3012.lst	02/12/2019 11:32 a. m.	Archivo LST	15 KB
Semaforo_3012.map	02/12/2019 11:32 a. m.	Archivo MAP	4 KB

4.- Damos clic en escribir para guardar el archivo



5.- Damos clic en verificar para comprobar si todo está correcto y listo!!!



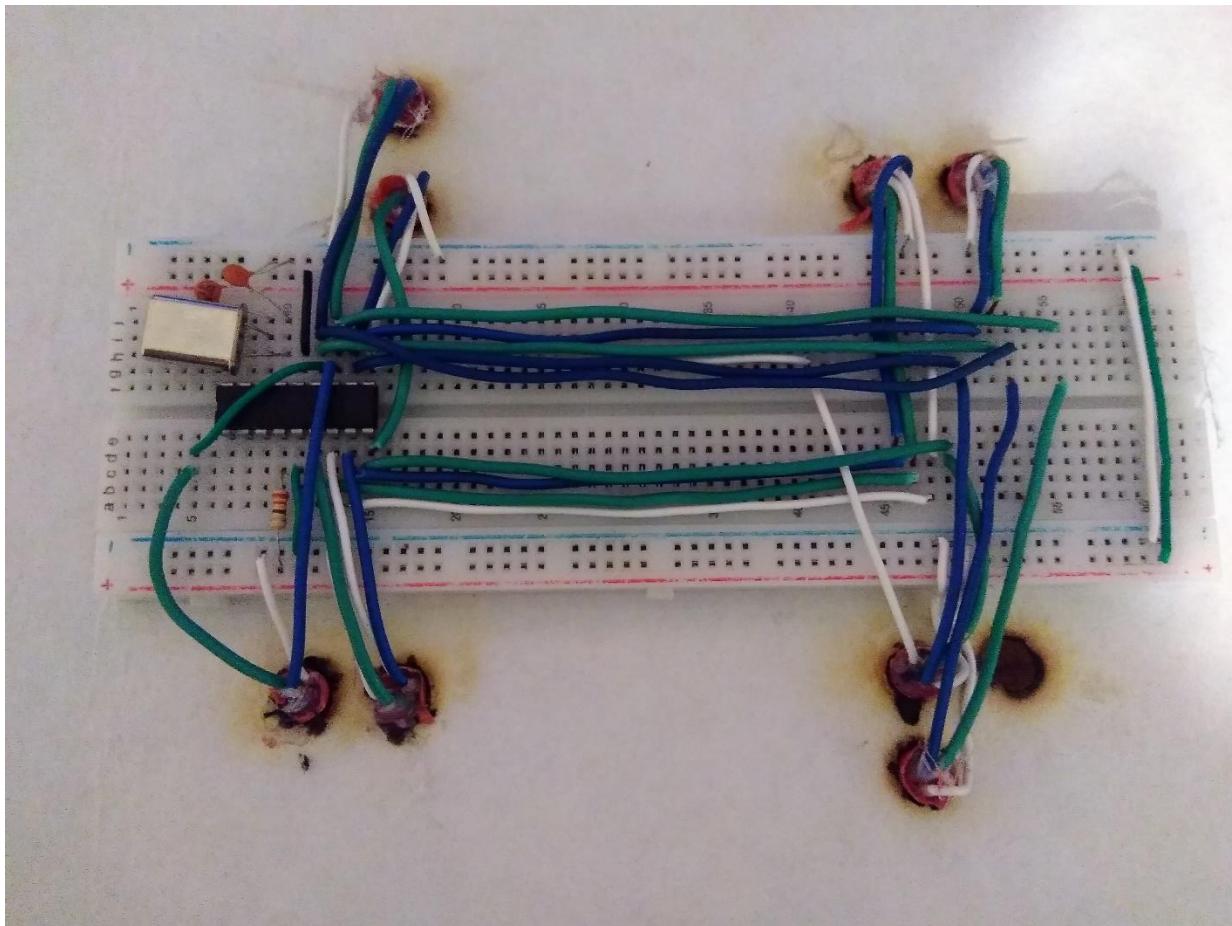
MANUAL DE PRÁCTICAS



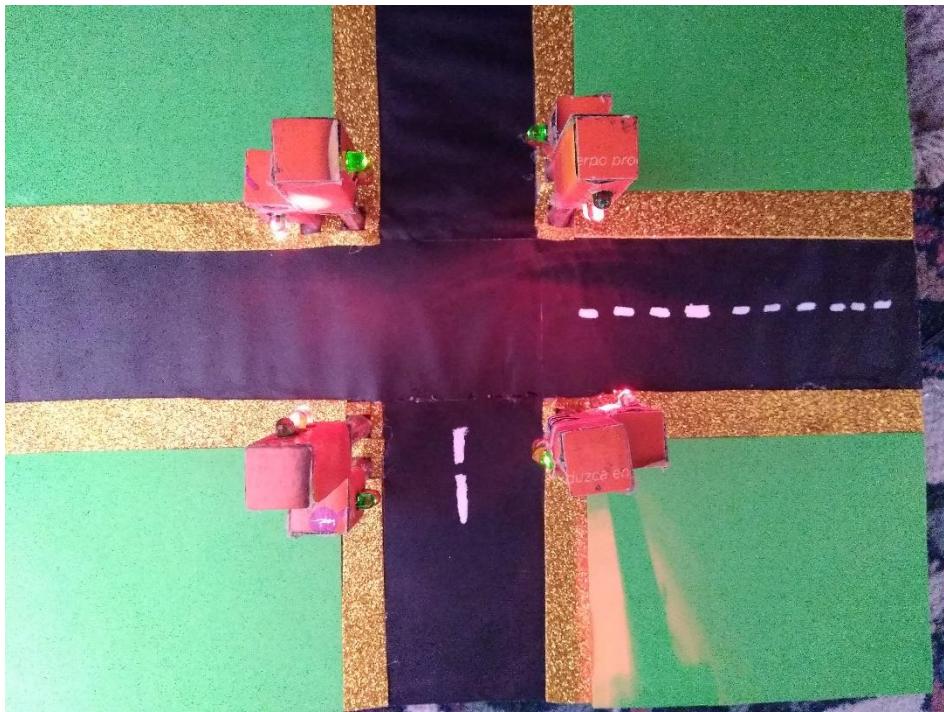
Nuestro PIC esta programado!



PARA LA PRUEBA FISICA, NECESITAMOS ENSAMBLAR EN LA PROTOBOARD LOS COMPONENTES AL IGUAL QUE LO HICIMOS EN PROTEUS, EL RESULTADO SERIA ALGO ASI:



LO CONECTAREMOS A UNA FUENTE DE PODER, EN ESTE CASO UN CARGADOR DE CELULAR CON LOS CABLES PELADOS PARA SABER SI FUNCIONA:



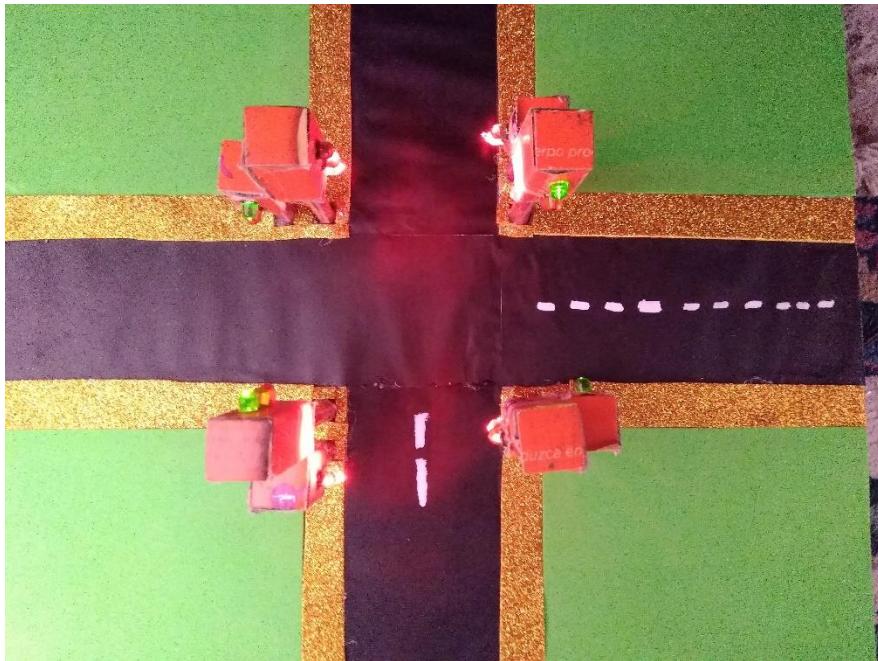
El semáforo 1 y 3 de peatones se encuentra en rojo, por lo tanto, su semáforo para autos está en verde, el semáforo 2 y 4 de peatones se encuentra en verde, por lo tanto, su semáforo de autos está en rojo.



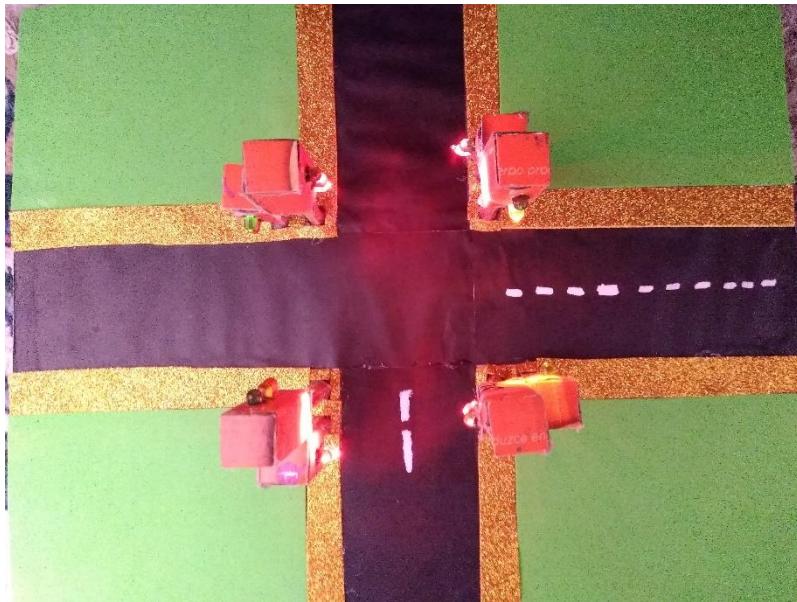
Al cabo de 20 segundos los semáforos 1 y 3 pasaran a color amarillo por 5 segundos después de eso cambiaron a rojo, además su semáforo de peatones pasara a encenderse la luz verde.



Los semáforos 2 y 4 de autos pasaran a cambiar a color verde y sus semáforos de peatones pasaran a color rojo.



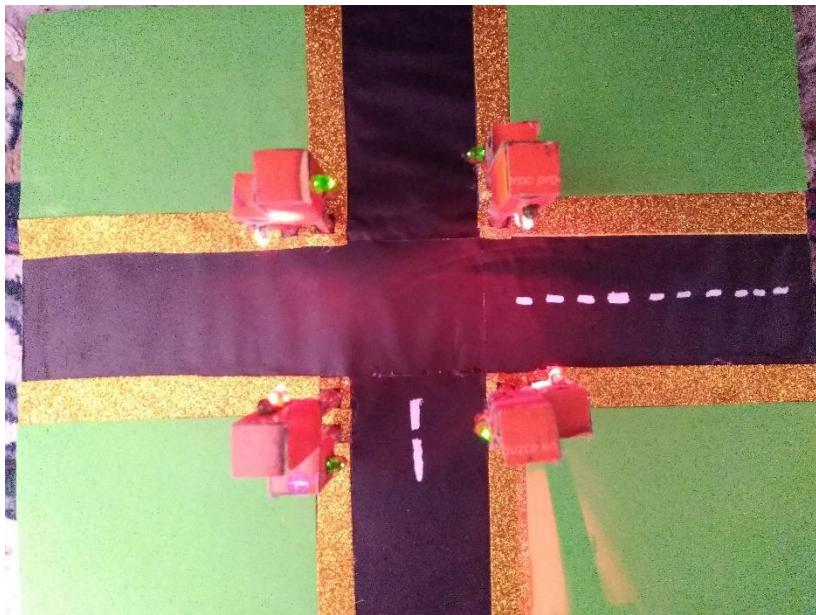
Ahora los semáforos 2 y 4 pasaran a color amarillo al cabo de 20 segundos para despues cambiar a rojo y sus semáforos de peatones en verde, los semáforos 1y 3 pasaran a ser de color verde y sus semáforos de peatones de rojo.





GOBIERNO DEL
ESTADO DE MÉXICO

MANUAL DE PRÁCTICAS



Esro coincide con las pruebas realizadas en proteus, por lo tanto el proyecto esta correcto.

V. Conclusiones:

Esto nos demuestra el inicio de la combinación entre el hardware y el software, el cómo es que algo que comenzamos a escribir en un código en la computadora es realizado por el hardware que construimos para ello, la programación del pic es un poco tediosa pero no imposible, esto me hace ver que muchas de las cosas que me rodean deben estar programadas de cierto parecido con lo que hice, también me muestra que el más mínimo error en alguna conexión puede descomponer todo, por lo que el equivocarse ya no es una opción.