Facultad de Instrumentación Electrónica y Ciencias Atmosféricas

UNIVERSIDAD VERACRUZANA

MANUAL TECNICO

Dale vida a tu casa

MICROPROCESADORES Y MICROCONTROLADORES

INDICE

Introducción

El proyecto dale vida a tu casa es un sistema de automatización del hogar, el cual manipula elementos importantes tales como luces, alarmas y puertas. Su función principal es el brindar seguridad y comodidad al usuario.

Los elementos de los que consta dicho proyecto son:

- 2 Luces a través de interruptores
- Luz automática al momento de presenciar oscuridad
- Luz a través del cambio de intensidad de la misma con un elemento potenciómetro
- Control eléctrico de plumilla de estacionamiento
- Control eléctrico de la puerta principal de la vivienda
- Alarma antiincendios
- Alarma anti-intrusos

Objetivo

El objetivo de este proyecto es que el usuario pueda tener el control de las distintas partes de su hogar al alcance de un click y también pueda sentirse seguro ante algún descuido como por ejemplo un posible incendio. También podrá estar tranquilo en casa ante una posible intrusión. Podrá manipular luces y puertas a su antojo.

Contenido técnico

Para la parte de software es necesario contar con la plataforma de Atmel studio 7 ya que en dicho programa se realiza el código programado en el micro para el control del sistema. De la misma manera es vital contar con Proteus se recomienda la versión 7, ya que es el software en donde se prueba el funcionamiento del sistema de manera simulada para posteriormente pasar a armarlo físicamente teniendo la certeza de que funciona. También se puede hacer uso de la plataforma Arduino para verificar el código antes de pasar a Atmel studio

Cabe mencionar que en el programa Proteus se debe de contar con los elementos de Arduino de no contar con ellos deberá de instalarse la librería.

Para la instalación de estos programas se requiere tener una computadora con al menos Windows 7, es importante mencionar que dependiendo la velocidad de su procesador correrá de mejor manera los programas sobre todo se hace énfasis en Atmel studio 7.

Responsables

son dos los elementos que conforman el equipo en cargado de realizar este proyecto, por un lado tenemos a Osiel de Jesús López López y por otro lado a Luis Fernando Suarez del Carmen ambos fungen la tarea de coordinación del proyecto así como también son equipo técnico encargado de realizar todas las actividades dentro del mismo.

En cuanto a las responsabilidades que toco a cada uno son las siguientes:

Luis Fernando fue el encargado de probar los códigos del proyecto uno por uno individualmente en la plataforma Arduino para posteriormente realizarlos en lenguaje c dentro de Atmel, también fue el encargado de recopilar parte del material para la parte física del proyecto a excepción de servomotores y jumpers, estos últimos fueron recopilados por Osiel.

Osiel fue el encargado de una vez teniendo las partes del proyecto listas en la plataforma Arduino, realizarlos en lenguaje c y simularlas en Proteus, primero parte por parte y luego todo el proyecto en conjunto.

Análisis de costos

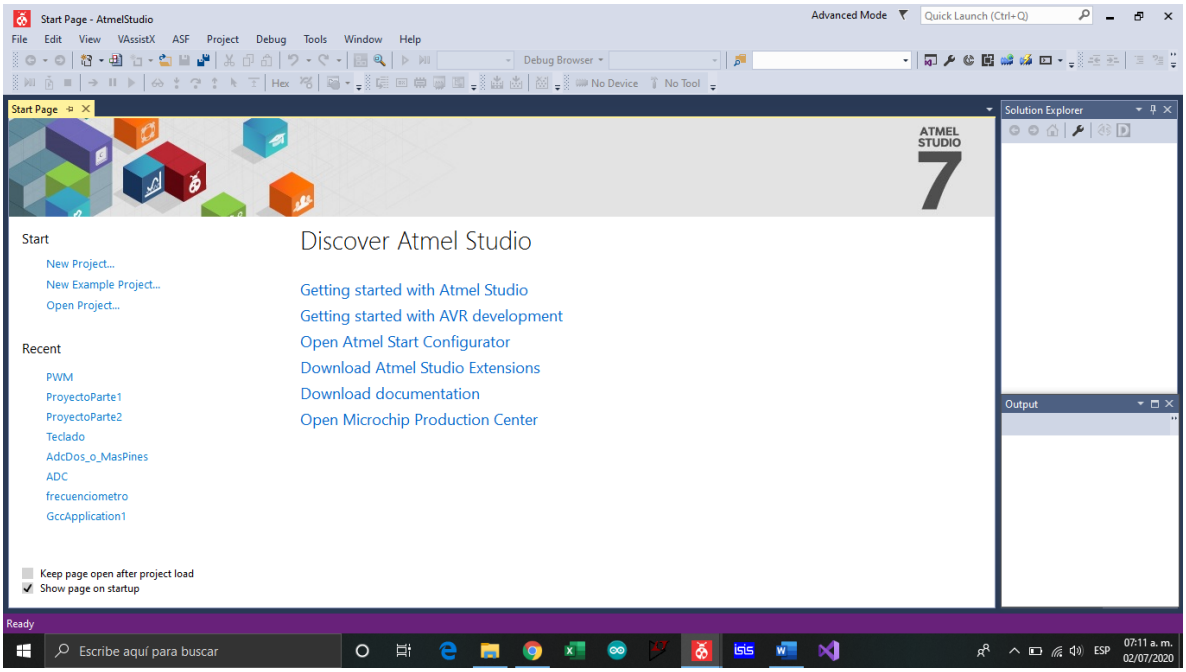
La siguiente tabla muestra el análisis de precios en cuanto al material a utilizar al momento de montar el proyecto.

Materiales	Precio
2 Buzzer	$70.00
2 Fotorresistencias	$24.00
Jumpers 40 piezas.	$22.00
4 Leds.	$12.00
Papel batería 1 pliego (recomendado para maqueta).	$150.00
3 placa Arduino uno.	$216.00
Protoboard.	$50.00

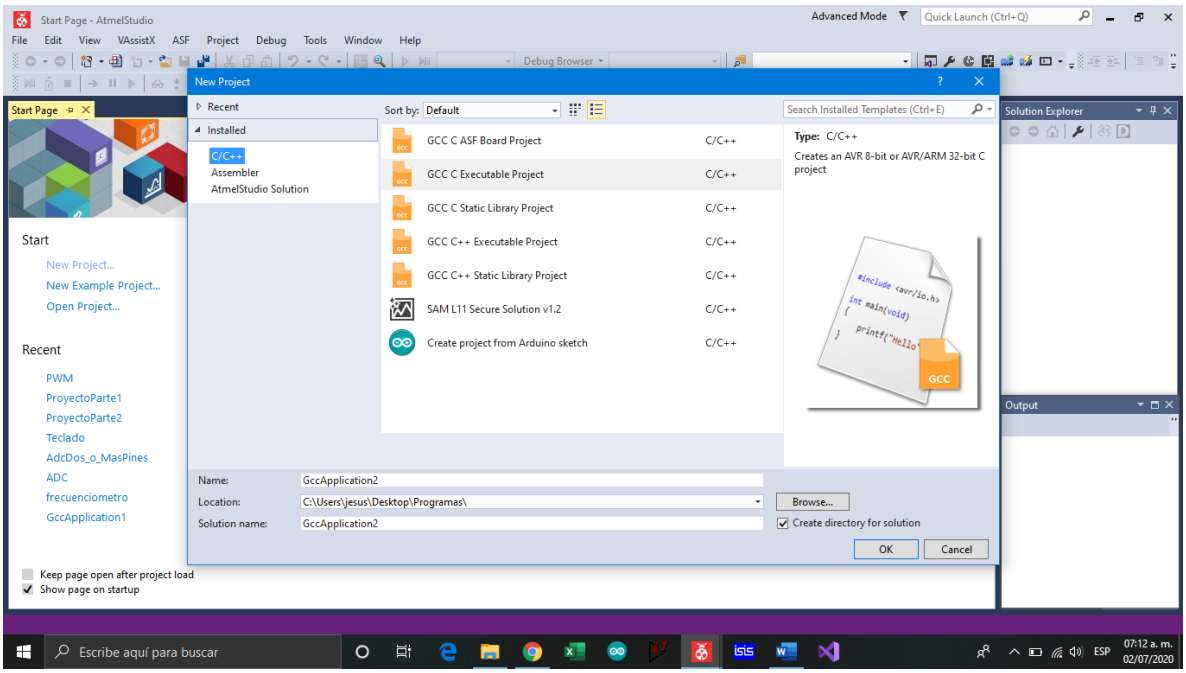
La realidad es que no es un proyecto barato, pero la seguridad que brinda es notable y vale la pena invertir en ello.

Desarrollo

Para comenzar a trabajar en Atmel studio procedemos a abrir el programa y en a esquina superior izquierda damos seleccionamos la opción de file y posteriormente new Project para realizar un nuevo proyecto.



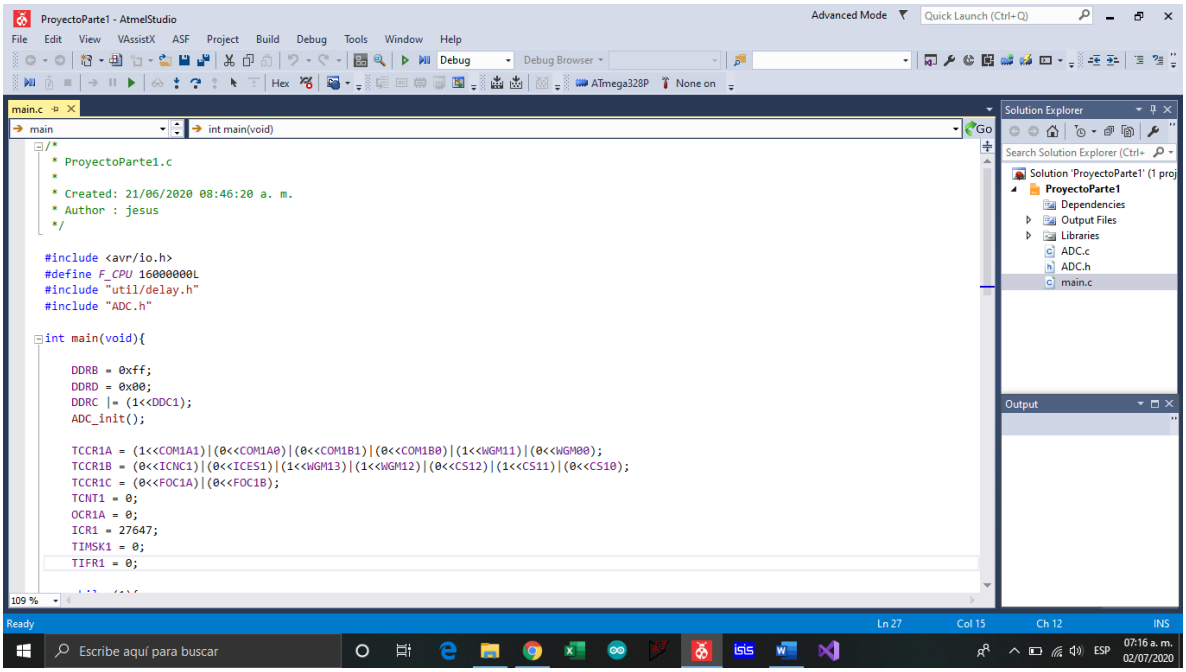
Posteriormente se elije realizar un programa en lenguaje c



Finalmente se elije el microcontrolador con el que se va a trabajar en el caso de este proyecto se hace uso del atmega328p, después de eso damos click y pasamos al área de trabajo.

Por default en todos los programas aparece la librería de arv.io, pero es necesario agregar un par más, tal es el caso de la velocidad del micro, una librería para delays y cualquier otra externa que ocupe su programa.

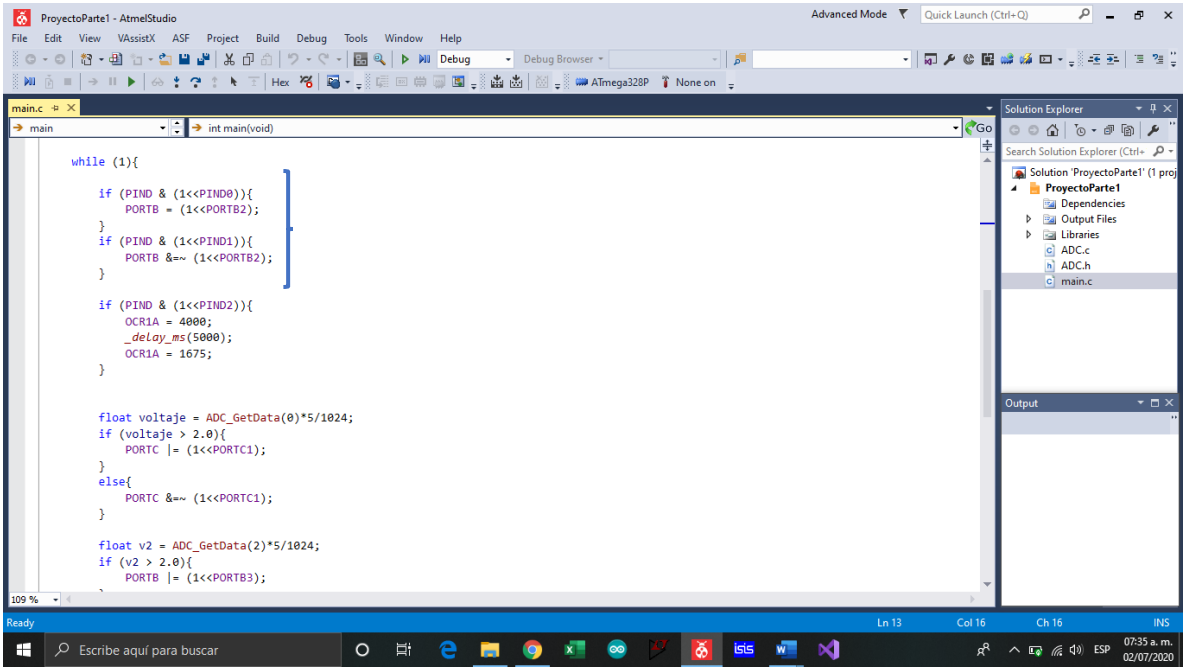
A continuación se procederá a explicar el código del proyecto.



La imagen de arriba muestra las librerías mencionadas anteriormente, además de una librería para controlar el adc, dicha librería esta incluida en los archivos del proyecto, esta librería contiene dos funciones adc_init() que se encarga de inicializar el adc y la otra adc_getdata() a la cual se le pasa un parámetro en este caso el numero del canal del adc a utilizar y dentro de esa función se lleva a cabo al conversión.

Dentro de la función int main el primer grupo de instrucciones se encarga de configuras los puertos b y d como salida y entrada respectivamente, también se configura el puerto 1 del puerto c como salida. Finalmente se inicializa el adc con a función ya antes mencionada.

El segundo grupo de instrucciones contiene los comandos necesarios para que el timer 1 genere salida PWM a través de su salida OCR1A, dicho PWM deberá ser de una frecuencia de 50 Hz la necesaria para que función un servomotor.

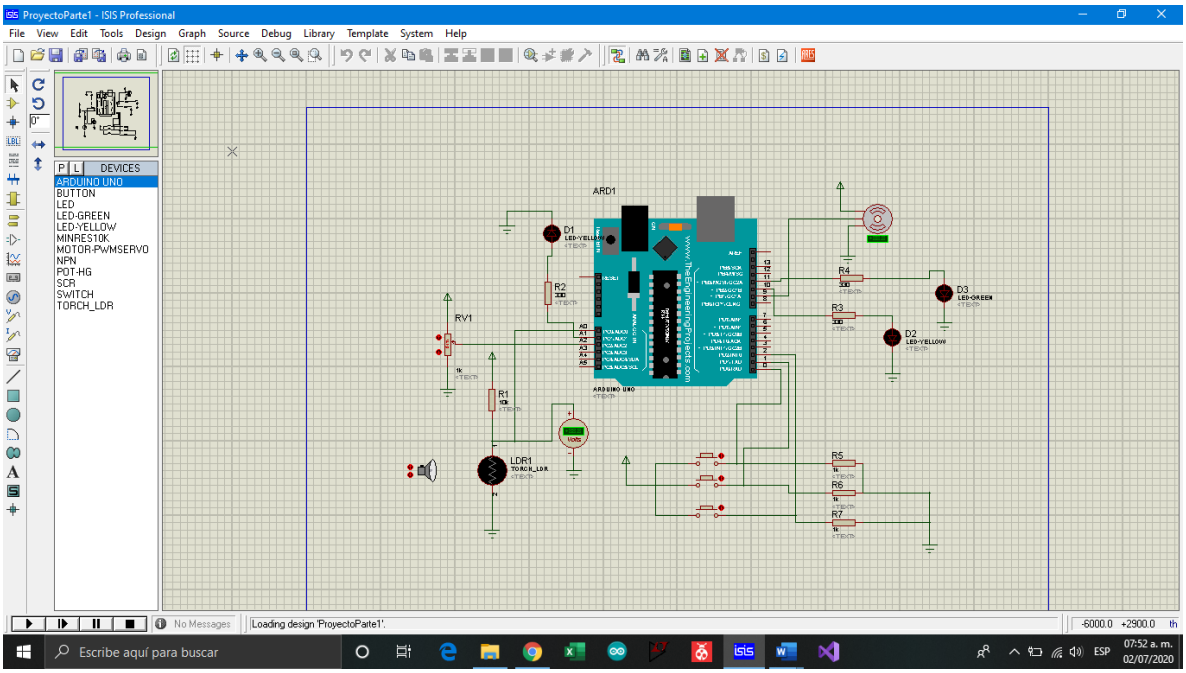


Ya dentro del bucle del código en las primeras instrucciones con if marcadas en la imagen, se indica que presionando el botón conectado al pind0 encendera el led en el pin 2 del puerto b, además de que al momento de presionar el botón conectado al pin d1 el led se apagara.

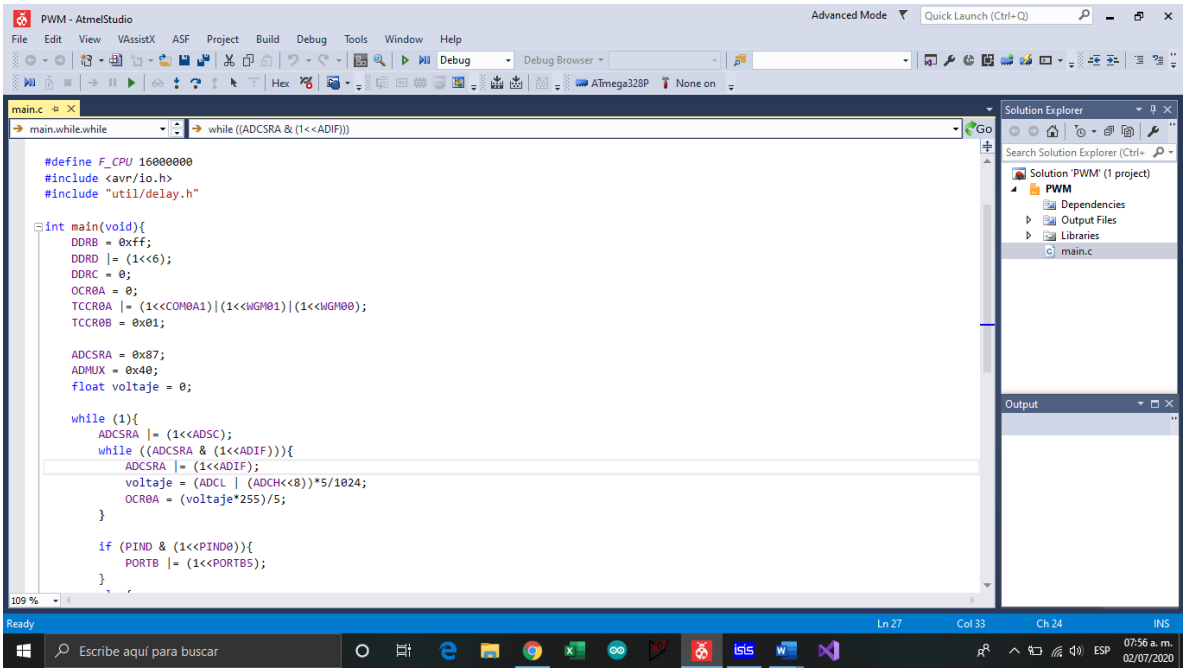
Le tercer if corresponde a la acción de controlar el servomotor tanto del estacionamiento como de la puerta principal, la instrucción dice que presionando el botón conectado al pin d2 el servomotor conectado al pin con nombre OCR1A girara 90° y después de 5 segundos regresara a su posición inicial 0°.

Por ultimo tenemos dos condicionales if sujetas al adc, dos variables (voltaje y v2) que guardaran el valor del adc del respectivo pin indicado en la función adc_getdata() en valor de voltaje. Cada conversión de adc cuenta con un if que indica que si sobrepasa el valor de 2.0 encenderá el led indicado de l contrario se apagará.

La siguiente imagen muestra la interfaz de esta primera parte del proyecto.



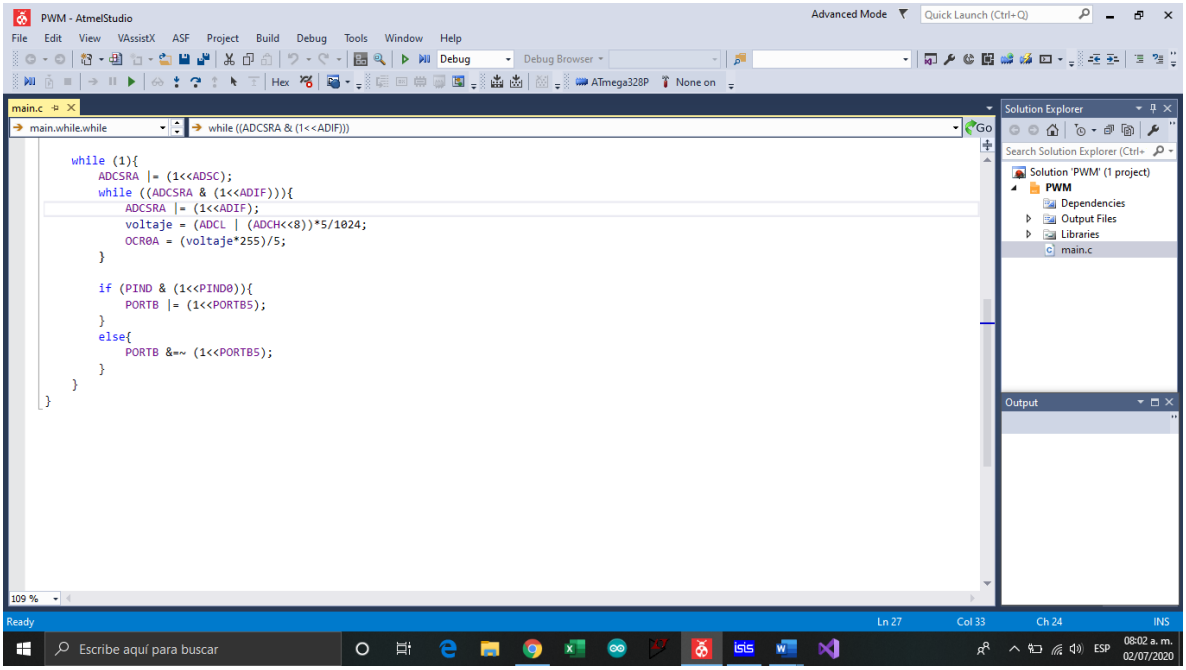
La segunda parte del proyecto es mostrada a continuación:



Esta segunda parte tiene como principio la inclusión de librerías necesarias mismas que están incluidas en la parte uno a excepción de la librería adc.

Dentro del int main el primer grupo de instrucciones se encarga de configuras el puerto b como salida, el puerto c como entrada, además de las configuraciones necesaria en el timer0 para generar PWM por su salida OCR0A.

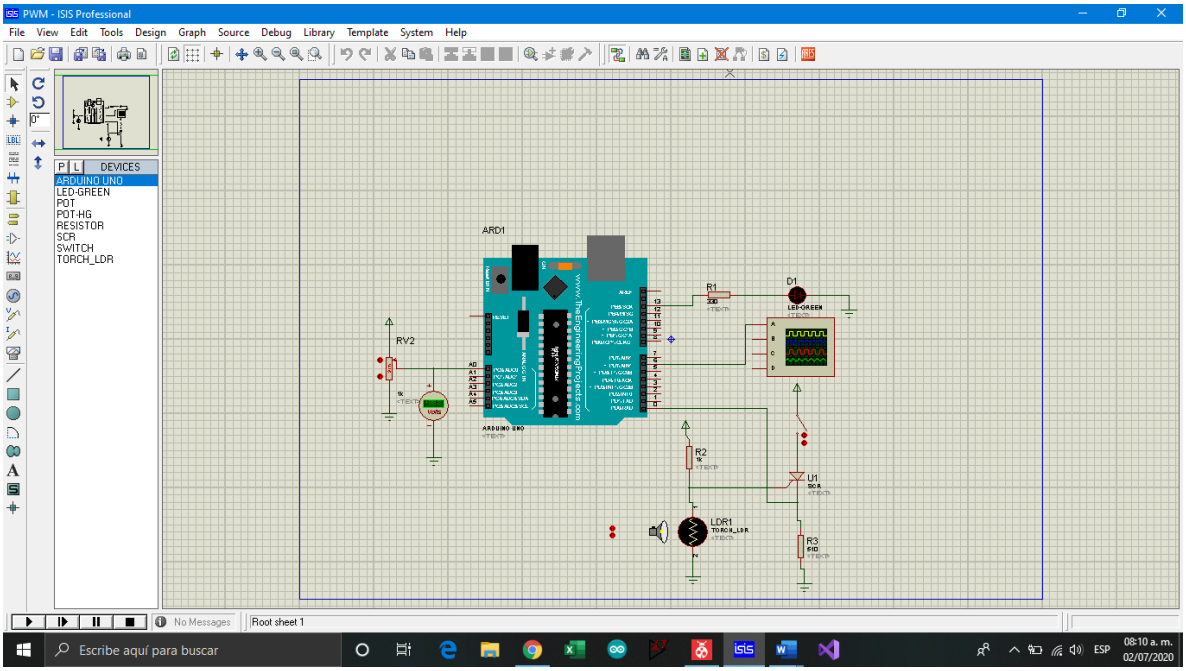
El segundo grupo de instrucciones se encarga de configurar el adc, cosas como la referencia de voltaje, el pin de lectura, etc. Además de una variable de voltaje que servirá para almacenar lo obtenido del adc.



Ya dentro del bucle while, el primer grupo de instrucciones indica que se inicializa el adc, se genera la conversión y después de tenerla se almacena en la variable voltaje y dependiendo del valor de dicha variable el timer0 generará PWM con diferente ancho de pulso.

El segundo grupo es un if pertenece al circuito de la alarma anti intrusos, la cual manda un pulso al pind0 y con esto hace que encienda el led del pin b5, para hacer que se apague la alarma es necesario cortar la corriente del scr abriendo el switch.

La siguiente imagen es la interfaz de esta segunda parte, para una mejor comprensión de lo explicado.


PWM - ISIS Professional
File View Edit Tools Design Graph Source Debug Library Template System Help
DEVICES
ARDUINO UNO
LED-GREEN
POT
POT-HG
RESISTOR
SCR
SWITCH
TORCH_LDR
ARD1
RV2
R1
D1
LED-GREEN
R2
U1
LDR1
R3
No Messages
Root sheet 1
Escribe aquí para buscar
ESP
08:10 a. m.
02/07/2020