VNiVERSiDAD
D SALAMANCA

SISTEMA DE ALARMA CON SENSOR DE MOVIMIENTO

Práctica Final Periféricos

Juan José López Gómez

Álvaro García Sánchez

Contenido

1. MOTIVACIÓN

Nuestra motivación con este proyecto está basada en la creación de un sistema de alarma doméstica que incluye un sensor de movimiento, unos diodos led y un zumbador, de modo que, si se detecta cualquier movimiento una vez el sistema esté activado, se enviará un mensaje al email del usuario especificando fecha y hora en la que se ha detectado dicho movimiento y activando también el aviso sonoro incorporado mediante el zumbador.

2. MATERIAL (HARDWARE) EMPLEADO

El material utilizado se encuentra en el apartado de Anexo en la subsección Materiales utilizados

3. SOFTWARE EMPLEADO

a. Lenguajes de programación

- Python: Empleado para comunicarnos con la placa de Arduino y recoger los datos de los sensores y también para enviar los avisos al usuario vía email. Versión 3.9.6

- Arduino: Empleado para poder darle utilidad a los elementos del hardware y programarlos acorde a nuestro sistema.

b. Entornos de programación utilizados

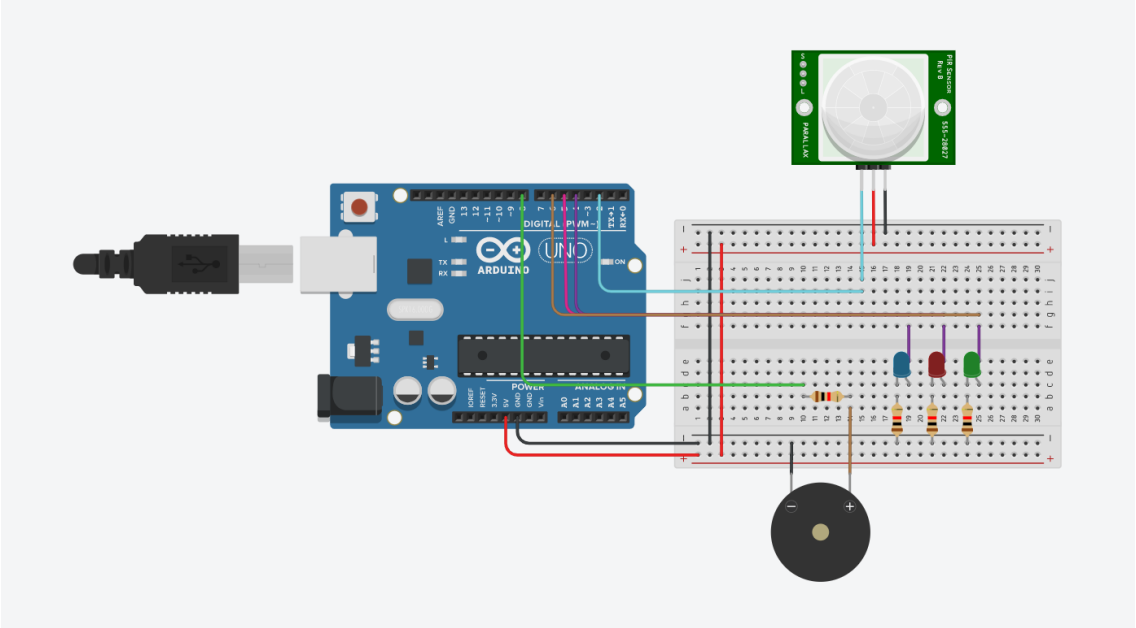
- Visual Studio Code: Empleado para la programación del código python debido a su sencillez y la posibilidad de incluir complementos para el reconocimiento de código.

- Arduino IDE para windows: Empleado para subir el proyecto de arduino a la placa y así comprobar la funcionalidad del mismo en un entorno real.

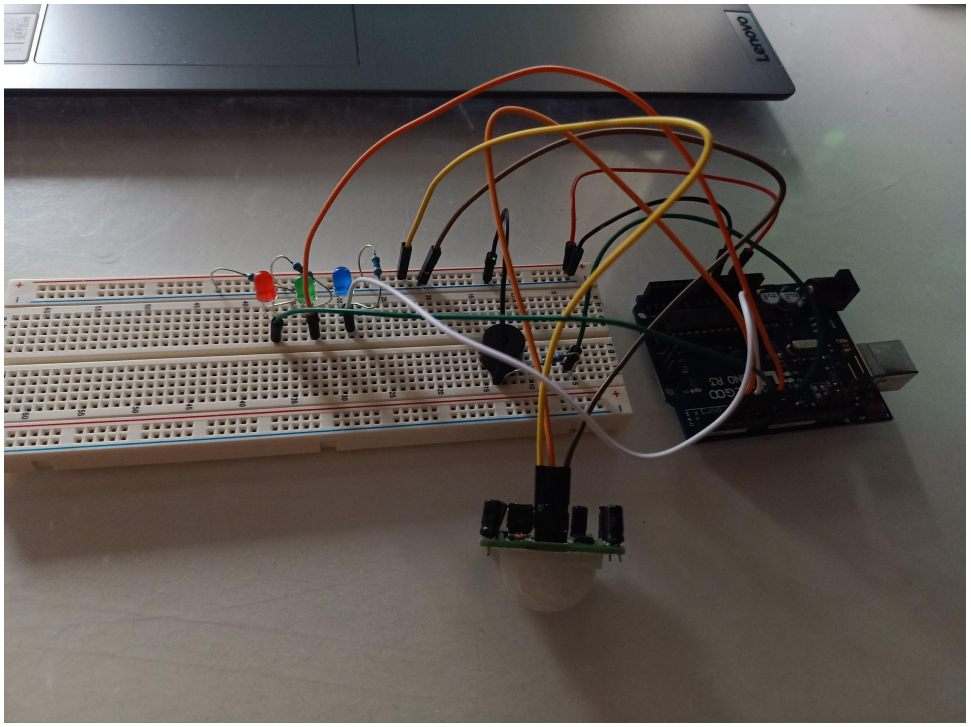
- Tinkercad: Simulador online utilizado para un primer prototipo del proyecto.

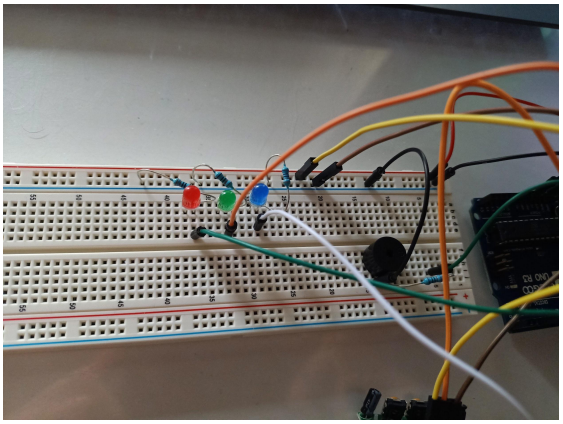
4. IMPLEMENTACIÓN HARDWARE DEL DISPOSITIVO

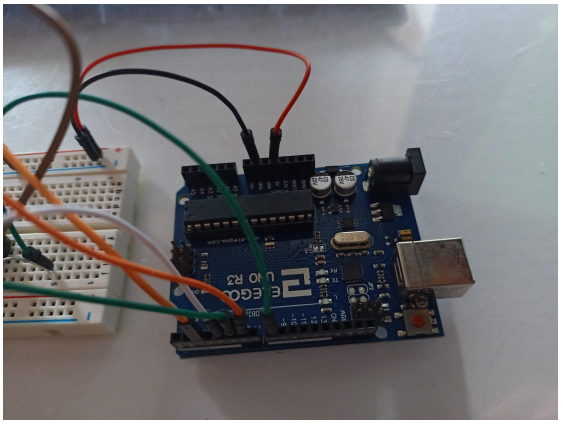
El ensamblaje y conexión de las piezas de nuestro circuito a la placa Arduino se puede ver representados en la siguiente imagen realizada en el simulador Tinkercad.

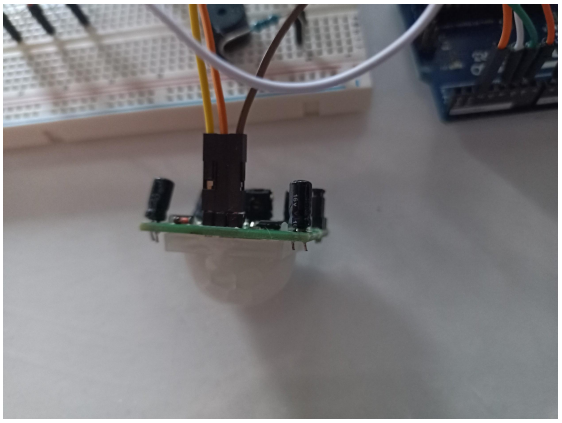


La implementación real del proyecto es como en las siguientes imágenes:









Para la implementación del proyecto hemos utilizado las conexiones siguientes:

Para los LEDs, el ánodo de cada uno está conectado a un pin digital del microcontrolador, mientras que el cátodo está conectado en serie con una resistencia que a su vez está conectada a tierra para que le llegue el voltaje correcto al LED y así no quemarlo.

En cuánto al zumbador dos cables, el unido a tierra y a una resistencia para el correcto funcionamiento, y el positivo a un pin digital del microcontrolador.

El sensor de movimiento tiene tres pines, izquierdo es el que se conecta a la corriente, el central es el que genera la señal al detectar movimiento por lo que va a un pin digital y el derecho está conectado a tierra.

5. CÓDIGO

Para la implementación del Código hemos optado por dividirl varios ficheros para cada componente del hardware. Siendo así, especificamos el funcionamiento individual de cada uno de los ficheros:

- Main.ino: Programa principal con la plantilla base de arduino (setup(), loop()), en el cual se van a inicializar tanto el modo de los pines y el puerto serie. Seguidamente se inicia el bucle de ejecución del proyecto.

- Main.h: Archivo de cabecera en el que se definen todas las variables a utilizar por el resto de los archivos y se incluye la librería de “pitches.h” que se encarga de la definición de las notas musicales para el zumbador.

- Pitches.h: Archivo de cabecera en el que se van a definir las macros a utilizar con el buzzer esto es opensource a través de la página de arduino.

- Sensor.ino: Se encarga de dejar el programa a la espera y en caso de que reciba movimiento el sensor, cambiará los leds de color y devolverá un valor a main.ino, para continuar con la ejecución del programa.

- Message.ino: Se encargará de comunicarse por el puerto serie con el script de python para poder enviar un email al correo del usuario y también se encargará de encender y apagar el led azul, mientras que no haya recibido la confirmación por el puerto serie.

- Buzzer.ino: Se encargará de hacer sonar el zumbador con la melodía que deseemos una vez se haya completado el proceso de notificación.

- Main.py: Script de pyhon que usaremos para comunicarnos con el puerto serie de nuestro sistema y enviar un email al usuario especificado mediante protocolo smtp, dados unos parámetros de usuarios predeterminados.

6. FUNCIONAMIENTO

Nuestro sistema funcionará y actuará de la siguiente manera dependiendo de los distintos casos que se pueden dar. El funcionamiento del proyecto se puede comprobar en el vídeo demostración subido a studium, el vídeo consta de dos partes: la primera el funcionamiento normal mostrando la consola de python y la bandeja de entrada del correo, y la otra es de más cerca para apreciar el sonido del zumbador ya que con el ruido que había no se apreciaba correctamente.

El funcionamiento es el siguiente: primeramente debemos cargar en el arduino el proyecto, se encenderá el LED verde (indicando sin actividad), y seguidamente hay que ejecutar el código python (importante el orden y no tener abierto el monitor serie del IDE de arduino ya que no se permite tener dos abiertos dos a la vez). Una vez hecho esto, la placa se reinicia y ya están sincronizados tanto el arduino como el script python.

Cuando se detecta el movimiento se apaga el LED verde y se enciende el Rojo, mientras que se manda por el puerto serie el mensaje hacia el script, que lo recoge y empieza a generar los datos necesarios. Mientras el microcontrolador alterna el LED azul, indicando un estado de espera.

Cuando se ha enviado correctamente el email, se envía una confirmación por el puerto serie, la placa la recibe y cambia el estado, se apagan todos los LEDs y el zumbador emite la melodía.

Cuando todo este proceso ha finalizado se vuelve al estado de parada (LED verde activo) y el script python a la espera de datos.

7. POSIBLES MEJORAS

Alguna mejora que se podría haber introducido es incorporar una especie de cámara de vigilancia que se active cuando el script ha recibido información con la cual se pueda grabar y en cierto modo enviar en el correo un link para poder ver la transmisión en directo.

Otra mejora posible sería el uso de hilos, para poder recibir más avisos de detección mientras se está enviando el correo, ya que con nuestro diseño hay “puntos ciegos”, además de configurar el sensor de movimiento para que solo envíe señal cuando cambie de estado y no mientras se detecta el movimiento.

8. REFERENCIAS

- Tutorial del sensor de movimiento, https://mschoeffler.com/2021/05/15/arduino-tutorial-hc-sr501-passive-infrared-sensor-pir-sensor/

- DataSheet del sensor, https://components101.com/sensors/hc-sr501-pir-sensor

- Tutorial del uso del sensor, https://www.makerguides.com/hc-sr501-arduino-tutorial/

- Funciones implicitas en arduino, https://www.arduino.cc/reference/en/language/functions/advanced-io/tone

- Tutorial de como utilizar el zumbador, http://arduino-tutorials.eu/creating-sounds-with-arduino-buzzer

- Posible melodía para el buzzer, https://create.arduino.cc/projecthub/Raushancpr/3-some-noise-with-buzzer-f2b2fa

- Posible melodía para el buzzer, https://create.arduino.cc/projecthub/410027/rickroll-piezo-buzzer-a1cd11

- Posible melodía para el buzzer, https://create.arduino.cc/projecthub/2-bros-team/arduino-buzzer-coffin-dance-funny-7a6329

- Diapositivas en Studium sobre arduino

9. ANEXOS

a. Materiales Utilizados

- Microcontrolador Arduino UNO
- 830 Tie-Points Protoboard
- Led Verde
- Led Azul
- Led Rojo
- Zumbador Pasivo
- HC-SR501 PIR
- Resistencias * 4
- 10 cables de unión