VNiVERSiDAD
D SALAMANCA
Facvltad D Ciencias

PERIFÉRICOS

Práctica: Práctica Final. Control de Riego

Grupo de laboratorio: Grupo 1

Autores: De la Peña Ramos, Jaime (70921486C)
Martín García, Juan Carlos (70882826T)
Montero Fernández, Alberto (80093303N)

Fecha de la defensa: 22/05/2019

Año académico: 2018 – 2019

Tabla de contenidos

1. Explicación de la práctica.....3

2. Código de ejecución.....40

3. Implementación Software.....41

4. Implementación Física.....43

5. Explicación de su Funcionamiento.....46

6. Referencias.....47

1. Explicación de la práctica

La práctica asignada al grupo 1 del laboratorio, por sorteo, ha sido "Control de riego" de forma obligatoria se pide utilizar los siguientes sensores: flexómetro [1], sensor de humedad y temperatura (DHT11 Biblioteca) [2] y un motor Biblioteca [3].

No obstante, se ha decidido agregar más funcionalidad al sistema planteado, por tanto, se van a agregar más periféricos, estos son:

1. LCD1602 [4]
2. Módulo I2C adaptador LCD1602 Biblioteca [5]
3. Módulo HC-SR501 [6]
4. Modulo I2C Tiny DS3231 AT24C32 Biblioteca [7]
5. LED 5mm [8]
6. Teclado de membrana matricial 4x4 Biblioteca [9]
7. Módulo FC28 o Higrómetro [10]
8. GY-NEO6MV2 Biblioteca [11]
9. ESP8266 Biblioteca [12]
10. Buzzer [13]
11. Power MB V2 [14]
12. Detector de Llama Óptico [15]

Tras conocer los periféricos que serán utilizados para la realización de la práctica, a continuación, se va a proceder a explicar cada uno de ellos y la función que realizarán en el sistema.

Flexómetro

Este sensor de unos 8cm de largo aumenta su resistencia al ser flexionado. El conector tiene un espaciado de 0.1 pulgadas por lo que se puede conectar directamente a una placa protoboard. La forma del sensor es la siguiente.

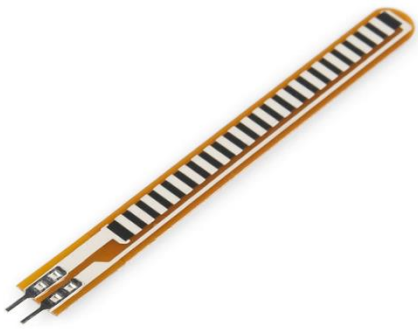

Figura 1. Fléxometro // Sensor de flexión.

Cabe mencionar, que la forma en la cual se debe doblar este sensor es la siguiente.

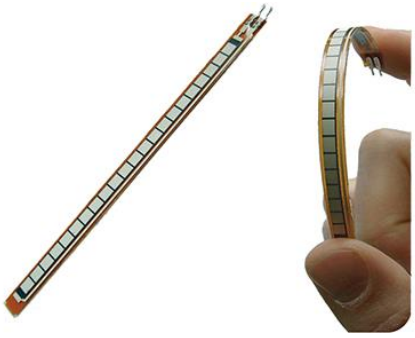

Figura 2. Forma en la que debe doblarse el flexómetro.

Las dimensiones son 8 cm x 0.75 cm, aunque la parte útil que medirá la resistencia doblado es de 6 cm.

El esquema de montaje es sencillo, a continuación, se muestra una imagen de cómo sería el montaje con la herramienta de simulación "Fritzing".

Nota: La resistencia empleada en el ejemplo es de 2.2 kΩ.

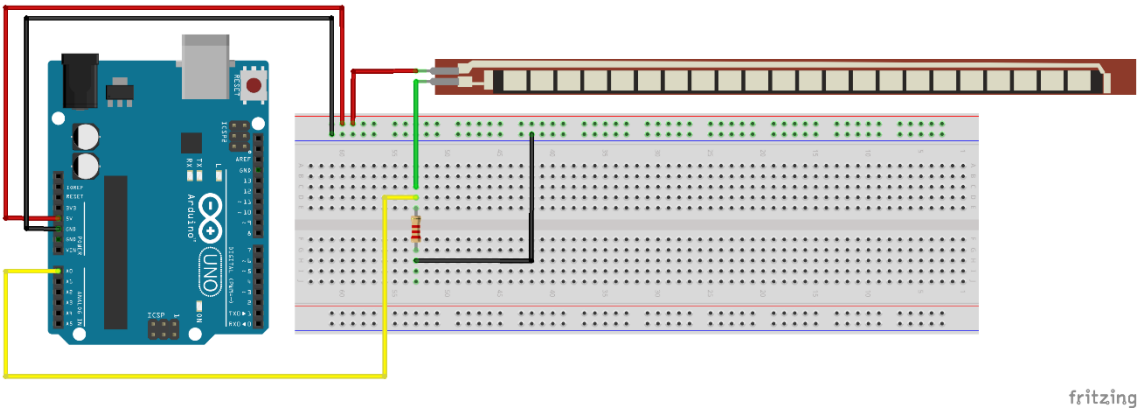

Figura 3. Esquema de montaje flexómetro

El Vcc del flexómetro (línea continua) conectada 5v, el GND a tierra y por último estará comunicada en serie, la entrada analógica A0 a una resistencia de 2.2 kΩ con tierra.

Sensor de humedad y temperatura (DHT11)

El sensor DHT11 permite medir la humedad y temperatura del Arduino, este sensor es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviar la información y, por lo tanto, estará más protegido frente al ruido. El aspecto de este sensor es el siguiente.

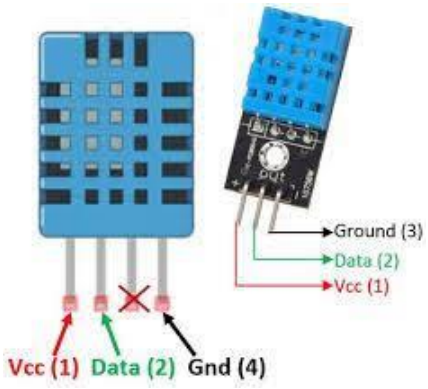

Figura 4. DHT11 // Sensor de humedad y temperatura.

Aunque se conecte a un pin digital, se trata de un dispositivo analógico. Dentro del propio dispositivo se hace la conversión entre analógico y digital. Por lo tanto, se parte de una señal analógica que luego es convertida en formato digital y se enviará al microcontrolador. La trama de datos es de 40 bits correspondiente a la información de humedad y temperatura del DHT11.

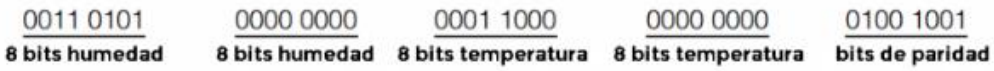

Figura 5. Trama de datos humedad y temperatura enviada por el sensor DHT11.

El primer grupo de 8-bits es la parte entera de la humedad y el segundo grupo la parte decimal. Lo mismo ocurre con el tercer y cuarto grupo, la parte entera de la temperatura y la parte decimal. Por último, los bits de paridad sirven para confirmar que no hay datos corruptos.

Estos bits de paridad lo único que hacen es asegurar que la información es correcta, sumando los 4 primeros grupos de 8-bits. Esta suma debe ser igual a los bits de paridad.

0011 0101 + 0000 0000 + 0001 1000 + 0000 0000 = 0100 1101

Figura 6. Verificación funcionamiento bits de paridad del sensor DHT11.

El esquema de montaje es sencillo, a continuación, se muestra una imagen de cómo sería el montaje con la herramienta de simulación "Fritzing".

Nota: La resistencia empleada en el ejemplo es de 5 kΩ.

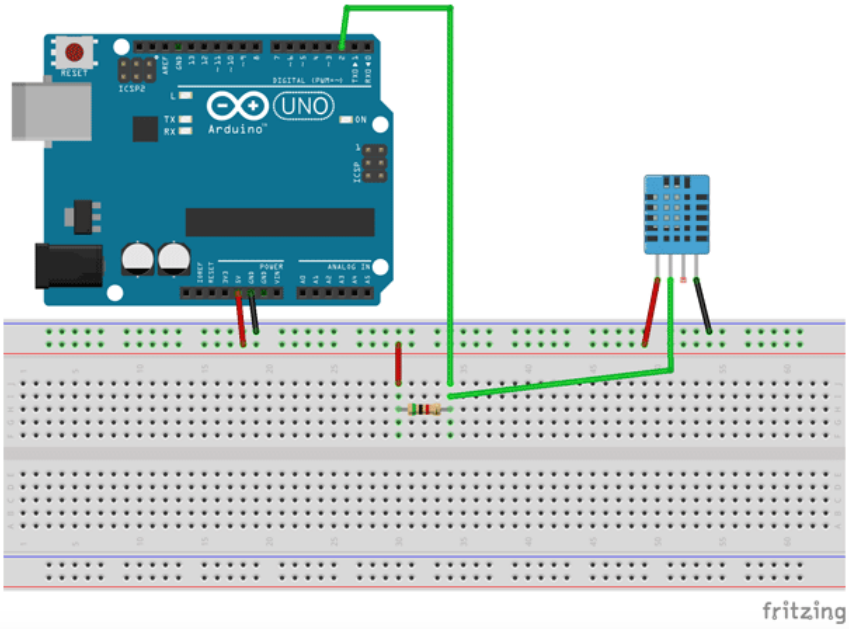

Figura 7. Esquema de montaje DHT11

Para conectar el circuito de una aplicación típica con un DHT11, es necesario tener una resistencia pull-up conectada a la salida digital. La recomendación es utilizar una resistencia de 5 kΩ. Se dispone de 4 pines el VCC (de 3,5V a 5V), la salida digital I/O, el pin no conectado NC y la toma de tierra GND.

Motor (servomotor)

Un servomotor o servo. Es un motor eléctrico, pero con dos características especiales. Por un lado, permite mantener la posición que se le indique, siempre que esté dentro del rango de operación del propio dispositivo. Por otro lado, permite controlar la velocidad de giro, se puede hacer que antes de que se mueva a la siguiente posición espere un tiempo. Hay varios modelos de servomotor con Arduino. En este caso se va a utilizar un Micro Servo 9g SG90 de Tower Pro, cuyo aspecto es el siguiente.

Figura 8. Micro Servo 9g SG90 de Tower Pro // motor.

El ángulo de giro, en este caso, permite hacer un barrido entre -90º y 90º. Lo que viene a ser un ángulo de giro de 180º.

Aunque el servo puede moverse con una resolución de más de 1 grado, este es el máximo de resolución que se puede conseguir debido a la limitación de la señal PWM que es capaz de generar Arduino UNO.

Estos motores funcionan con una señal PWM, con un pulso de trabajo entre 1 ms y 2 ms y con un periodo de 20 ms (50 Hz). Solo se podrá cambiar de posición cada 20 ms.

El esquema de montaje de este componente es el siguiente.

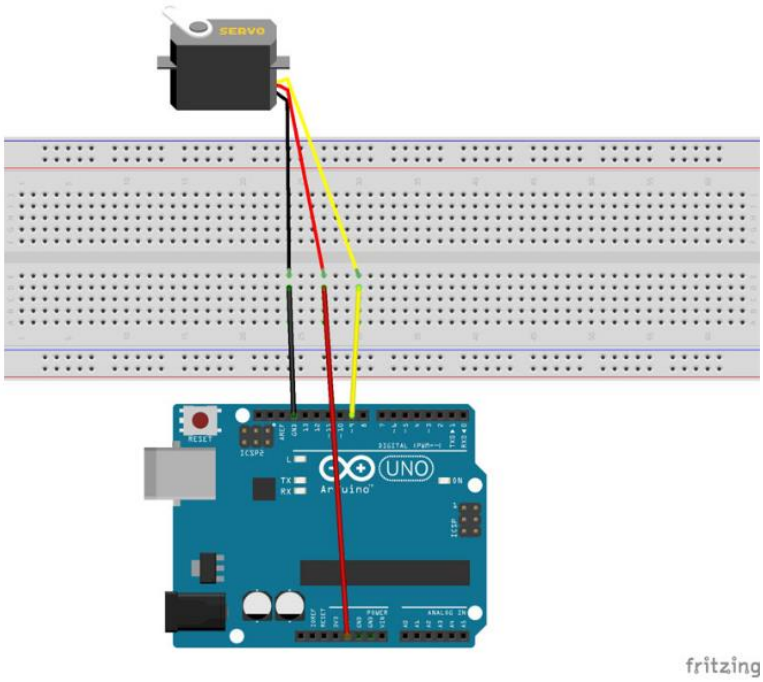

Figura 9. Esquema de montaje servomotor

Todos los servos deben tener 3 cables. Uno irá a tierra (negro/marrón), otro a la alimentación de 5 Voltios (rojo) y el tercero a un pin PWM (amarillo/naranja).

LCD1602

Este componente se encarga de convertir las señales eléctricas de la placa en información visual fácilmente entendible por los seres humanos. Hay una amplia gama de pantallas LCDs que se pueden utilizar con Arduino. Aparte de las funcionalidades extra que puedan dar cada una de ellas, se pueden diferenciar por el número de filas y columnas, su tamaño.

Por ejemplo, una pantalla LCD de 16×1 tendrá una fila de 16 caracteres, es decir, solo se podrán mostrar 16 caracteres simultáneamente, al igual que un LCD de 20×4 tendrá 4 filas de 20 caracteres cada una.

En esta práctica se va a trabajar con una pantalla LCD1602, esto significa que se va a poder mostrar 16 caracteres en la primera fila y los mismos en la segunda fila.

El aspecto de este componente es el siguiente.

Figura 10. LCD1602 // pantalla LCD

En la siguiente imagen, se muestran los pines físicos del LCD.

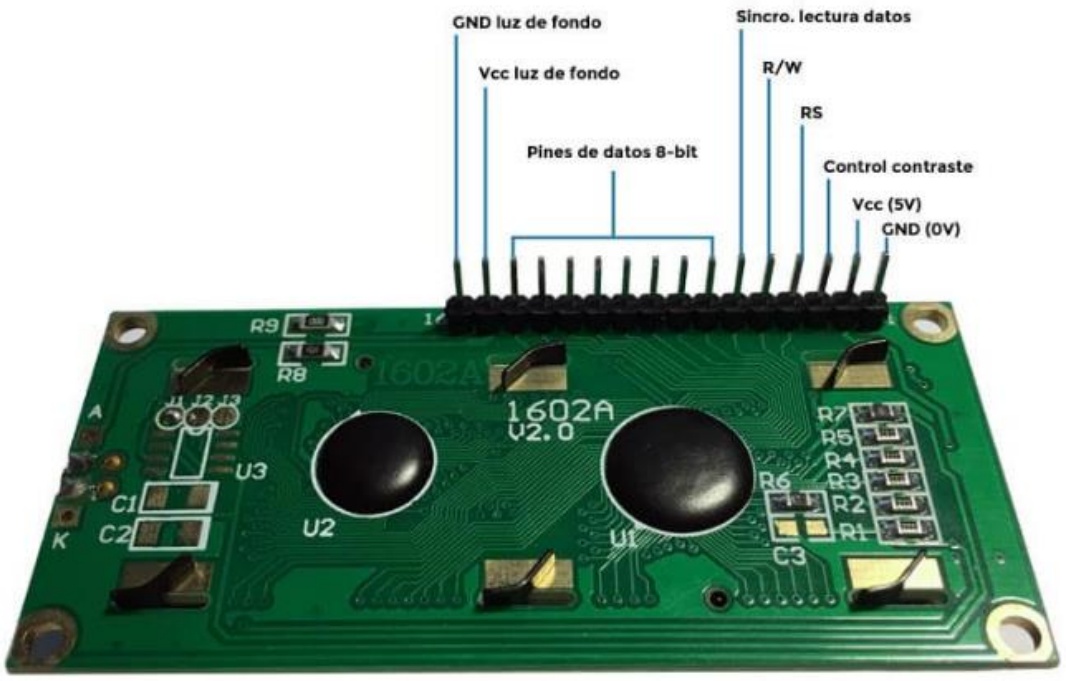

Figura 11. Pines físicos del LCD1602

Donde la función de los pines es la siguiente.

PIN	**FUNCIÓN**
1	GND (Tierra)
2	5 Voltios
3	Control de contraste pantalla
4	RS – Selector entre comandos y datos
5	RW – Escritura y lectura de comandos y datos
6	Sincronización de lectura de datos
7-14	Pines de datos de 8-bit
15	Alimentación luz de fondo (5V)
16	GND (Tierra) luz de fondo (0V)

A continuación, se procede a mostrar el montaje de este componente. Para ellos será necesario, además del LCD, un potenciómetro de 10 kΩ que será utilizado para regular el contraste de la pantalla y una resistencia de 200 Ω que permite regular el voltaje a la entrada de la alimentación del LCD.

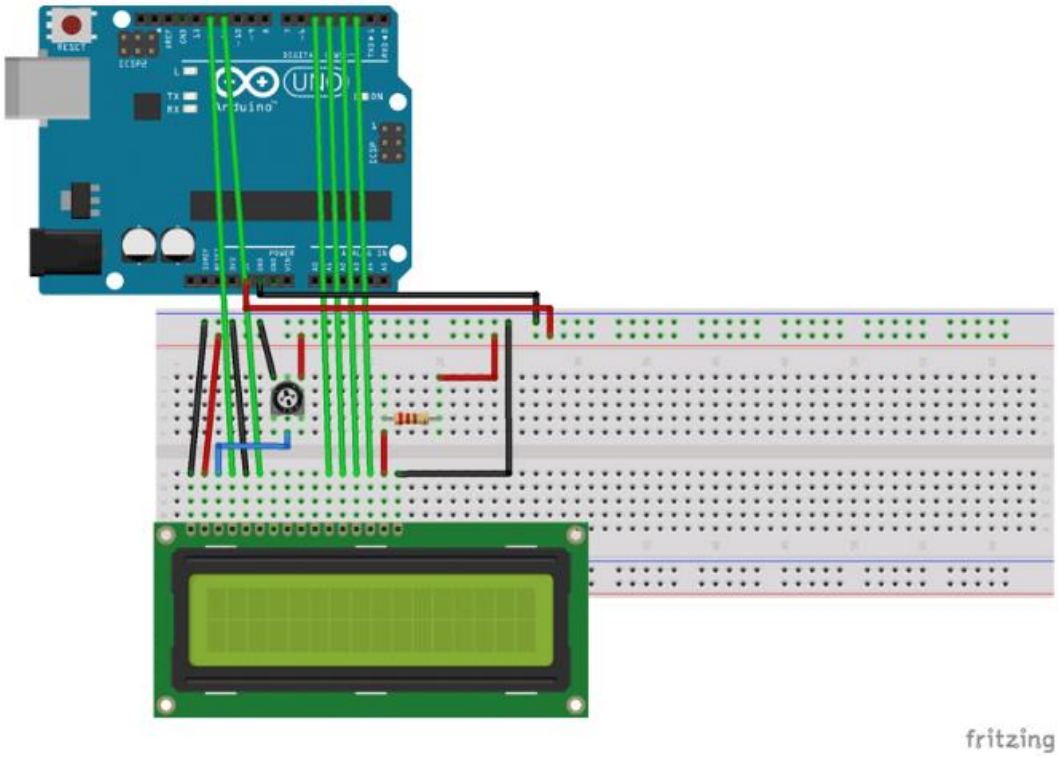

Figura 12. Esquema de montaje LCD1602.

Módulo I2C adaptador para LCD1602

El Módulo adaptador LCD a I2C está basado en el controlador I2C PCF8574, el cual es un Expansor de Entradas y Salidas digitales controlado por I2C. Por el diseño del PCB este módulo se usa especialmente para controlar un LCD Alfanumérico.

La dirección I2C por defecto del módulo puede ser 0x3F o en otros casos 0x27. Es muy importante identificar correctamente la dirección I2C de nuestro modulo, pues de otra forma el programa no funcionará correctamente, el aspecto de este módulo es el siguiente.

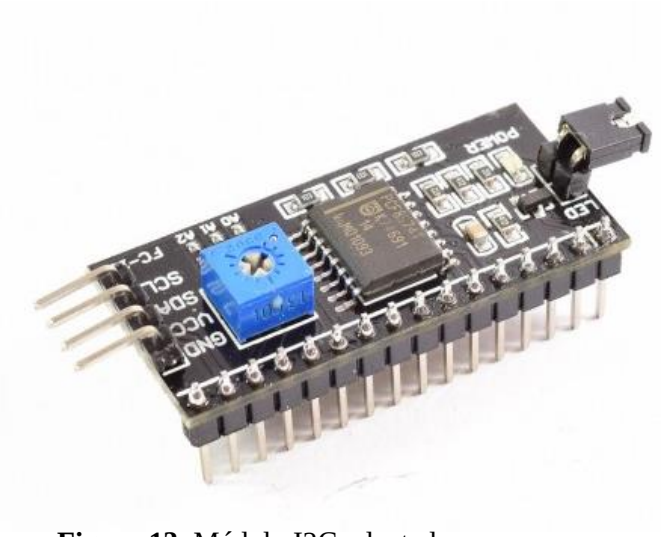

Figura 13. Módulo I2C adaptador para LCD1602.

Para controlar el contraste de los dígitos en el LCD solo se necesita girar el potenciómetro que se encuentra en el módulo.

El esquema de montaje de este módulo con el Arduino y con el LCD es el siguiente.

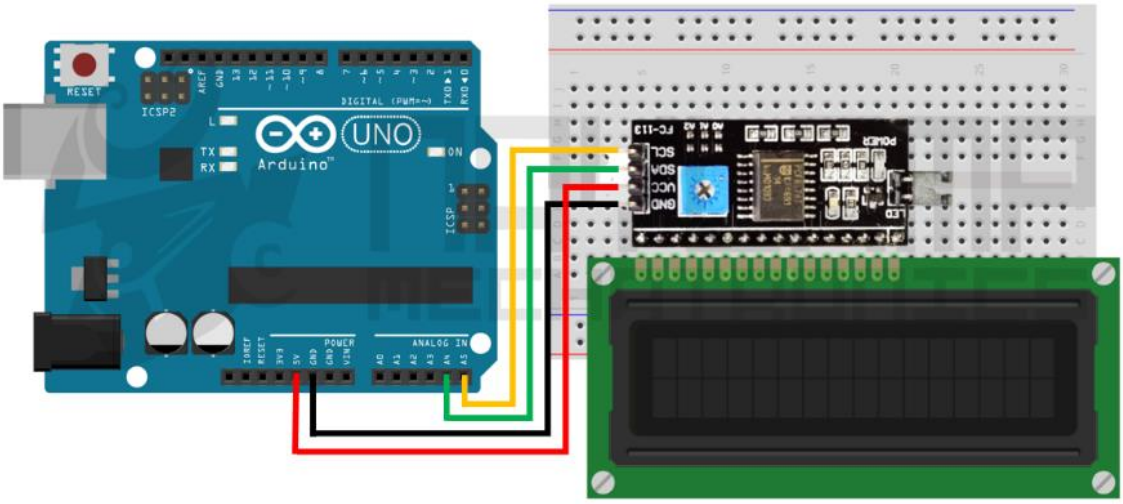

Figura 13. Esquema de montaje módulo I2C adaptador para LCD1602.

Para conectar el módulo con el Arduino solo se utilizan los pines analógicos del Arduino (SDA y SCL) y alimentación (5V y GND), los pines I2C varían de acuerdo al modelo de Arduino con el que se trabaje, en la siguiente tabla se pueden ver cuáles son los pines I2C para cada modelo de Arduino.

Adaptador LCD a I2C	Arduino Uno, Nano, Mini.	Arduino Mega , DUE	Arduino Leonardo
GND	GND	GND	GND
VCC	5V	5V	5V
SDA	A4	20	2
SCL	A5	21	3

Figura 14. Tabla de compatibilidad de pines con el I2C.

HC-SR501

Este sensor, también llamado PIR (sensor infrarrojo pasivo) es un dispositivo utilizado para la detección de movimiento. Los sensores PIR se basan en la medición de la radiación infrarroja. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piezo-eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

Cada sensor está dividido en dos campos y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

El otro elemento restante para que todo funcione es la óptica del sensor. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

El aspecto de este sensor es el siguiente.

Figura 15. HC-SR501 // Sensor PIR.

La conexión es sencilla, simplemente se alimenta el módulo desde Arduino mediante GND y 5V y se conecta un pin digital del Arduino, por ejemplo, el 2, con el pin central de este sensor.

Las conexiones con el Arduino son las siguientes.

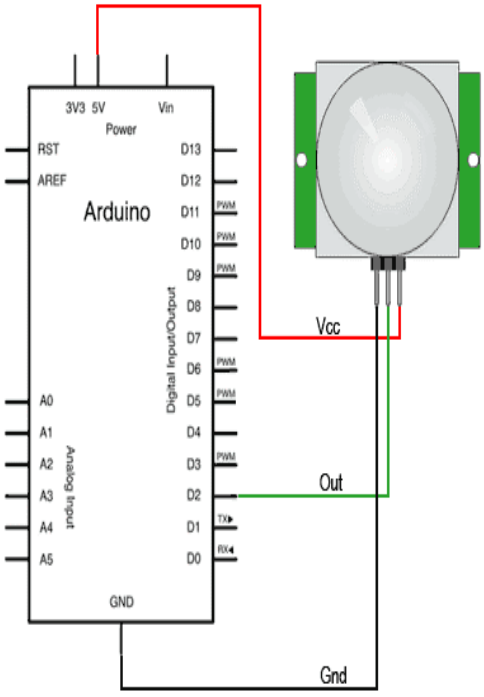

Figura 16. Conexiones HC-SR501 con el Arduino

A continuación, puede verse un esquema eléctrico de la composición de este sensor.

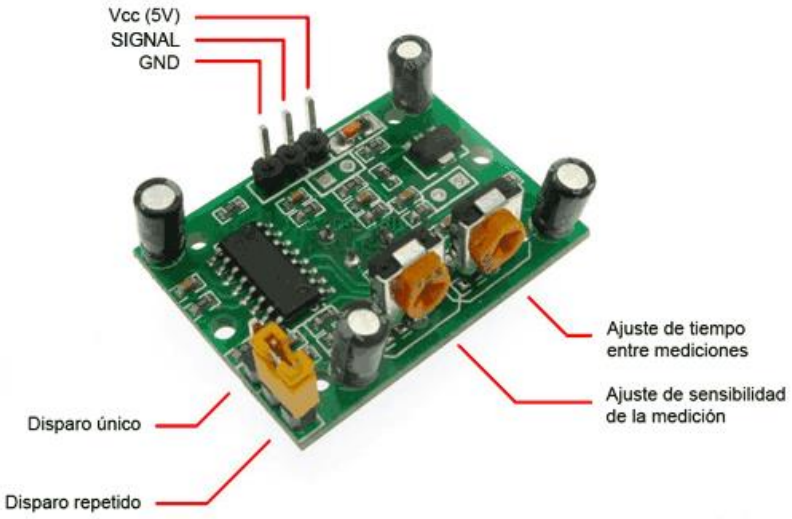

Figura 17. Esquema eléctrico HC-SR501

Finalmente, el esquema de montaje con el Arduino es el siguiente.

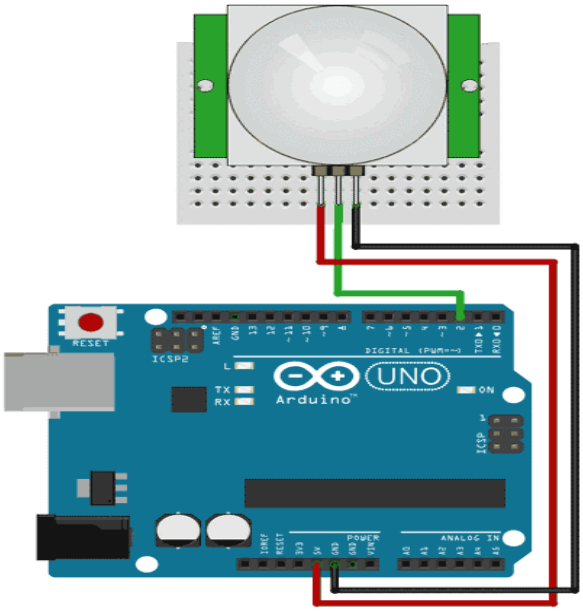

Figura 18. Esquema de montaje HC-SR501

I2C Tiny DS3231 AT24C32

Un reloj de tiempo real (RTC) es un dispositivo electrónico que permite obtener mediciones de tiempo en las unidades temporales que se emplean de forma cotidiana.

Los RTC normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo. La electrónica de los RTC tiene en cuenta las peculiaridades de la forma de medir el tiempo, como por ejemplo el sistema sexagesimal, los meses con diferentes días, o los años bisiestos.

Los RTC aportan la ventaja de reducir el consumo de energía, aportar mayor precisión y liberar a Arduino de tener que realizar la contabilización del tiempo. Además, frecuentemente los RTC incorporan algún tipo de batería que permite mantener el valor del tiempo en caso de pérdida de alimentación.

Existen dos RTC habituales el DS1307 y el DS3231. El DS3231 tiene una precisión muy superior y puede considerarse sustituto del DS1307. En esta práctica se utilizará el modelo DS3231.

También incorporan una batería CR2032 para mantener el dispositivo en hora al retirar la alimentación.

A continuación, se procede a mostrar una imagen del aspecto del DS3231.

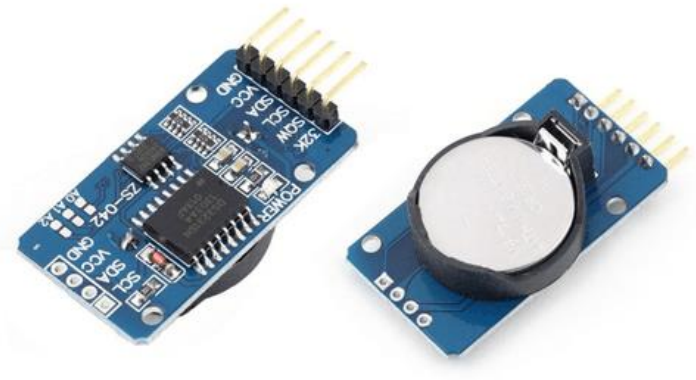

Figura 19. DS3231 // Reloj y calendario

La conexión es sencilla y similar tanto para el DS1307 como el DS3231. Simplemente se alimenta el módulo desde Arduino mediante 5V y Gnd. Por otro lado, se conectan los pines del bus I2C a los correspondientes de Arduino. Las conexiones del modelo DS1307 deberían ser similares a las de la siguiente figura.

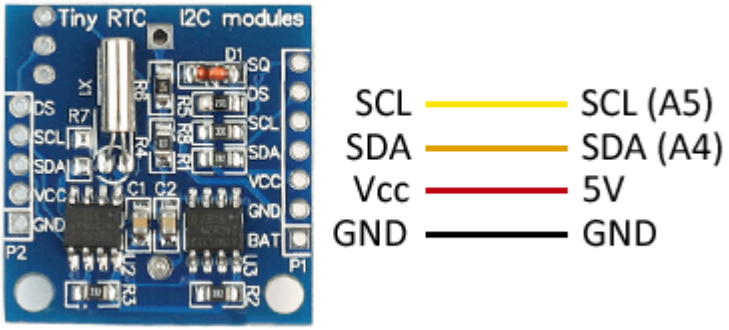

Figura 20. Conexiones DS1307

Y las del modelo DS3231, como las de la siguiente figura.

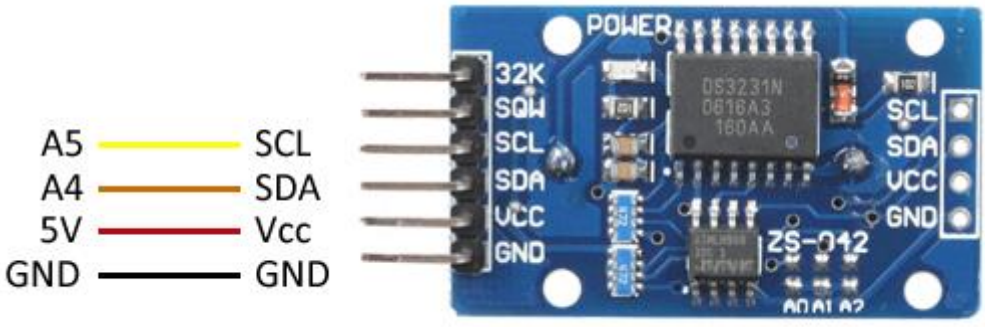

Figura 21. Conexiones DS3231

Finalmente, el esquema de montaje con el Arduino es el siguiente.

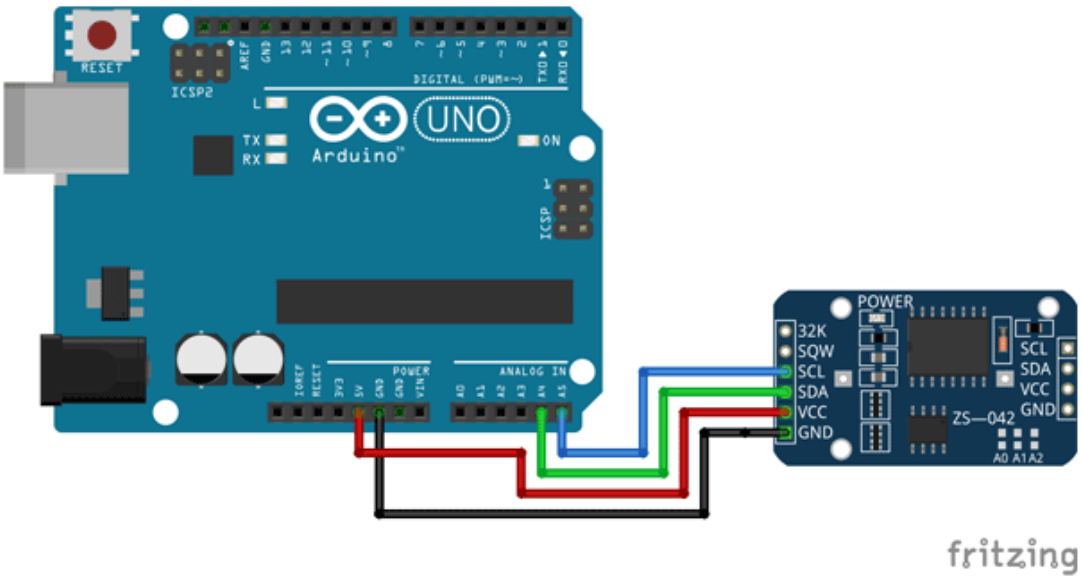

Figura 22. Esquema de montaje DS3231

LED 5mm

Este elemento se trata de un diodo emisor de luz, también conocido por sus siglas en inglés, LED (light-emitting diode), es una fuente de luz constituida por un material semiconductor dotado de dos terminales. Se trata de un diodo de unión p-n, que emite luz cuando está activado. Cuando el ánodo del plomo de un LED tiene una tensión que es más positiva que su cátodo de plomo por lo menos de caída de tensión directa del diodo de luz, la corriente fluye. Los electrones son capaces de recombinarse con agujeros dentro del dispositivo, la liberación de energía se produce en forma de fotones. Este efecto se denomina electroluminiscencia y el color de la luz (correspondiente a la energía del fotón) se determina por el intervalo de banda de energía del semiconductor.

En función de la longitud de onda que emite cada LED, se pueden diferenciar LEDs de diferentes colores, esta banda de longitud de onda y los colores de los LEDs es la siguiente:

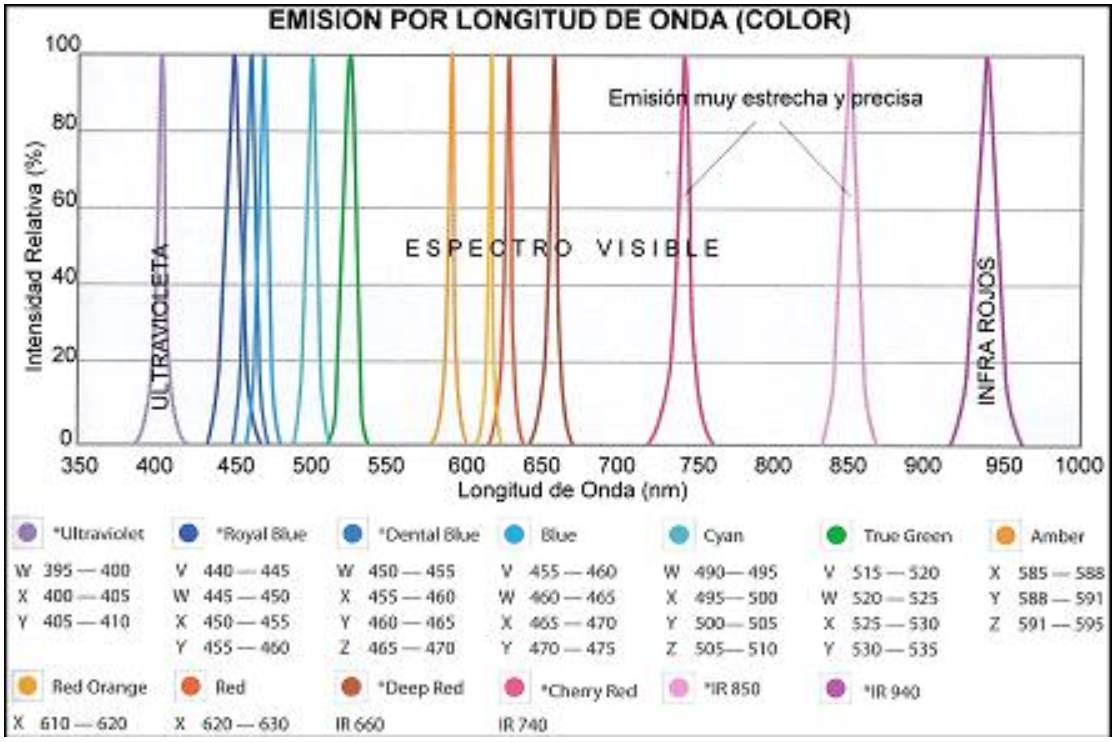

Figura 23. Ondas y color de cada LED.

El aspecto de este componente es el siguiente.

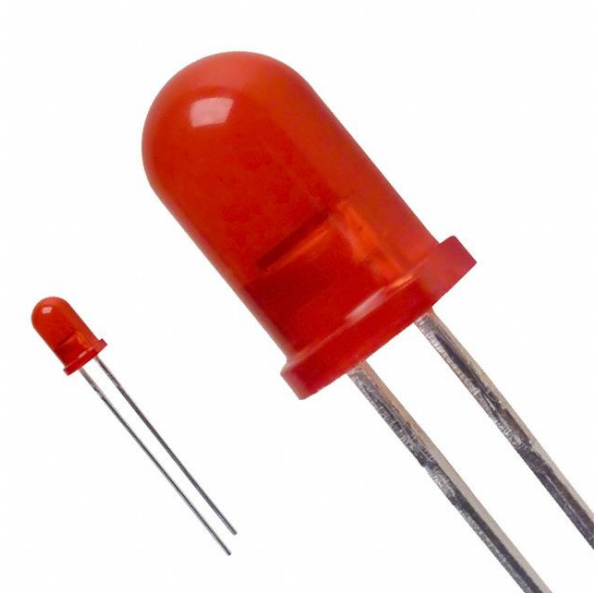

Figura 24. LED rojo 5mm

Las conexiones de este elemento con el Arduino son sencillas y se representan a continuación.

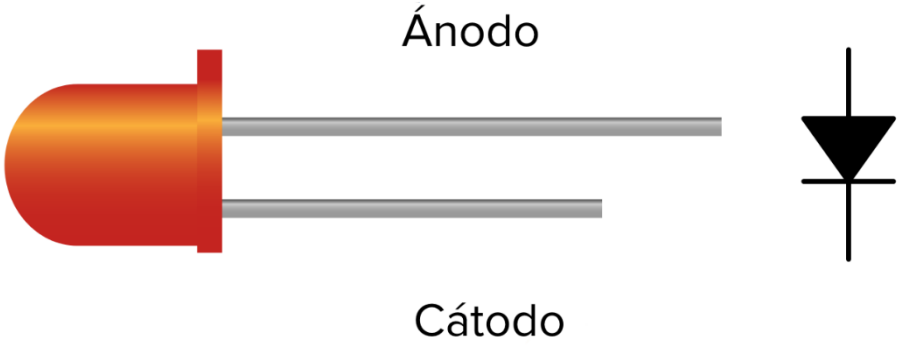

Figura 25. Esquema de montaje LED rojo 5mm

Finalmente, el esquema de montaje con el Arduino es el siguiente.

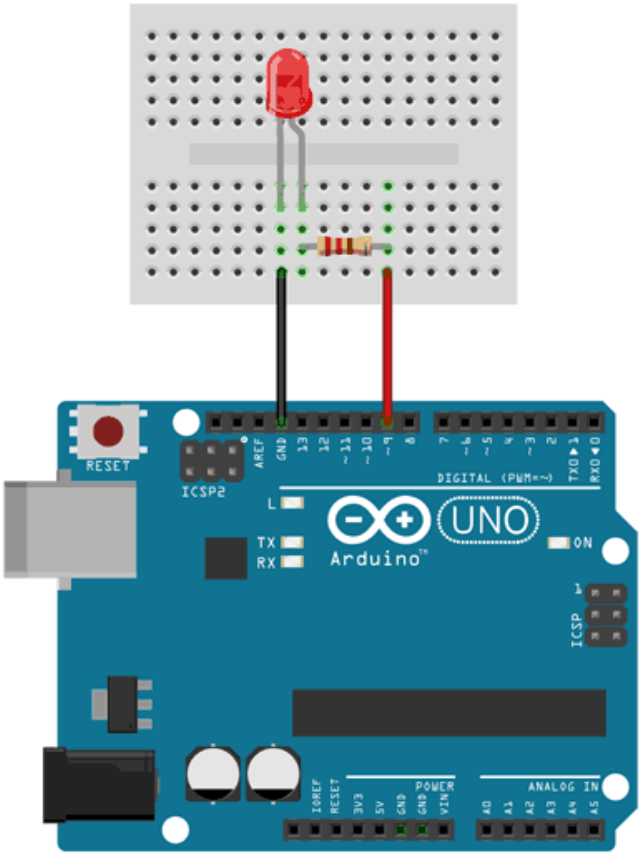

Figura 26. Esquema de montaje del LED con el Arduino.

Teclado de membrana matricial 4x4

Este teclado está compuesto por una matriz de orden 4x4 (4 filas y 4 columnas) donde en cada posición se encuentra un símbolo. Ej. Símbolo [1][3] = B. Por notación, en el ejemplo anterior la matriz comienza en el Símbolo [0][0] = 1.

La representación de este teclado corresponde a la de la siguiente figura.

Figura 27. Teclado de membrana matricial 4x4.

El teclado está formado por pulsadores en cada una de las posiciones de la matriz, estos pulsadores están organizados por filas y columnas, de este modo como resultado final forman la matriz 4x4. Por lo tanto, al detectar la pulsación en la fila i columna j, se puede saber que se ha pulsado la tecla [i][j]. Por lo tanto, para detectar la pulsación de una tecla se actuará de la misma forma que en la detección de un pulsador (1 -> ON ó 0 -> OFF). Para leer todas las teclas, inicialmente hay que realizar un barrido por filas. En primer lugar, se ponen todas las filas a 5V, se pasa a la siguiente fila y así de forma sucesiva hasta recorrer todas las filas.

Internamente la disposición de los pulsadores es la siguiente.

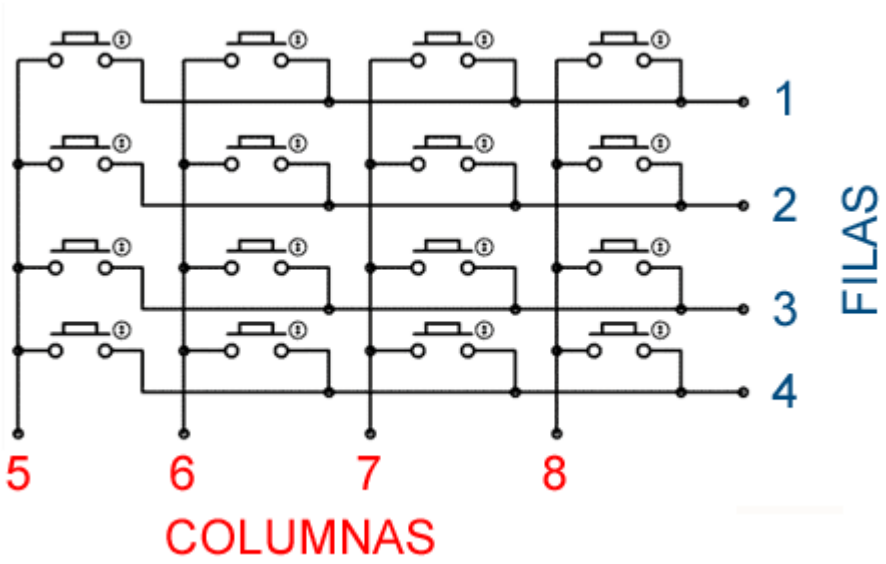

Figura 28. Disposición de los pulsadores en un teclado de membrana 4x4.

El papel de filas y columnas es intercambiable, pudiendo realizar un barrido por las columnas, y lectura en las filas.

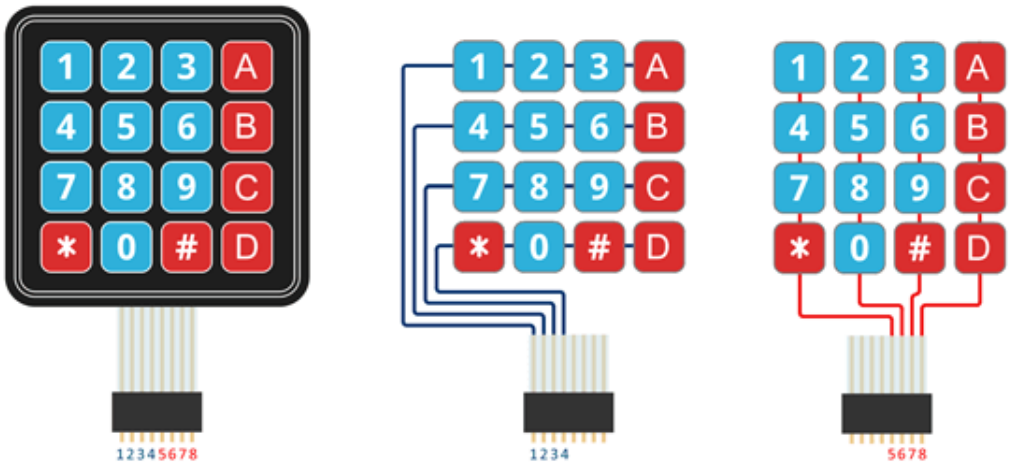

Figura 29. Conexiones filas y columnas en un teclado de membrana 4x4.

Las conexiones de este teclado con el Arduino son sencillas, simplemente se conectan todos los pines a entradas digitales del Arduino, estas se representan a continuación.

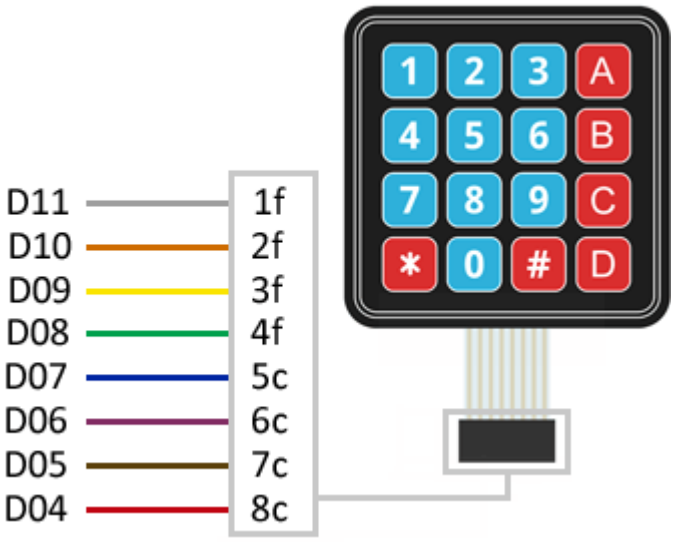

Figura 30. Conexiones teclado de membrana 4x4.

Finalmente, el esquema de montaje con el Arduino es el siguiente.

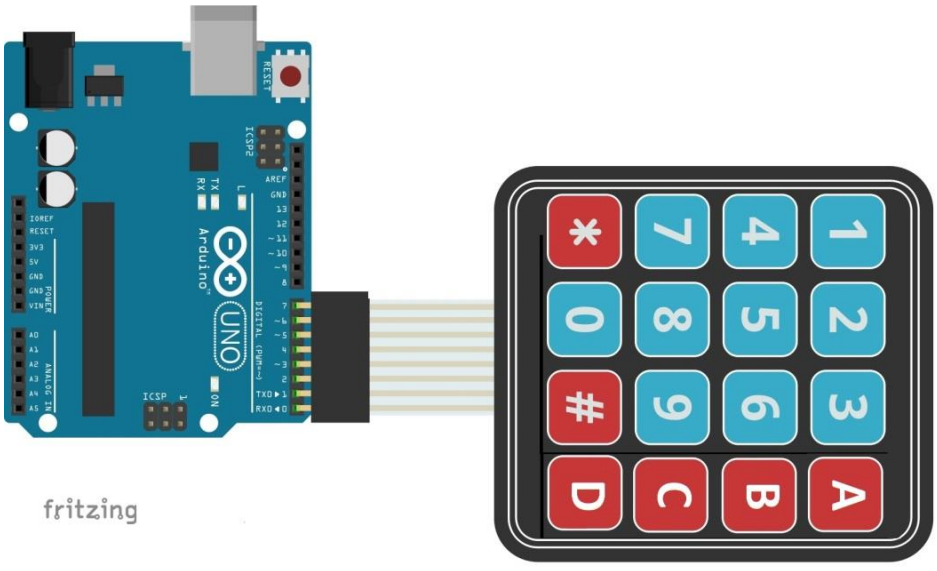

Figura 31. Esquema de montaje teclado matricial 4x4.

Módulo FC28 o Higrómetro

Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo.

El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo

El FC-28 se distribuye con una placa de medición estándar que permite obtener la medición como valor analógico o como una salida digital, activada cuando la humedad supera un cierto umbral.

Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmero daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023.

La salida digital se dispara cuando el valor de humedad supera un cierto umbral, que se ajusta mediante el potenciómetro. Por tanto, se obtendrá una señal LOW cuando el suelo no está húmedo, y HIGH cuando la humedad supera el valor de consigna.

El valor concreto dependerá del tipo de suelo y la presencia de elementos químicos, como fertilizantes.

A continuación, se muestra una figura de como sería la representación de este sensor.

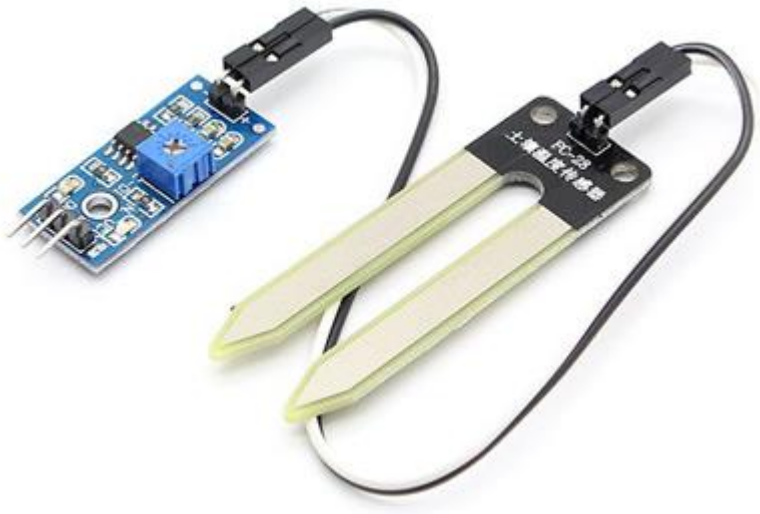

Figura 32. FC28 // Higrómetro

El esquema eléctrico es sencillo. Se alimenta el módulo conectando GND y 5V a los pines correspondientes de Arduino.

Si se quiere usar la lectura analógica, hay que conectar la salida A0 a una de las entradas analógicas de Arduino.

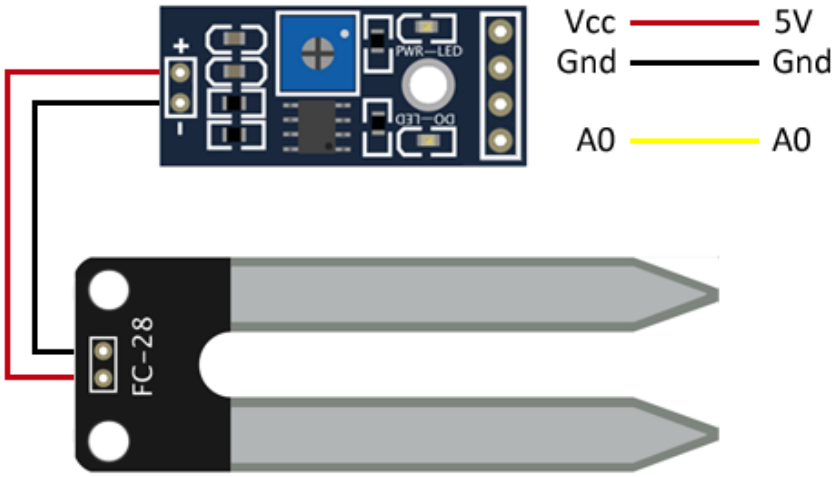

Figura 33. Conexiones FC28.

Finalmente, el esquema de montaje con el Arduino es el siguiente.

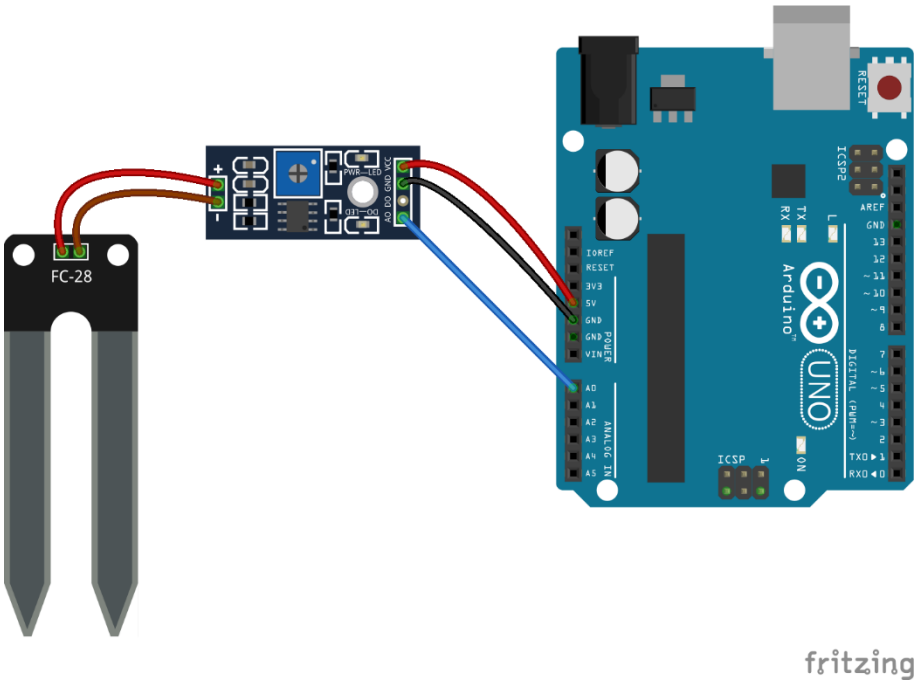

Figura 34. Esquema de montaje FC28.

GY-NEO6MV2

Los dispositivos NEO-6 son una familia de receptores, disponen de interfaz de comunicación UART, SPI, DDC (I2C) y USB. Soportan los protocolos NMEA, UBX binary y RTCM.

La familia de receptores GPS NEO-6 están diseñados para tener un pequeño tamaño, pequeño coste, y pequeño consumo. La intensidad de corriente necesaria es de unos 37 mA en modo de medición continuo.

La tensión de alimentación es de 2.7 a 3.6V para los modelos NEO-6Q/6M, y 1.75-2.0V para los modelos NEO-6G.

El GPS NEO-6 tiene un tiempo de encendido “cold” y “warm” de unos 30 segundos, y en “hot” de 1 segundo. La frecuencia máxima de medición es de 5 Hz.

La precisión que en posición es de 2.5 m, en velocidad 0,1 m/s y en orientación 0.5º, valores más que aceptables para un sistema de posicionamiento GPS.

La representación de este receptor GPS es la siguiente.

Figura 35. GY-NEO6MV2 // Receptor GPS

La conexión es sencilla, en primer lugar, se alimenta el módulo conectando Vcc y Gnd del módulo, a 5 V y Gnd de Arduino.

En esta práctica se utilizará el Pin 3 para RX y el Pin 4 para TX. El esquema es el siguiente.

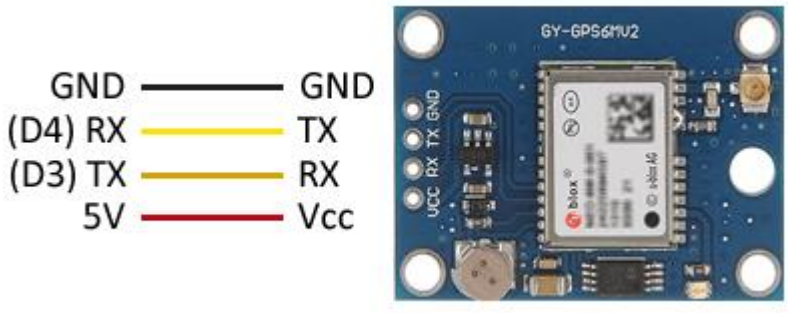

Figura 36. Conexiones GY-NEO6MV2.

Finalmente, el esquema de montaje con el Arduino es el siguiente.

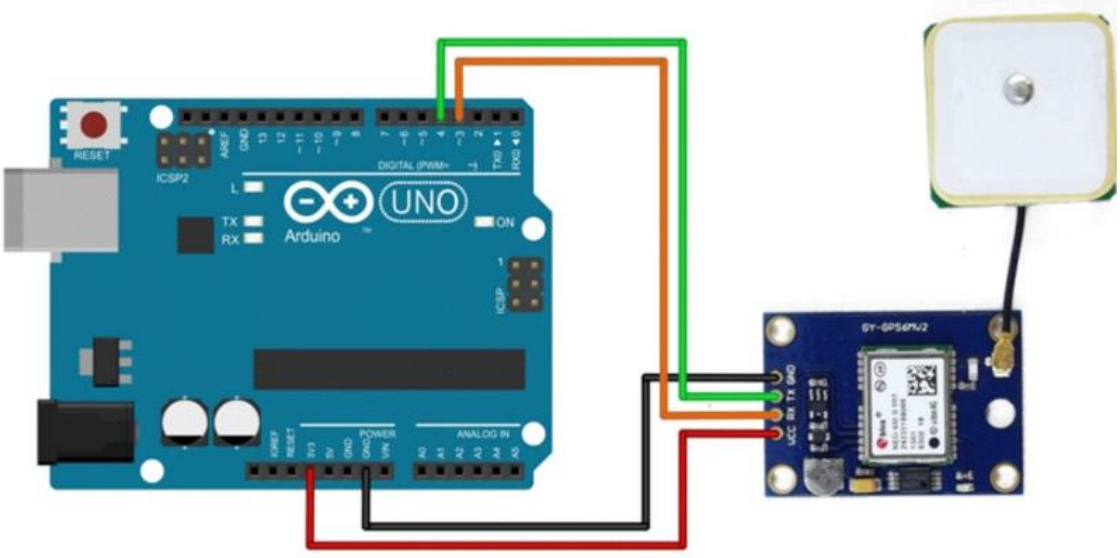

Figura 37. Esquema de montaje GY-NEO6MV2.

ESP8266

El ESP8266 es un microprocesador de bajo coste con Wi-Fi integrado. El ESP01 monta la versión más sencilla del ESP8266. Aun así, sigue siendo un procesador de 32bits a 80 Mhz, con 512kB o 1MB de memoria según modelo. Dispone de 2 pines GPIO, SPI, I2C y UART.

En cuanto a comunicación Wi-Fi, el ESP01 tiene comunicación integrada 802.11 b/g/n, incluidos modos Wi-Fi Direct (P2P) y soft-Ap. Incluye una pila de TCP/IP completa, lo que libera de la mayor parte del trabajo de comunicación al procesador.

A continuación, se procede a mostrar una imagen de cómo sería la representación de este componente.

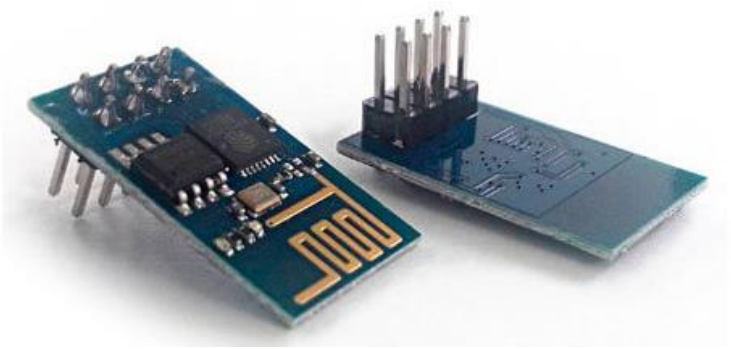

Figura 38. ESP8266 // Móudlo Wi-Fi en ESP01.

Por un lado, el pin CH_PD que apaga o enciende el módulo conectándolo, respectivamente, a Gnd o 3.3V. Por su parte, el pin RST reinicia el módulo si se conecta a Gnd. En algunas versiones del módulo se puede dejarlo sin conexión, pero, en general, se tiene que conectar a 3.3V para que el módulo arranque. Finalmente, la comunicación con el módulo se realiza mediante puerto serie.

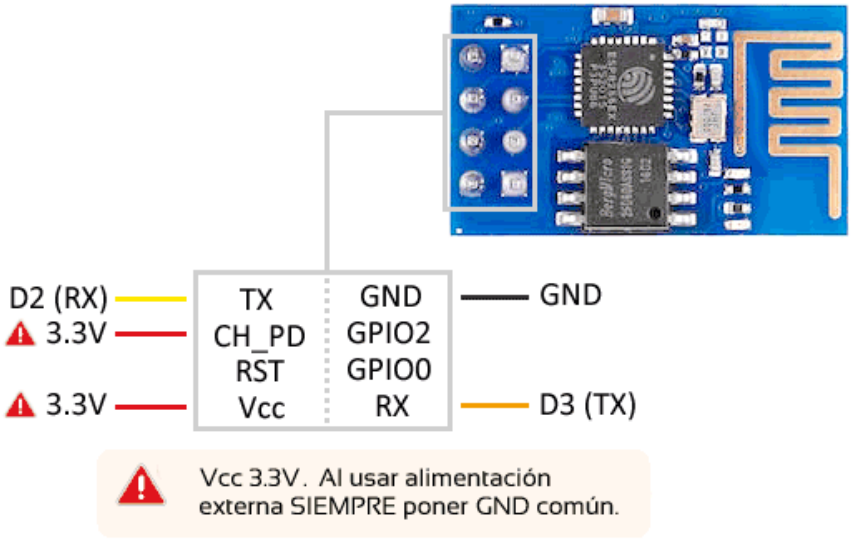

Figura 39. Conexiones ESP8266.

Finalmente, el esquema de montaje con el Arduino es el siguiente.

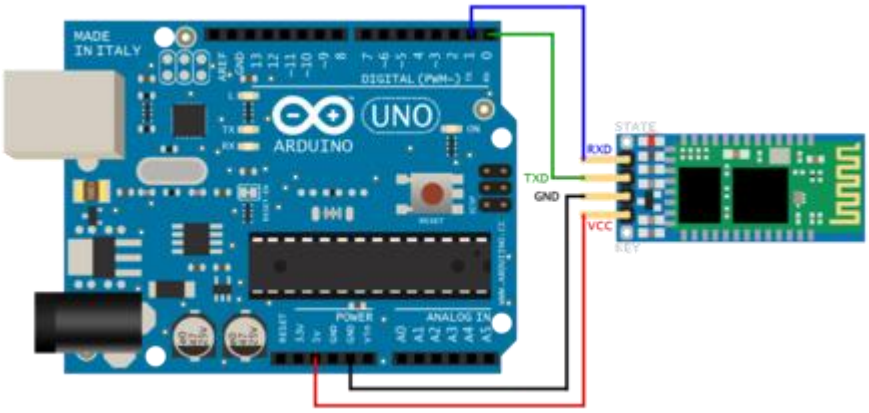

Figura 40. Esquema de montaje ESP8266.

Buzzer

Un buzzer pasivo o un altavoz son dispositivos que permiten convertir una señal eléctrica en una onda de sonido. Estos dispositivos no disponen de electrónica interna, por lo que es necesario que proporcionar una señal eléctrica para conseguir el sonido deseado.

En oposición, los buzzer activos disponen de un oscilador interno, por lo que únicamente es necesario alimentar el dispositivo para que se produzca el sonido.

Pese a tener la complejidad de tener que proporcionar y controlar la señal eléctrica, los buzzer pasivos y de los altavoces tienen la ventaja de que pueden variar el tono emitido modificando la señal que se aplica al altavoz, lo que permite generar melodías.

A continuación, se procede a mostrar una imagen de este componente.

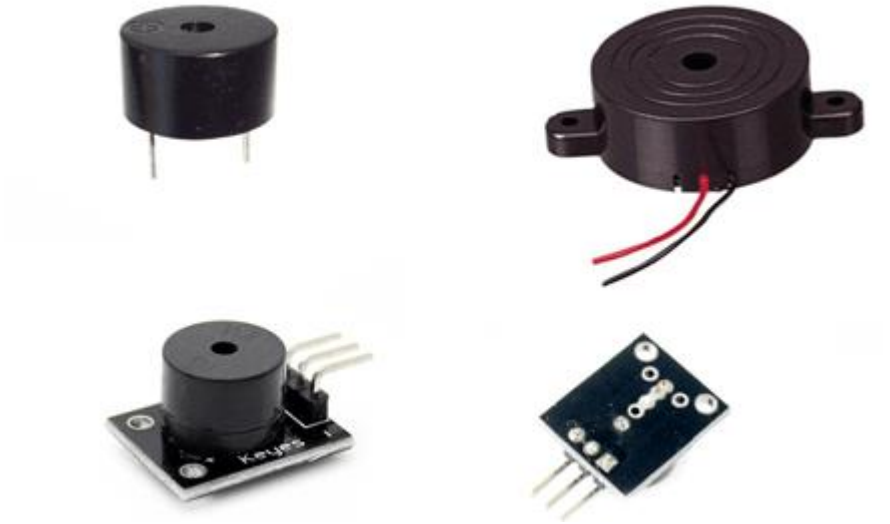

Figura 41. Buzzer // Activo o Pasivo

Las conexiones del buzzer con el Arduino son sencillas, simplemente hay que alimentar el módulo conectando Vcc y GND a Arduino, y la entrada de señal a cualquier salida digital de Arduino.

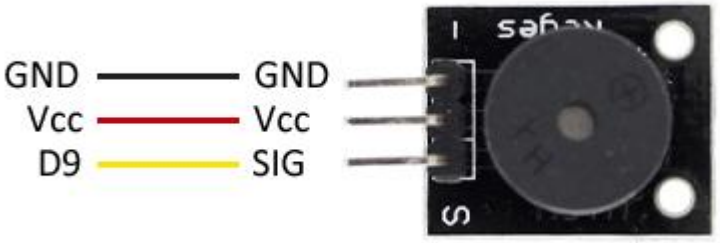

Figura 42. Conexiones Buzzer.

Finalmente, el esquema de montaje con el Arduino es el siguiente.

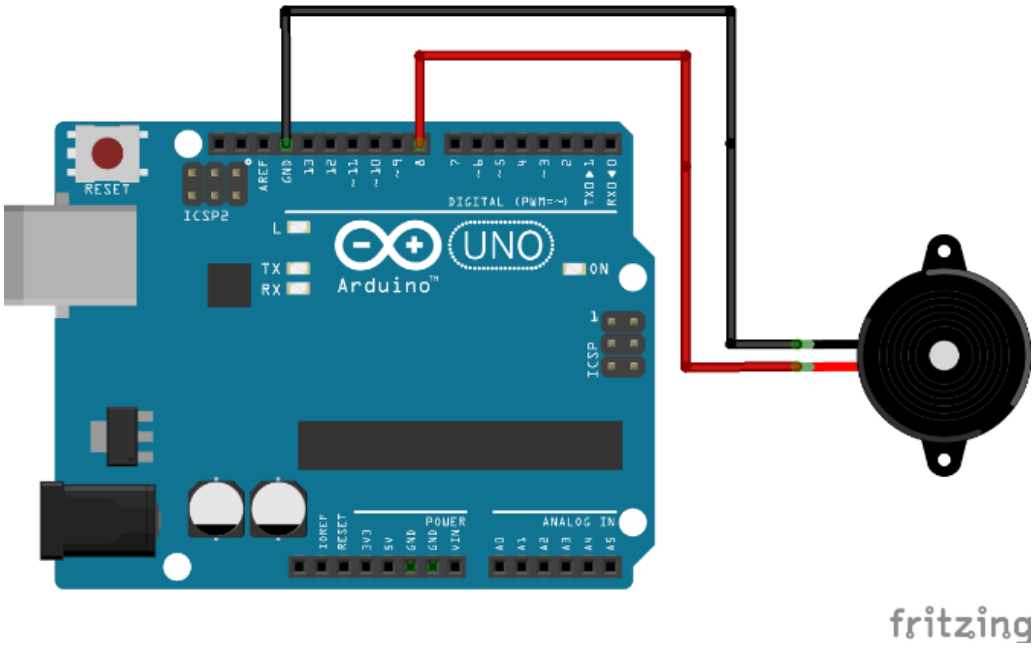

Figura 43. Esquema de montaje Buzzer.

Power MB V2

Módulo que permite alimentar una protoboard MB-102 mediante plug a un transformador de pared hasta 12v. Proporciona dos salidas independientes que son seleccionables mediante jumpers y permiten suministrar 5V o 3.3V. Cuenta además con salida de 5V por conector USB.

Simplifica la alimentación de la protoboard y circuitos electrónicos especialmente los de carácter digital.

Cada lado de la fuente tiene un jumper de encendido / apagado y de selección de voltaje.

A continuación, se procede a mostrar una imagen de este componente.

Figura 44. Power MB V2 // Fuente alimentación

A continuación, se procede a mostrar un esquema eléctrico de este componente.

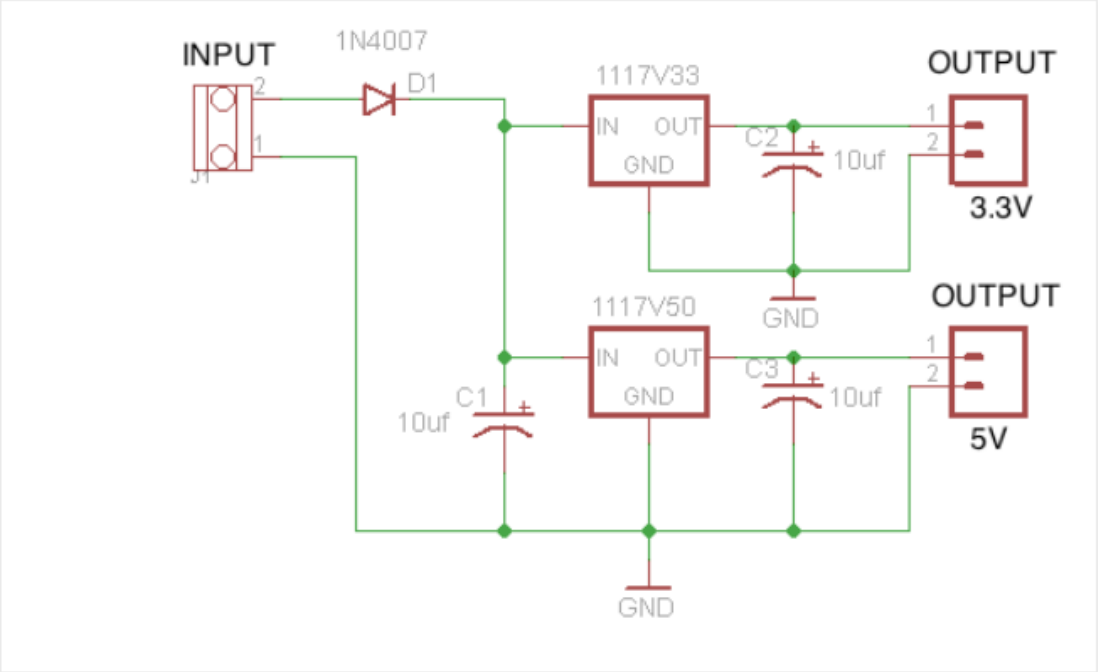

Figura 45. Esquema eléctrico Power MB V2

Detector de Llama Infrarrojo

Un sensor de llama óptico es un dispositivo que permite detectar la existencia de combustión por la luz emitida por la misma. Esta luz puede ser detectada por un sensor óptico, y ser capturado por las entradas digitales y las entradas analógicas de Arduino.

La llama es un fenómeno de emisión de luz asociado a los procesos de combustión. La combustión es un proceso que desprende grandes cantidades de energía en forma de calor. Durante la reacción se generan compuestos intermedios que liberan parte de su energía mediante la emisión de luz.

El espectro de emisión de llama depende de los elementos que intervienen en la reacción. En el caso de combustión de productos con carbón en presencia del oxígeno tenemos dos picos característicos en ultravioleta en longitudes de onda de 185nm-260nm y en infrarrojo en longitudes de onda 4400-4600nm.

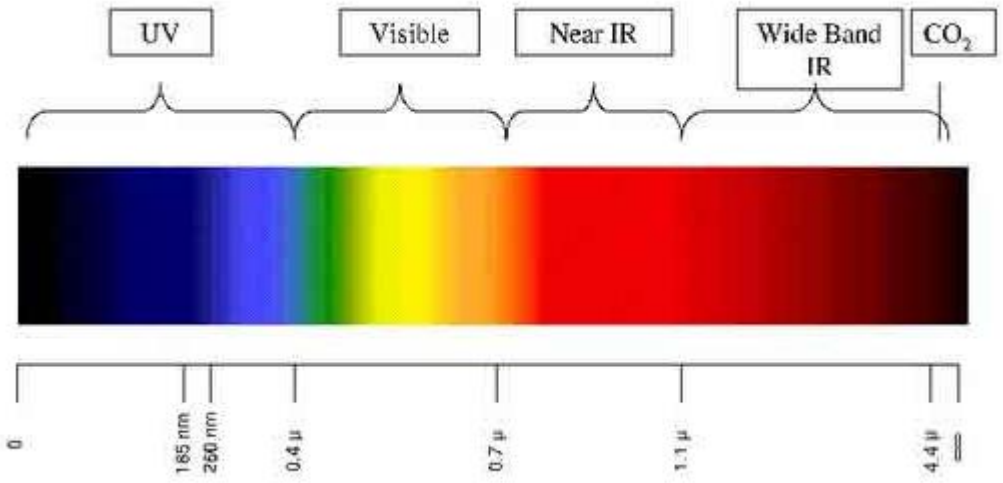

Figura 46. Espectro de ondas.

A continuación, se procede a mostrar una imagen de este componente.

Figura 47. Sensor de Llama
Infrarrojo.

El esquema eléctrico es sencillo. Se el módulo conectando GND y 5V a los pines correspondientes de Arduino. Si se quiere usar la lectura digital, se conecta la salida DO a una de las entradas digitales de Arduino.

Figura 48. Conexiones
detector de Llama Infrarrojo

Finalmente, el esquema de montaje con el Arduino es el siguiente.

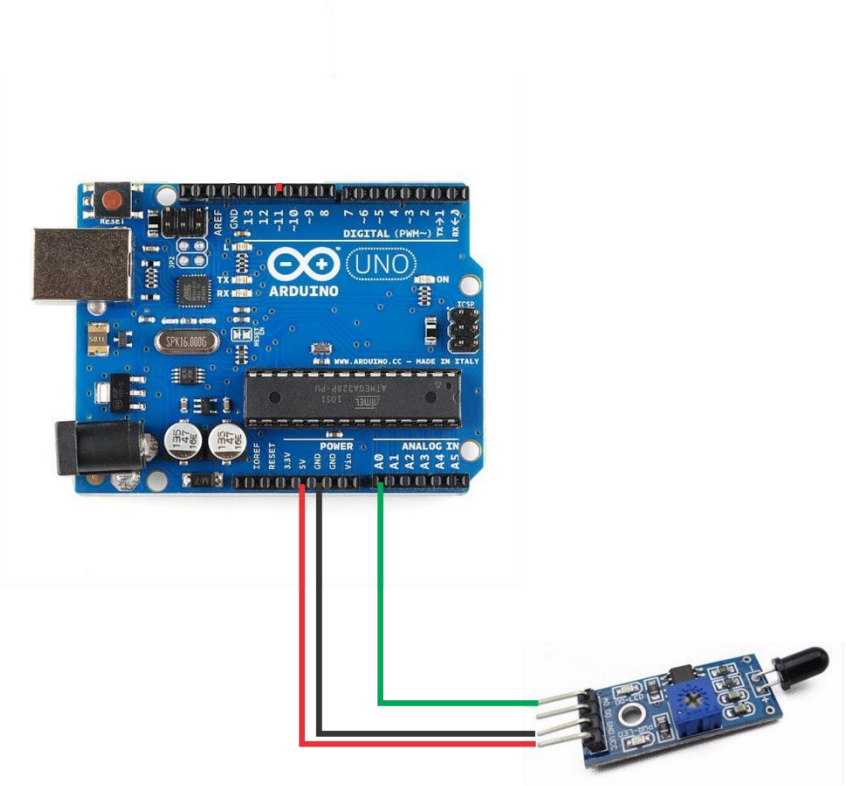

Figura 49. Esquema de montaje del sensor Detector de Llama Infrarrojo

2. Código de ejecución

Debido a la extensa longitud del código empleado para la elaboración de la práctica este se adjuntará en un directorio llamado "Código", el cual contendrá tres archivos, los cuales son los siguientes:

- Maestro.ino: Contiene el código de ejecución del Arduino que actuará como maestro, es el encargado de obtener la información de los diferentes sensores utilizados y enviar esta información, con un formato de estructura (struct), hacia el Arduino que actuará como esclavo.

- Esclavo.ino: Contiene el código encargado de recibir la información obtenida a través del Arduino maestro y mostrar esta por una pantalla LCD, adicionalmente cuenta con un sistema de alarma, el cual cuando el usuario presione la tecla adecuada en el teclado de membrana (*) se iniciará una cuenta atrás de 10 segundos, tiempo suficiente para que el usuario se aleje. En este momento, ya sea porque el sensor PIR detecta la presencia de alguna persona o animal o porque se active del sensor infrarrojo de fuego, se emitirá un sonido que actúa como alarma por un buzzer y se le enviará al usuario un mensaje vía WiFi por la plataforma Telegram. Para detener esta alarma será necesario introducir la contraseña correcta, en este caso 3515, ya que en caso de introducirla incorrectamente la alarma seguirá sonando.

- WiFi.ino: Contiene el código encargado de enviar una alerta por Telegram al usuario, esto sucede cuando se activa el LED incorporado en la práctica, este LED únicamente se activará en dos ocasiones, si el sensor PIR detecta alguna presencia o si el sensor infrarrojo de fuego se activa.

3. Implementación Software

A continuación, se presenta una implementación en software, realizada con Fritzing en la cual se muestra el esquema de montaje empleado en la elaboración de la práctica.

Arduino Maestro

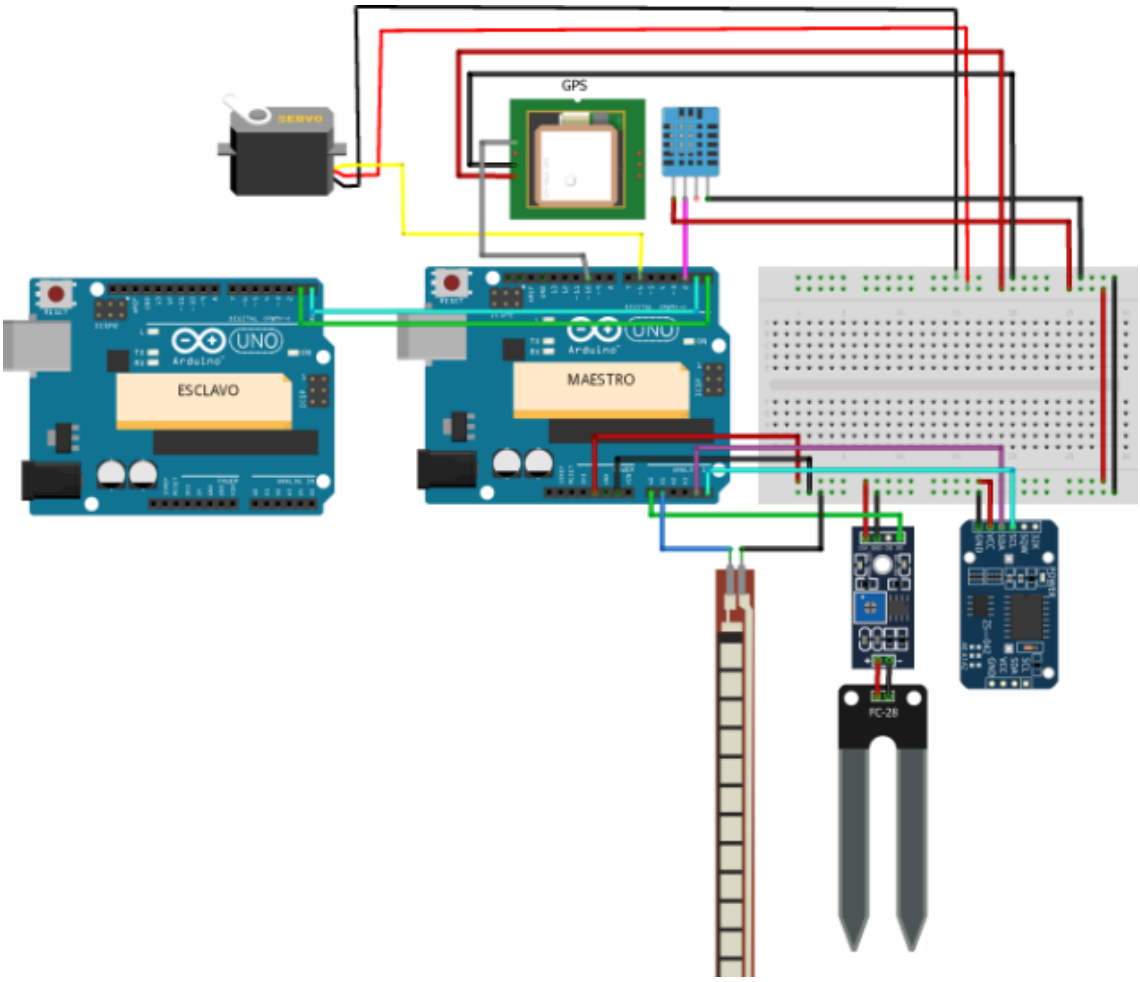

Figura 50. Esquema de montaje del Arduino Maestro.

Arduino Esclavo

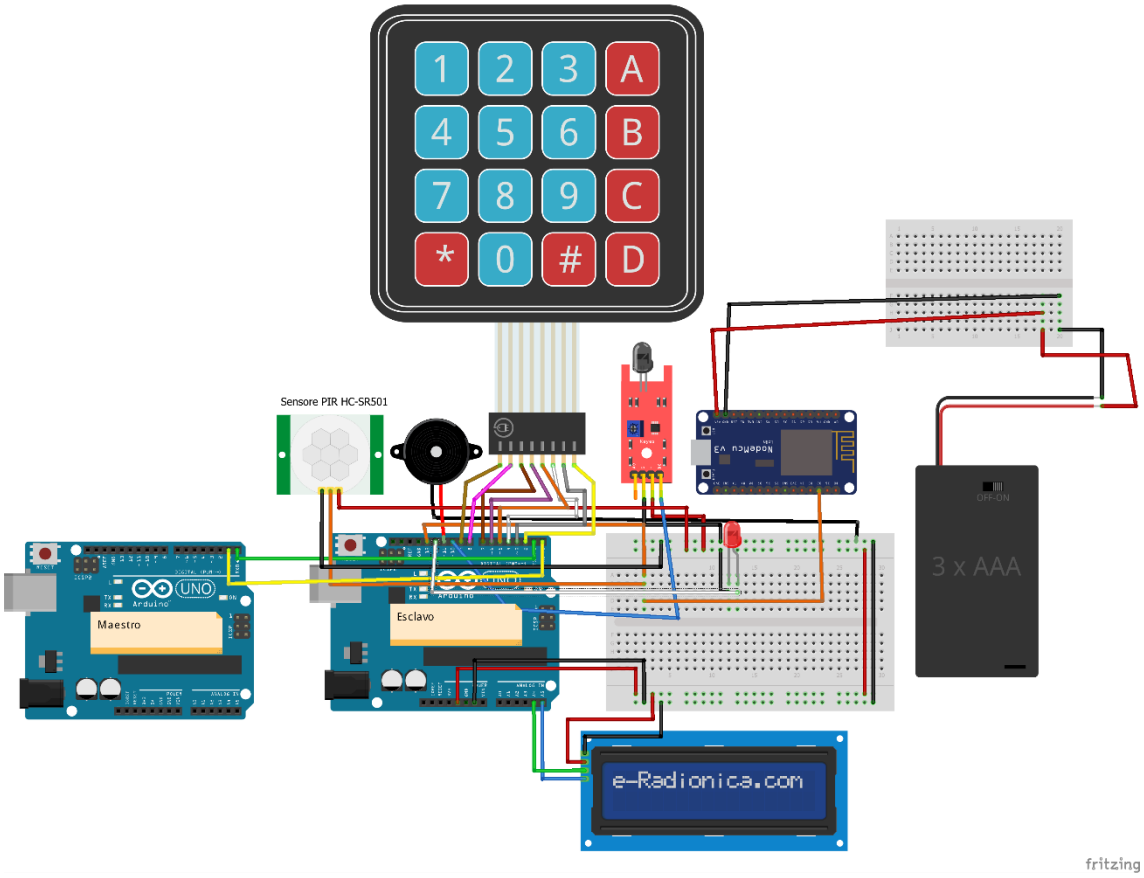

Figura 51. Esquema de montaje del Arduino Esclavo.

4. Implementación Física

El día 22 de mayo de 2019 se ha realizado la defensa de esta práctica, los profesores a cargo de la defensa fueron Antonio y Ricardo, adicionalmente se adjuntará dos videos donde se muestra su correcto funcionamiento, aunque debido al excesivo tamaño de estos, para poder visualizarlos es necesario entrar en el siguiente vínculo, el cual está vinculado con un directorio de Google Drive donde se encuentras dichos videos Vídeos. La implementación final presentada es la siguiente.

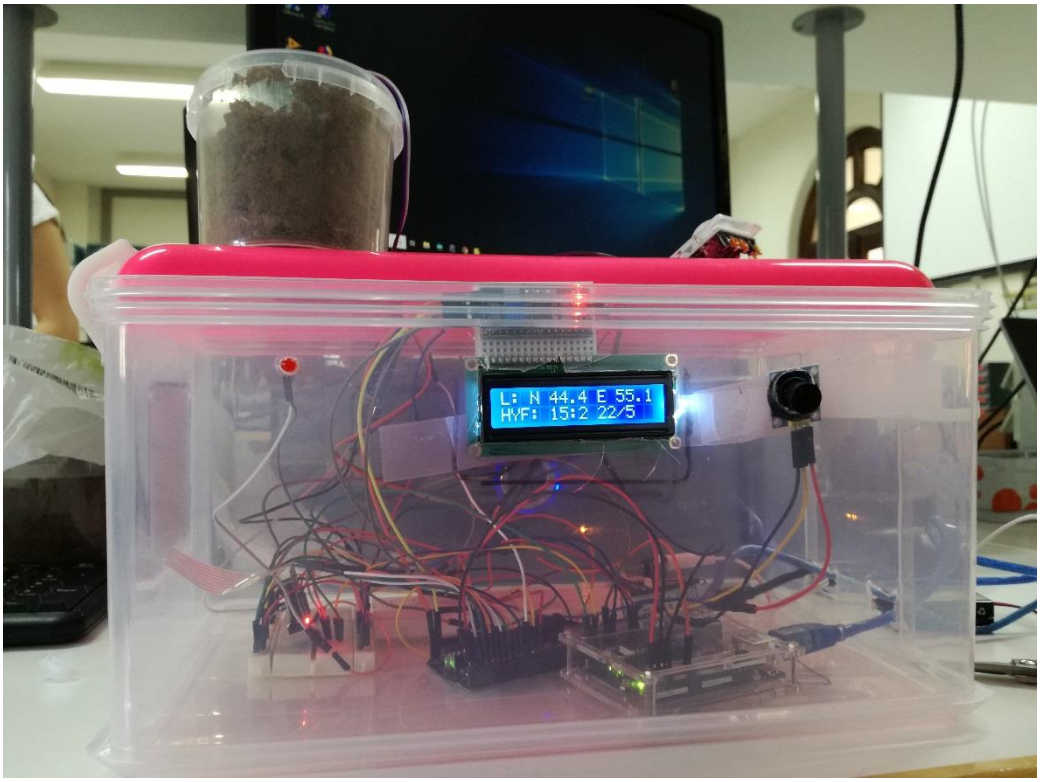

Figura 52. Imagen 1 de la implementación final

Figura 53. Imagen 2 de la implementación final

Figura 54. Mensaje enviado por un bot de Telegram en caso de que salte la alarma

5. Explicación de su Funcionamiento

Finalmente, se procede a explicar el funcionamiento de esta implementación:

En primer lugar el Arduino maestro es el encargado de obtener los datos de la temperatura del ambiente y la humedad (DHT11), el estado del tallo de la planta (Flexómetro), la hora y fecha (DS3231), la localización (GPS) y la temperatura de la tierra (Higrómetro), una vez obtiene estos datos los mapea y almacena en una estructura de información denominada estructura (struct), pero antes de almacenar estos datos es necesario transformar aquellos que sean del tipo "String" en vectores de caracteres (Char Array) el motivo es que para poder enviar la información de un Arduino a otro (conexión maestro-esclavo) no es posible pasar más de 288 bytes por lo que los tipos Strings no se pueden enviar así que es necesario transformarlos en vectores de caracteres para poder enviarlos. Este Arduino emite la orden al servo de que gire cierta cantidad de grados (riego) en función del estado del tallo y de la humedad de la tierra.

El Arduino esclavo recibe estos datos y los saca por la pantalla LCD, de forma periódica recibe estos, por tanto, se irán actualizando en función de la información que capten los sensores del Arduino maestro. Cuando el usuario, por ejemplo, decida irse conectará una alarma la cual se activa al introducir el carácter (*) en el teclado de membrana durante un intervalo de tiempo, en el momento en el que aparezca por la pantalla un aviso de cuenta atrás el usuario tendrá 10 segundos para alejarse, una vez transcurra esta cuenta atrás la alarma estará conectada así que esta saltará por dos motivos, el primero es que el sensor PIR detecte la presencia de algo, el segundo es que el sensor infrarrojo de fuego capte alguna combustión. En ambos casos saltará la alarma (se enciende el LED rojo y comienza a sonar el buzzer) a continuación se le notificará al usuario esta incidencia a través de un mensaje enviado por Telegram. En caso de que se introduzca mal la contraseña, la cual se podrá ir visualizando por pantalla, la alarma seguirá sonando, cuando se introduzca de forma correcta esta parará y se continuará mostrando la información que este Arduino obtiene a través del maestro.

6. Referencias

1. Enlace
2. Enlace
3. Enlace
4. Enlace
5. Enlace
6. Enlace
7. Enlace
8. Enlace
9. Enlace
10. Enlace
11. Enlace
12. Enlace
13. Enlace
14. Enlace
15. Enlace