



VNIVERSIDAD  
D SALAMANCA  
Facultad 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: 03/04/2019

Año académico: 2018 – 2019



## **Tabla de contenidos**

1. Explicación de la práctica.....	3
2. Código de ejecución.....	34
3. Implementación Software.....	35
4. Implementación Física .....	36
5. Resultados.....	37
6. Conclusiones.....	38
7. Referencias .....	39



## **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 -> “Servo.h”) [3].

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

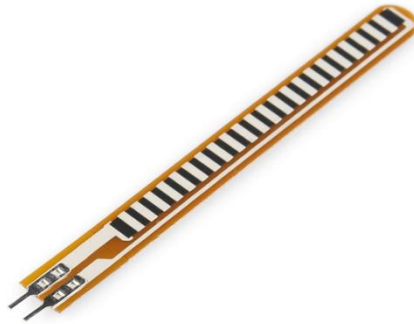
1. LCD1602 Biblioteca -> “LiquidCrystal.h” [4]
2. Módulo I2C adaptador LCD1602 [Biblioteca](#) [5]
3. Módulo MPU-6050 [Biblioteca](#) [6]
4. Modulo I2C Tiny DS3231 AT24C32 [Biblioteca](#) [7]
5. Módulo FC37 [8]
6. Teclado de membrana matricial 4x4 [Biblioteca](#) [9]
7. Módulo FC28 o Higrómetro [10]
8. GY-NEO6MV2 [Biblioteca](#) [11]
9. ESP8266 [Biblioteca](#) [12]
10. Buzzer [13]

Tras conocer los componentes 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.



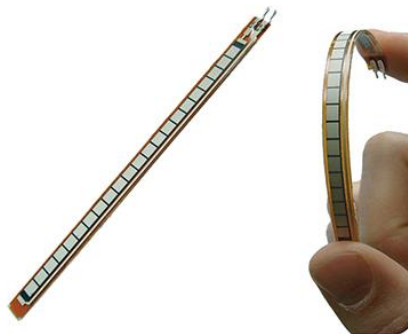
## Flexómetro

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.



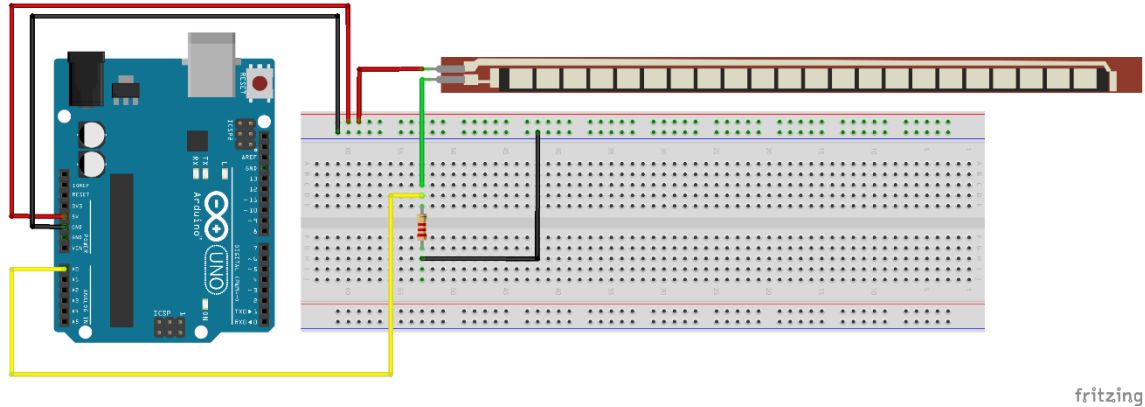
**Figura 2.** Forma en la que debe doblarse el flexómetro.

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  $2.2\text{ k}\Omega$ .



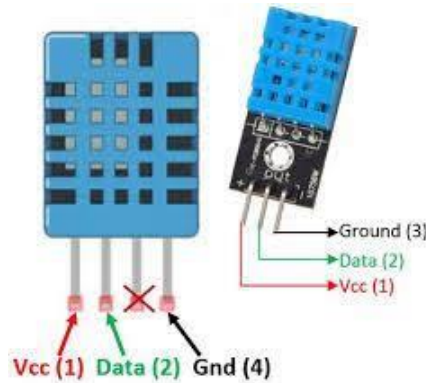
**Figura 3.** Esquema de montaje flexómetro

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  $2.2\text{ k}\Omega$  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.

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1001</u>
8 bits humedad	8 bits humedad	8 bits temperatura	8 bits temperatura	bits de paridad

**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.



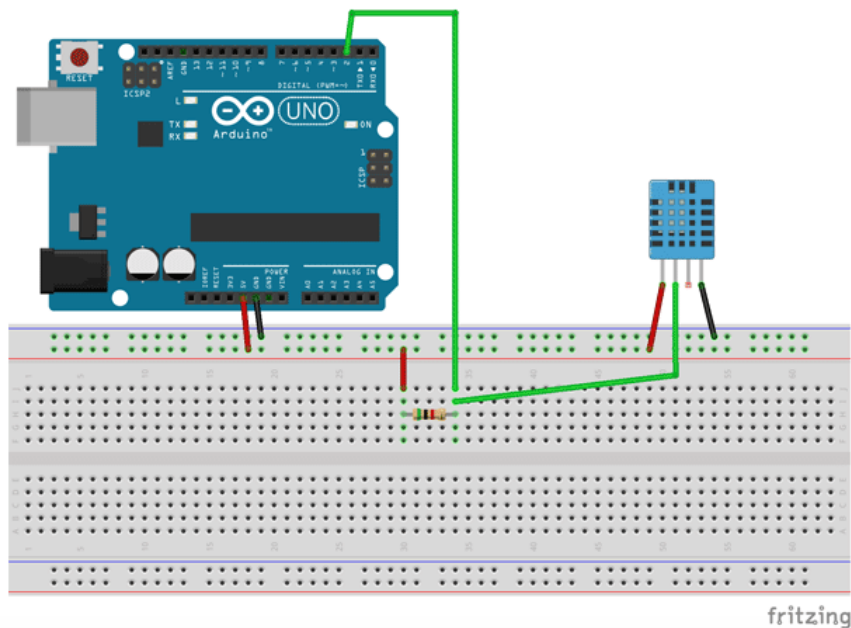
Estos bits de paridad lo único que hacen es asegurar que la información es correcta, sumando los 4 primeros grupos de 8-bits. Esta suma debe ser igual a los bits de paridad.

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

**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  $-90^\circ$  y  $90^\circ$ . Lo que viene a ser un ángulo de giro de  $180^\circ$ .

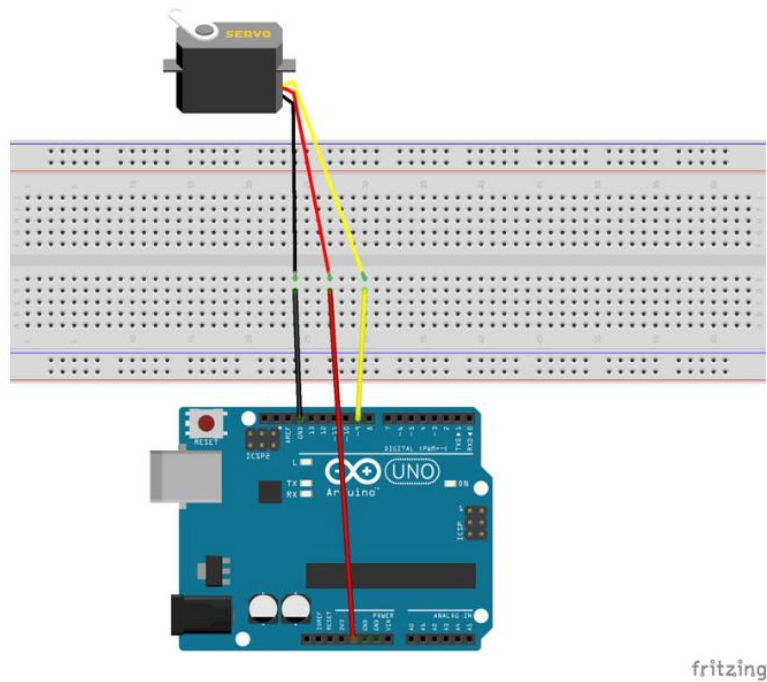
Aunque el servo puede moverse con una resolución de más de 1 grado, este es el máximo de resolución que se puede conseguir debido a la limitación de la señal PWM que es capaz de generar Arduino UNO.

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.

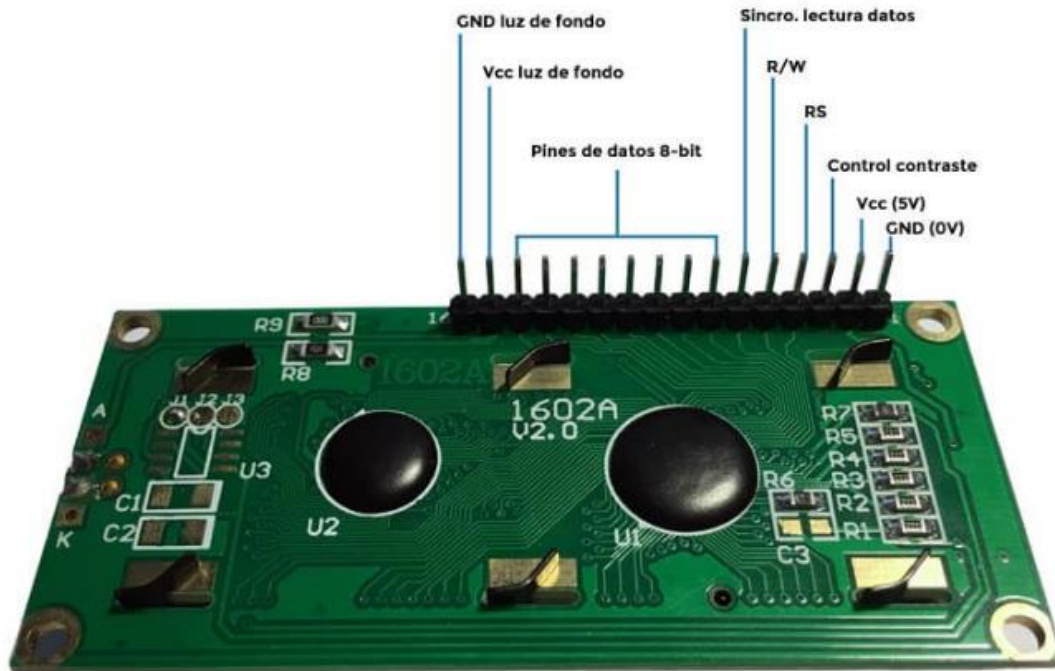
El aspecto de este componente es el siguiente.



**Figura 10.** LCD1602 // pantalla LCD



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



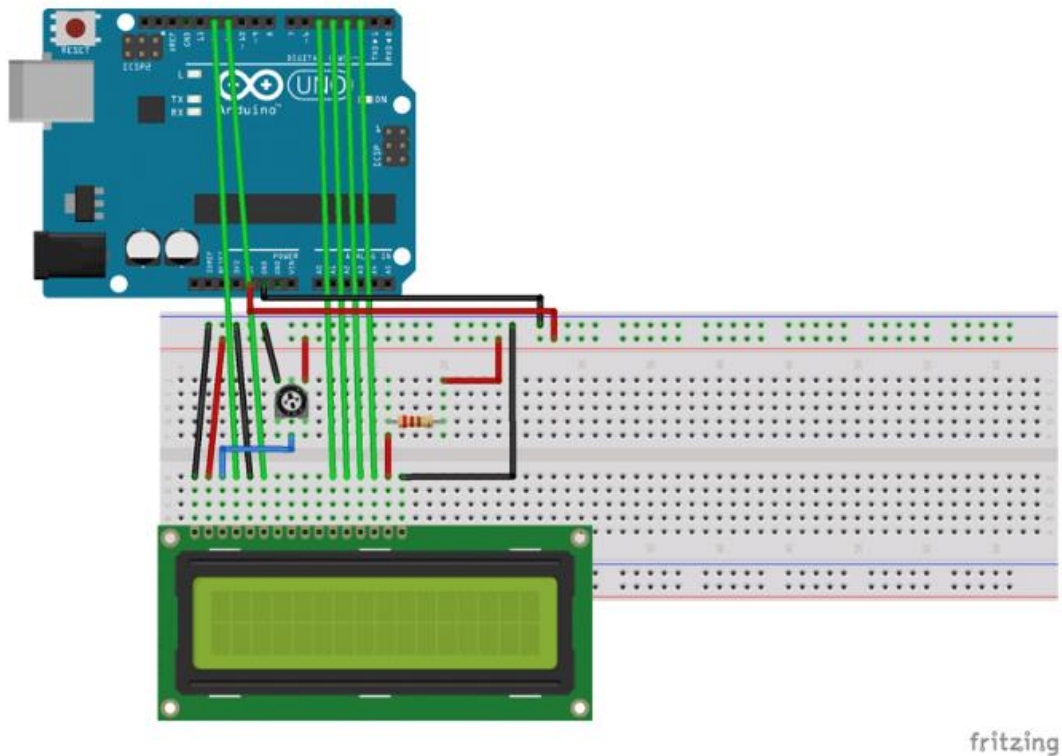
**Figura 11.** Pines físicos del LCD1602

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 k $\Omega$  que será utilizado para regular el contraste de la pantalla y una resistencia de 200  $\Omega$  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.

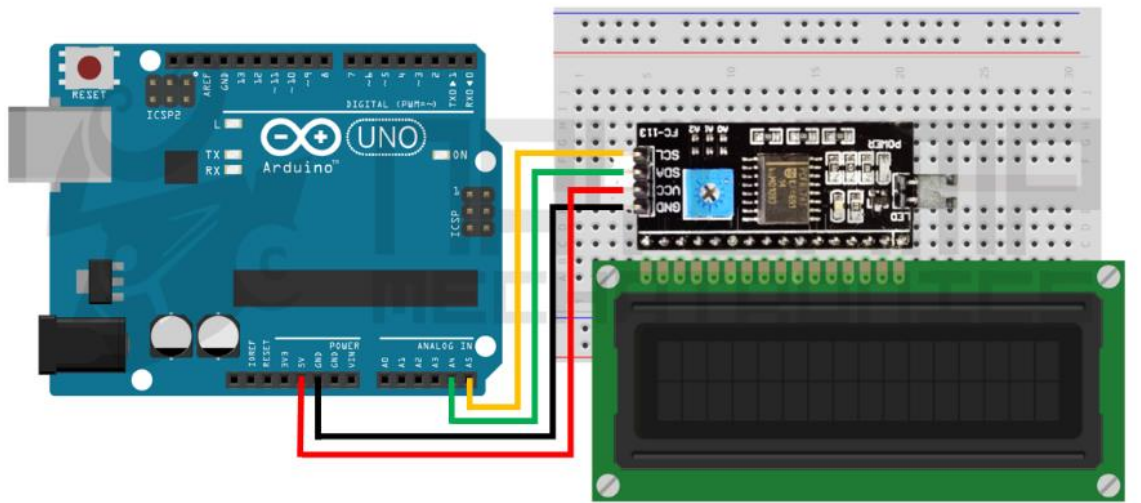


**Figura 13.** Módulo I2C adaptador para LCD1602.

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.



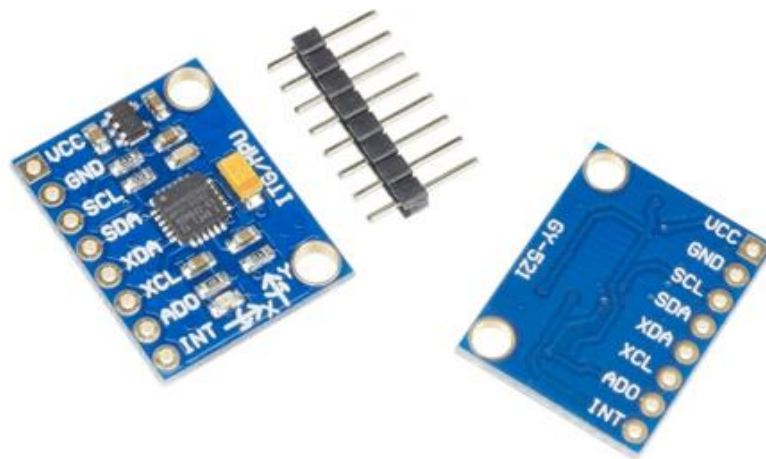


## MPU-6050

El MPU-6050 es una unidad de medición inercial (IMU) de seis grados de libertad (6DOF), que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Dispone de conversores analógicos digitales (ADC) de 16bits. El rango del acelerómetro puede ser ajustado a  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , y  $\pm 16g$ , el del giroscopio a  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$ .

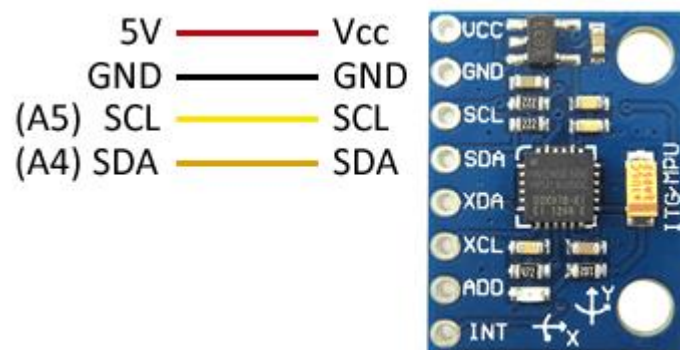
El MPU-6050 incorpora un procesador interno (DMP Digital Motion Processor) que ejecuta complejos algoritmos de MotionFusion para combinar las mediciones de los sensores internos, evitando tener que realizar los filtros de forma exterior.

El aspecto de este sensor es el siguiente.



**Figura 15.** MPU-6050 // Acelerómetro y Giroscopio.

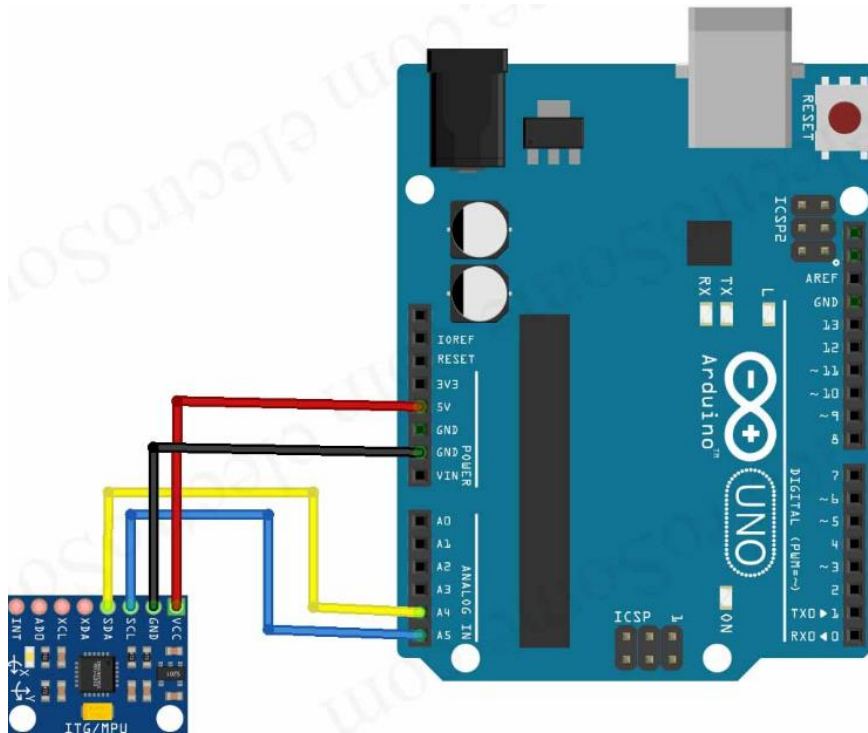
Las conexiones con el Arduino son las siguientes.



**Figura 16.** Conexiones MPU-6050 con el Arduino



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



**Figura 17.** Esquema de montaje MPU-6050

La conexión es sencilla, simplemente se alimenta el módulo desde Arduino mediante GND y 5V y se conecta el pin A4 y A5 del Arduino con los pines correspondientes del sensor SDA y SCL respectivamente.





## *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.

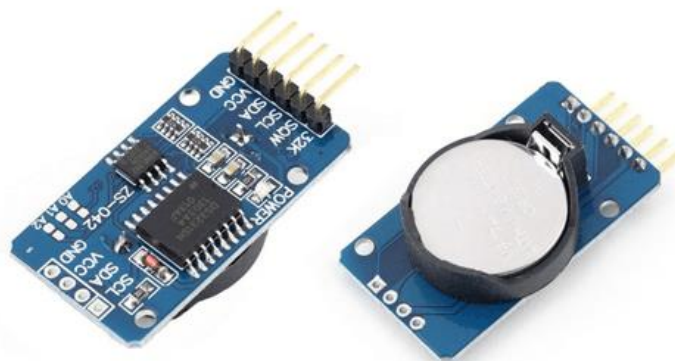
Los RTC normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo. La electrónica de los RTC tiene en cuenta las peculiaridades de la forma de medir el tiempo, como por ejemplo el sistema sexagesimal, los meses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

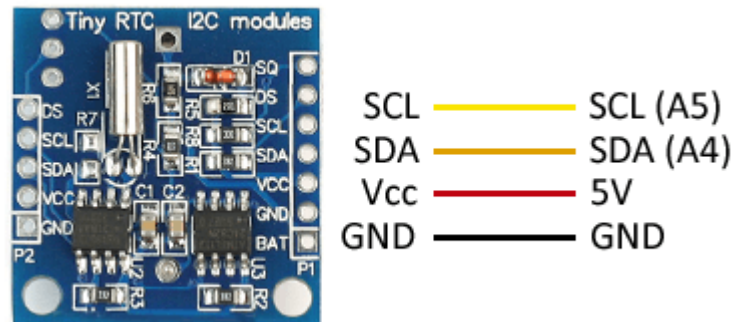
A continuación, se procede a mostrar una imagen del aspecto del DS3231.



**Figura 18.** DS3231 // Reloj y calendario

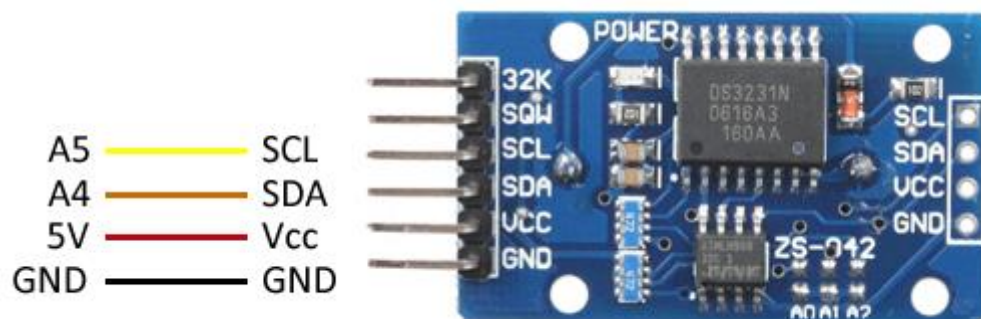


La conexión es sencilla y similar tanto para el DS1307 como el DS3231. Simplemente se alimenta el módulo desde Arduino mediante 5V y Gnd. Por otro lado, se conectan los pines del bus I2C a los correspondientes de Arduino. Las conexiones del modelo DS1307 deberían ser similares a las de la siguiente figura.



**Figura 19.** Conexiones DS1307

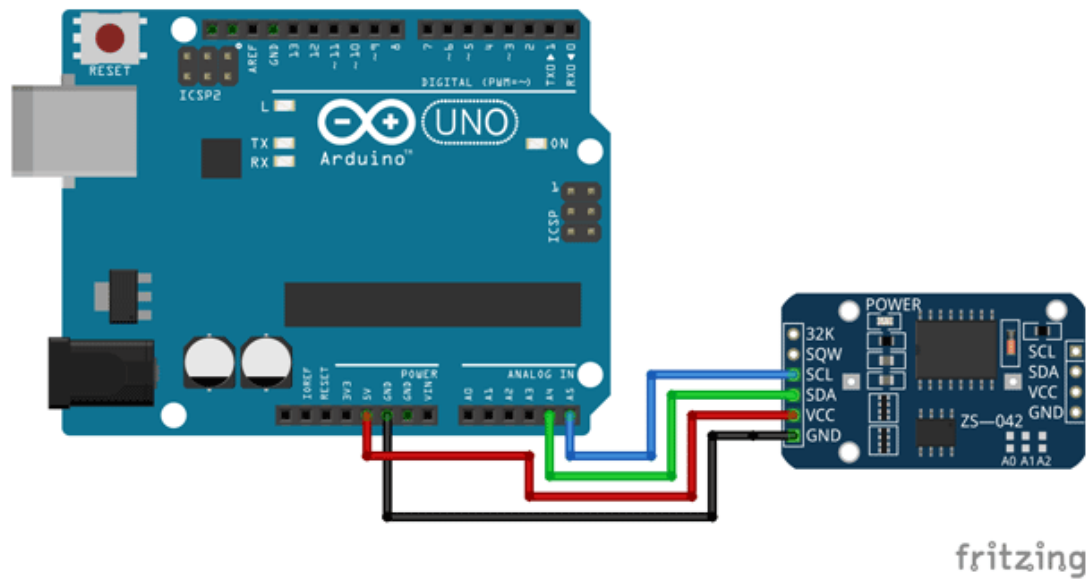
Y las del modelo DS3231, como las de la siguiente figura.



**Figura 20.** Conexiones DS3231



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



**Figura 21.** Esquema de montaje DS3231



## **FC37**

Este tipo de sensores detectan la presencia de lluvia por la variación de conductividad del sensor al entrar en contacto con el agua. Existen varios modelos de sensores similares, como el FC-37 y el YL-83.

Constructivamente son sensores sencillos. Se dispone de dos contactos, unidos a unas pistas conductoras entrelazadas entre sí a una pequeña distancia, sin existir contacto entre ambas. Al depositarse agua sobre la superficie, se pone en contacto eléctrico ambos conductores, lo que puede ser detectado por un sensor.

Ambos modelos utilizan una placa de medición estándar con el comparador LM393, que permite obtener la lectura tanto como un valor analógico como de forma digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la propia placa.

Los valores analógicos medidos varían desde 0 para una placa totalmente mojada, a 1023 para una placa totalmente seca.

La salida digital se dispara cuando el valor de humedad supera un cierto valor, que se ajusta mediante el potenciómetro. Por tanto, se obtendrá una señal LOW en ausencia de lluvia, y HIGH con presencia de lluvia.

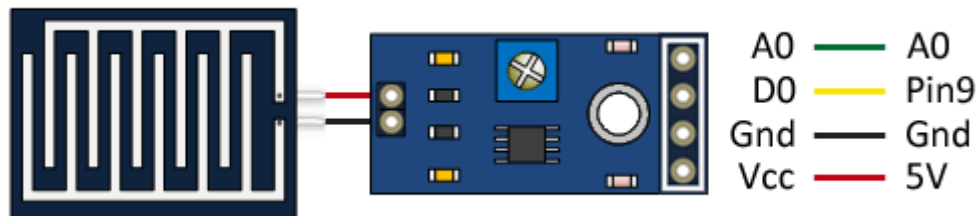
El aspecto de este sensor es el siguiente.



**Figura 22.** FC37 // Detector de agua

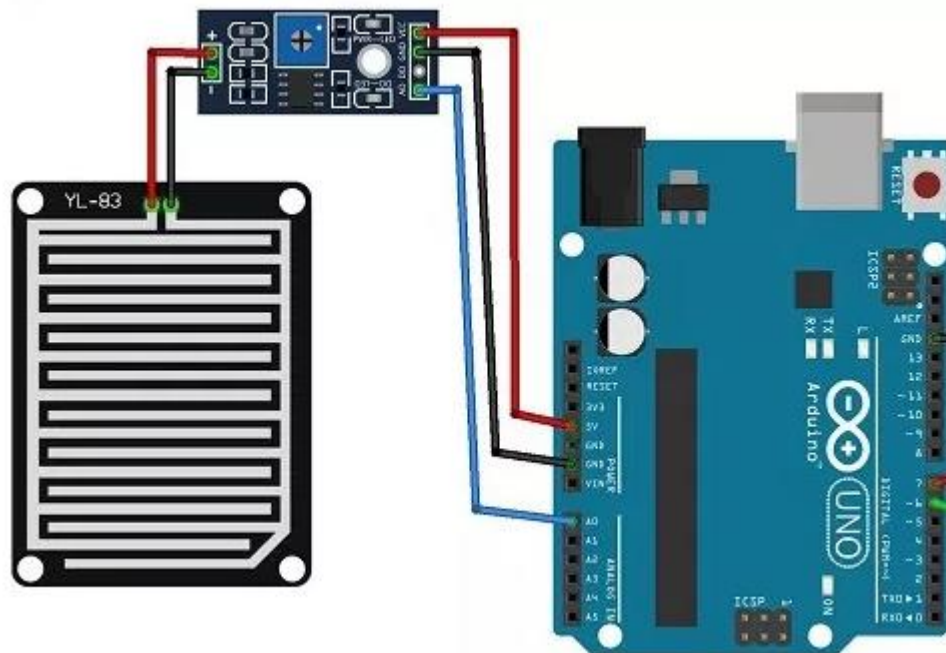


Las conexiones de este sensor con el Arduino son sencillas y se representan a continuación.



**Figura 23.** Conexiones FC37

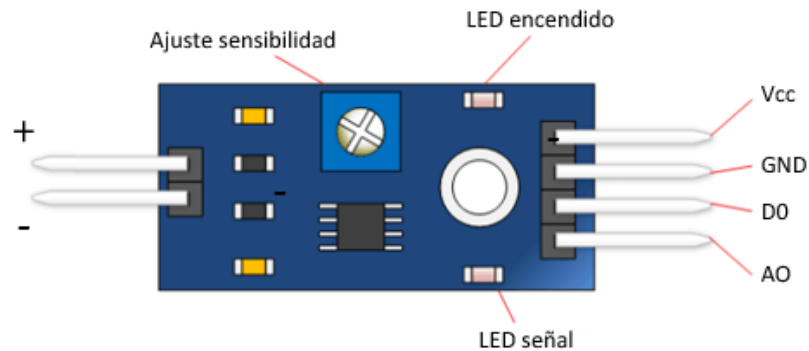
A continuación, el esquema de montaje con el Arduino es el siguiente.



**Figura 24.** Esquema de montaje FC37



Finalmente, se conecta la salida digital del sensor a una entrada digital para leer el estado del sensor.



**Figura 25.** Esquema componentes FC37

Si se quisiera obtener el valor analógico de la medición, se conectaría la salida analógica del sensor a una entrada analógica de Arduino, aunque este sensor no dispone de la precisión suficiente para que el valor analógico sea realmente de utilidad.

Opcionalmente, se puede calibrar el umbral de disparo de la salida digital con el potenciómetro.



## 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.

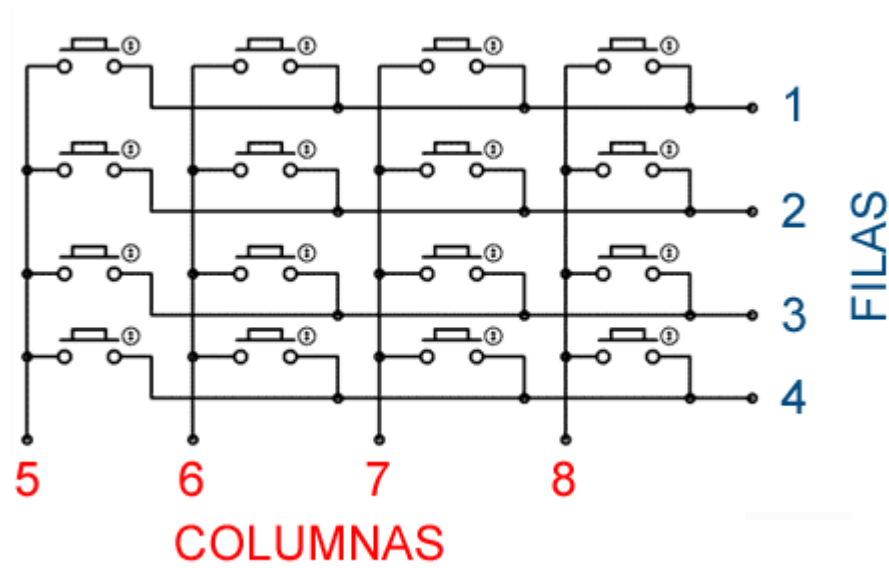


**Figura 26.** Teclado de membrana matricial 4x4.

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  $\rightarrow$  ON ó 0  $\rightarrow$  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.

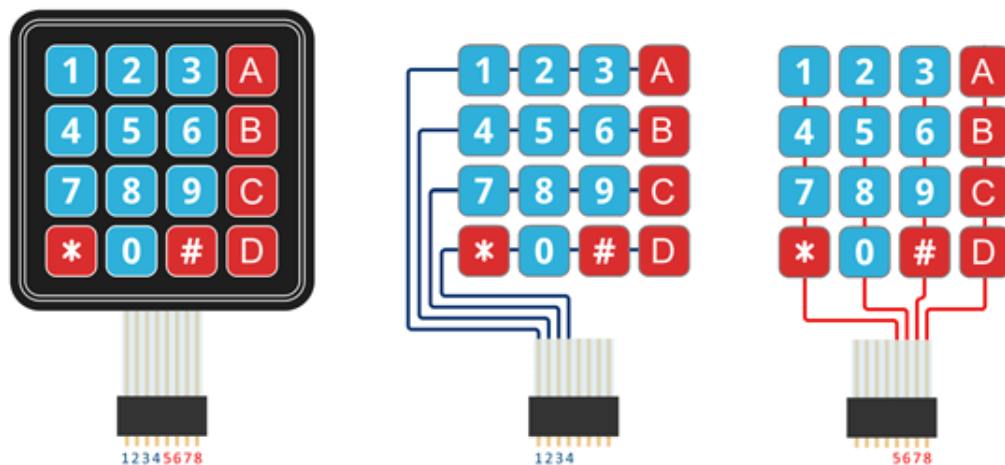


Internamente la disposición de los pulsadores es la siguiente.



**Figura 27.** 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.

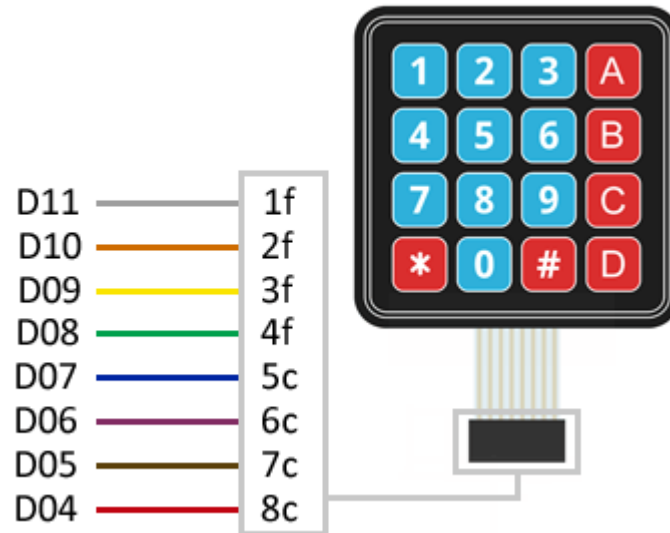


**Figura 28.** Conexiones filas y columnas en un teclado de membrana 4x4.



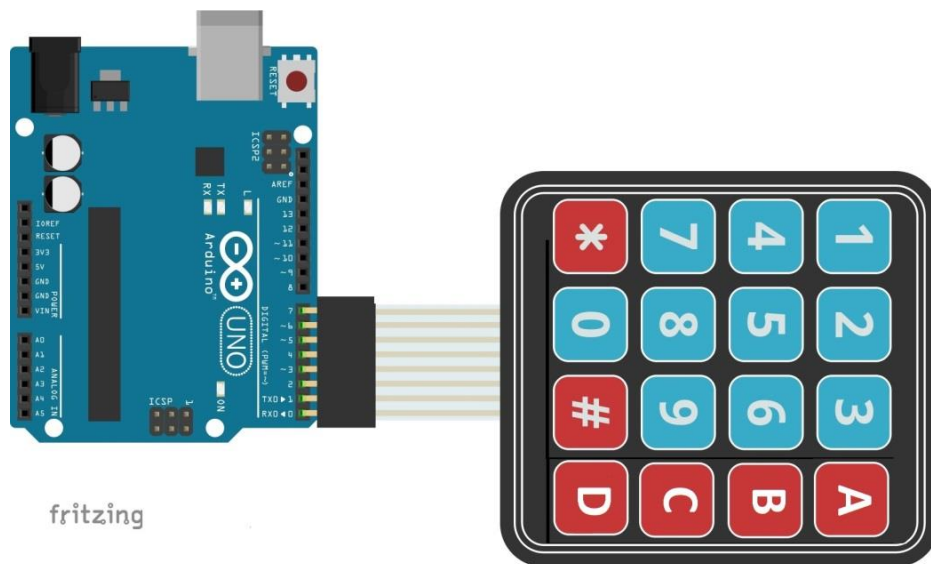


Las conexiones de este teclado con el Arduino son sencillas, simplemente se conectan todos los pines a entradas digitales del Arduino, estas se representan a continuación.



**Figura 28.** Conexiones teclado de membrana 4x4.

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



**Figura 29.** 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

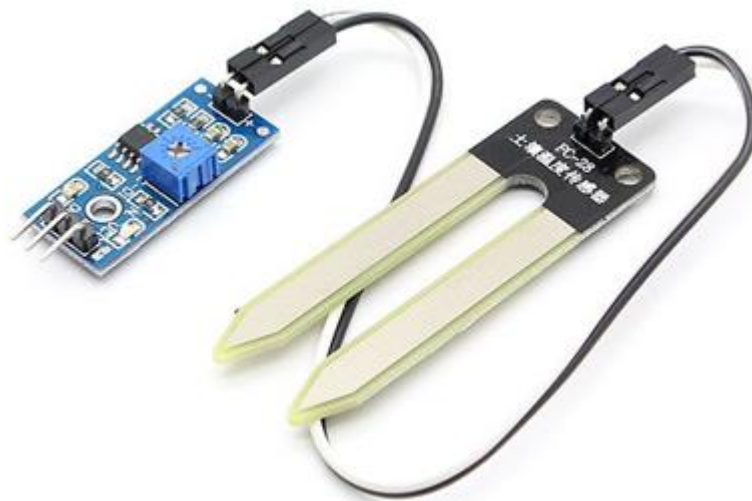
El FC-28 se distribuye con una placa de medición estándar que permite obtener la medición como valor analógico o como una salida digital, activada cuando la humedad supera un cierto umbral.

Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo 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.

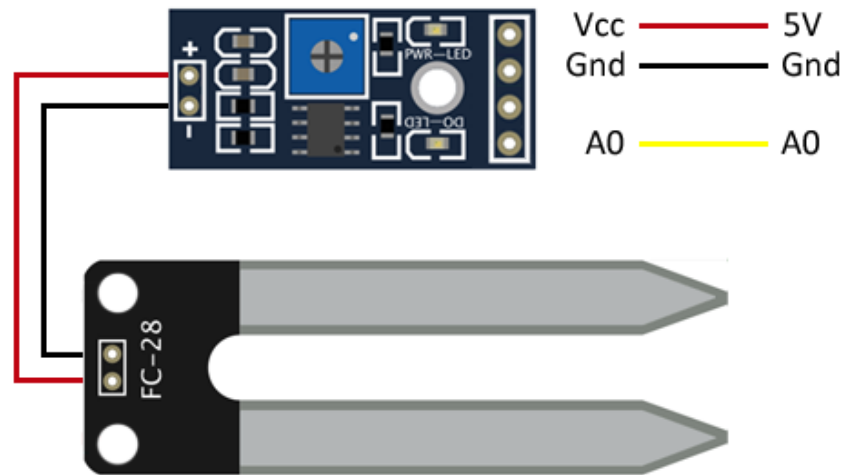


**Figura 30.** FC28 // Higrómetro



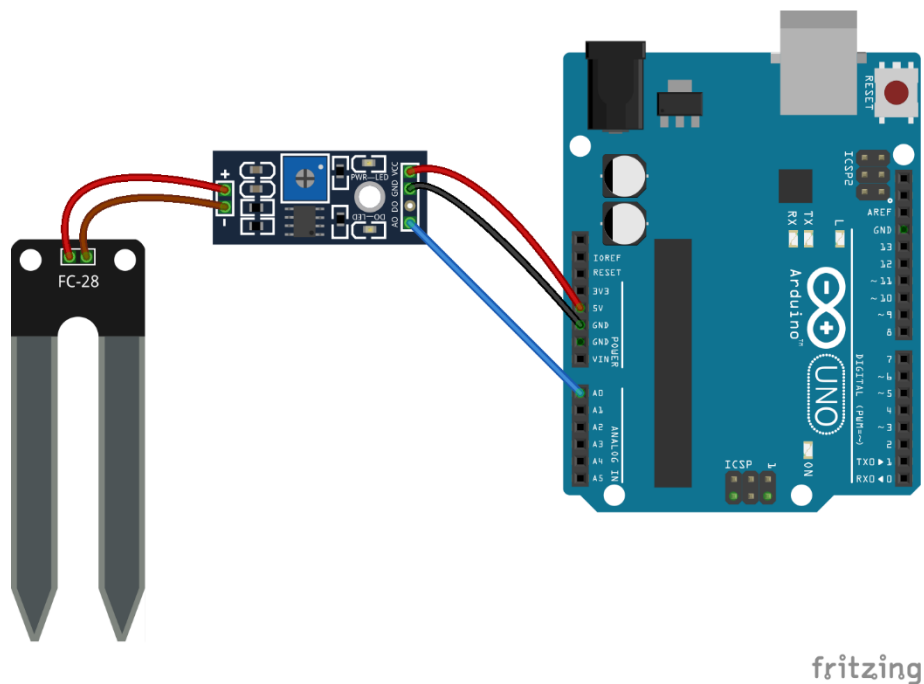
El esquema eléctrico es sencillo. Se alimenta el módulo conectando GND y 5V a los pines correspondientes de Arduino.

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



**Figura 31.** Conexiones FC28.

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



fritzing

**Figura 32.** 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.

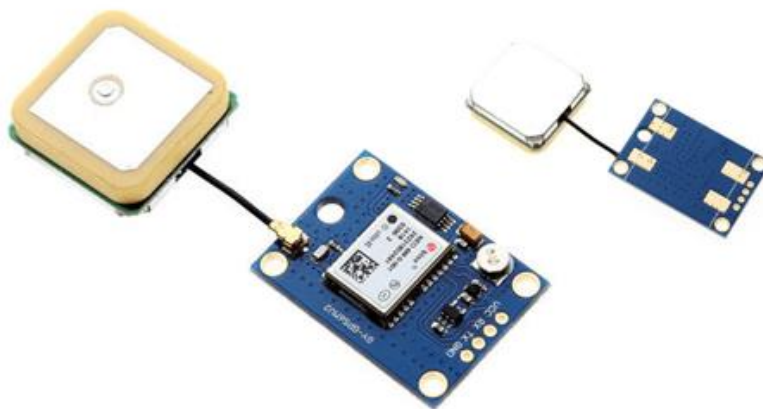
La familia de receptores GPS NEO-6 están diseñados para tener un pequeño tamaño, pequeño coste, y pequeño consumo. La intensidad de corriente necesaria es de unos 37 mA en modo de medición continuo.

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 0.5°, valores más que aceptables para un sistema de posicionamiento GPS.

La representación de este receptor GPS es la siguiente.

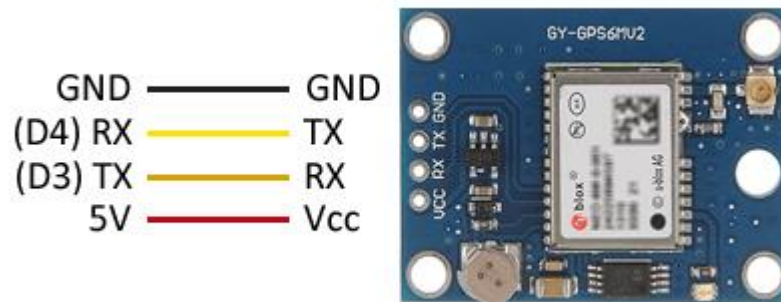


**Figura 33.** GY-NEO6MV2 // Receptor GPS



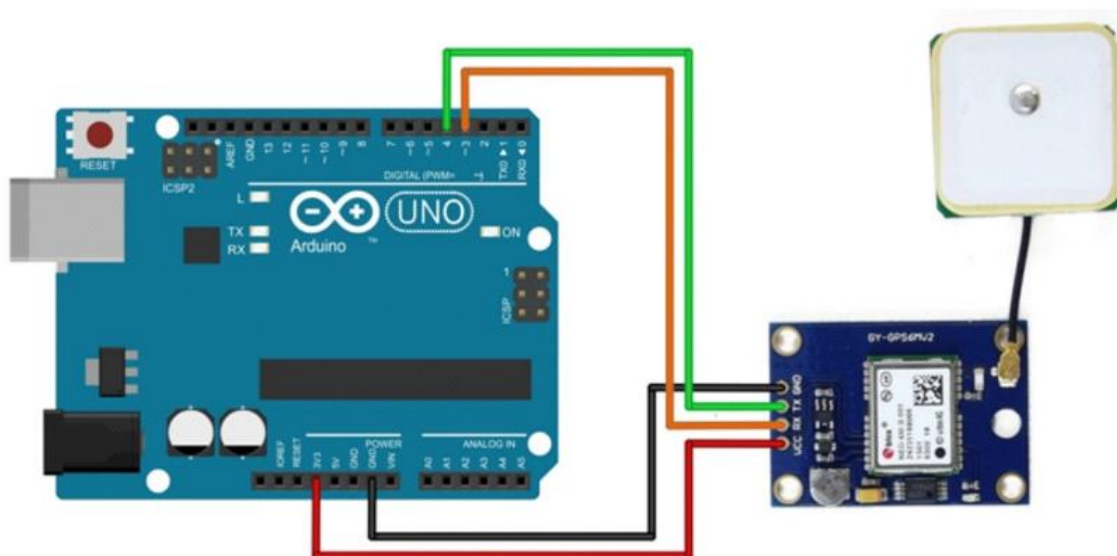
La conexión es sencilla, en primer lugar, se alimenta el módulo conectando Vcc y Gnd del módulo, a 5 V y Gnd de Arduino.

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



**Figura 34.** Conexiones GY-NEO6MV2.

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



**Figura 35.** Esquema de montaje GY-NEO6MV2.

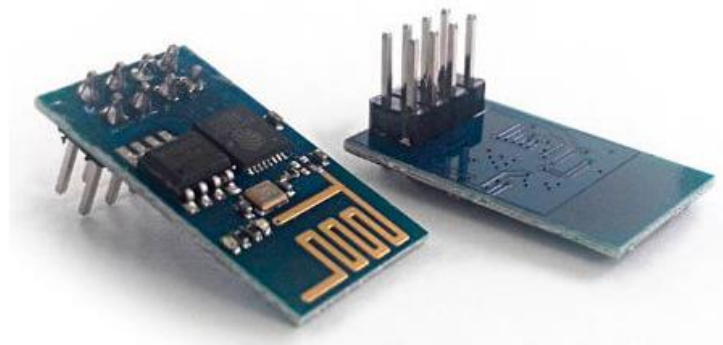


## **ESP8266**

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.

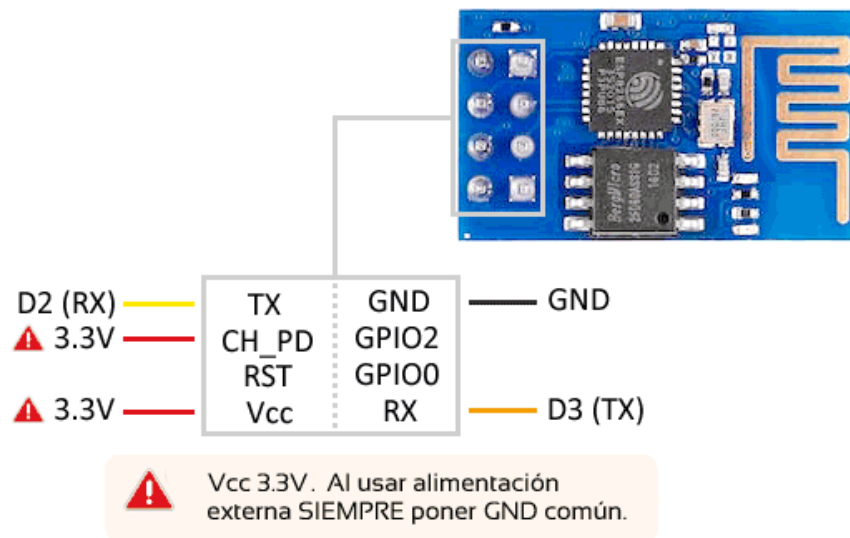
A continuación, se procede a mostrar una imagen de cómo sería la representación de este componente.



**Figura 36.** ESP8266 // Módulo Wi-Fi en ESP01.

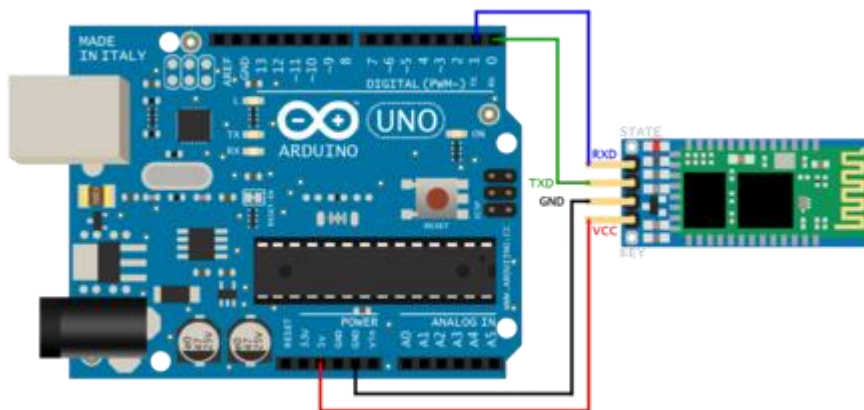
Por un lado, el pin CH\_PD que apaga o enciende el módulo conectándolo, respectivamente, a Gnd o 3.3V. Por su parte, el pin RST reinicia el módulo si se conecta a Gnd. En algunas versiones del módulo se puede dejarlo sin conexión, pero, en general, se tiene que conectar a 3.3V para que el módulo arranque.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





**Figura 37.** Conexiones ESP8266.

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



**Figura 38.** Esquema de montaje ESP8266.





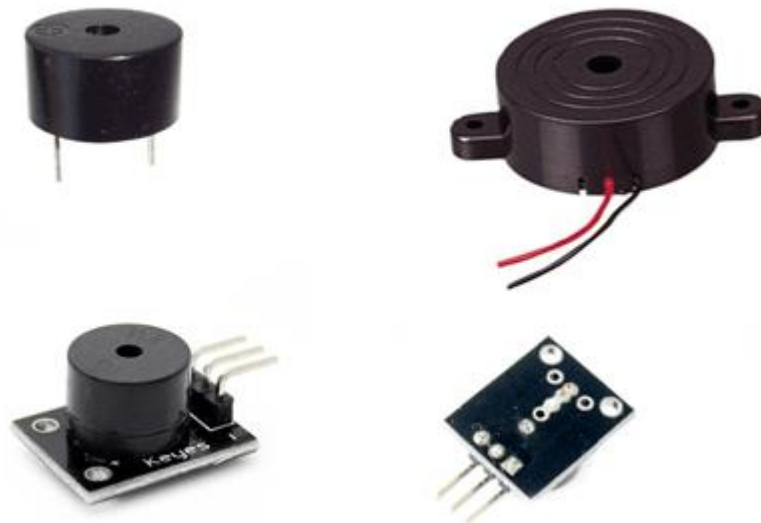
## Buzzer

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.

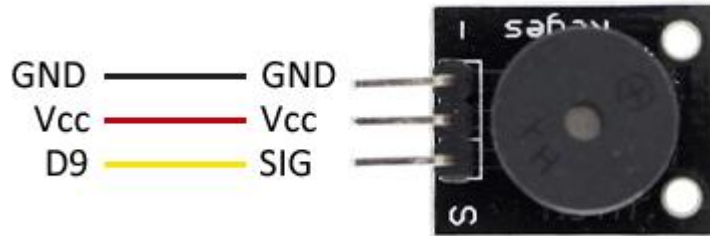


**Figura 39.** Buzzer // Activo o Pasivo



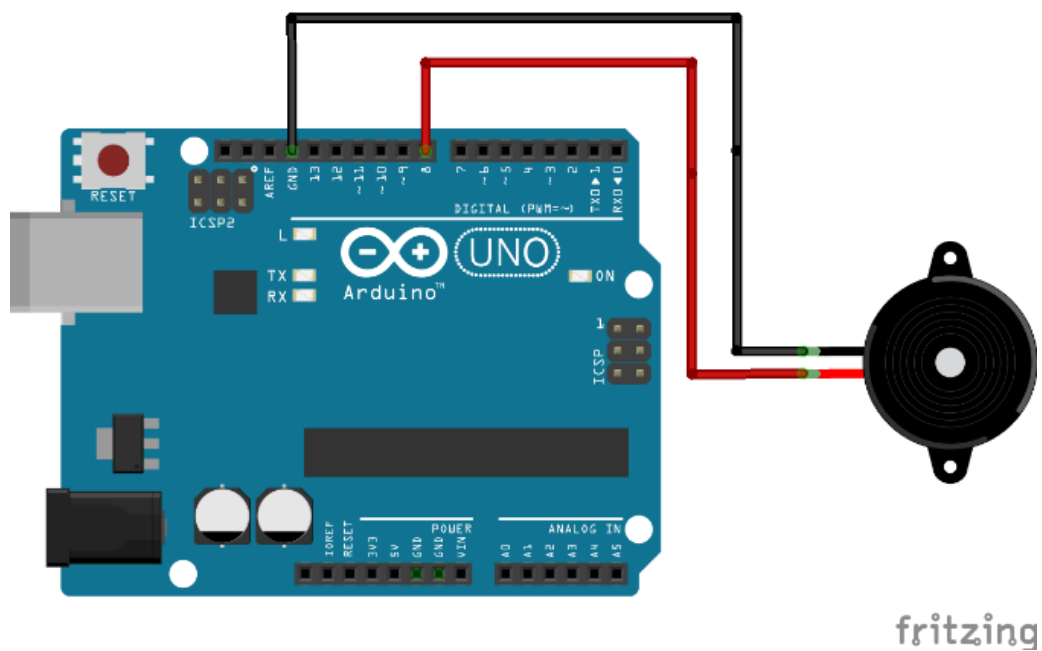


Las conexiones del buzzer con el Arduino son sencillas, simplemente hay que alimentar el módulo conectando Vcc y GND a Arduino, y la entrada de señal a cualquier salida digital de Arduino.



**Figura 40.** Conexiones Buzzer.

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



**Figura 41.** Esquema de montaje Buzzer.



## ***2. Código de ejecución***



### **3. Implementación Software**



## **4. Implementación Física**



## **5. Resultados**



## **6. Conclusiones**



## **7. Referencias**

1. [http://www.lunegate.net/2016/07/tutorial-aprender-usar-un-sensor-flex\\_31.html](http://www.lunegate.net/2016/07/tutorial-aprender-usar-un-sensor-flex_31.html)
2. <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>
3. <https://programarfacil.com/tutoriales/fragmentos/servomotor-con-arduino/>
4. <https://programarfacil.com/tutoriales/fragmentos/arduino/texto-en-movimiento-en-un-lcd-con-arduino/>
5. [https://www.naylampmechatronics.com/blog/35\\_Tutorial--LCD-con-I2C-control-a-un-LCD-con-so.html](https://www.naylampmechatronics.com/blog/35_Tutorial--LCD-con-I2C-control-a-un-LCD-con-so.html)
6. <https://www.luisllamas.es/arduino-orientacion-imu-mpu-6050/>
7. <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>
8. <https://www.luisllamas.es/arduino-lluvia/>
9. <https://www.luisllamas.es/arduino-teclado-matricial/>
10. <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>
11. <https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-gps-neo-6/>
12. <https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>
13. <https://www.luisllamas.es/reproducir-sonidos-arduino-buzzer-pasivo-altavoz/>
- 14.

