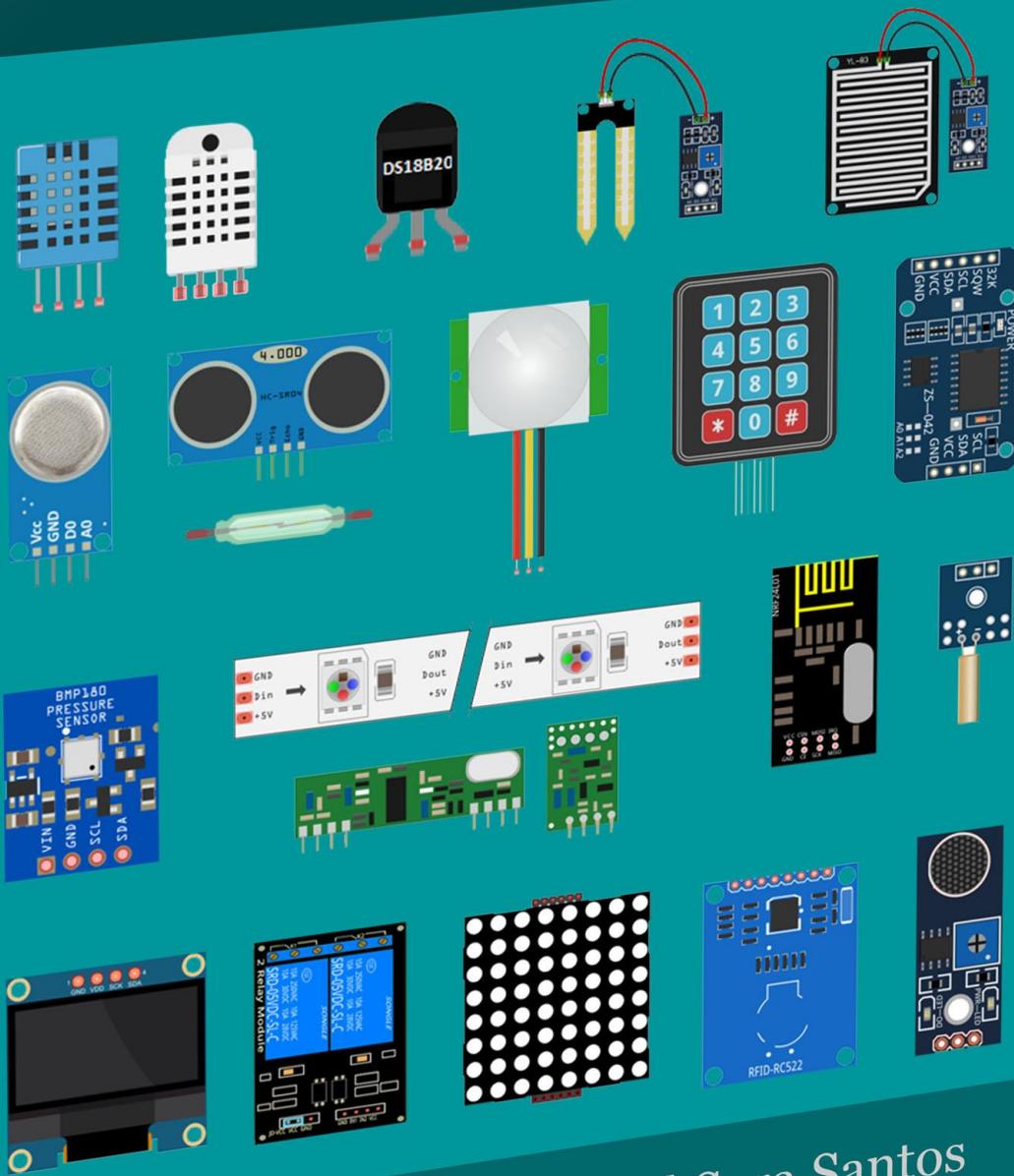


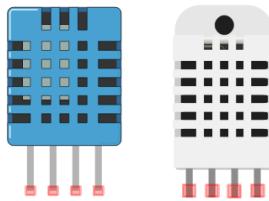
ULTIMATE GUIDE FOR ARDUINO SENSORS/MODULES



Written By Rui Santos and Sara Santos

Tabla de contenido

DHT11 / DHT22 temperatura y sensor de humedad	7
Sensor barométrico BMP180.....	13
FC-37 o YL-83 Sensor de lluvia	21
YL-69 o sensor de humedad HL-69 Soil.....	26
DS18B20 sensor de temperatura	31
DS1307 o DS3231 reloj de tiempo real	37
MQ-2 Gas / sensor de humo.....	43
El sensor ultrasónico HC-SR04.....	49
Sensor de movimiento PIR.....	53
Tilt Sensor.....	57
Sensor de sonido del micrófono.....	62
Reed Switch.....	66
MRFC522 RFID.....	70
Módulo de relé	77
nRF24L01	83
433 MHz Transmisor / Receptor.....	91
8x8 matriz de puntos	102
WS2812B Addressable RGB LED Strip.....	108
Teclado de membrana.....	114
1.8 Pantalla TFT	118
SIM900 GSM GPRS Shield	127
Módulo SD Card.....	146
sensor de color TCS3200.....	152

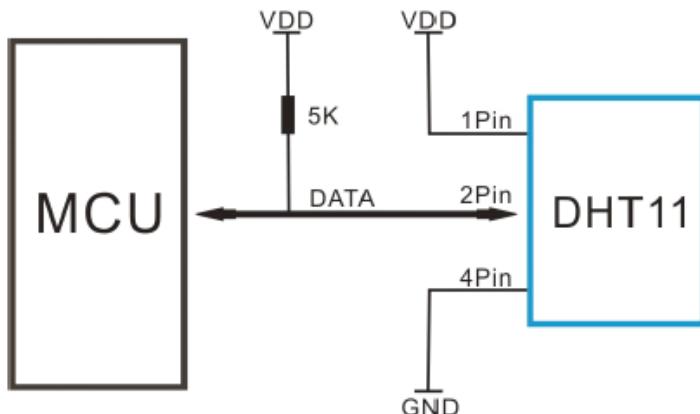


DHT11/DHT22 Sensor de Temperatura y Humedad

Estos sensores DHTXX son muy populares entre los manitas Arduino. Los sensores de DHT son sensores de bajo costo para la medición de temperatura y humedad.

Estos sensores contienen un chip que hace conversión analógica a digital y escupe una señal digital con la temperatura y la humedad.

Estas señales son fáciles de leer con cualquier microcontrolador (MCU).



Especificaciones DHT11 vs DHT22

Hay dos versiones del sensor de DHT:

DHT11

- Rango: 20-90%
- La precisión absoluta: $\pm 5\%$
- Repetibilidad: $\pm 1\%$
- Estabilidad a largo plazo: $\pm 1\%$ por año
- Precio: \$ 1 a \$ 5

DHT22

- Rango: 0-100%
- La precisión absoluta: $\pm 2\%$
- Repetibilidad: $\pm 1\%$
- Estabilidad a largo plazo: $\pm 0,5\%$ por año
- Precio: \$ 4 a \$ 10

Como se puede ver en las especificaciones anteriores, el DHT22 es un poco más preciso.

Donde comprar?

Haga clic en los enlaces a continuación para encontrar el mejor precio :

- Haga clic aquí para ver DHT11
- Haga clic aquí para ver DHT22

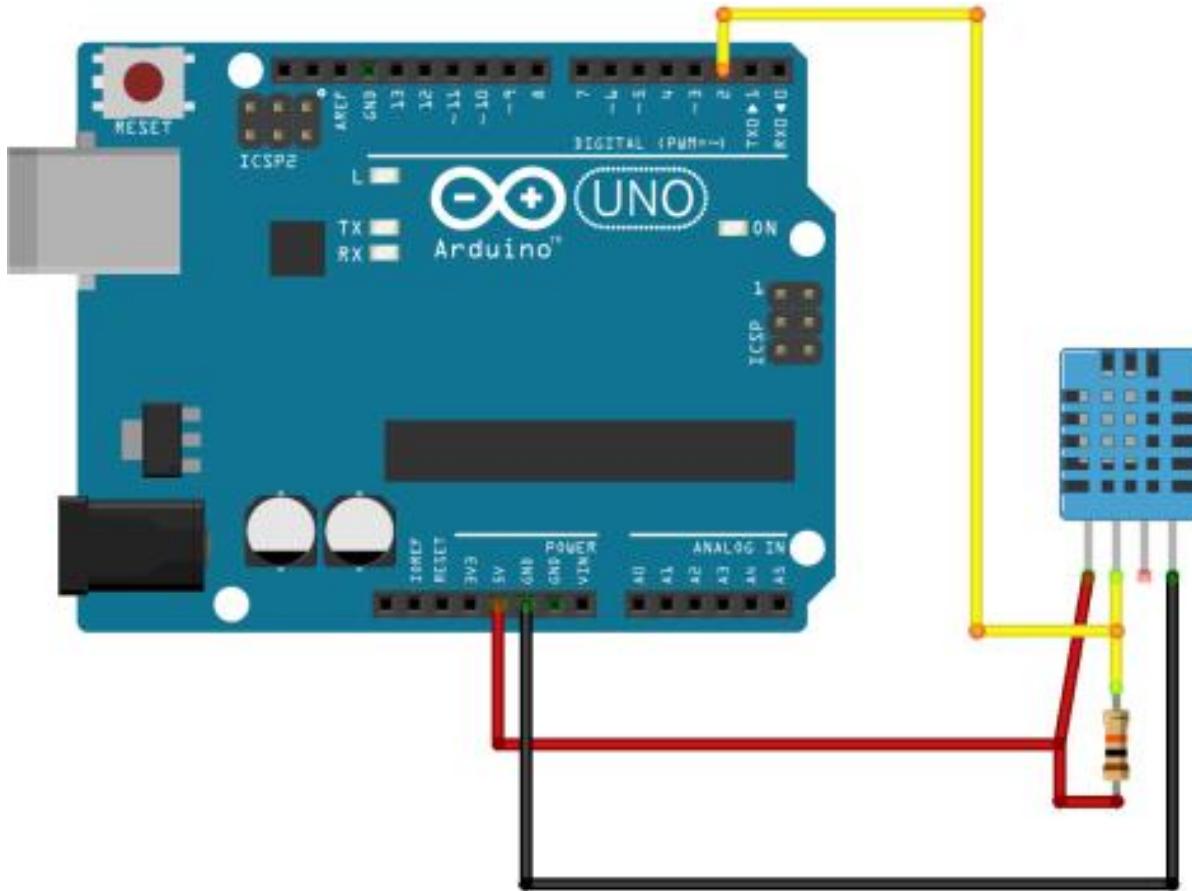
Arduino con el sensor DHT11

Para este ejemplo se necesitan los siguientes componentes:

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	1x DHT11	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	10KΩ Resistor (or 4.7KΩ)	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Círcito

Aquí es cómo conectar el DHT11 a un Arduino:



Pin	Wiring to Arduino UNO
1 st pin - VCC	5V
2 nd pin - Data OUT	Digital pin 2
3 rd pin	Don't connect
4 th pin - GND	GND

Código

A continuación puede encontrar el código que necesita para este proyecto. Pero primero es necesario instalar la librería DHT11.

1. Descarga la librería aquí [DHT11](#)
2. **Descomprimi la libreria**
3. Cambiar el nombre de la carpeta extraída a DHT y retire el "-". De lo contrario el Arduino IDE no reconocerá la librería
4. Instalar el DHT11 en el IDE de Arduino: ir a **Sketch ▶ Include Library ▶ Add .ZIP library** y seleccione la biblioteca que acabas de descargar
5. Reiniciar el IDE Arduino
6. Ir Archivo • Ejemplos • DHT • DHTtester
7. Sube el código

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN 2      // what digital pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11    // DHT 11
//##define DHTTYPE DHT22    // DHT 22  (AM2302), AM2321
//##define DHTTYPE DHT21    // DHT 21  (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors. This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}


```

```

void loop() {
    // Wait a few seconds between measurements.
    delay(2000);
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);
    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    // Compute heat index in Fahrenheit (the default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahrenheit = false)
    float hic = dht.computeHeatIndex(t, h, false);
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.print(" *C ");
    Serial.print(f);
    Serial.print(" *F\t");
    Serial.print("Heat index: ");
    Serial.print(hic);
    Serial.print(" *C ");
    Serial.print(hif);
    Serial.println(" *F");
}

```

{;} SOURCE CODE

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/DHTtester.ino>

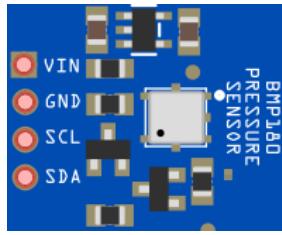
Demostración

En este proyecto el Arduino es la medición de la temperatura y la humedad. Esos dos mediciones se muestran en el monitor de serie. Esto es lo que debe ver en su monitor serie Arduino IDE.

DHTxx test!

Humidity: 75.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.76 *F
Humidity: 75.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.76 *F
Humidity: 75.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.76 *F
Humidity: 75.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.76 *F
Humidity: 75.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.76 *F
Humidity: 76.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.30 *F
Humidity: 76.00 %	Temperature: 18.00 *C	64.40 *F	Heat index: 72.30 *F
Humidity: 77.00 %	Temperature: 19.00 *C	66.20 *F	Heat index: 71.32 *F
Humidity: 77.00 %	Temperature: 19.00 *C	66.20 *F	Heat index: 71.32 *F
Humidity: 77.00 %	Temperature: 20.00 *C	68.00 *F	Heat index: 71.31 *F
Humidity: 78.00 %	Temperature: 20.00 *C	68.00 *F	Heat index: 70.92 *F
Humidity: 78.00 %	Temperature: 21.00 *C	69.80 *F	Heat index: 71.46 *F
Humidity: 78.00 %	Temperature: 21.00 *C	69.80 *F	Heat index: 71.46 *F
Humidity: 78.00 %	Temperature: 22.00 *C	71.60 *F	Heat index: 72.51 *F
Humidity: 78.00 %	Temperature: 22.00 *C	71.60 *F	Heat index: 72.51 *F
Humidity: 78.00 %	Temperature: 22.00 *C	71.60 *F	Heat index: 72.51 *F
Humidity: 77.00 %	Temperature: 22.00 *C	71.60 *F	Heat index: 72.77 *F

Autoscroll Both NL & CR 9600 baud



Sensor barométrico BMP180

El sensor barométrico BMP180 (modelo GY-68) es la de la figura siguiente (vista frontal y posterior). Es un módulo muy pequeño, con 1,1 mm x 1 mm (0.039in x 0.043in).

Front view



Back view



Se mide la presión absoluta del aire que lo rodea. Tiene un rango de medición de 300 hPa a 1100hPa con una precisión de hasta 0.02hPa. También se puede medir la altitud y la temperatura.

El sensor barométrico BMP180 comunica a través de la interfaz I2C. Esto significa que se comunica con el Arduino usando sólo 2 pines.

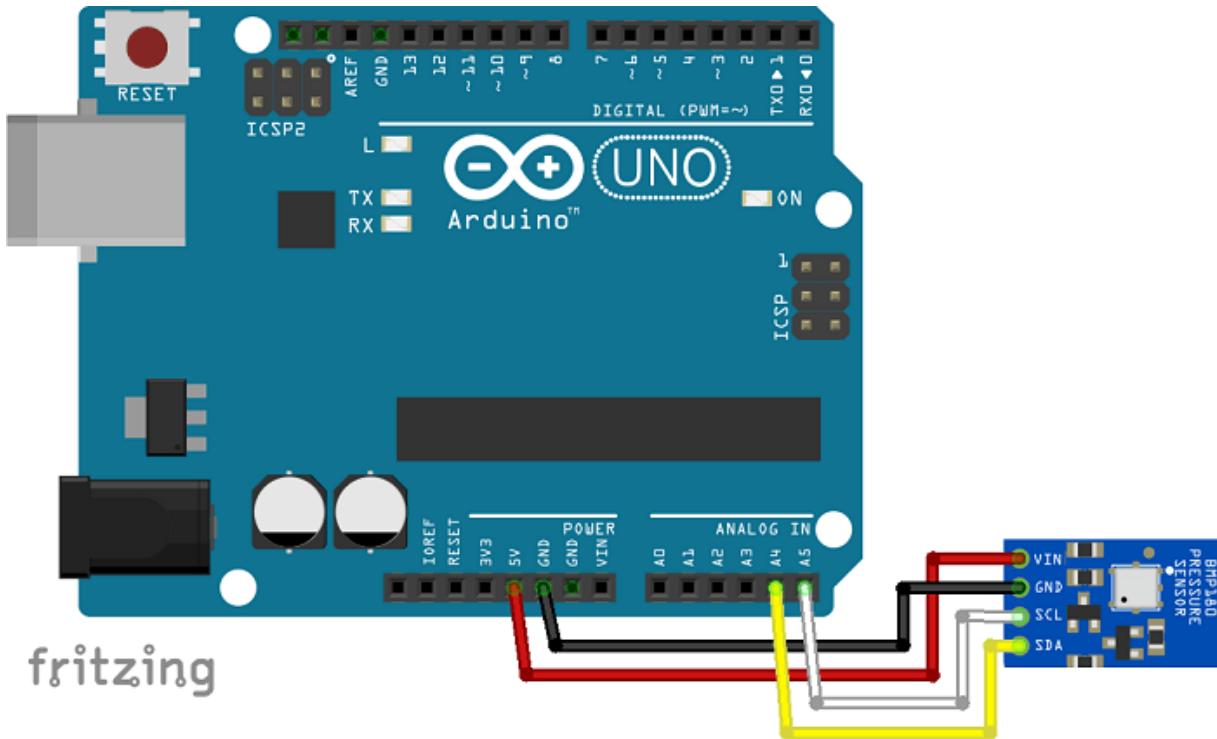
¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [BMP180 barometric sensor](#)

Círculo

Cablear el sensor para el Arduino como se muestra en el diagrama esquemático siguiente



El cableado del sensor a la placa Arduino UNO es bastante sencillo:

Pin	Wiring to Arduino Uno
Vin	5V
GND	GND
SCL	A5
SDA	A4

Código

Para controlar el sensor barométrico BMP180, es necesario instalar la librería SFE_BMP180

Instalación de la librería SFE_BMP180

1. [Click here to download the SFE_BMP180 library](#). Usted debe tener una carpeta .zip en la carpeta de Descargas
2. Descomprimir la carpeta .zip y usted debe obtener BMP180_Breakout_Arduino_Library-master folder
3. Cambiar el nombre de la carpeta de ~~BMP180_Breakout_Arduino_Library-master~~ a ~~BMP180_Breakout_Arduino_Library~~
4. Mover la carpeta BMP180_Breakout_Arduino_Library a la carpeta de bibliotecas de instalación del Arduino IDE
5. Por último, vuelva a abrir el IDE de Arduino

Subir el código

Ir Archivo • Ejemplos • SparkfunBMP180 • SFE_BMP180_example. Este ejemplo está muy bien comentado y explicado en cómo el sensor lee la presión, temperatura y calcula la altitud. No descargue el código de ahora, es necesario establecer primero la altitud.

```
/*
```

```
/* SFE_BMP180 library example sketch
```

```
This sketch shows how to use the SFE_BMP180 library to read the
Bosch BMP180 barometric pressure sensor.
https://www.sparkfun.com/products/11824
```

```
Like most pressure sensors, the BMP180 measures absolute pressure.
This is the actual ambient pressure seen by the device, which will
vary with both altitude and weather.
```

```
Before taking a pressure reading you must take a temperature reading.
This is done with startTemperature() and getTemperature().
The result is in degrees C.
```

```
Once you have a temperature reading, you can take a pressure reading.
This is done with startPressure() and getPressure().
The result is in millibar (mb) aka hectopascals (hPa).
```

```
If you'll be monitoring weather patterns, you will probably want to
remove the effects of altitude. This will produce readings that can
```

be compared to the published pressure readings from other locations. To do this, use the `sealevel()` function. You will need to provide the known altitude at which the pressure was measured.

If you want to measure altitude, you will need to know the pressure at a baseline altitude. This can be average sealevel pressure, or a previous pressure reading at your altitude, in which case subsequent altitude readings will be + or - the initial baseline. This is done with the `altitude()` function.

Hardware connections:

- (GND) to GND
- + (VDD) to 3.3V

(WARNING: do not connect + to 5V or the sensor will be damaged!)

You will also need to connect the I₂C pins (SCL and SDA) to your Arduino. The pins are different on different Arduinos:

Any Arduino pins labeled:	SDA	SCL
Uno, Redboard, Pro:	A4	A5
Mega2560, Due:	20	21
Leonardo:	2	3

Leave the IO (VDDIO) pin unconnected. This pin is for connecting the BMP180 to systems with lower logic levels such as 1.8V

Have fun! -Your friends at SparkFun.

The `SFE_BMP180` library uses floating-point equations developed by the Weather Station Data Logger project: <http://wmx00.sourceforge.net/>

Our example code uses the "beerware" license. You can do anything you like with this code. No really, anything. If you find it useful, buy me a beer someday.

V1.0 Mike Grusin, SparkFun Electronics 10/24/2013

V1.1.2 Updates for Arduino 1.6.4 5/2015

*/

```
// Your sketch must #include this library, and the Wire library.  
// (Wire is a standard library included with Arduino.):
```

```
#include <SFE_BMP180.h>  
#include <Wire.h>  
  
// You will need to create an SFE_BMP180 object, here called "pressure":  
  
SFE_BMP180 pressure;
```

```

#define ALTITUDE 1655.0 // Altitude of SparkFun's HQ in Boulder, CO. in meters

void setup()
{
    Serial.begin(9600);
    Serial.println("REBOOT");

    // Initialize the sensor (it is important to get calibration values stored on the
    device).

    if (pressure.begin())
        Serial.println("BMP180 init success");
    else
    {
        // Oops, something went wrong, this is usually a connection problem,
        // see the comments at the top of this sketch for the proper connections.

        Serial.println("BMP180 init fail\n\n");
        while(1); // Pause forever.
    }
}

void loop()
{
    char status;
    double T,P,p0,a;

    // Loop here getting pressure readings every 10 seconds.
    // If you want sea-level-compensated pressure, as used in weather reports,
    // you will need to know the altitude at which your measurements are taken.
    // We're using a constant called ALTITUDE in this sketch:
    Serial.println();
    Serial.print("provided altitude: ");
    Serial.print(ALTITUDE,0);
    Serial.print(" meters, ");
    Serial.print(ALTITUDE*3.28084,0);
    Serial.println(" feet");

    // If you want to measure altitude, and not pressure, you will instead need
    // to provide a known baseline pressure. This is shown at the end of the sketch.

    // You must first get a temperature measurement to perform a pressure reading.

    // Start a temperature measurement:
    // If request is successful, the number of ms to wait is returned.
    // If request is unsuccessful, 0 is returned.

    status = pressure.startTemperature();
    if (status != 0)
    {
        // Wait for the measurement to complete:

```

```

delay(status);

// Retrieve the completed temperature measurement:
// Note that the measurement is stored in the variable T.
// Function returns 1 if successful, 0 if failure.

status = pressure.getTemperature(T);
if (status != 0)
{
    // Print out the measurement:
    Serial.print("temperature: ");
    Serial.print(T,2);
    Serial.print(" deg C, ");
    Serial.print((9.0/5.0)*T+32.0,2);
    Serial.println(" deg F");
    // Start a pressure measurement:
    // The parameter is the oversampling setting, from 0 to 3 (highest res, longest
    wait).

    // If request is successful, the number of ms to wait is returned.
    // If request is unsuccessful, 0 is returned.
    status = pressure.startPressure(3);
    if (status != 0)
    {
        // Wait for the measurement to complete:
        delay(status);
        // Retrieve the completed pressure measurement:
        // Note that the measurement is stored in the variable P.
        // Note also that the function requires the previous temperature measurement
        (T).

        // (If temperature is stable, you can do one temperature measurement for a
        number of pressure measurements.)
        // Function returns 1 if successful, 0 if failure.
        status = pressure.getPressure(P,T);
        if (status != 0)
        {
            // Print out the measurement:
            Serial.print("absolute pressure: ");
            Serial.print(P,2);
            Serial.print(" mb, ");
            Serial.print(P*0.0295333727,2);
            Serial.println(" inHg");
            // The pressure sensor returns absolute pressure, which varies with altitude.
            // To remove the effects of altitude, use the sealevel function and your
            current altitude.
            // This number is commonly used in weather reports.
            // Parameters: P = absolute pressure in mb, ALTITUDE = current altitude in
            m.
            // Result: p0 = sea-level compensated pressure in mb

            p0 = pressure.sealevel(P,ALTITUDE); // we're at 1655 meters (Boulder, CO)
            Serial.print("relative (sea-level) pressure: ");

```

```

    Serial.print(p0,2);
    Serial.print(" mb, ");
    Serial.print(p0*0.0295333727,2);
    Serial.println(" inHg");

    // On the other hand, if you want to determine your altitude from the
    pressure reading,
    // use the altitude function along with a baseline pressure (sea-level or
other).
    // Parameters: P = absolute pressure in mb, p0 = baseline pressure in mb.
    // Result: a = altitude in m.

    a = pressure.altitude(P,p0);
    Serial.print("computed altitude: ");
    Serial.print(a,0);
    Serial.print(" meters, ");
    Serial.print(a*3.28084,0);
    Serial.println(" feet");
}
else Serial.println("error retrieving pressure measurement\n");
}
else Serial.println("error starting pressure measurement\n");
}
else Serial.println("error retrieving temperature measurement\n");
}
else Serial.println("error starting temperature measurement\n");
delay(5000); // Pause for 5 seconds.
}

```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/SFE_BMP180_example.ino

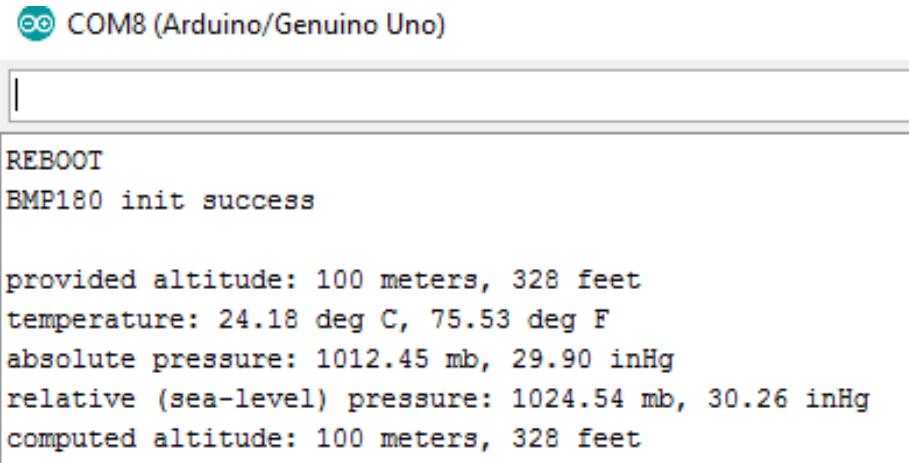
Ajustar la altitud

Antes de cargar el código , es necesario configurar su altitud actual . Ir a [elevationmap .net](#), Introduzca su dirección y comprobar la ubicación de su altura. Fijar su altura en el código. El lugar donde se debe escribir su altitud se comentó

```
#define ALTITUDE 1655.0 // Altitude of SparkFun's HQ in Boulder, CO. in meters
```

Demostración

Después de cargar el código, abra el monitor en serie a una velocidad de transmisión de 9600. Usted verá sus lecturas de los sensores.

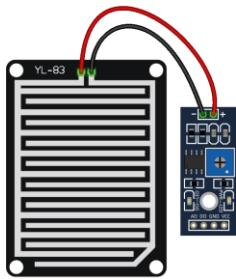


The screenshot shows the Arduino Serial Monitor interface. At the top, it says "COM8 (Arduino/Genuino Uno)". Below the header is a text area containing the following output:

```
|  
REBOOT  
BMP180 init success  
  
provided altitude: 100 meters, 328 feet  
temperature: 24.18 deg C, 75.53 deg F  
absolute pressure: 1012.45 mb, 29.90 inHg  
relative (sea-level) pressure: 1024.54 mb, 30.26 inHg  
computed altitude: 100 meters, 328 feet
```

Terminando

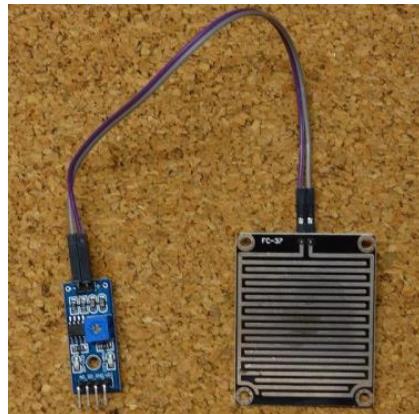
El BMP180 es un sensor interesante para ser utilizado en su propia estación meteorológica. Debido a que la presión cambia con la altitud, este sensor también puede calcular la altitud.



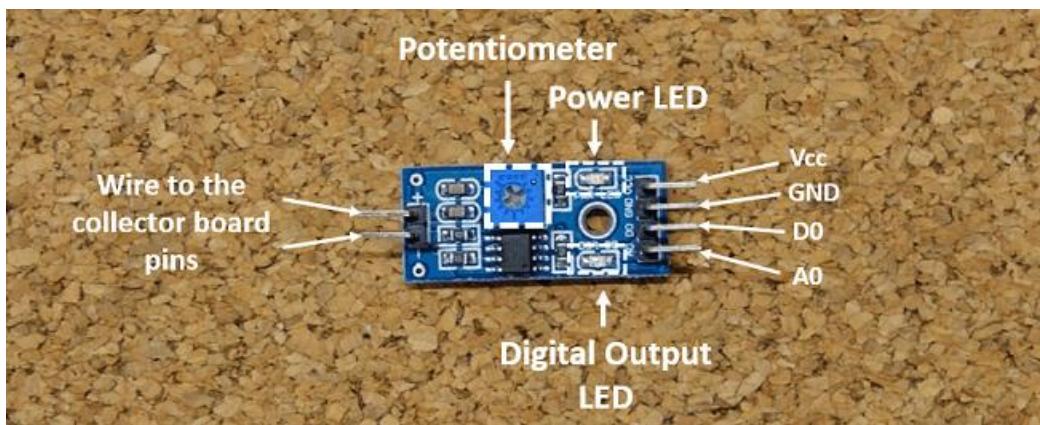
FC-37 o YL-83 Sensor de lluvia

El sensor de lluvia se utiliza para detectar agua y puede detectar más allá de lo que una lata sensor de humedad.

El sensor FC-37 de lluvia (u otras versiones como YL-83) está configurado por dos piezas: la tarjeta electrónica (a la izquierda) y la placa de colector (a la derecha) que recoge las gotas de agua, como se puede ver en la siguiente figura:



El sensor de lluvia tiene un potenciómetro incorporado para la regulación de la sensibilidad de la salida digital (D0). También tiene un LED de encendido que se ilumina cuando el sensor está activado y una salida digital LED.

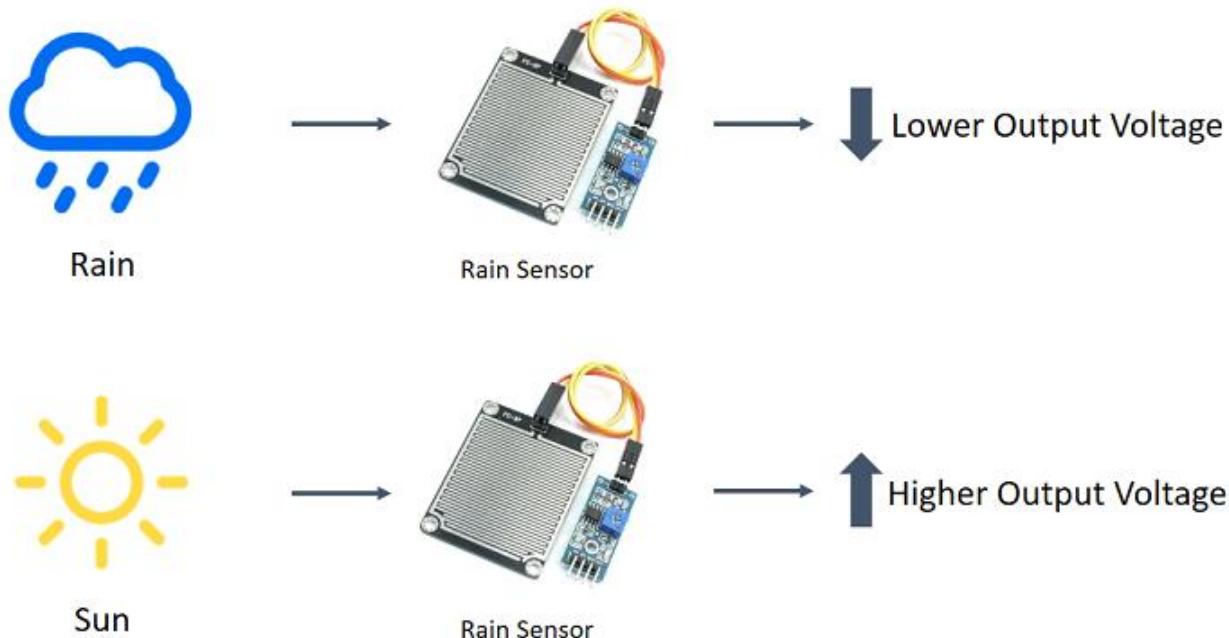


¿Como funciona?

En términos simples, la resistencia de la placa de colector varía de acuerdo a la cantidad de agua en su superficie.

Cuando la placa esta:

- Mojada: la resistencia aumenta, y la tensión de salida disminuye
- Seca: la resistencia es más baja, y la tensión de salida es mayor



¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [YL-83 rain sensor](#)

Sensor de lluvia con Arduino: Ejemplo

Este es un ejemplo sencillo para mostrar cómo se puede utilizar el sensor de lluvia en sus proyectos con Arduino . Usted va a leer los valores de los sensores analógicos e imprimirllos en el monitor serie Arduino IDE.

Para este ejemplo, tendrá las siguientes partes:

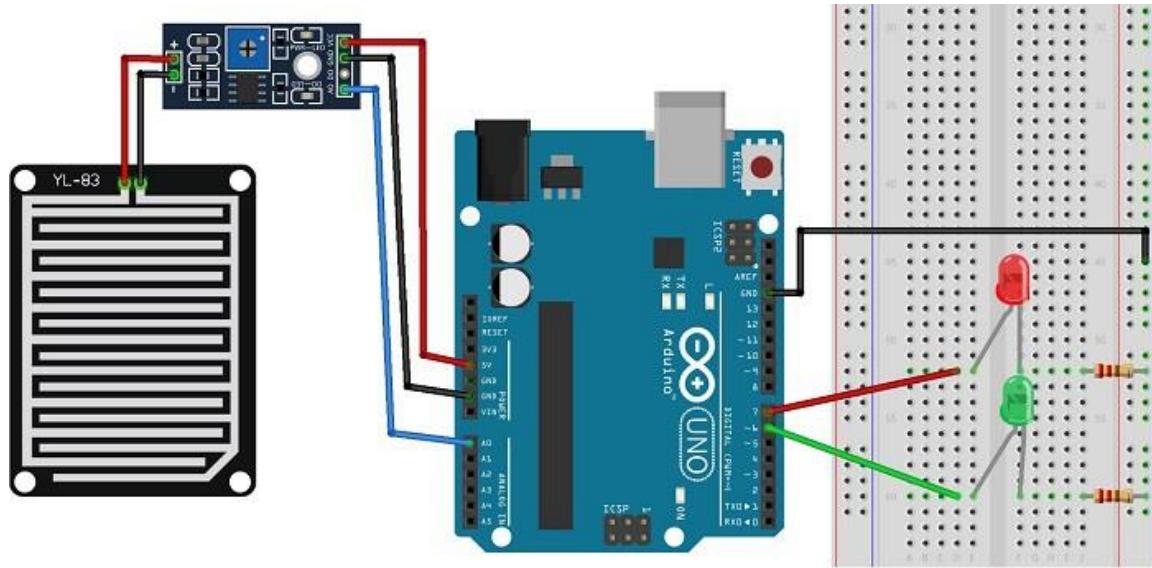
Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	YL-83 Rain Sensor	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	2× 220Ω resistor	Find best price on Maker Advisor
	2× LEDs	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Pin de cableado

Sensor Pin	Wiring to Arduino Uno
A0	Any analog pin (A5 in this example)
D0	Digital pins
GND	GND
VCC	5V

Esquemático

Siga el siguiente diagrama esquemático para completar el proyecto:



Código

Sube el siguiente esquema para la placa Arduino (no dude en ajustar la variable thresholdValue con un valor de umbral diferente)::

```
/*
All the resources for this project:
http://randomnerdtutorials.com/
*/
int rainPin = A0;
int greenLED = 6;
int redLED = 7;
// you can adjust the threshold value
int thresholdValue = 500;

void setup() {
    pinMode(rainPin, INPUT);
    pinMode(greenLED, OUTPUT);
    pinMode(redLED, OUTPUT);
    digitalWrite(greenLED, LOW);
    digitalWrite(redLED, LOW);
    Serial.begin(9600);
}

void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(rainPin);
    Serial.print(sensorValue);
    if(sensorValue < thresholdValue) {
        Serial.println(" - It's wet");
        digitalWrite(greenLED, LOW);
        digitalWrite(redLED, HIGH);
    }
    else {
        Serial.println(" - It's dry");
    }
}
```

```
    digitalWrite(greenLED, HIGH);
    digitalWrite(redLED, LOW);
}
delay(500);
}
```

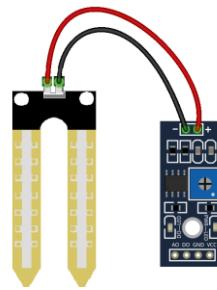
{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/rain_sensor_arduino.ino

Abra el monitor serie Arduino IDE para ver los resultados . A continuación , añadir gotas de agua a la junta del colector y ver cómo cambia las lecturas . Cuando el valor está por debajo de un cierto umbral , un LED rojo se enciende , y cuando el valor sube por encima de un cierto umbral, un LED verde se iluminará.

Terminando

Si quieres saber cuando está lloviendo, es necesario configurar el sensor de lluvia con el Arduino exterior. Tenga en cuenta que usted debe proteger su Arduino y su circuito de agua. Una caja impermeable proyecto puede ser útil en esta situación.



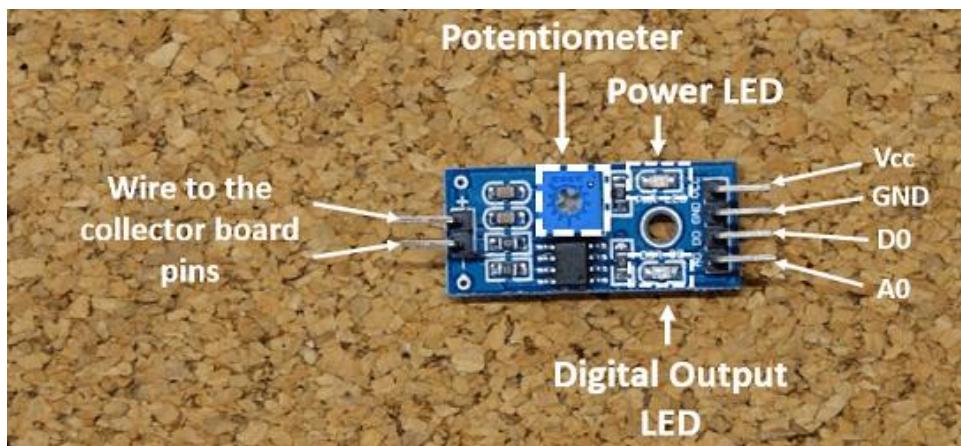
YL-69 o HL-69 sensor de humedad de suelo

El sensor de humedad del suelo o el higrómetro se utiliza generalmente para detectar la humedad del suelo. Por lo tanto, es perfecto para construir un sistema de riego automático o para controlar la humedad del suelo de sus plantas.

El sensor está configurado por dos piezas: el tablero electrónico (a la derecha), y la sonda con dos almohadillas, que detecta el contenido de agua (a la izquierda).



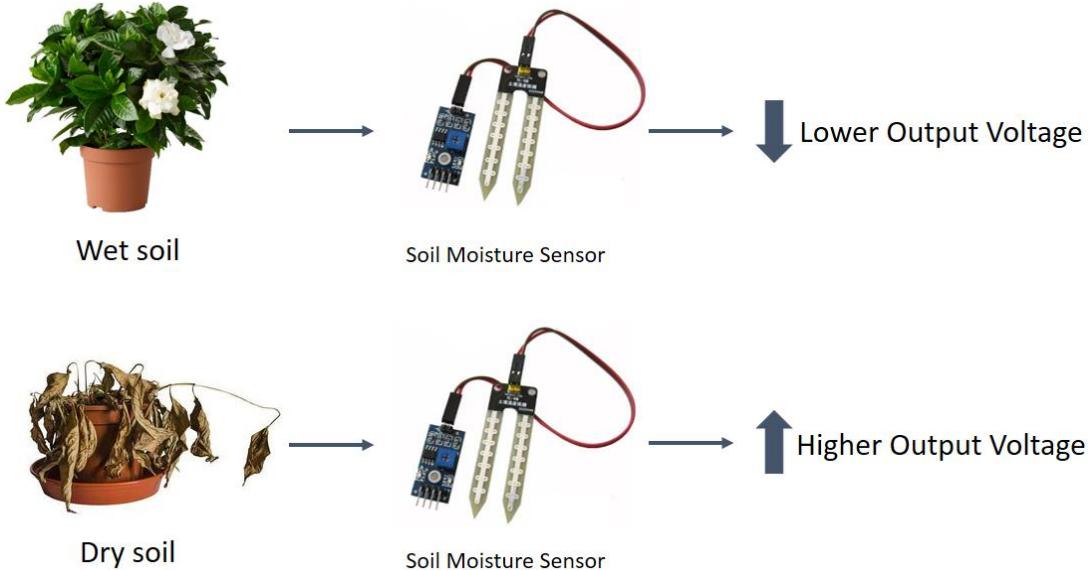
El sensor tiene un potenciómetro incorporado para la regulación de la sensibilidad de la salida digital (D0), un LED de alimentación y una salida digital LED, como se puede ver en la siguiente figura..



¿Como funciona?

El voltaje que las salidas del sensor cambia en consecuencia para el contenido de agua en el suelo. Cuando el suelo esta:

- **Mojado:** la tensión de salida disminuye
- **Seco:** La tensión de salida aumenta



La salida puede ser una señal digital l (D0) alta o baja, dependiendo del contenido de agua. Si la humedad del suelo excede un cierto valor umbral predefinido , los módulos de salidas de bajo, de lo contrario salidas de alta. El valor umbral para la señal digital se puede ajustar mediante el potenciómetro. La salida también puede ser una **Señal analogica** y de manera que obtendrá un valor entre 0 y 1023

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [HL-69 soil moisture sensor](#)

Sensor de humedad del suelo con el Arduino

En este ejemplo, podrá leer los valores de salida del sensor analógico utilizando el Arduino e imprimir las lecturas en el monitor serie Arduino IDE.

Para este ejemplo, necesitará los siguientes componentes:

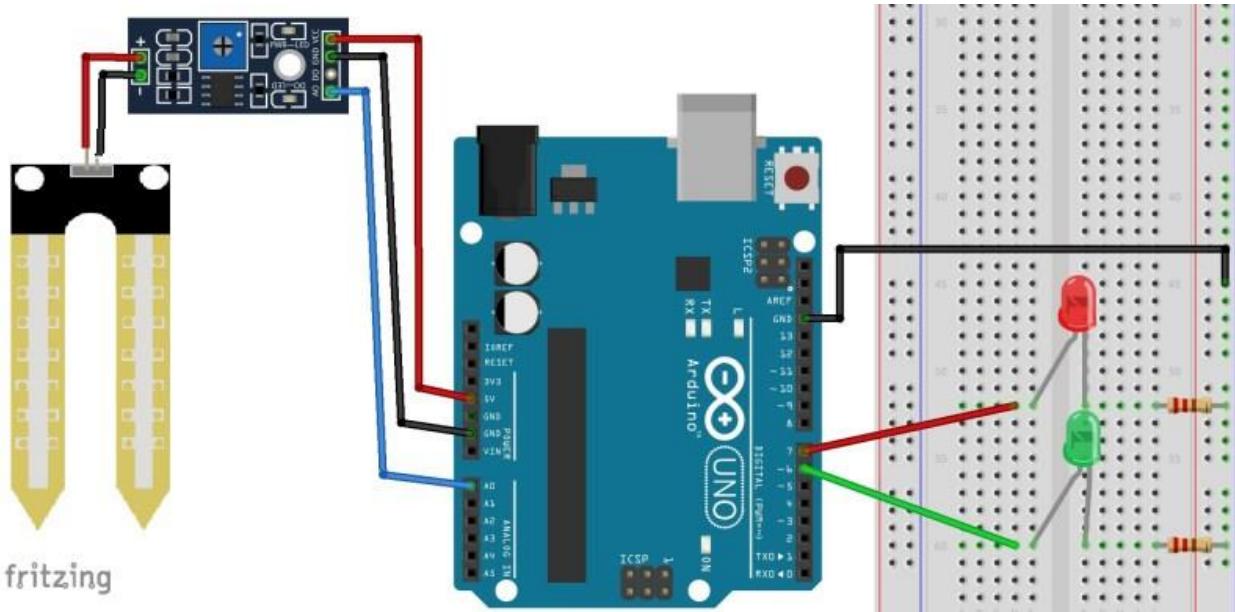
Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	HL-69 Rain Sensor	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	2× 220Ω Resistor	Find best price on Maker Advisor
	2× LEDs	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Pin de cableado

Sensor Pin	Wiring to Arduino Uno
A0	Any analog pin (A5 in this example)
D0	Digital pins
GND	GND
VCC	5V

Círcito

Para completar el proyecto, siga el siguiente esquema:



Código

Sube el siguiente esquema para la placa Arduino:

```
/* All the resources for this project:  
http://randomnerdtutorials.com/  
*/  
  
int rainPin = A0;  
int greenLED = 6;  
int redLED = 7;  
// you can adjust the threshold value  
int thresholdValue = 800;  
  
void setup(){  
  pinMode(rainPin, INPUT);  
  pinMode(greenLED, OUTPUT);  
  pinMode(redLED, OUTPUT);  
  digitalWrite(greenLED, LOW);  
  digitalWrite(redLED, LOW);  
  Serial.begin(9600);  
}  
  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(rainPin);  
  Serial.print(sensorValue);  
  if(sensorValue < thresholdValue){
```

```
    Serial.println(" - Doesn't need watering");
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, HIGH);
}
else {
    Serial.println(" - Time to water your plant");
    digitalWrite(redLED, HIGH);
    digitalWrite(greenLED, LOW);
}
delay(500);
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/soil_moisture_sensor_arduino.ino

Abra el monitor serie Arduino IDE para ver los valores. A continuación, probar su sensor en un húmedo y en un suelo seco y ver lo que sucede.

Cuando el valor analógico pasa por encima de un cierto umbral, un LED rojo se encenderá (indica que la planta necesita riego), y cuando el valor cae por debajo de un cierto umbral, un LED verde se enciende (indica que la planta es aceptable).

Terminando

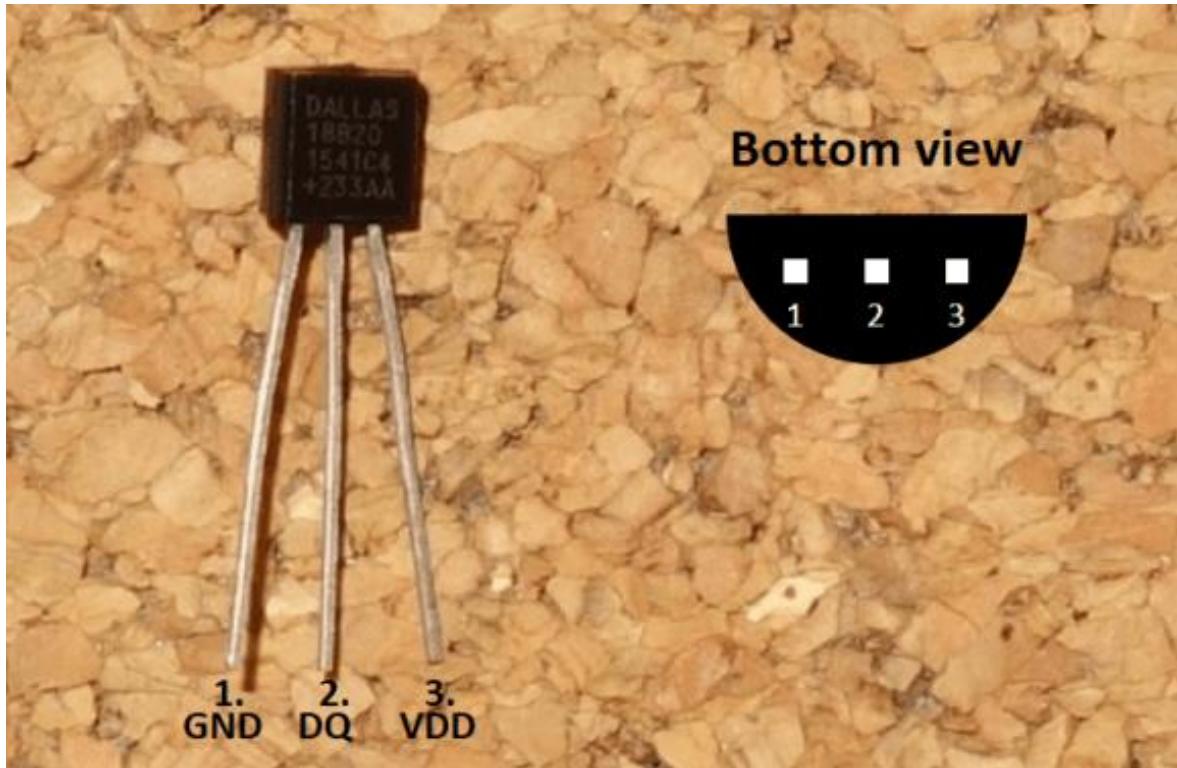
El sensor de humedad permite monitorizar el contenido de agua en el suelo. Esto es útil si usted quiere construir un sistema de riego automático . También puede utilizarlo para simplemente controlar su humedad plantas del suelo.



Sensor de temperatura DS18B20

El sensor de temperatura DS18B20 es un sensor digital de la temperatura 1-wire. Esto significa que se puede leer la temperatura con una configuración de circuito muy simple. Se comunica el bus común, lo que significa que se puede conectar varios dispositivos y leer sus valores usando sólo un pin digital del Arduino.

El sensor tiene sólo tres pines como se puede ver en la siguiente figura:



El DS18B20 también está disponible en versión resistente al agua:



Características

He aquí algunas de las características principales del sensor de temperatura DS18B20:

- Comunica a través de la comunicación del bus 1-wire
- temperatura Rango de funcionamiento: -55°C a 125°C
- Precisión +/- 0,5 °C (entre el rango de -10°C a 85°C)

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [DS18B20 digital temperature sensor](#)

Sensor de temperatura DS18B20 con el Arduino

En este ejemplo, usted leerá la temperatura con el sensor DS18B20 y el Arduino. Las lecturas se muestran en el monitor de serie Arduino.

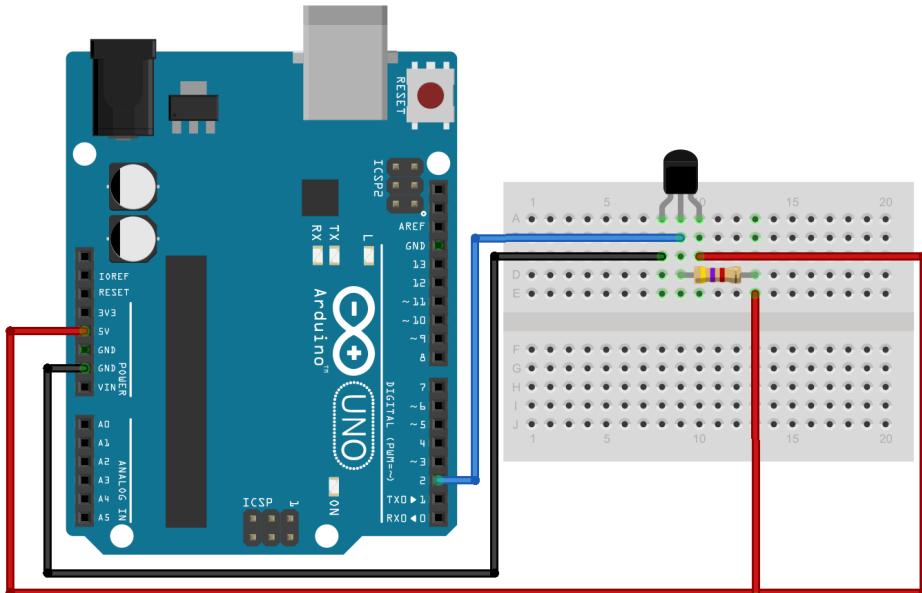
Para este ejemplo, tendrá las siguientes partes:

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	DS18B20 temperature sensor	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	4.7kΩ Resistor	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

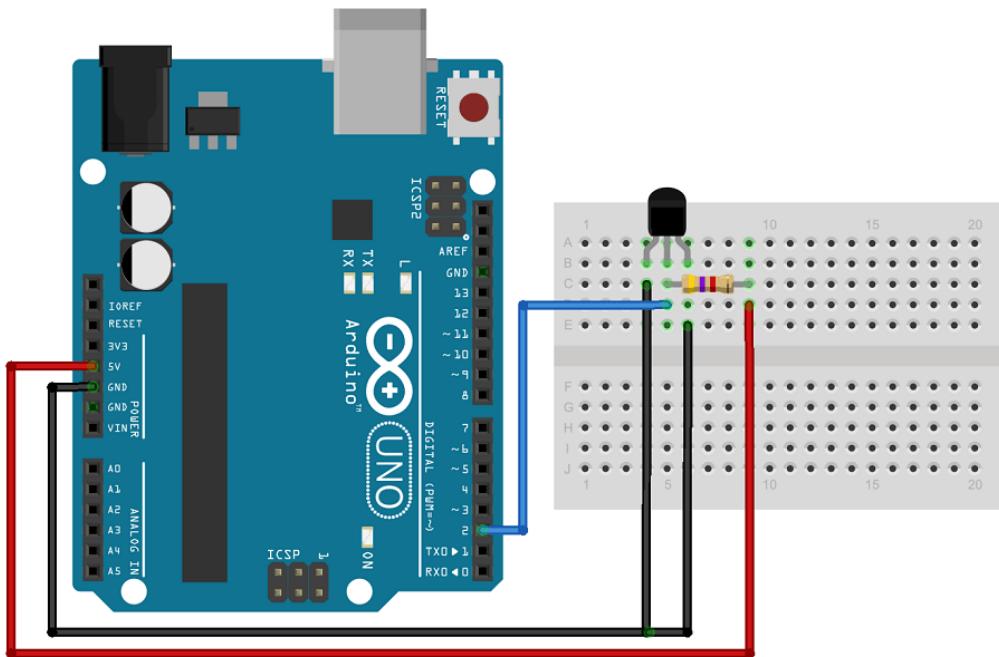
Circuito

El sensor puede funcionar en dos modos diferentes:

- **Modo normal:** se necesita conexión de 3 hilos. Aquí está el esquema que debe seguir



- **Modo parásito:** sólo 2 cables requeridos, los datos y la tierra. El sensor deriva su poder de la línea de datos. En este caso, aquí está el esquema que debe seguir:



fritzing

Usted puede leer la temperatura de más de un sensor a la vez con un solo PIN de Arduino digital. Para ello, sólo tiene que conectar todos los pines de DQ a cualquier pin de Arduino digital.

Código

Tendrá que instalar la Biblioteca OneWire y Biblioteca DallasTemperature.

Instalación de la biblioteca OneWire

1. [Click here to download the OneWire library](#). Usted debe tener una carpeta .zip en sus descargas
2. Descomprimir la carpeta .zip y usted debe obtener OneWire-master carpeta
3. Cambiar el nombre de la carpeta de OneWire-master a OneWire
4. Mover la carpeta OneWire a su carpeta de bibliotecas de instalación Arduino IDE
5. Por último, vuelva a abrir el IDE de Arduino

Instalación de la biblioteca DallasTemperature

1. [Click here to download the DallasTemperature library](#). Usted debe tener una carpeta .zip en sus descargas

- 2.** Descomprimir la carpeta .zip y usted debe obtener la carpeta Arduino-Temperature-Control-Library-master
 - 3.** Rebautizar tu carpeta from Arduino-Temperature-Control-Library-master a DallasTemperature
 - 4.** Mover la carpeta DallasTemperature a tu Arduino IDE carpeta de instalación de librerías.
- 5. Por último, vuelva a abrir el IDE de Arduino**

Después de instalar las librerías necesarias, subir el siguiente código a la placa Arduino.

```
/* Rui Santos
   Complete project details at http://randomnerdtutorials.com
   Based on the Dallas Temperature Library example
*/
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is connected to the Arduino digital pin 2
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

void setup(void)
{
    // Start serial communication for debugging purposes
    Serial.begin(9600);
    // Start up the library
    sensors.begin();
}

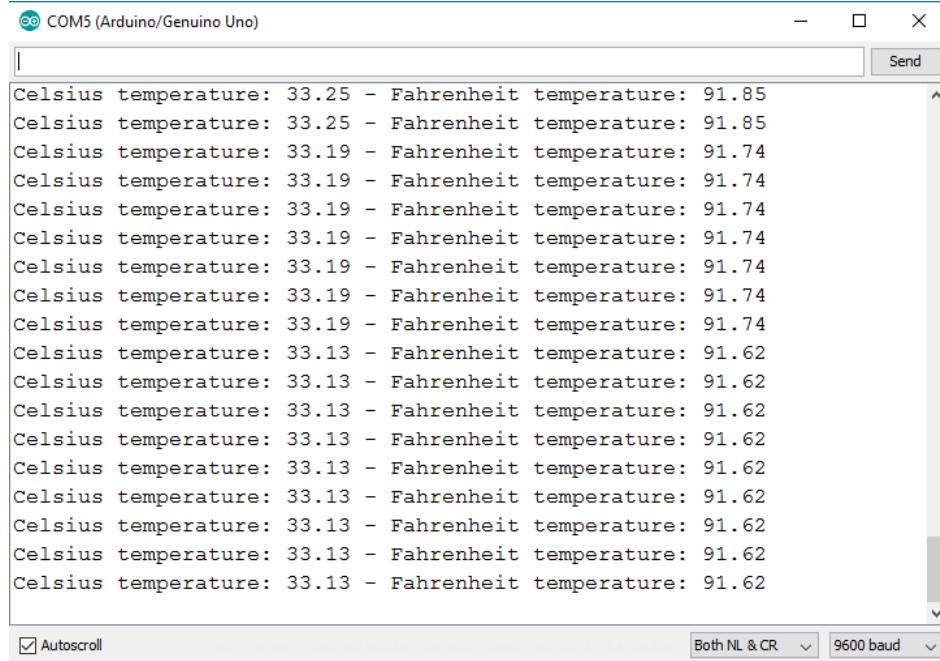
void loop(void) {
    // Call sensors.requestTemperatures() to issue a global temperature and Requests to
    // all devices on the bus
    sensors.requestTemperatures();

    Serial.print("Celsius temperature: ");
    // Why "byIndex"? You can have more than one IC on the same bus. 0 refers to the
    // first IC on the wire
    Serial.print(sensors.getTempCByIndex(0));
    Serial.print(" - Fahrenheit temperature: ");
    Serial.println(sensors.getTempFByIndex(0));
    delay(1000);
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/DS18B20_Temperature_Sensor_Arduino.ino

Por último, se debe abrir el monitor serie Arduino IDE a una velocidad de transmisión 9600 y verá la temperatura mostrada tanto en grados Celsius y Fahrenheit:



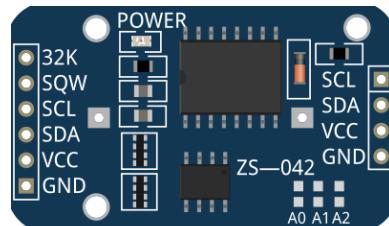
The screenshot shows the Arduino Serial Monitor window titled "COM5 (Arduino/Genuino Uno)". The window displays a series of temperature readings in Celsius and Fahrenheit. The text in the monitor reads:

```
Celsius temperature: 33.25 - Fahrenheit temperature: 91.85
Celsius temperature: 33.25 - Fahrenheit temperature: 91.85
Celsius temperature: 33.19 - Fahrenheit temperature: 91.74
Celsius temperature: 33.13 - Fahrenheit temperature: 91.62
```

At the bottom of the window, there are three buttons: "Autoscroll" (checked), "Both NL & CR", and "9600 baud".

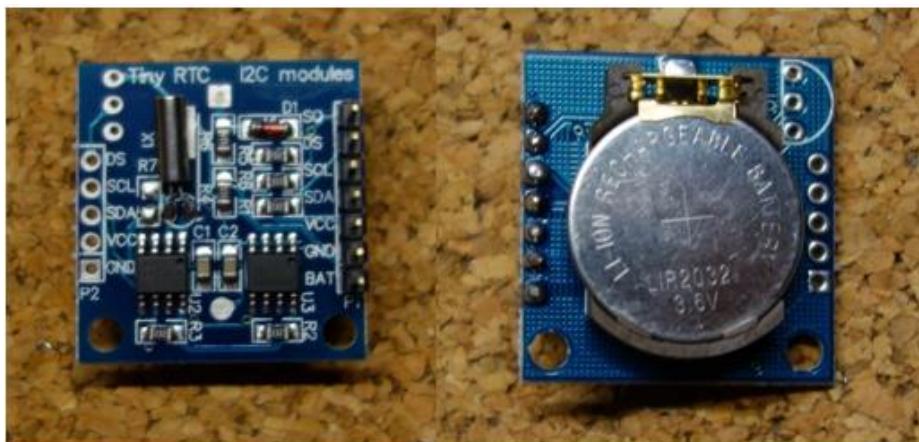
Terminando

El sensor de temperatura DS18B20 es muy útil. Se le permite leer la temperatura en grados Celsius y Fahrenheit a través de una comunicación por cable. Esto significa que puede tener lecturas de varios sensores sin multiplicar las conexiones con el Arduino.



DS1307 o DS3231 reloj de tiempo real (RTC)

El módulo de reloj en tiempo real es el de la figura siguiente (frontal y posterior).



La primera vez que utiliza este módulo, es necesario soldar algunos pines de cabecera.

Como se puede ver en la imagen superior, el módulo tiene una batería de reserva instalado. Esto permite que el módulo de retener el tiempo, incluso cuando no está siendo alimentado por el Arduino. De esta manera, cada vez que se enciende y apaga su módulo, el tiempo no se restablece.

Este módulo utiliza la comunicación I2C. Esto significa que se comunica con el Arduino usando sólo 2 pines.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [DS1307 Real Time Clock Module](#)

Cableado de Pines

Module Pin	Wiring to Arduino Uno
SCL	A5
SDA	A4
VCC	5V
GND	GND

Si está utilizando otra placa Arduino en lugar de la ONU, echa un vistazo a lo que son sus pines SCL y SDA..

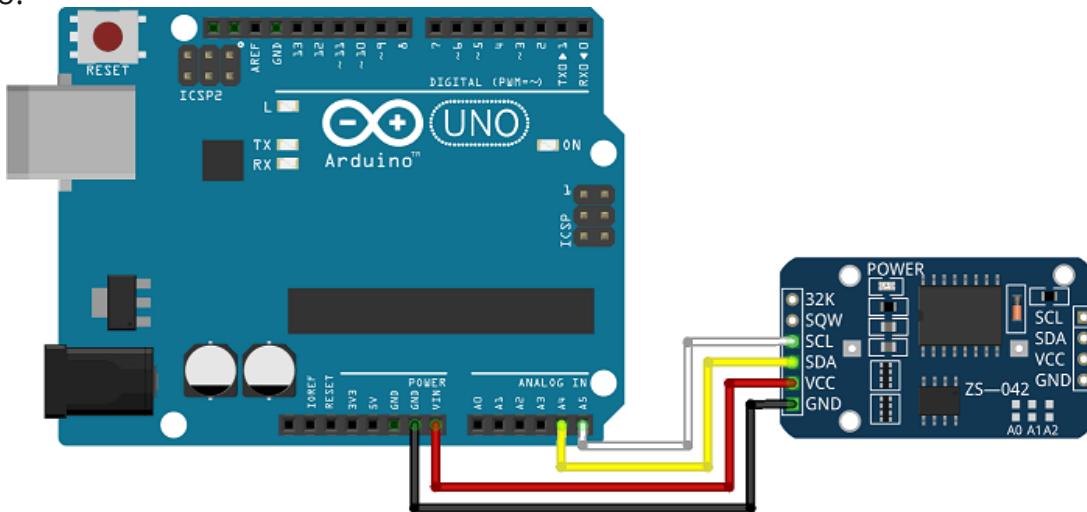
- **Nano:** SDA (A4); SCL(A5)
- **MEGA:** SDA (20); SCL(21)
- **Leonardo:** SDA (20); SCL(21)

Viendo la fecha y la hora en el monitor serie

En este ejemplo se va a mostrar los datos y el tiempo en el monitor serie.

Circuito

Conectar el módulo de reloj en tiempo real a tu Arduino como en los esquemas de abajo.



fritzing

Código

Trabajar con el RTC requiere dos pasos importantes:

- ajuste de la hora actual , de manera que el RTC sabe qué hora es
- reteniendo el momento , de manera que el RTC siempre da la hora correcta, incluso cuando está apagado

Ajustar la hora actual en el reloj de tiempo real

Para el ajuste de la hora actual necesita cambiar el código proporcionado. Ajustar la hora actual en la función setDS3231time ()

```
// set the initial time here:  
// DS3231 seconds, minutes, hours, day, date, month, year  
setDS3231time(33,53,16,1,25,9,16);
```

Los parámetros para la función aparecen resaltados en rojo: segundos, minutos, horas, días de la semana, día, mes y año (en este orden). El domingo es el día 1 de la semana y el sábado es 7. No se olvide de descomentar esa línea de código.

Después de ajustar la hora actual, puede cargar el código proporcionado con las modificaciones necesarias.

El código proporcionado fue escrito por John Boxall de tronixstuff . Puede leer su tutorial [aquí](#).

```
// Written by John Boxall from http://tronixstuff.com

---

  
#include "Wire.h"  
#define DS3231_I2C_ADDRESS 0x68  
// Convert normal decimal numbers to binary coded decimal  
byte decToBcd(byte val){  
    return( (val/10*16) + (val%10) );  
}  
// Convert binary coded decimal to normal decimal numbers  
byte bcdToDec(byte val){  
    return( (val/16*10) + (val%16) );  
}  
void setup(){  
    Wire.begin();  
    Serial.begin(9600);  
    // set the initial time here:  
    // DS3231 seconds, minutes, hours, day, date, month, year  
    setDS3231time(30,42,16,5,13,10,16);  
}  
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte  
dayOfMonth, byte month, byte year){
```

```

// sets time and date data to DS3231
Wire.beginTransmission(DS3231_I2C_ADDRESS);
Wire.write(0); // set next input to start at the seconds register
Wire.write(decToBcd(second)); // set seconds
Wire.write(decToBcd(minute)); // set minutes
Wire.write(decToBcd(hour)); // set hours
Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday, 7=Saturday)
Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
Wire.write(decToBcd(month)); // set month
Wire.write(decToBcd(year)); // set year (0 to 99)
Wire.endTransmission();
}

void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set DS3231 register pointer to 00h
    Wire.endTransmission();
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
    // request seven bytes of data from DS3231 starting from register 00h
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
}
void displayTime(){
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    // retrieve data from DS3231
    readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
    &year);
    // send it to the serial monitor
    Serial.print(hour, DEC);
    // convert the byte variable to a decimal number when displayed
    Serial.print(":");
    if (minute<10){
        Serial.print("0");
    }
    Serial.print(minute, DEC);
    Serial.print(":");
    if (second<10){
        Serial.print("0");
    }
    Serial.print(second, DEC);
    Serial.print(" ");
}

```

```

Serial.print(dayOfMonth, DEC);
Serial.print("/");
Serial.print(month, DEC);
Serial.print("/");
Serial.print(year, DEC);
Serial.print(" Day of week: ");
switch(dayOfWeek) {
case 1:
    Serial.println("Sunday");
    break;
case 2:
    Serial.println("Monday");
    break;
case 3:
    Serial.println("Tuesday");
    break;
case 4:
    Serial.println("Wednesday");
    break;
case 5:
    Serial.println("Thursday");
    break;
case 6:
    Serial.println("Friday");
    break;
case 7:
    Serial.println("Saturday");
    break;
}
}

void loop() {
displayTime(); // display the real-time clock data on the Serial Monitor,
delay(1000); // every second
}

```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/rtc_example.ino

Retener el tiempo en el reloj de tiempo real

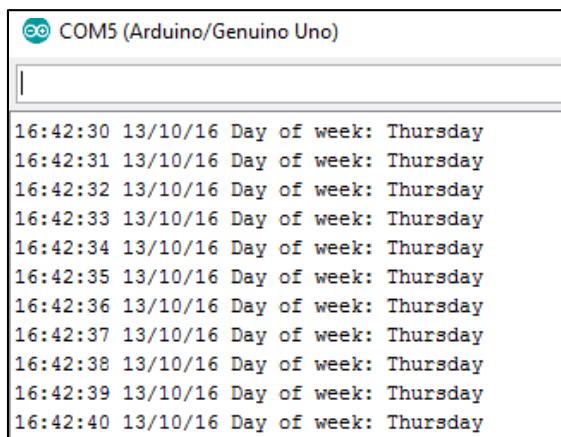
Si no desea restablecer el tiempo cada vez que el RTC está apagado , usted debe hacer lo siguiente : después de configurar el tiempo , usted debe comentar la función que ajusta el tiempo y cargar el código de nuevo.

```
// set the initial time here:  
// DS3231 seconds, minutes, hours, day, date, month, year  
//setDS3231time(33,53,16,1,25,9,16);
```

Este es un paso muy importante para establecer el tiempo en su RTC. Si no se hace esto, cada vez que se reinicia el RTC, se mostrará el tiempo que ha establecido con anterioridad en lugar de la hora actual.

Demostración

Abrir el monitor de puerto serie a una velocidad de transmisión de 9600 y verá los resultados. Aquí está el monitor de serie que muestra la fecha y hora actuales.



```
16:42:30 13/10/16 Day of week: Thursday  
16:42:31 13/10/16 Day of week: Thursday  
16:42:32 13/10/16 Day of week: Thursday  
16:42:33 13/10/16 Day of week: Thursday  
16:42:34 13/10/16 Day of week: Thursday  
16:42:35 13/10/16 Day of week: Thursday  
16:42:36 13/10/16 Day of week: Thursday  
16:42:37 13/10/16 Day of week: Thursday  
16:42:38 13/10/16 Day of week: Thursday  
16:42:39 13/10/16 Day of week: Thursday  
16:42:40 13/10/16 Day of week: Thursday
```

Terminando

El módulo RTC es útil y se puede utilizar como un reloj, temporizador, etc. Un proyecto muy interesante que ver con el RTC es un reloj despertador con pantalla OLED y, por ejemplo.

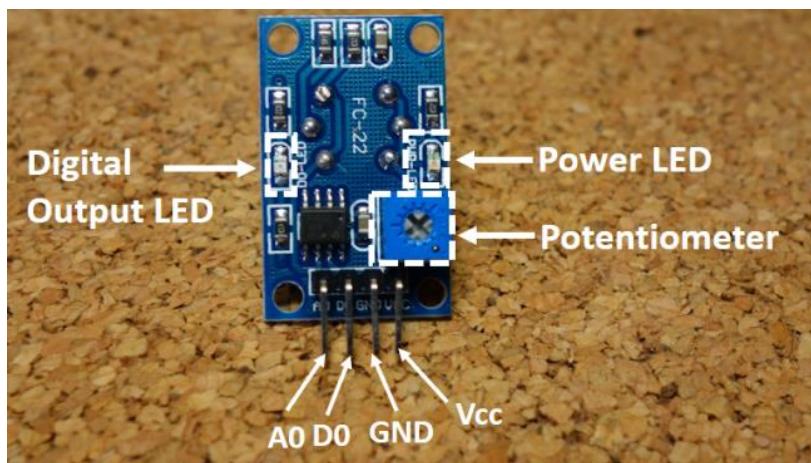


MQ-2 Gas / sensor de humo

Esta guía muestra cómo construir un detector de humo que emite un sonido cuando se detecta humo o gases inflamables. El sensor MQ-2 humo se muestra en la siguiente figura:



El sensor de humo MQ-2 es sensible al humo y a los siguientes gases inflamables: LPG, butano , propano , metano , alcohol e hidrógeno . La resistencia a través del sensor es diferente dependiendo del tipo de gas. El sensor de humo tiene un potenciómetro incorporado que le permite ajustar el umbral de salida digital del sensor(D0). Este umbral se establece el valor por encima del cual el pin de salida digital de una señal de voluntad ALTO.

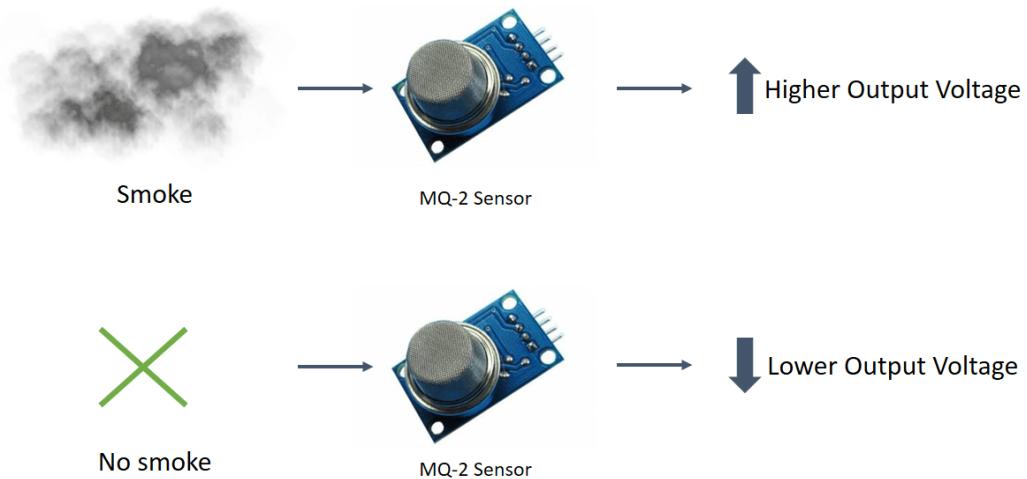


¿Como funciona?

El voltaje que las salidas del sensor cambia en consecuencia para el nivel de humo / gas que existe en la atmósfera. El sensor genera una tensión que es proporcional a la concentración de humo / gas.

En otras palabras , la relación entre la tensión y la concentración de gas es la siguiente:

- El la concentración de gas, mayor es la tensión de salida mayor
- El la concentración de gas, menor será la tensión de salida más baja



La salida puede ser una señal analógica (A0) que se puede leer con una entrada analógica de la Arduino o una salida digital (D0) que se puede leer con una entrada digital de la Arduino.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [MQ-2 Gas and Smoke Sensor](#)

Sensor de gas con Arduino

En este ejemplo, se lee la tensión de salida analógica del sensor. Cuando el humo se alcanza un cierto nivel, que hará sonar un zumbador y un LED rojo se encenderá .Cuando la tensión de salida está por debajo de ese nivel, un LED verde se enciende . Para este ejemplo, tendrá las siguientes partes:

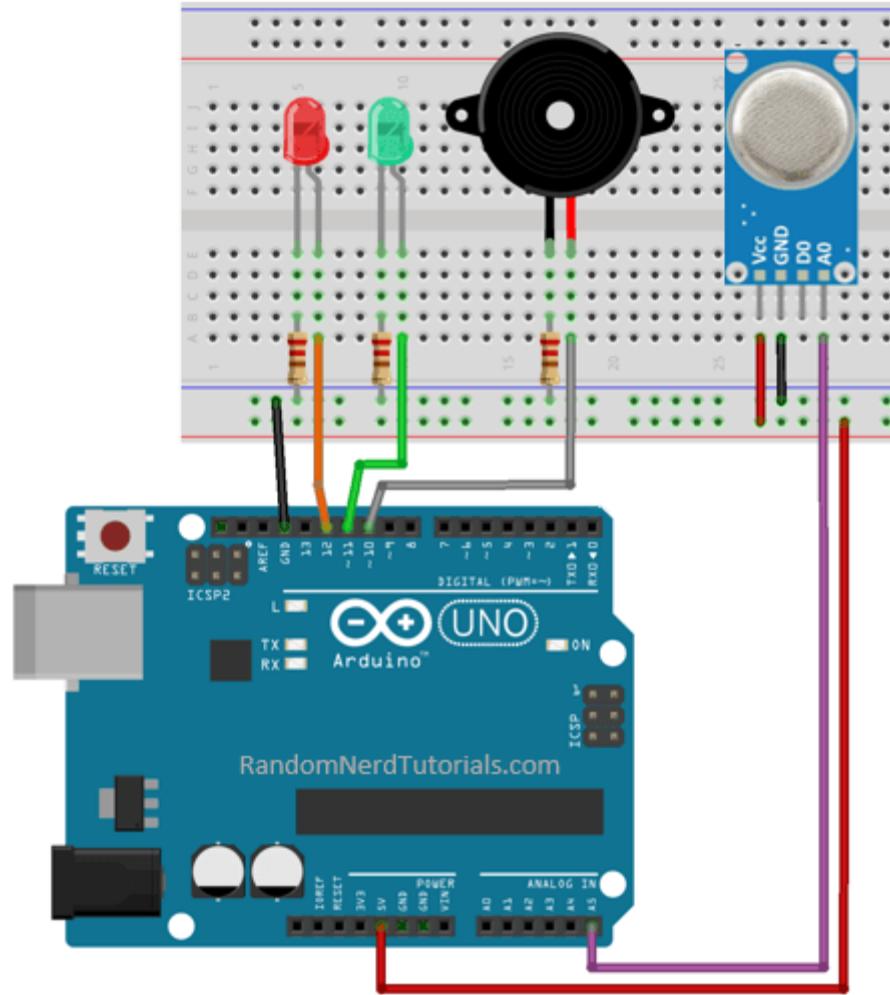
Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	MQ-2 gas/smoke sensor	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	2 x LEDs	Find best price on Maker Advisor
	3 x 220kΩ Resistor	Find best price on Maker Advisor
	Piezo buzzer	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Cableado de los Pines

Sensor Pin	Wiring to Arduino Uno
A0	Analog pins
D0	Digital pins
GND	GND
VCC	5V

Círculo

Siga el siguiente diagrama esquemático para completar e proyecto:



Código

Sube el siguiente esquema para la placa Arduino (no dude en ajustar la variable sensorThres con un valor de umbral diferente):

```
*****
All the resources for this project:  
http://randomnerdtutorials.com/
*****  
  
int redLed = 12;  
int greenLed = 11;  
int buzzer = 10;  
int smokeA0 = A5;  
// Your threshold value  
int sensorThres = 400;
```

```

void setup() {
    pinMode(redLed, OUTPUT);
    pinMode(greenLed, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(smokeA0, INPUT);
    Serial.begin(9600);
}

void loop() {
    int analogSensor = analogRead(smokeA0);

    Serial.print("Pin A0: ");
    Serial.println(analogSensor);
    // Checks if it has reached the threshold value
    if (analogSensor > sensorThres)
    {
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, LOW);
        tone(buzzer, 1000, 200);
    }
    else
    {
        digitalWrite(redLed, LOW);
        digitalWrite(greenLed, HIGH);
        noTone(buzzer);
    }
    delay(100);
}

```

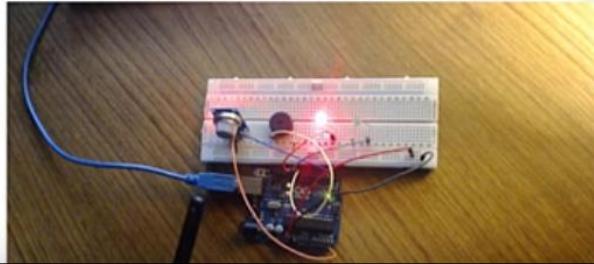
{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/smoke_detector.ino

Demostración

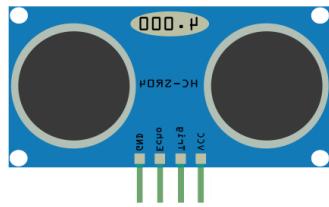
Cuando se presiona un encendedor al lado del sensor, el LED rojo se enciende y el zumbador emite un pitido. También se puede ver en el monitor serie, el cambio de valores y superando el valor umbral.

```
Pin A0: 280
Pin A0: 279
Pin A0: 280
Pin A0: 279
Pin A0: 278
Pin A0: 278
Pin A0: 278
Pin A0: 277
Pin A0: 278
Pin A0: 277
Pin A0: 277
Pin A0: 277
Pin A0: 277
Pin A0: 276
Pin A0: 275
Pin A0: 275
Pin A0: 275
Pin A0: 274
Pin A0: 274
Pin A0: 274
Pin A0: 300
Pin A0: 355
Pin A0: 402
Pin A0: 437
Pin A0: 461
Pin A0: 461
Pin A0: 494
Pin A0: 504
Pin A0: 512
Pin A0: 517
Pin A0: 520
```



Terminando

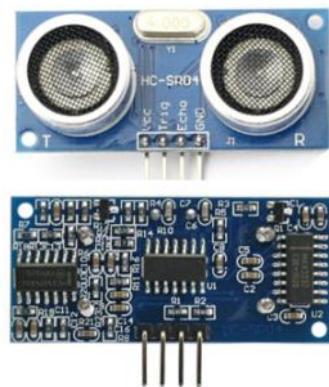
El sensor de gas MQ-2 le permite detectar una amplia variedad de gases en el medio ambiente. Es muy útil para construir un detector de humo en su casa.



El sensor ultrasónico HC-SR04

El sensor ultrasónico HC-SR04 utiliza el sonar para determinar la distancia a un objeto como murciélagos hacen. Ofrece una excelente detección gama de no contacto con una alta precisión y lecturas estables en un paquete fácil de usar. De 2 cm a 400 cm o 1" a 13 pies. Se completa con transmisor ultrasónico y el módulo receptor.

El sensor ultrasónico HC-SR04 se muestra en la siguiente figura:



Características

- Fuente de alimentación: + 5V DC
- Corriente en reposo: <2 mA
- Corriente de trabajo: 15mA
- Ángulo Effectual: <15 °
- Extendiéndose Distancia: 2 cm - 400 cm / 1 "- 13 pies
- Resolución: 0,3 cm
- Ángulo de medición: 30 grados
- Entrada activadora ancho de pulso: 10uS
- Dimensiones: 45mm x 20mm x 15mm

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [HC-SR04 Ultrasonic Sensor](#)

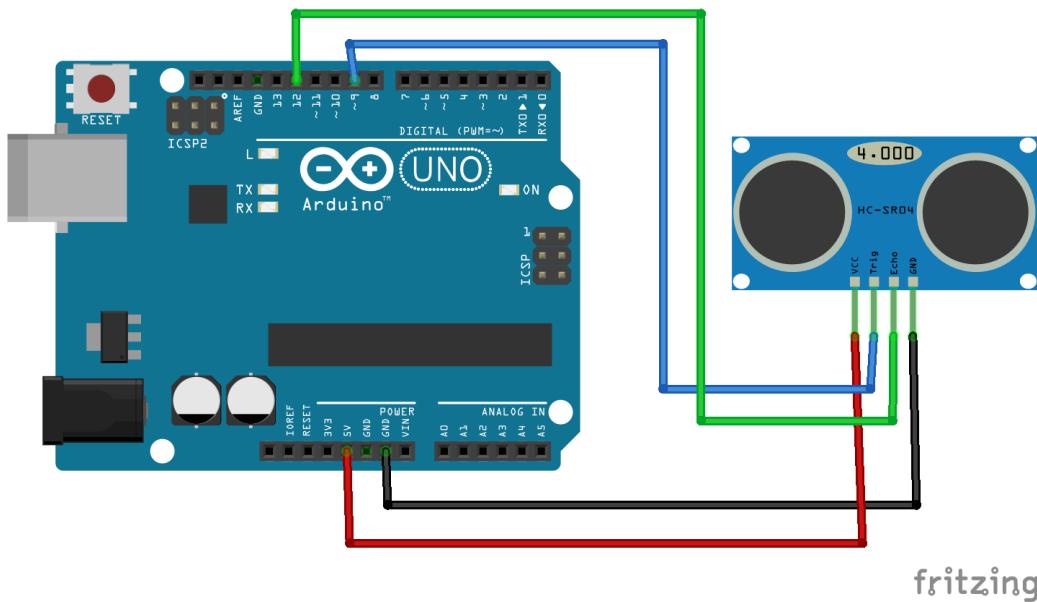
Arduino con HC - SR04 Sensor

En este proyecto el sensor ultrasónico lee y escribe la distancia en el monitor serie.

Nota: Hay una biblioteca llamada Arduino NewPing que pueden hacer su vida más fácil cuando se utiliza este sensor.

Circuito

Cablear el sensor siguiendo el diagrama esquemático siguiente.



Sensor Pin	Wiring to Arduino Uno
VCC	5V
Trig	Trigger (INPUT) D9
Echo	Echo (OUTPUT) D12
GND	GND

Código fuente

Sube el siguiente código a su IDE Arduino.

```
/*
 * created by Rui Santos, http://randomnerdtutorials.com
 *
 * Complete Guide for Ultrasonic Sensor HC-SR04
 *
 Ultrasonic sensor Pins:
 VCC: +5VDC
 Trig : Trigger (INPUT) - Pin11
 Echo: Echo (OUTPUT) - Pin 12
 GND: GND
 */
int trigPin = 11;
int echoPin = 12;
long duration, cm, inches;

void setup() {
    //Serial Port begin
    Serial.begin (9600);
    //Define inputs and outputs
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop()
{
    // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
    // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Read the signal from the sensor: a HIGH pulse whose
    // duration is the time (in microseconds) from the sending
    // of the ping to the reception of its echo off of an object.
    pinMode(echoPin, INPUT);
    duration = pulseIn(echoPin, HIGH);
    // convert the time into a distance
    cm = (duration/2) / 29.1;
    inches = (duration/2) / 74;

    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(250);
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Ultrasonic_Sensor_HC-SR04.c

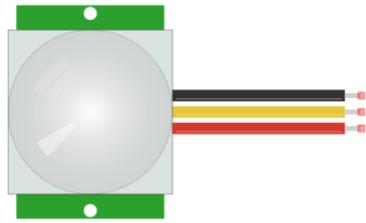
Código fuente con NewPing

A continuación hay un ejemplo usando la biblioteca NewPing. Descargar la biblioteca [aqui](#).

```
/*
 * Posted on http://randomnerdtutorials.com
 * created by http://playground.arduino.cc/Code/NewPing
 */
#include <NewPing.h>
#define TRIGGER_PIN 12
#define ECHO_PIN 11
#define MAX_DISTANCE 200
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum
distance.
void setup() {
    Serial.begin(9600);
}
void loop() {
    delay(50);
    unsigned int us = sonar.ping_cm();
    Serial.print(us);
    Serial.println("cm");
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Ultrasonic_Sensor_HC-SR04_with_NewPing.c



Sensor de movimiento PIR

El sensor de movimiento PIR es ideal para detectar el movimiento . PIR significan “ infrarrojo pasivo ”. Básicamente , las medidas del sensor de movimiento PIR luz infrarroja de los objetos en su campo de visión. Por lo tanto, se puede detectar el movimiento en base a los cambios en la luz infrarroja en el medio ambiente . Es ideal para detectar si un humano se ha movido en o fuera del rango del sensor.



El sensor en la figura anterior tiene dos potenciómetros incorporados para ajustar el tiempo de retardo (el potenciómetro a la izquierda) y la sensibilidad (el potenciómetro a la derecha).

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [PIR Motion Sensor \(HC-SR501\)](#)

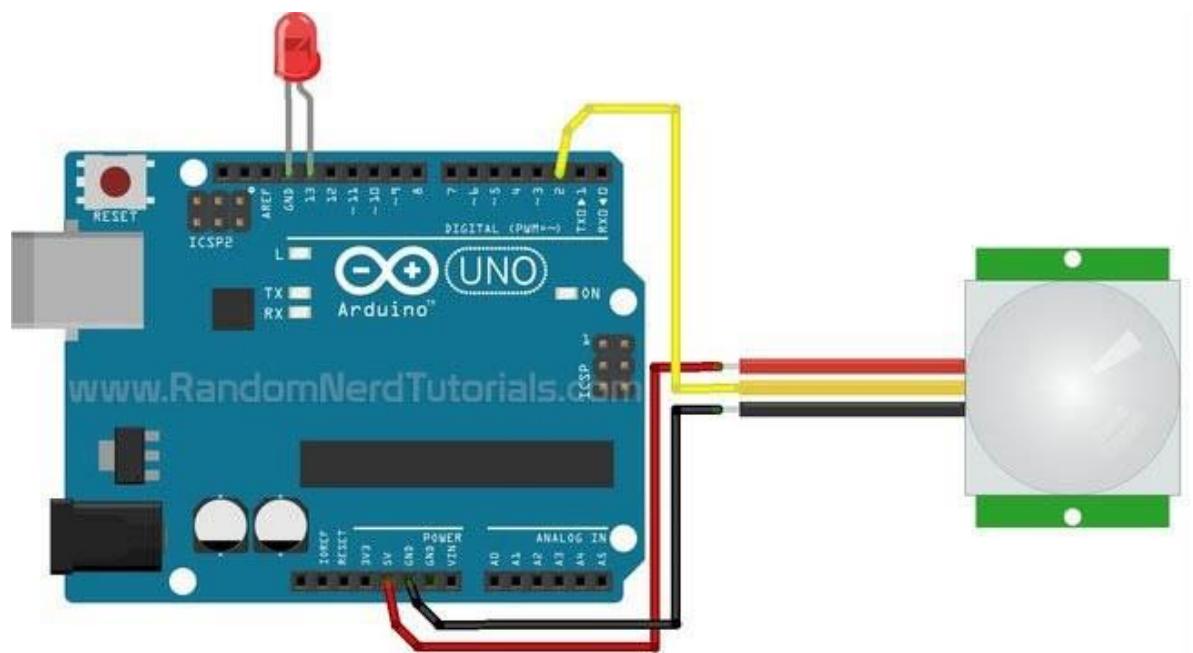
La detección de movimiento con el sensor de movimiento PIR

En este ejemplo un LED se iluminará cuando se detecta movimiento. Para este ejemplo, se necesitan los siguientes componentes:

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	PIR motion sensor	Find best price on Maker Advisor
	2 x LEDs	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Circuito

Montar todas las piezas siguiendo el esquema siguiente.



Pin wiring

Sensor Pin	Wiring to Arduino Uno
GND	GND
OUT	Arduino digital pin
5V	5V

Código

Sube el siguiente código.

```
/*
  Arduino with PIR motion sensor
  For complete project details, visit: http://RandomNerdTutorials.com/pirsensor
  Modified by Rui Santos based on PIR sensor by Limor Fried
*/

int led = 13;                      // the pin that the LED is attached to
int sensor = 2;                     // the pin that the sensor is attached to
int state = LOW;                    // by default, no motion detected
int val = 0;                        // variable to store the sensor status (value)

void setup() {
    pinMode(led, OUTPUT);           // initialize LED as an output
    pinMode(sensor, INPUT);         // initialize sensor as an input
    Serial.begin(9600);             // initialize serial
}

void loop(){
    val = digitalRead(sensor);     // read sensor value
    if (val == HIGH) {              // check if the sensor is HIGH
        digitalWrite(led, HIGH);     // turn LED ON
        delay(100);                 // delay 100 milliseconds

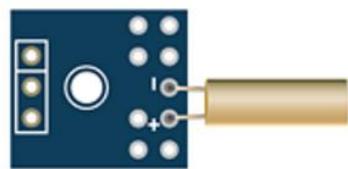
        if (state == LOW) {
            Serial.println("Motion detected!");
            state = HIGH;           // update variable state to HIGH
        }
    }
    else {
        digitalWrite(led, LOW);    // turn LED OFF
        delay(200);                // delay 200 milliseconds

        if (state == HIGH){
            Serial.println("Motion stopped!");
            state = LOW;           // update variable state to LOW
        }
    }
}
```

```
    }  
}  
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Arduino_with_PIR_motion_sensor.ino



Sensor de inclinación

El sensor de inclinación es muchas veces se hace referencia como inclinómetro, interruptor de inclinación o rodando sensor de pelota. El uso de un sensor de inclinación es una manera simple para detectar la orientación o la inclinación.

El módulo de sensor de inclinación es el uno en la siguiente figura.

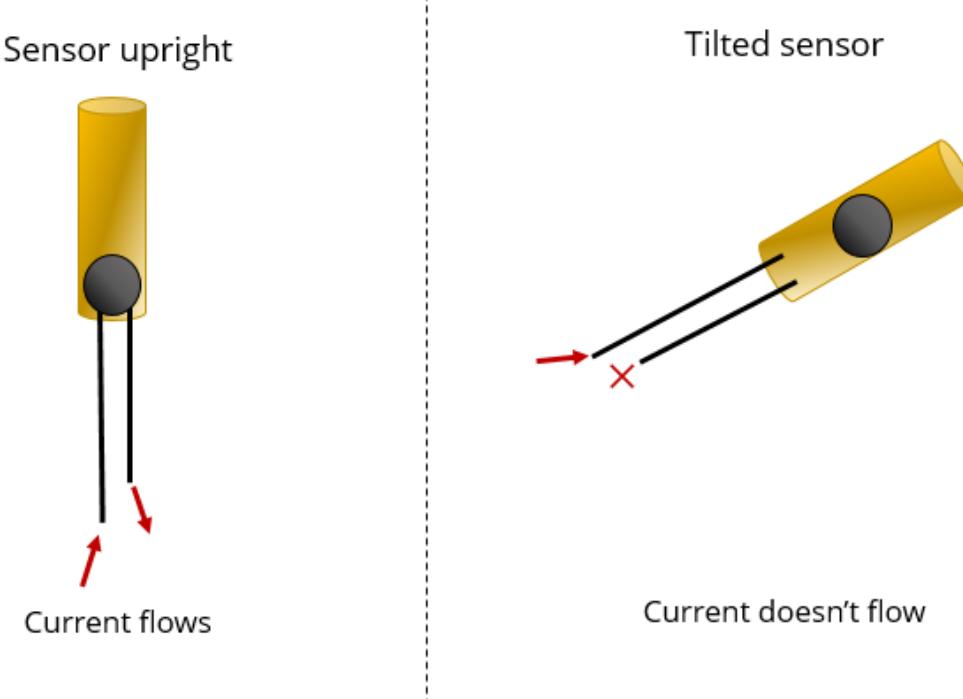


El sensor de inclinación permite detectar la orientación o la inclinación. Detecta si el sensor está completamente en posición vertical o si está inclinada.

Esto hace que sea muy útil para ser utilizado, por ejemplo, en los juguetes, robots y otros aparatos cuya metodología de trabajo depende de la inclinación.

¿Cómo funciona?

El sensor de inclinación es cilíndrica y contiene una bola rodante conductora libre en el interior con dos elementos conductores (polos) debajo.



Así es como funciona:

- Cuando el sensor está completamente en posición vertical, la bola cae a la parte inferior del sensor y conecta los polos, lo que permite que la corriente fluya.
- Cuando se inclina el sensor, la pelota no toque los polos, el circuito está abierto, y la corriente no fluye.

De esta manera, el sensor de inclinación actúa como un interruptor que se enciende o se apaga en función de su inclinación. Por lo tanto, se le dará la información digital a la placa Arduino, ya sea alto o una señal baja.

¿Donde comprar?

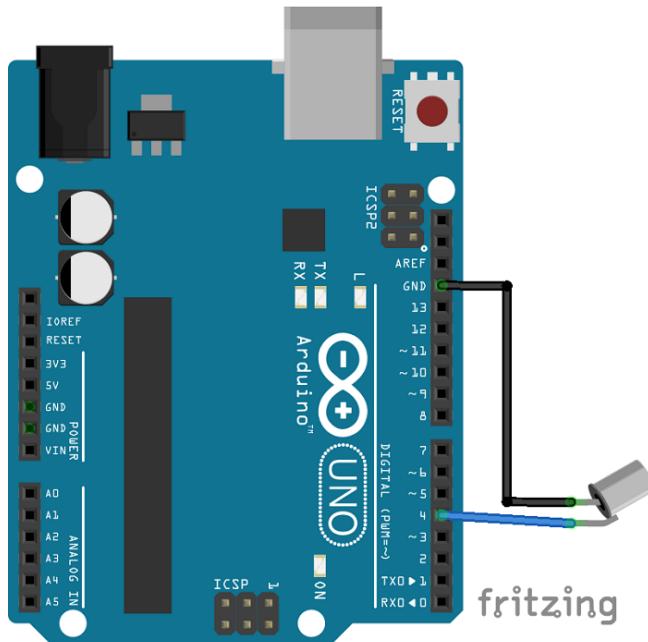
Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Tilt Sensor](#)

Cableado

Cableado del sensor de inclinación para que Arduino es sencillo. Sólo tiene que conectar una clavija a los pines digitales de Arduino y el GND a GND.

Si se conecta el sensor como tal, es necesario activar la resistencia pull-up interna de Arduino para el pin digital a la que el sensor está conectado. De lo contrario, se debe utilizar un 10kOhm tire hacia arriba de la resistencia en el circuito.



Ejemplo: LED sensible a la inclinación

Esto es sólo un ejemplo sencillo para empezar a poner las manos en el sensor de inclinación. En este ejemplo, un LED se apagará si el sensor está en posición vertical y se encenderá si se inclina el sensor.

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	Sensor de inclinación	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	2 x LEDs	Find best price on Maker Advisor
	220Ω Resistor	Find best price on Maker Advisor

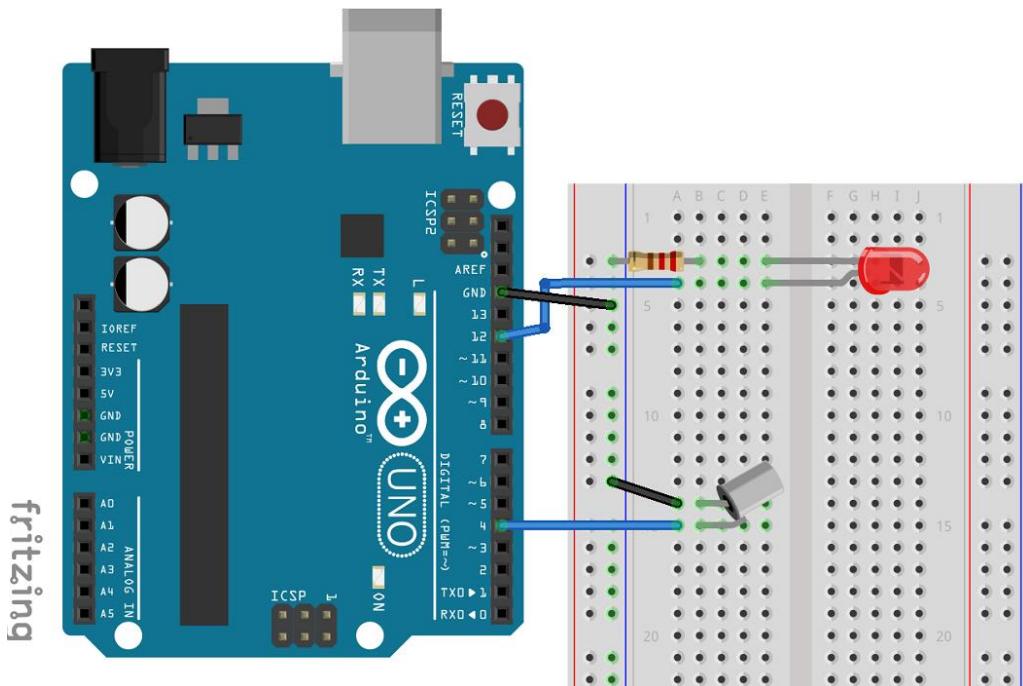


Jumper Wires

[Find best price on
Maker Advisor](#)

Círculo

Para este ejemplo, sólo tiene que añadir un LED para los esquemas en la sección "Pin de conexión".



Código

Para completar este ejemplo, subir el siguiente código.

```
/*
 * Rui Santos
 * Complete Project Details http://randomnerdtutorials.com
 */
int ledPin = 12;
int sensorPin = 4;
int sensorValue;
int lastTiltState = HIGH; // the previous reading from the tilt sensor

// the following variables are long's because the time, measured in miliseconds,
// will quickly become a bigger number than can be stored in an int.
long lastDebounceTime = 0; // the last time the output pin was toggled
long debounceDelay = 50; // the debounce time; increase if the output flickers

void setup() {
    pinMode(sensorPin, INPUT);
    digitalWrite(sensorPin, HIGH);
```

```

pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}

void loop() {
    sensorValue = digitalRead(sensorPin);
    // If the switch changed, due to noise or pressing:
    if (sensorValue == lastTiltState) {
        // reset the debouncing timer
        lastDebounceTime = millis();
    }
    if ((millis() - lastDebounceTime) > debounceDelay) {
        // whatever the reading is at, it's been there for longer
        // than the debounce delay, so take it as the actual current state:
        lastTiltState = sensorValue;
    }
    digitalWrite(ledPin, lastTiltState);
    Serial.println(sensorValue);
    delay(500);
}

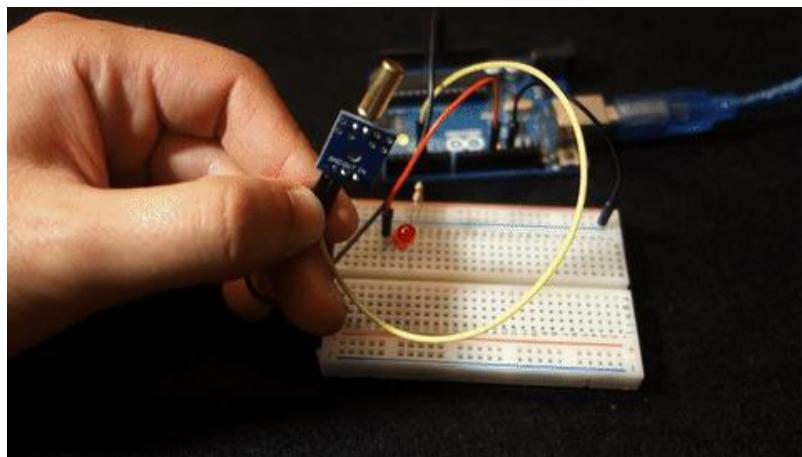
```

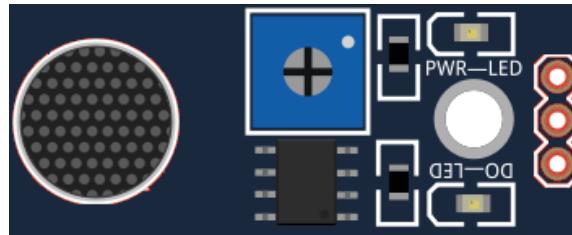
{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Arduino_tilt_sensor.ino

Demostración

Al final, esto es lo que tendrá.

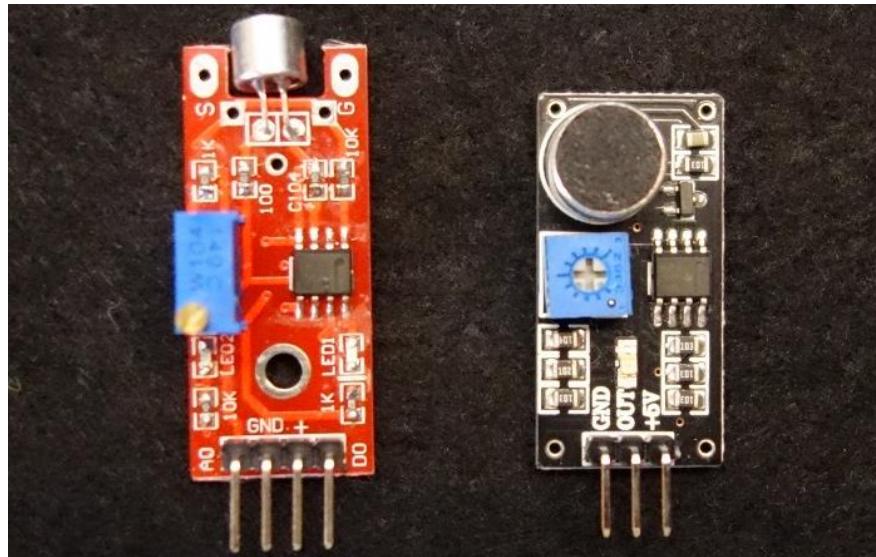




Sensor de sonido del micrófono

Esta guía muestra cómo utilizar el sensor de sonido del micrófono con el Arduino. El sensor de sonido del micrófono, como su nombre lo dice, detecta sonido. Se da una medida de qué tan fuerte es un sonido.

Hay una amplia variedad de estos sensores. En la figura siguiente se puede ver las más comunes que se utilizan con el Arduino.



A la izquierda se puede ver la KY-038 y en la derecha los sensores de sonido del micrófono LM 393 . Ambos módulos de sensor tienen un potenciómetro incorporado para ajustar la sensibilidad de los pines de salida digitales.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Microphone sound sensor KY-038](#)

Cableado

Sensor Pin	Wiring to Arduino Uno
A0	Analog pins
D0	Digital pins
GND	GND
VCC	5V

Si está utilizando el módulo LM 393, debería conectar la clavija OUT a un pin digital Arduino.

Example: Luces sensibles al sonido

En este ejemplo, un sensor de micrófono detecta la intensidad del sonido de su entorno y se iluminará y el LED si la intensidad del sonido es superior a un cierto umbral.

Puede modificar este proyecto mediante la adición de más LEDs. También puede ajustar la sensibilidad del sensor de forma que los LEDs están encendidos o apagados en consecuencia al ritmo de una música.

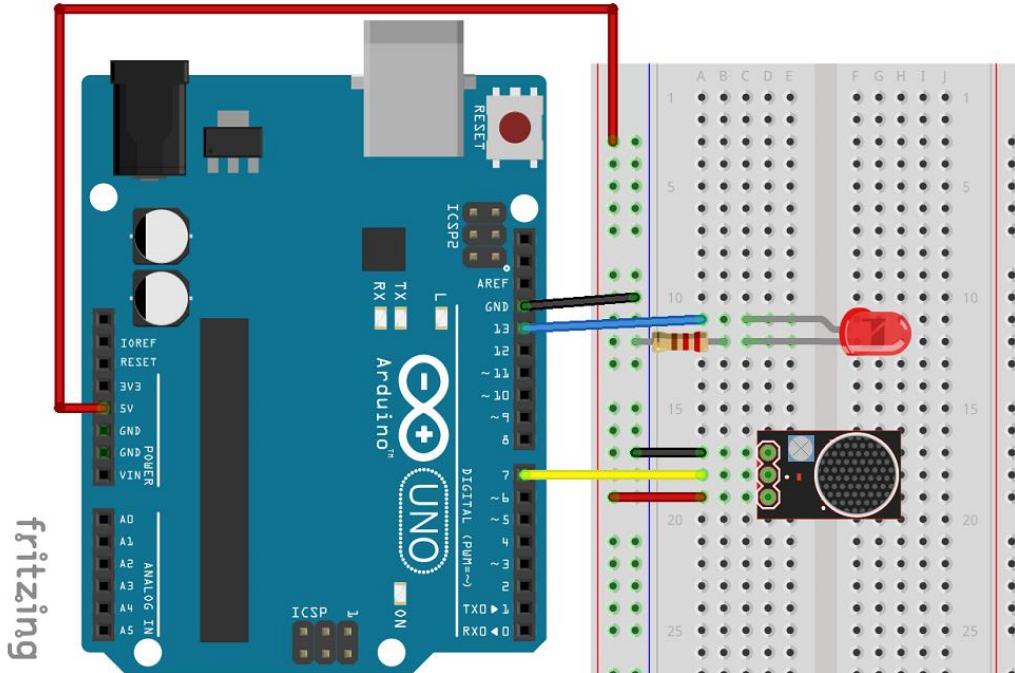
Para este ejemplo, necesitará los siguientes componentes.

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	Microphone sound sensor	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	LED	Find best price on Maker Advisor
	220Ω Resistor	Find best price on Maker Advisor



Circuito

Montar todas las piezas siguiendo el esquema siguiente:



Código

Sube el siguiente código a tu Arduino.

```
/*
 * Rui Santos
 * Complete Project Details http://randomnerdtutorials.com
*/
```

```
int ledPin=13;  
int sensorPin=7;  
boolean val =0;  
  
void setup(){  
    pinMode(ledPin, OUTPUT);  
    pinMode(sensorPin, INPUT);  
    Serial.begin (9600);  
}  
  
void loop (){
```

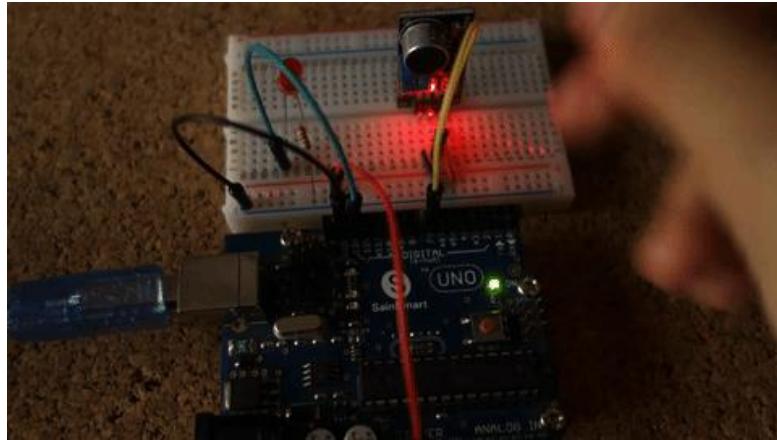
```
val =digitalRead(sensorPin);
Serial.println (val);
// when the sensor detects a signal above the threshold value, LED flashes
if (val==HIGH) {
  digitalWrite(ledPin, HIGH);
}
else {
  digitalWrite(ledPin, LOW);
}
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Arduino_microphone_sensor.ino

Demostración

Después de cargar el código, puede aplaudir al lado del sensor. Si el LED no se enciende, es necesario cambiar la sensibilidad del sensor girando el potenciómetro.



También puede ajustar la sensibilidad para que el LED siga el ritmo de una música determinada.

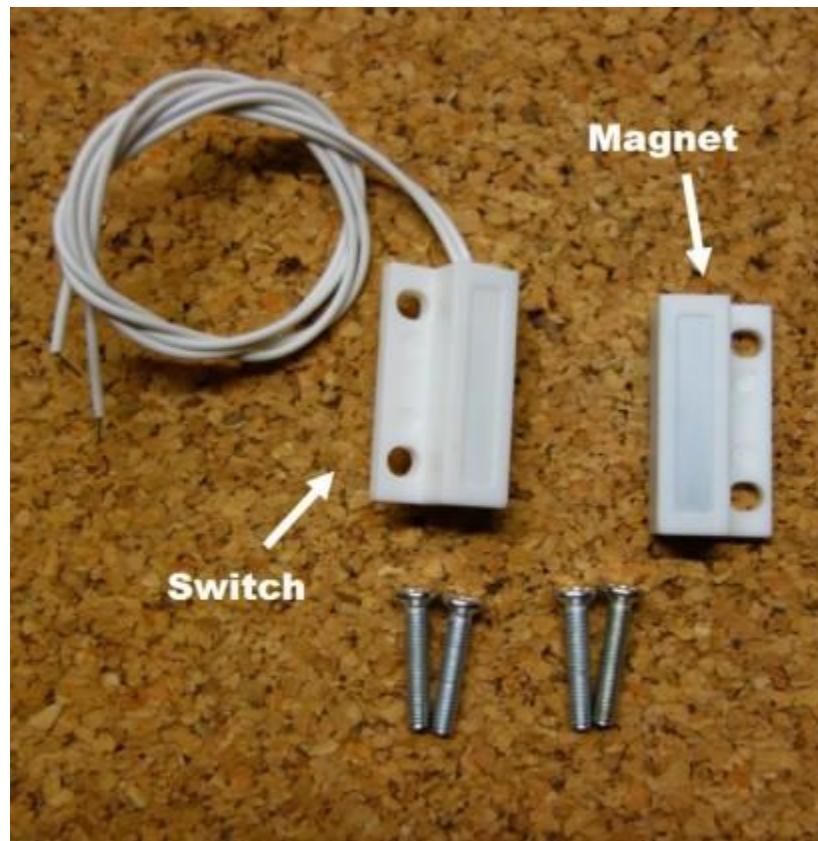
Añadir más LEDs para un efecto más espectacular.



Reed Switch

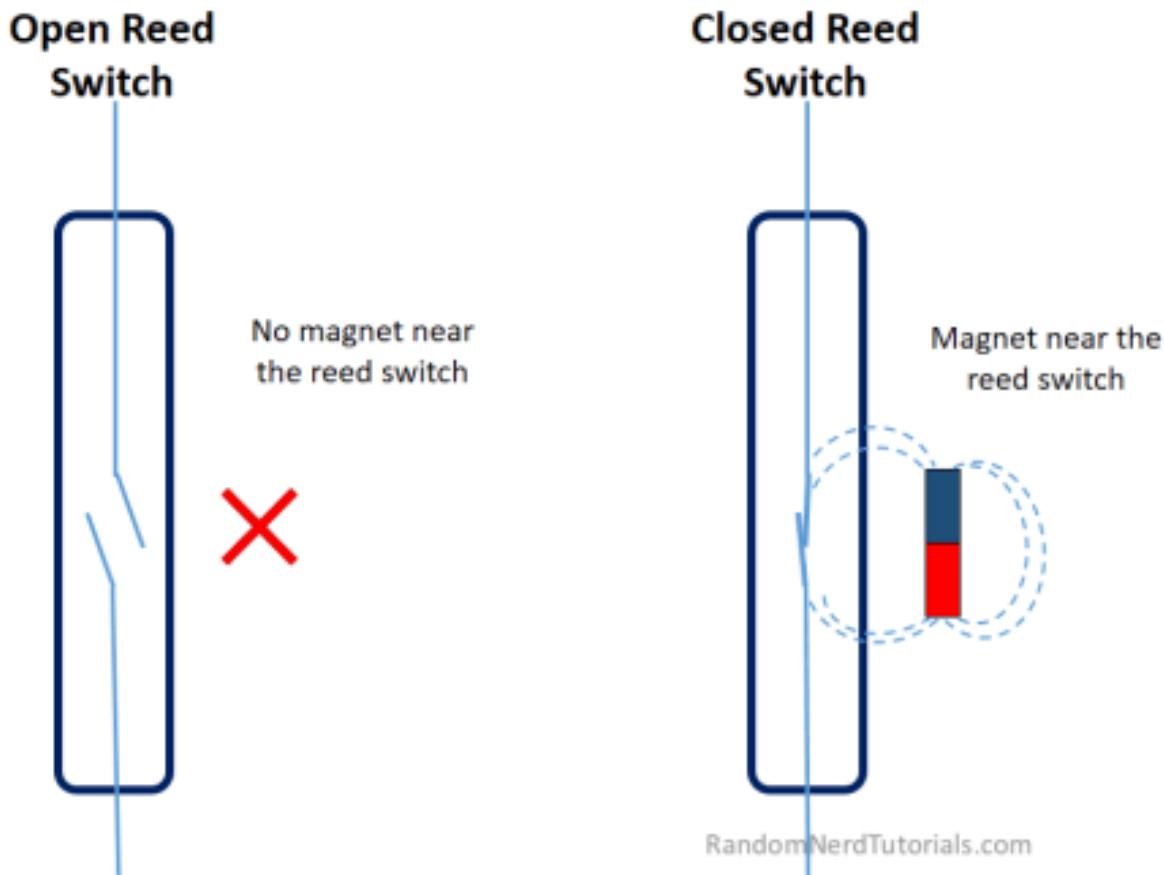
Un interruptor de contacto magnético es básicamente un interruptor de láminas encerrado en una carcasa de plástico de modo que se puede aplicar fácilmente en una puerta , una ventana o un cajón para detectar si la puerta está abierta o cerrada.

El interruptor que vamos a utilizar tiene dos partes: el propio interruptor, que suele venir abierto y el imán. Al comprar este interruptor, que también viene con 4 tornillos , por lo que puede adjuntar a su puerta.



¿Como funciona?

Es muy sencillo . El circuito eléctrico está cerrado cuando un imán se encuentra cerca del interruptor (menos de 13 mm (0,5") de distancia). Cuando el imán está lejos del interruptor, el circuito está abierto. Véase la siguiente figura.



¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Magnetic reed switch](#)

Ejemplo de proyecto

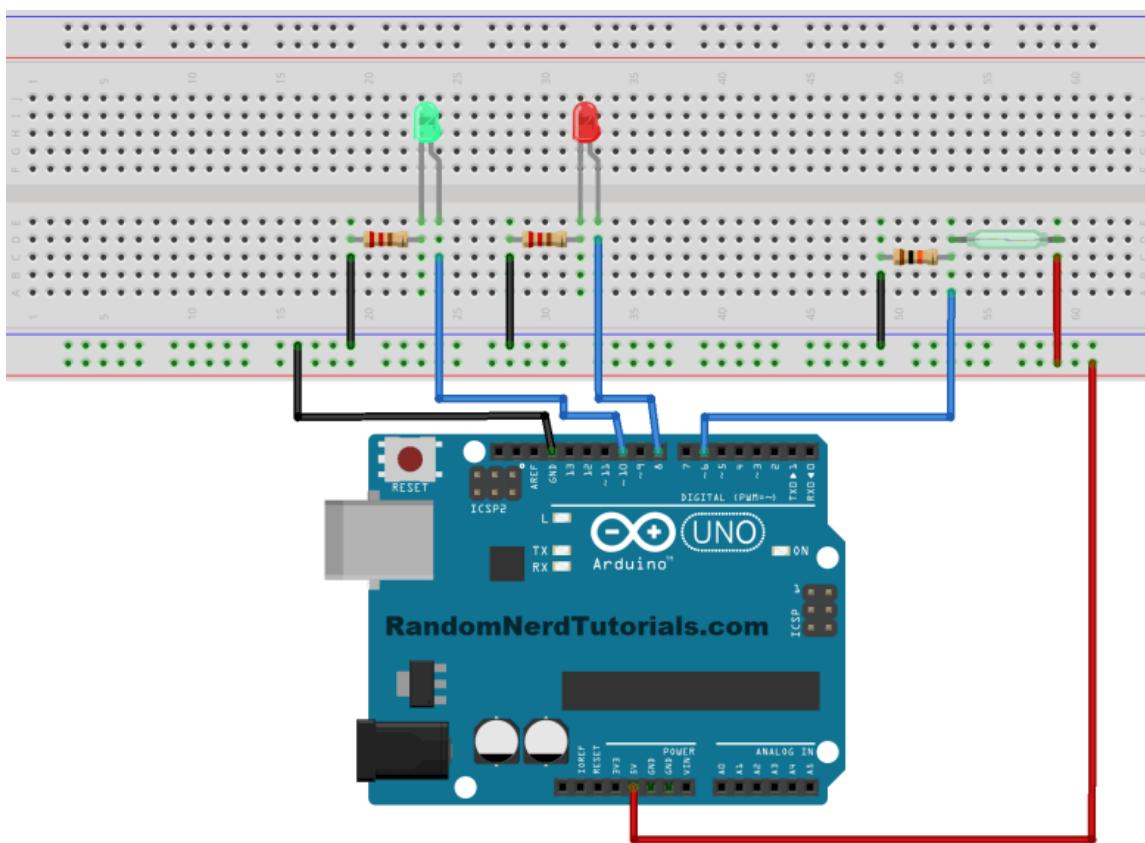
En este ejemplo, vamos a encender un LED rojo si la puerta está abierta y un LED verde si la puerta está cerrada.

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	Magnetic reed switch	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor

	LED	Find best price on Maker Advisor
	2x 220Ω Resistor, 1x 10kΩ Resistor	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Circuito

Siga el siguiente diagrama esquemático para completar este proyecto.



Código

Para este ejemplo, subir el siguiente código:

```
/*
Created by Rui Santos
All the resources for this project:
http://randomnerdtutorials.com/
*/
int ledOpen=8;
```

```

int ledClose=10;
int switchReed=6;

void setup() {
  pinMode(ledOpen, OUTPUT);
  pinMode(ledClose, OUTPUT);
  pinMode(switchReed, INPUT);
  Serial.begin(9600);
}

void loop() {

  if (digitalRead(switchReed)==HIGH) {
    digitalWrite(ledOpen, LOW);
    digitalWrite(ledClose, HIGH);
    Serial.println("Your Door is Closed");
  }
  else {
    digitalWrite(ledOpen, HIGH);
    digitalWrite(ledClose, LOW);
    Serial.println("Your Door is Open");
  }
  delay(1);
}

```

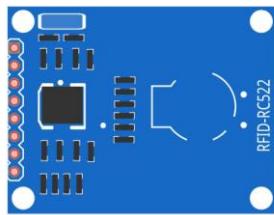
{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Magnetic_Contact_Switch.ino

Demostración

Usted puede ver este vídeo de demostración para ver el proyecto en acción:

<https://youtu.be/yY0DK8fe04w>



RFID MRFC522

RFID medios de identificación de radio-frecuencia . RFID utiliza campos electromagnéticos para transferir datos a través de distancias cortas. RFID es útil para identificar a las personas, para hacer transacciones, etc.

Se puede utilizar un sistema RFID para abrir una puerta . Por ejemplo , sólo la persona con la información correcta en su tarjeta puede entrar. Un sistema RFID utiliza:

- etiquetas unido al objeto a identificar , en este ejemplo tenemos un llavero y una tarjeta electromagnética. Cada etiqueta tiene su propia identificación única (UID).



- Radio de dos vías transmisor-receptor, el lector , que envían una señal a la etiqueta y leer su respuesta.



Especificaciones

- Voltaje de entrada: 3.3V
- Frecuencia: 13,56 MHz

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [MRFC522 RFID](#)

Cableado

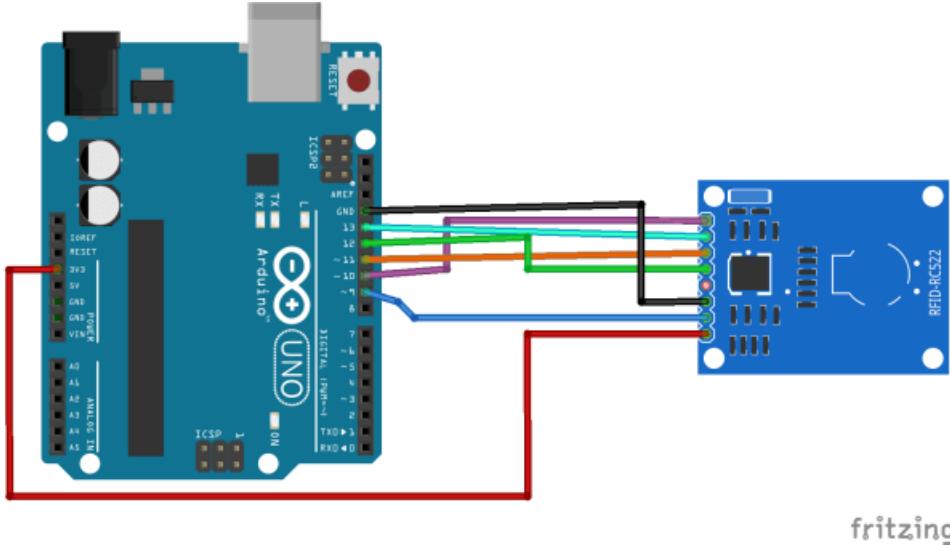
Sensor Pin	Wiring to Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	Don't connect
GND	GND
RST	Digital 9
3.3V	3.3V

Leer datos de una etiqueta RFID

En este ejemplo va a leer datos de una etiqueta RFID. Va a autorizar una etiqueta predeterminada y negar a los demás.

Circuito

Cablear el módulo lector RFID como en el esquema siguiente.



fritzing

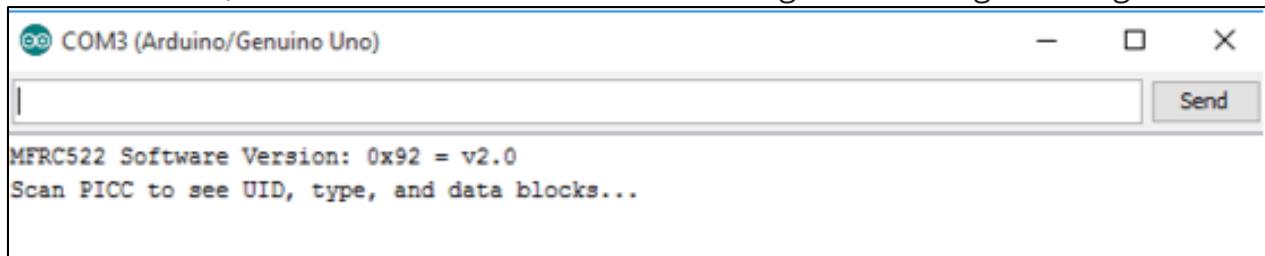
Descarga de la biblioteca

Es necesario la biblioteca RFID para este proyecto:

1. Haga clic aquí para descargar el [RFID library here](#) creado por miguelbalboa. Tú debe tener una carpeta .zip en la carpeta de Descargas.
2. Descomprimir la carpeta .zip y usted debe obtener la carpeta RFID-maestro.
3. Cambiar el nombre de la carpeta de RFID-maestro para RFID.
4. Mover la carpeta de RFID a la carpeta de bibliotecas de instalación Arduino IDE.
5. Por último, vuelva a abrir el IDE de Arduino.
6. Reiniciar el IDE Arduino

Ir Archivo • Ejemplos • MFRC522 • DumpInfo y cargar el código. Este código estará disponible en el IDE de Arduino (después de instalar la biblioteca RFID).

A continuación, abra el monitor serie. Debería ver algo como la siguiente figura:



Aproximar la tarjeta RFID o el llavero para el lector. Dejar que el lector y la etiqueta más cerca hasta que se muestre toda la información

```

COM3 (Arduino/Genuino Uno)

MFRC522 Software Version: 0x92 = v2.0
Scan PICC to see UID, type, and data blocks...
Card UID: BD 31 15 2B
PICC type: MIFARE 1KB

Sector Block  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 AccessBits
  15    63  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          62  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          61  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  14    59  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          58  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          57  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          56  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  13    55  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          54  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          53  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          52  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  12    51  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          49  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          48  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  11    47  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          46  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          45  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          44  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  10    43  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          42  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          41  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
   9     39  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF FF [ 0 0 1 ]
          38  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 [ 0 0 0 1 ]

```

Esta es la información que se puede leer desde la tarjeta, incluyendo el UID tarjeta que está resaltado en amarillo.

La información se almacena en la memoria que se divide en segmentos y bloques como se puede ver en la imagen anterior . Usted tiene 1024 bytes de almacenamiento de datos divididos en 16 sectores.

Anote su tarjeta de UID porque se le necesite más adelante.

Entonces, es necesario el siguiente código (no subirlo ahora).

```
/*
 *
 * All the resources for this project: http://randomnerdtutorials.com/
 * Modified by Rui Santos
 *
 * Created by FILIPEFLOP
 *
 */

#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

void setup()
{
    Serial.begin(9600); // Initiate a serial communication
    SPI.begin(); // Initiate SPI bus
    mfrc522.PCD_Init(); // Initiate MFRC522
    Serial.println("Approximate your card to the reader...");
    Serial.println();

}

void loop()
{
    // Look for new cards
    if ( ! mfrc522.PICC_IsNewCardPresent() )
    {
        return;
    }
    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial() )
    {
        return;
    }
    //Show UID on serial monitor
    Serial.print("UID tag :");
    String content= "";
    byte letter;
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
    Serial.println();
    Serial.print("Message : ");
    content.toUpperCase();
}
```

```

if (content.substring(1) == "BD 31 15 2B") //change here the UID of the card/cards
that you want to give access
{
    Serial.println("Authorized access");
    Serial.println();
    delay(3000);
}

else {
    Serial.println(" Access denied");
    delay(3000);
}

```

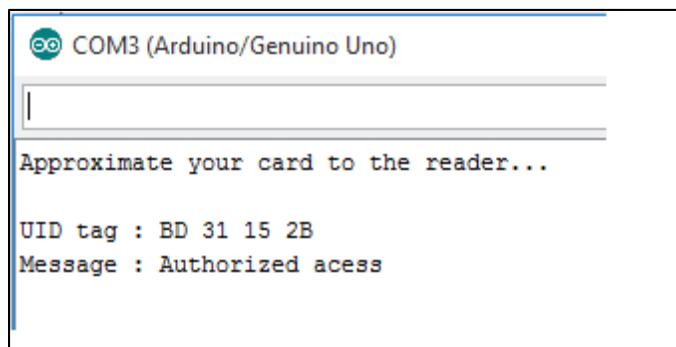
{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/MFRC522_RFID_Reader_with_Arduino.ino

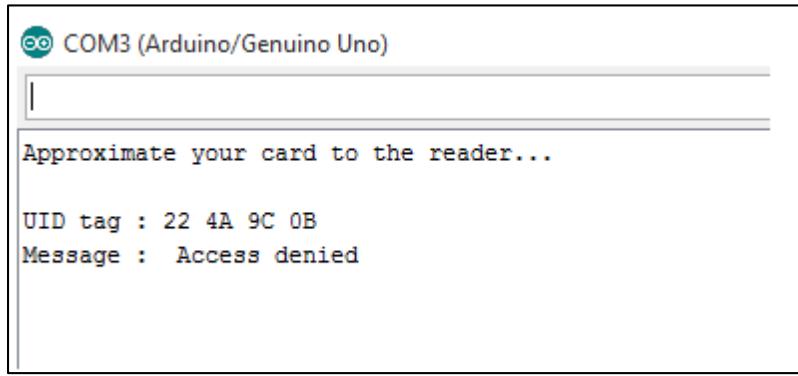
En el fragmento de código anterior es necesario cambiar la línea si (content.substring (1) == "sustituir con su UID") y escriba la tarjeta UID que has escrito anteriormente.

Demostración

Ahora, subir el código a su Arduino y abrir el monitor serie. Aproximar la tarjeta que ha elegido para dar acceso y verá:



Si aproximar otra etiqueta con otro UID, el mensaje de denegación se mostrará:



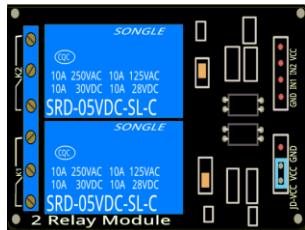
The screenshot shows a terminal window with the title "COM3 (Arduino/Genuino Uno)". The window displays the following text:
|
Approximate your card to the reader...
UID tag : 22 4A 9C 0B
Message : Access denied

Usted puede ver el siguiente vídeo de demostración para ver el proyecto en acción:

<https://youtu.be/mpLzcBDBI1U>

Terminando

El lector RFID MFRC522 le permite hacer un proyecto en el que tienes acceso de seguridad a algo. Por ejemplo , sólo la etiqueta correcta puede abrir una puerta específica . Esto es sólo una guía sencilla para mostrar cómo funciona este lector. Ahora , la idea es para que usted pueda aplicar este conocimiento y utiliza su imaginación para hacer sus propios proyectos.



Módulo de relé

Un relé es un interruptor accionado eléctricamente de la tensión de red. Se puede activar o desactivar , dejando ir la corriente a través o no. El módulo de relé se muestra en la figura siguiente.



Este módulo tiene dos canales (los cubos de color azul).

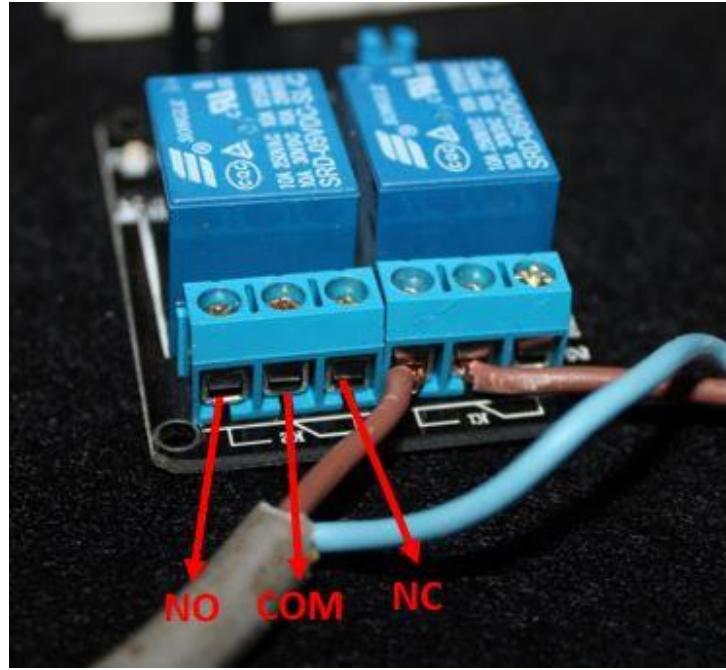
¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Relay module](#)

Tensión de red de conexiones

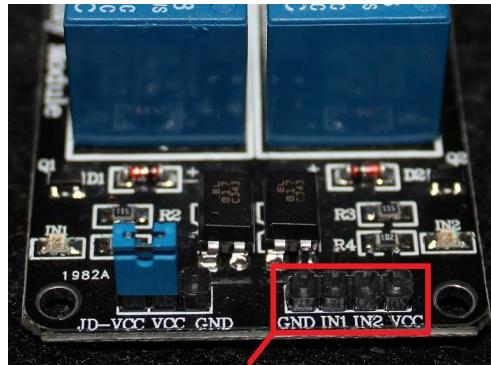
En relación con la tensión de la red, los relés tienen 3 conexiones posibles:



- **COM:** pin común
- **NO:** normalmente abierto - no hay contacto entre el pasador y el pasador común normalmente abierto. Por lo tanto, cuando se dispara el relé, se conecta a la clavija COM y se proporciona alimentación a la carga (una lámpara de escritorio, en nuestro caso).
- **NC:** normalmente cerrado - hay contacto entre el pasador común y el pasador normalmente cerrada. Siempre hay conexión entre el COM y NC pernos, incluso cuando el relé está desactivado. Cuando se dispara el relé, el circuito está abierto y no hay alimentación suministrada a la carga.

Si desea controlar una lámpara, por ejemplo, es mejor utilizar un circuito normalmente abierto, porque sólo queremos encender el indicador de vez en cuando.

Cableado



- GND: va a Tierra
- IN1: controla el primer relé. En caso de ser conectado a un pin digital Arduino
- IN2: controla el segundo relé. En caso de ser conectado a un pin digital Arduino si está utilizando este segundo relé. De lo contrario, no es necesario conectarlo.
- VCC: va a 5V.

Control de una lámpara con un módulo de relé y el sensor de movimiento PIR

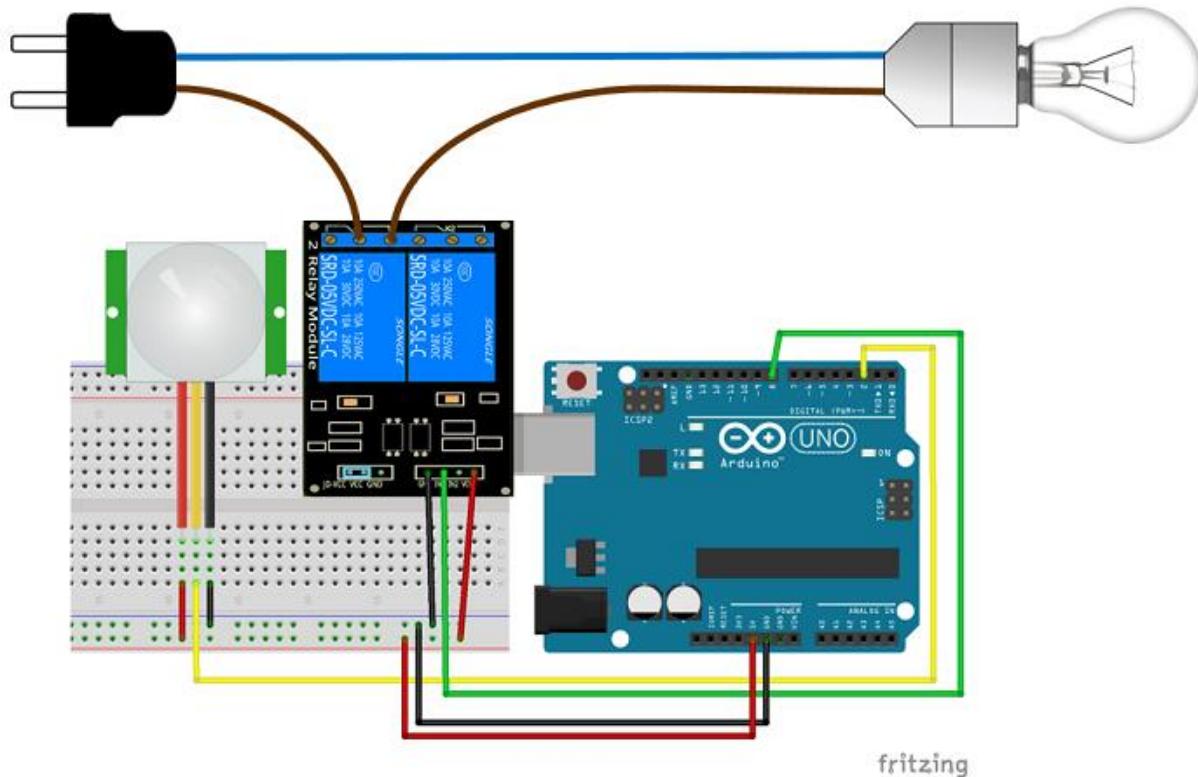
En este ejemplo se creará una lámpara sensible al movimiento . Una lámpara se encenderá durante 10 segundos se detecta cada movimiento tiempo . El movimiento se detecta mediante un sensor de movimiento PIR.

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	Relay Module	Find best price on Maker Advisor
	Lamp Cord Set	-
	PIR Motion Sensor	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Circuito

Montar todas las piezas siguiendo el diagrama esquemático siguiente.

Tenga mucho cuidado con las conexiones de tensión de red. Asegúrate de que tienes el relé desconectado de la tensión de red durante el montaje del circuito.



Código

Copia el siguiente código a su IDE Arduino y subirlo a tu placa Arduino.

```

/*
Rui Santos
Complete project details at http://randomnerdtutorials.com
*/
// Relay pin is connected to D8. The active wire is connected to Normally open and
common
int relay = 8;
volatile byte relayState = LOW;

// PIR Motion Sensor is connected to D2.
int PIRInterrupt = 2;

// Timer Variables
long lastDebounceTime = 0;
long debounceDelay = 10000;

void setup() {
// Pin for relay module set as output
pinMode(relay, OUTPUT);
digitalWrite(relay, HIGH);
// PIR motion sensor set as an input
pinMode(PIRInterrupt, INPUT);
}

```

```

// Triggers detectMotion function on rising mode to turn the relay on, if the
condition is met
attachInterrupt(digitalPinToInterruption(PIRInterrupt), detectMotion, RISING);
// Serial communication for debugging purposes
Serial.begin(9600);
}

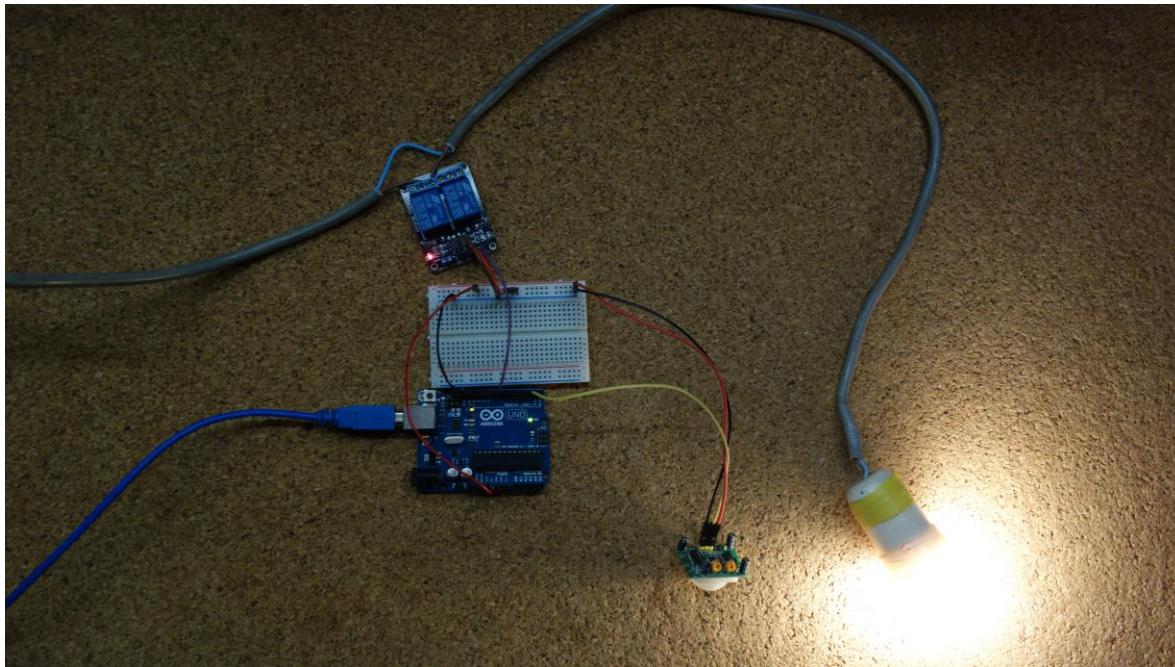
void loop() {
    // If 10 seconds have passed, the relay is turned off
    if((millis() - lastDebounceTime) > debounceDelay && relayState == HIGH) {
        digitalWrite(relay, HIGH);
        relayState = LOW;
        Serial.println("OFF");
    }
    delay(50);
}

void detectMotion() {
    Serial.println("Motion");
    if(relayState == LOW) {
        digitalWrite(relay, LOW);
    }
    relayState = HIGH;
    Serial.println("ON");
    lastDebounceTime = millis();
}

```

Demostración

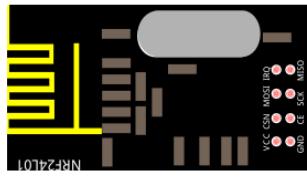
Ahora, cuando el movimiento se detecta su lámpara se enciende.



Terminando

Control de un módulo de relé con el Arduino es tan simple como controlar una salida.

Con el módulo de relé se puede controlar casi cualquier aparato de instalaciones eléctricas (no sólo las lámparas).



nRF24L01

Estos módulos de RF son muy populares entre los manitas Arduino. El nRF24L01 se utiliza en una amplia variedad de aplicaciones que requieren un control inalámbrico . Ellos son transceptores que significa que cada módulo puede transmitir y recibir datos.

Estos módulos son muy baratos y que ellos pueden usar con cualquier microcontrolador (MCU).



Especificaciones nRF24L01 - 2,4 GH transmisor-receptor de RF

- Bajo coste de un solo chip de 2,4 GHz IC transceptor RF GFSK
- Alcance con Antena: tasa de 250Kb (zona abierta)> 1.000 metros
- Energía: consumo de energía ultra bajo
- Voltaje de entrada: 3.3V
- Pins: tolerantes 5V

¿Donde comprar?

Haga clic en el enlace de abajo para el mejor precio :

- [nRF24L01/2.4GHz RF](#)

Arduino con nRF24L01

Necesita los componentes siguientes para completar las instrucciones de este ejemplo:

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	2x nRF24L01	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Biblioteca

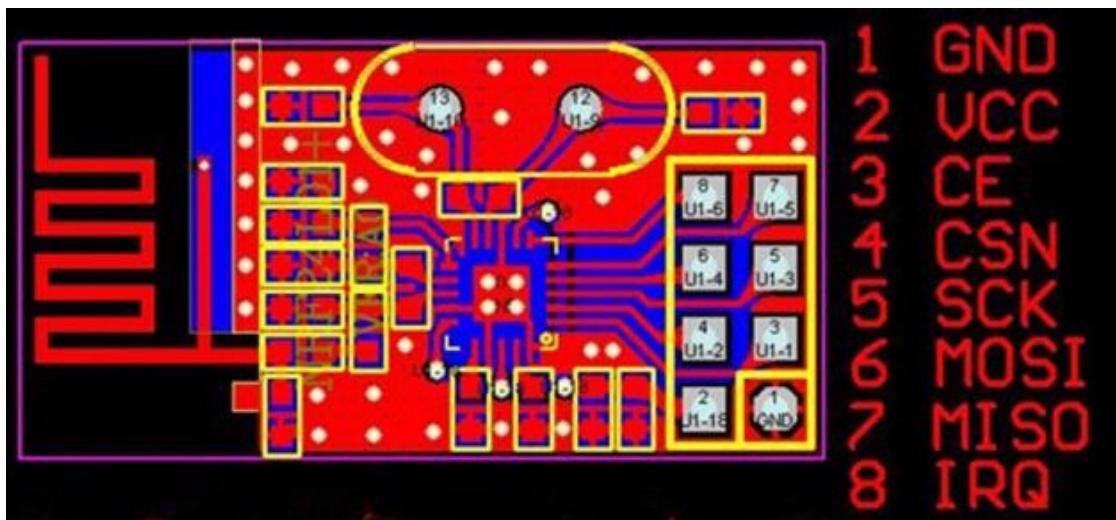
Para este proyecto se necesita la biblioteca de Radiohead.

1. Click here to download the [RadioHead library](#)
2. Descomprimir la carpeta .zip y usted debe obtener la carpeta RadioHead-1.46.
3. Cambiar el nombre de la carpeta de RadioHead-1,46 a Radiohead.
4. Mover la carpeta RadioHead a su carpeta de bibliotecas de instalación Arduino IDE.
5. Por último, vuelva a abrir el IDE de Arduino.

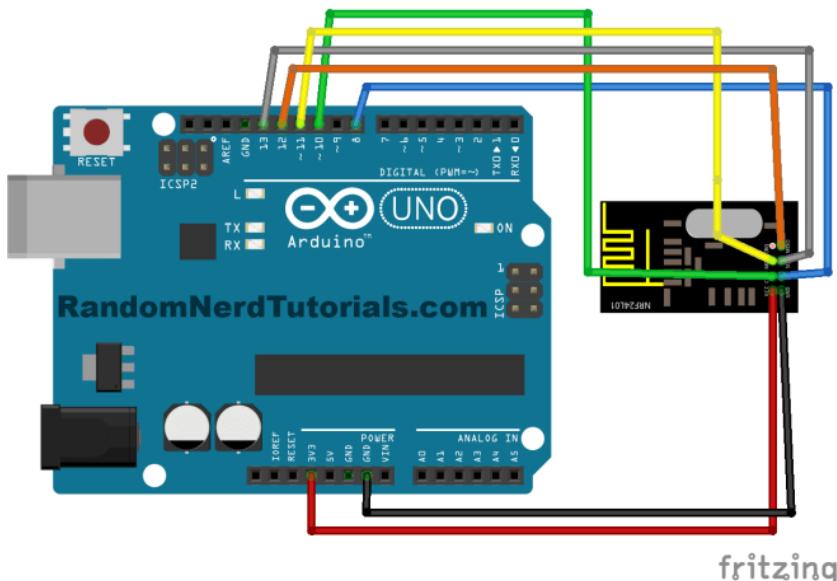
La biblioteca RadioHead es grande y funciona con casi todos los módulos de RF en el mercado. Puede leer más [sobre este proyecto here](#).

Pinout

Esta es la vista superior de NRF24L01



Círculo de cliente



ADVERTENCIA

voltaje de entrada es de 1.9V ~ 3.6V, no debe superarse dicho valor, de lo contrario se fríen su módulo.

Sigue el circuito anterior para su cliente. A continuación, cargar el código debajo de la cual se pueden encontrar en el IDE de Arduino (después de instalar la biblioteca RadioHead).

[Ir Archivo • Ejemplos • RadioHead • nrf24 • nrf24 client.](#)

```
// nrf24_client
#include <SPI.h>
#include <RH_NRF24.h>

// Singleton instance of the radio driver
RH_NRF24 nrf24;
// RH_NRF24 nrf24(8, 7); // use this to be electrically compatible with Mirf
// RH_NRF24 nrf24(8, 10); // For Leonardo, need explicit SS pin
// RH_NRF24 nrf24(8, 7); // For RFM73 on Anarduino Mini

void setup()
{
    Serial.begin(9600);
    while (!Serial)
        ; // wait for serial port to connect. Needed for Leonardo only
    if (!nrf24.init())
        Serial.println("init failed");
    // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
    if (!nrf24.setChannel(1))
        Serial.println("setChannel failed");
    if (!nrf24.setRF(RH_NRF24::DataRate2Mbps, RH_NRF24::TransmitPower0dBm))
        Serial.println("setRF failed");
}

void loop()
{
    Serial.println("Sending to nrf24_server");
    // Send a message to nrf24_server
    uint8_t data[] = "Hello World!";
    nrf24.send(data, sizeof(data));

    nrf24.waitPacketSent();
    // Now wait for a reply
    uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);

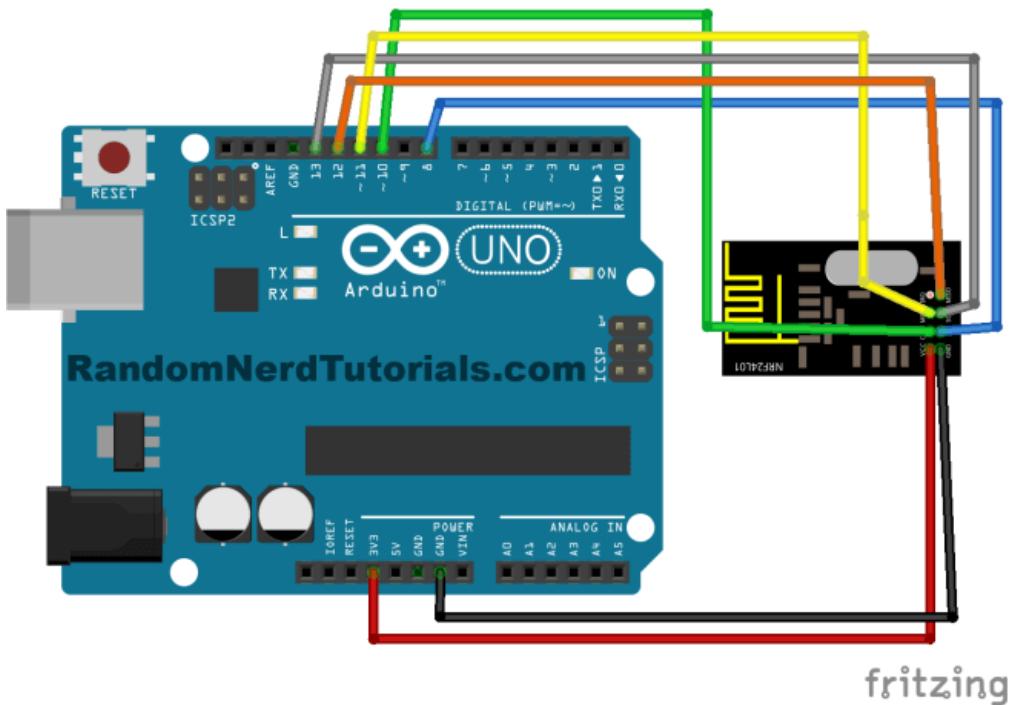
    if (nrf24.waitAvailableTimeout(500))
    {
        // Should be a reply message for us now
        if (nrf24.recv(buf, &len))
        {
            Serial.print("got reply: ");
            Serial.println((char*)buf);
        }
    }
}
```

```
        {
            Serial.println("recv failed");
        }
    }
else
{
    Serial.println("No reply, is nrf24_server running?");
}
delay(400);
}
```

{;} ADVERTENCIA

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/nrf24_client.ino

Círculo de servidor



ADVERTENCIA

voltaje de entrada es de 1.9V ~ 3.6V, no debe superarse dicho valor, de lo contrario se fríen su módulo.

Sigue el circuito anterior para su servidor. A continuación, cargar el código debajo de la cual se pueden encontrar en el IDE de Arduino (después de instalar la biblioteca RadioHead).

Ir Archivo • Ejemplos • RadioHead • nrf24 • nrf24_server.

```
// nrf24_server

#include <SPI.h>
#include <RH_NRF24.h>

// Singleton instance of the radio driver
RH_NRF24 nrf24;
// RH_NRF24 nrf24(8, 7); // use this to be electrically compatible with Mirf
// RH_NRF24 nrf24(8, 10); // For Leonardo, need explicit SS pin
// RH_NRF24 nrf24(8, 7); // For RFM73 on Anarduino Mini

void setup()
{
    Serial.begin(9600);
    while (!Serial)
        ; // wait for serial port to connect. Needed for Leonardo only
    if (!nrf24.init())
        Serial.println("init failed");
    // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
    if (!nrf24.setChannel(1))
        Serial.println("setChannel failed");
    if (!nrf24.setRF(RH_NRF24::DataRate2Mbps, RH_NRF24::TransmitPower0dBm))
        Serial.println("setRF failed");
}

void loop()
{
    if (nrf24.available())
    {
        // Should be a message for us now
        uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        if (nrf24.recv(buf, &len))
        {
            // NRF24::printBuffer("request: ", buf, len);
            Serial.print("got request: ");
            Serial.println((char*)buf);

            // Send a reply
            uint8_t data[] = "And hello back to you";
            nrf24.send(data, sizeof(data));
            nrf24.waitPacketSent();
            Serial.println("Sent a reply");
        }
    }
}
```

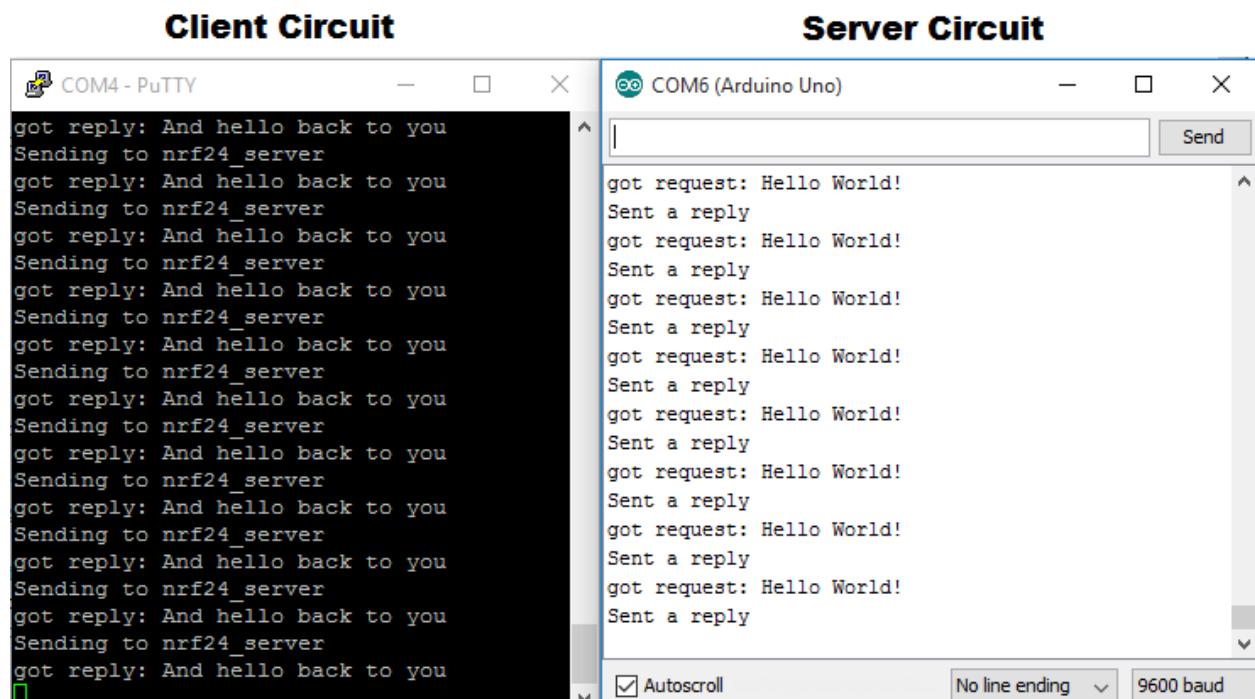
```
        }
    else
    {
        Serial.println("recv failed");
    }
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/nrf24_server.ino

Demostración

En este proyecto el cliente está enviando un mensaje "Hello World!" Al servidor a través de RF y el servidor envía de vuelta el siguiente mensaje "Y hola de nuevo a usted". Esos mensajes se visualizan en la pantalla de serie. Esto es lo que debería ver en el monitor de serie y ventanas de terminal (figura siguiente).

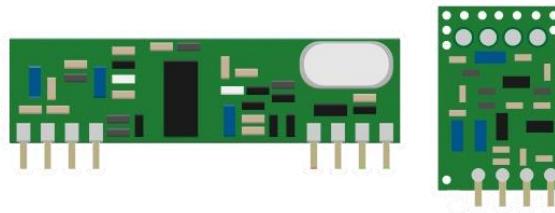


NOTA: En la ventana izquierda, estoy estableciendo una comunicación serie con PuTTY.org. En la ventana de la derecha, estoy usando el Monitor de serie Arduino IDE.

Terminando

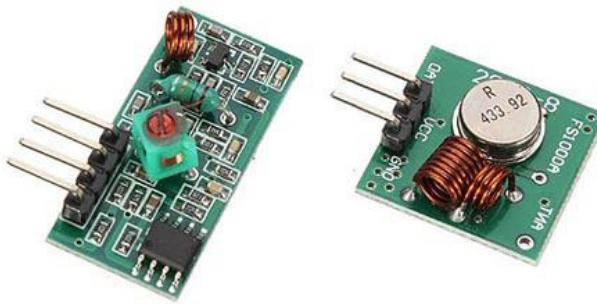
Es necesario tener algunas expectativas realistas cuando se utiliza este módulo. Ellos trabajan muy bien cuando el receptor y el transmisor están bastante cerca uno del otro. Si se separa demasiado lejos, si no se pierden la comunicación.

El alcance de comunicación puede variar. Depende de la cantidad de ruido en su entorno, si hay algún obstáculo y si está utilizando una antena externa.



433 MHz Transmisor / Receptor

Estos módulos de RF son muy populares entre los manitas Arduino. El 433 MHz se utiliza en una amplia variedad de aplicaciones que requieren un control inalámbrico.



Especificaciones del receptor de RF 433 MHz

- Rango de Frecuencia: 433.92 MHz
- Modulación: ASK
- Voltaje de entrada: 5V

Especificaciones 433MHz transmisor RF

- Rango de Frecuencia: 433,92 MHz
 - Voltaje de entrada: 3-12V
- Estos módulos son muy baratos, y usted puede utilizar con cualquier microcontrolador (MCU).

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [433 MHz Transmitter/Receiver](#)

Arduino con módulos de RF de 433 MHz

Este es un ejemplo sencillo para transmitir un mensaje a través de RF.

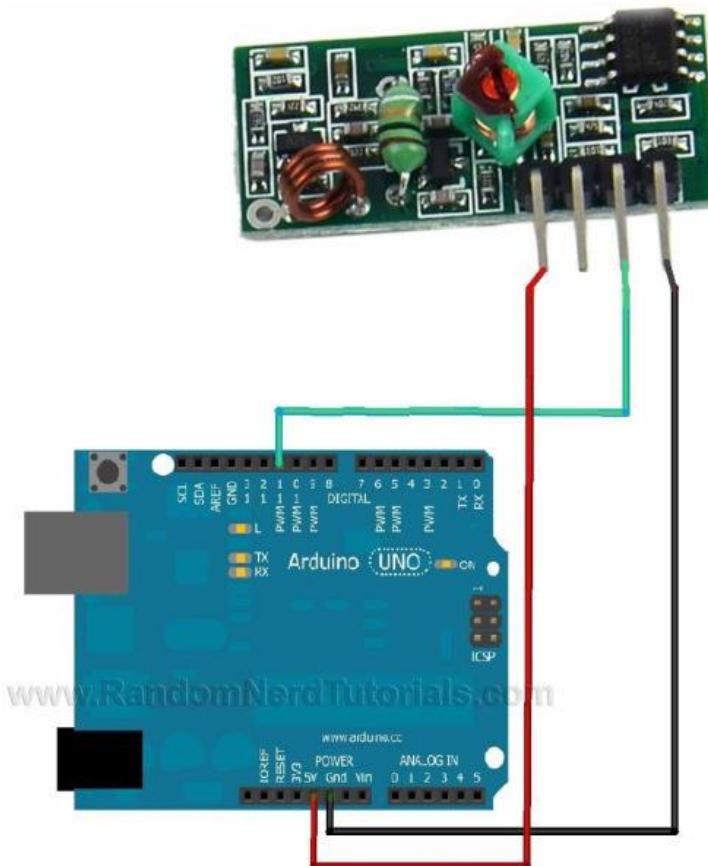
Descarga biblioteca

Aquí está la biblioteca que necesita para este proyecto:

1. Descargar el [RadioHead library](#)
2. Descomprimir la biblioteca RadioHead
3. Instalar la biblioteca RadioHead en su IDE Arduino
4. Reiniciar el IDE Arduino

TLa biblioteca RadioHead es grande y funciona con casi todos los módulos de RF en el mercado. Puede leer más [sobre este proyecto aquí](#).

Círcuito Receptor



Sigue el circuito anterior para su receptor. A continuación, cargar el código de abajo.

```
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile

RH_ASK driver;

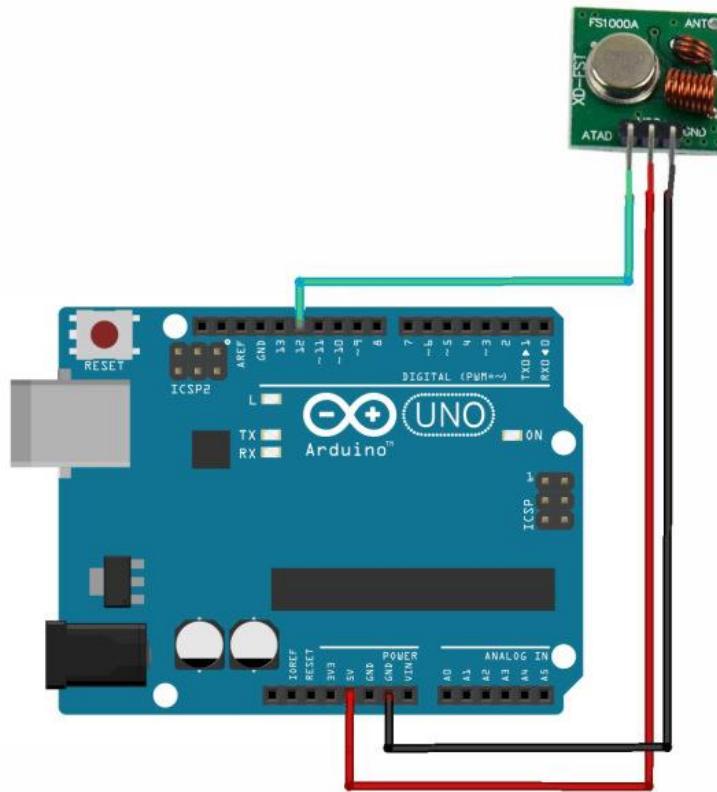
void setup()
{
    Serial.begin(9600);      // Debugging only
    if (!driver.init())
        Serial.println("init failed");
}

void loop()
{
    uint8_t buf[12];
    uint8_t buflen = sizeof(buf);
    if (driver.recv(buf, &buflen)) // Non-blocking
    {
        int i;
        // Message with a good checksum received, dump it.
        Serial.print("Message: ");
        Serial.println((char*)buf);
    }
}
```

{;} CÓDIGO FUENTE

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/433MHz/receiver.ino>

El Circuito Transmisor



Sigue el circuito anterior para su transmisor. A continuación, cargar el código de abajo.

```
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile

RH_ASK driver;

void setup()
{
    Serial.begin(9600);      // Debugging only
    if (!driver.init())
        Serial.println("init failed");
}

void loop()
{
    const char *msg = "Hello World!";
    driver.send((uint8_t *)msg, strlen(msg));
    driver.waitPacketSent();
    delay(1000);
}
```

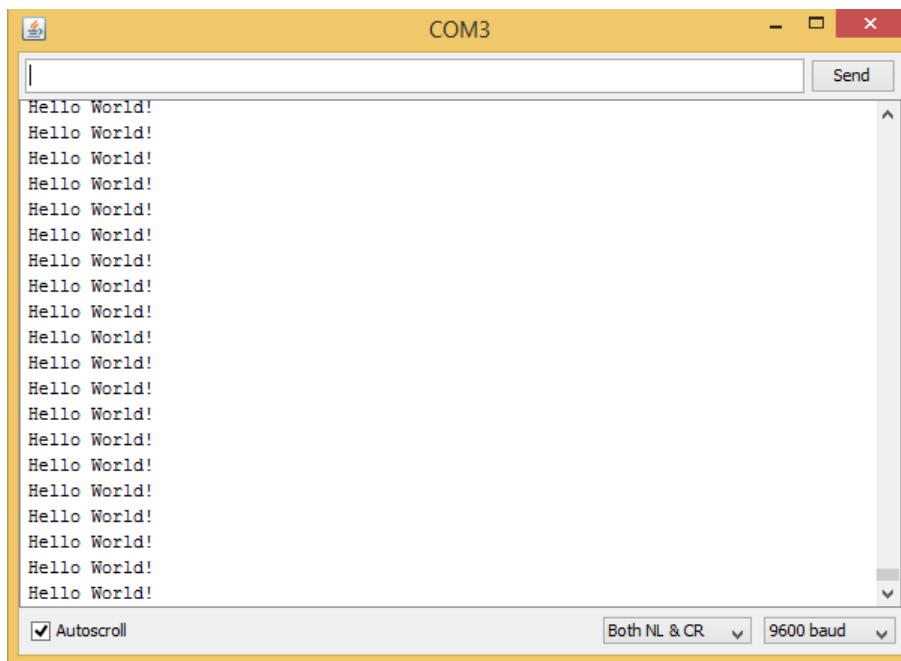


CÓDIGO FUENTE

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/433MHz/transmitter.ino>

Demostración

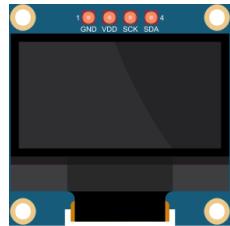
En este proyecto el transmisor está enviando un mensaje “Hello World!” Al receptor a través de radiofrecuencia. Esos mensajes se visualizan en la pantalla de serie del receptor. Esto es lo que debe ver en su monitor serie Arduino IDE.



Terminando

Es necesario tener algunas expectativas realistas cuando se utiliza este módulo. Ellos trabajan muy bien cuando el receptor y el transmisor están cerca uno del otro. Si se separa demasiado lejos si no se pierden la comunicación.

El alcance de comunicación puede variar. Depende de la cantidad de tensión que se está suministrando a su módulo de transmisión, el ruido de RF en su entorno y si está utilizando una antena externa.



Pantalla OLED

La pantalla OLED se muestra en la siguiente figura:



Es una muy pequeña pantalla hecha de 128 por 64 pixeles OLED individuales y no se requiere luz de fondo.

Esa pantalla OLED es monocromática (color blanco), pero hay otros modelos con varios colores.

Esta pantalla utiliza la comunicación I2C. Esto significa que se comunica con el Arduino usando sólo 2 pines.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [0.96 inch OLED display](#)

Cableado

OLED Pin	Wiring to Arduino Uno
Vin	5V
GND	GND
SCL	A5
SDA	A4

Si está utilizando otra placa Arduino en lugar de la Uno, echa un vistazo a lo que son sus pines SCL y SDA.

- Nano: SDA (A4); SCL(A5);
- MEGA: SDA (20); SCL(21);
- Leonardo: SDA (20); SCL(21);

Biblioteca

Para controlar la pantalla OLED que necesita la biblioteca “ **adafruit_GFX.h** ” y la biblioteca “adafruit_SSD1306.h ”

La instalación de la biblioteca adafruit_GFX

1. [CHaga click aquí para descargar la biblioteca GFX adafruit](#). Usted debe tener una carpeta .zip en la carpeta de Descargas
2. Descomprimir la carpeta .zip y usted debe obtener Adafruit-GFX-Biblioteca-master
3. Renombrar tu carpeta de Adafruit-GFX-Library-master a Adafruit_GFX_Library.
4. Mover la carpeta Adafruit_GFX_Library a su instalación Arduino IDE carpeta de bibliotecas
5. Por último, vuelva a abrir el IDE de Arduino

La instalación de la biblioteca adafruit_SSD1306

1. [Haga clic aquí para descargar la biblioteca adafruit_SSD1306](#) . Usted debe tener una carpeta .zip en la carpeta de Descargas
2. Descomprimir la carpeta .zip y usted debe obtener Adafruit-GFX-Biblioteca-master carpeta

3. Cambiar el nombre de la carpeta de Adafruit_SSD1306-master a Adafruit_SS1306
4. Mover la carpeta Adafruit_SS1306 a sus bibliotecas de instalación Arduino IDE
5. Por último, vuelva a abrir el IDE de Arduino

Consejos para la escritura de texto usando estas bibliotecas

He aquí algunas de las funciones que le ayudarán a manejar la biblioteca pantalla OLED para escribir texto o dibujar gráficos simples.

- `display.clearDisplay()` – todos los píxeles están fuera
- `display.drawPixel(x, y, color)` – trazar un píxel en la coordenadas x, y
- `display.setTextSize(n)` – fijar el tamaño de fuente, admite tamaños de 1 a 8
- `display.setCursor(x, y)` – establecer las coordenadas para comenzar a escribir el texto
- `display.print("message")` – imprimir los caracteres en la ubicación x,y

Ejemplo:Mostrar la temperatura y la humedad

En este ejemplo va a mostrar la temperatura y la humedad en la pantalla OLED. El objetivo de este proyecto es para familiarizarse con la pantalla OLED.

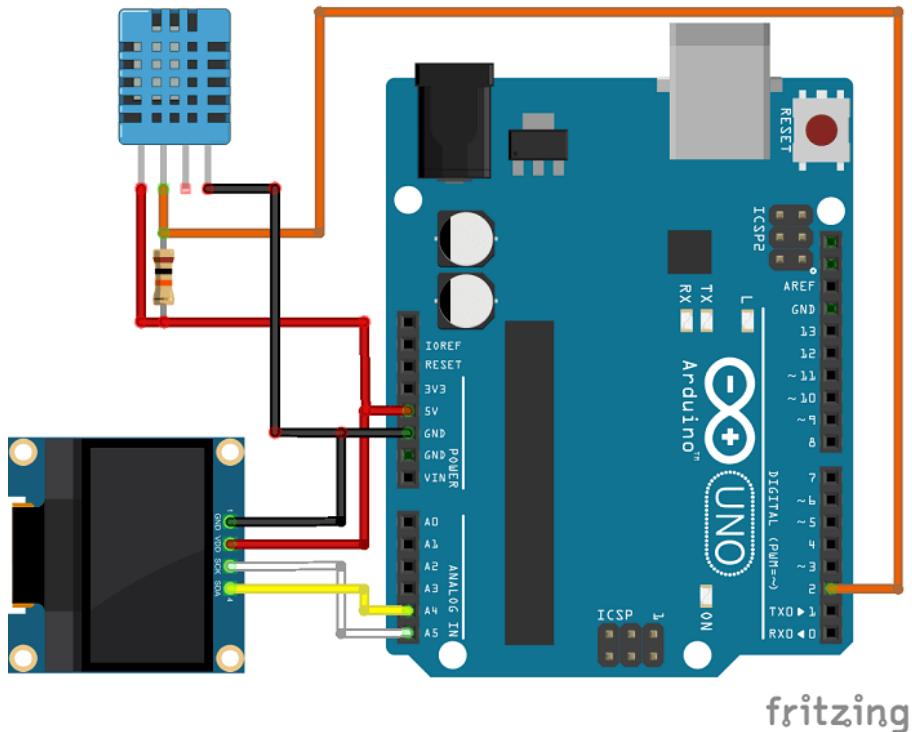
La temperatura y humedad se midieron usando el sensor de temperatura y humedad DHT11. Si usted no está familiarizado con el sensor DHT11 le recomendamos que consulte la guía de DHT11 / 22.

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	OLED Display	Find best price on Maker Advisor
	DHT11	Find best price on Maker Advisor
	Breadboard	Find best price on Maker Advisor
	10kΩ Resistor	Find best price on Maker Advisor



Circuitos

Monte todas las piezas como en los esquemas de abajo.



Código

Asegúrese de que ha instalado las bibliotecas necesarias para controlar la pantalla OLED. También es necesario instalar la biblioteca DHT. Compruebe el DHT11 / 22 temperatura y humedad guía de sensor.

A continuación, puede cargar el siguiente código.

```
/*
 * Random Nerd Tutorials - Rui Santos
 * Complete Project Details http://randomnerdtutorials.com
 */

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

#define DHTPIN 2      // what pin we're connected to
#define DHTTYPE DHT11  // DHT 11
#define OLED_RESET 4
```

```

Adafruit_SSD1306 display(OLED_RESET);
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
    Wire.begin();
    dht.begin(); // initialize dht
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // initialize with the I2C addr 0x3C (for
the 128x32) (initializing the display)
    Serial.begin(9600);
}

void displayTempHumid(){
    delay(2000);
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit
    float f = dht.readTemperature(true);
    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        display.clearDisplay(); // clearing the display
        display.setTextColor(WHITE); //setting the color
        display.setTextSize(1); //set the font size
        display.setCursor(5,0); //set the cursor coordinates
        display.print("Failed to read from DHT sensor!");
        return;
    }
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(0,0);
    display.print("Humidity: ");
    display.print(h);
    display.print(" %\t");
    display.setCursor(0,10);
    display.print("Temperature: ");
    display.print(t);
    display.print(" C");
    display.setCursor(0,20);
    display.print("Temperature: ");
    display.print(f);
    display.print(" F");
}
void loop()
{
    displayTempHumid();
    display.display();
}

```

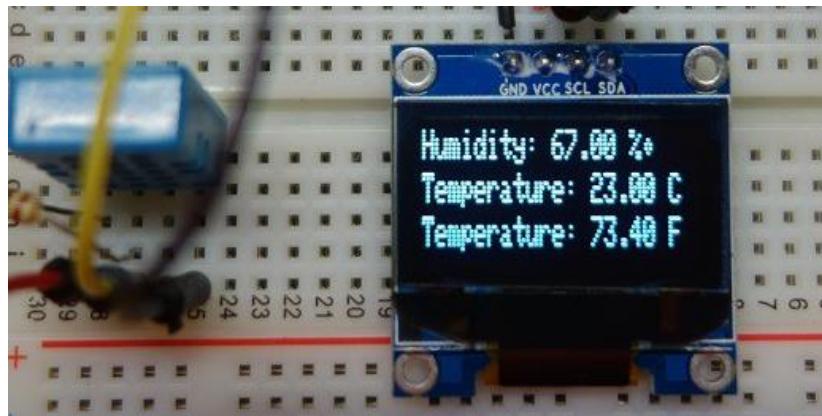


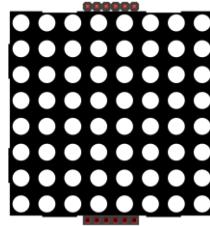
CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/display_tempreature_and_humidity_in_oled.ino

Demostración

Aquí está la pantalla OLED en acción después de subir el código..



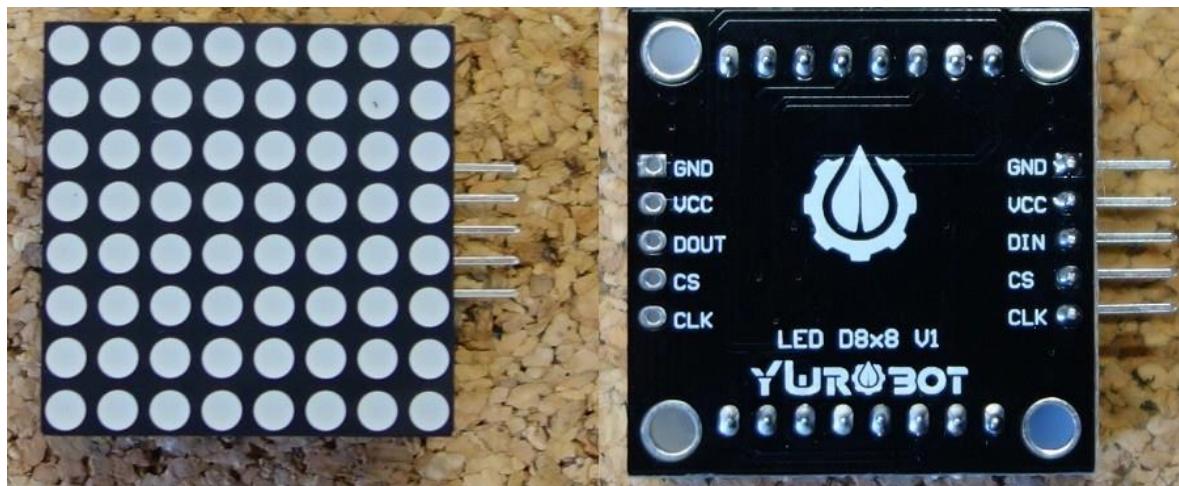


8x8 matriz de puntos

La matriz de puntos que vamos a utilizar en esta guía es una matriz de 8×8 , que significa que tiene 8 columnas y 8 filas, por lo que contiene un total de 64 LEDs.

El chip MAX7219 hace que sea más fácil controlar la matriz de puntos, sólo por el uso de 3 pines digitales de la tarjeta Arduino.

Creo que la mejor opción es comprar la matriz de puntos con el chip MAX7219 como un módulo, se va a simplificar el cableado.



Se puede controlar más de una matriz a la vez. Para que usted sólo tiene que conectarlos entre sí, ya que tienen pasadores en ambos lados para extender la matriz de puntos.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [MAX7219 LED matrix](#)

Cableado

Sólo tiene que conectar 5 pines de la matriz de puntos a la placa Arduino. El cableado es sencillo:

Pin de matriz de puntos	El cableado a Arduino Uno
GND	GND
VCC	5V
DIN	Digital pin
CS	Digital pin
CLK	Digital pin

Bibliotecas

Para hacer más fácil el control de la matriz de puntos, es necesario descargar e instalar en su IDE Arduino la LedControl biblioteca. Para instalar la biblioteca siga estos pasos:

1. [Haga clic aquí para descargar la biblioteca LedControl.](#) Usted debe tener una carpeta .zip en sus descargas
2. Descomprimir la carpeta .zip y usted debe obtener LedControl-master carpeta
3. Cambiar el nombre de la carpeta de LedControl-maestro para LedControl
4. Mover la carpeta LedControl a su carpeta de bibliotecas de instalación Arduino IDE
5. Por último, vuelva a abrir el IDE de Arduino

Usando las funciones de la biblioteca LedControl

La forma más sencilla de mostrar algo en la matriz de puntos es mediante el uso de las funciones setLed () , setRow() o setColumn () . Estas funciones le permiten controlar un solo LED, una fila o una columna a la vez. He aquí los parámetros para cada función:

```
setLed(addr, row, col, state)
```

- **addr** es la dirección de su matriz, por ejemplo, si usted tiene sólo 1 de la matriz, el addr será cero.
- **row** es la fila donde se encuentra el LED

- **col** es la columna donde se encuentra el LED
- **state**
 - Es cierto o 1 si desea activar el led
 - Es falso o 0 si desea apagarlo

```
setRow(addr, row, value)
setCol(addr, column, value)
```

Ejemplo: Viendo iconos en la matriz de puntos

En este ejemplo, podrás mostrar varias caras secuencialmente en una matriz de puntos: un feliz, neutral, y una cara triste.

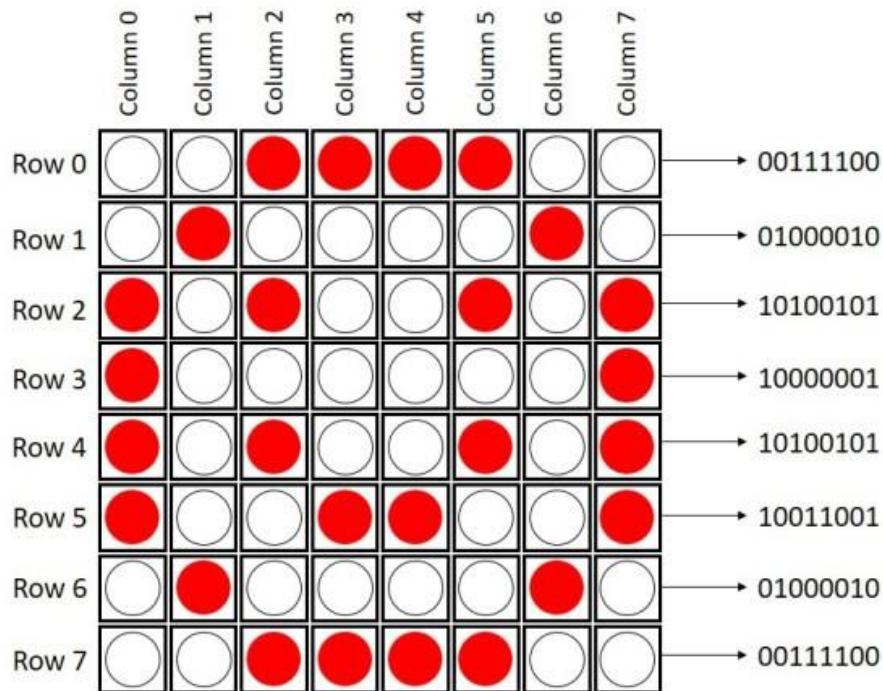
Index

Como se dijo anteriormente, esta matriz tiene 8 columnas y 8 filas. Cada uno está indexado de 0 a 7. Esto es una figura para una mejor comprensión:

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Row 0	●	●	●	●	●	●	●	●
Row 1	●	●	●	●	●	●	●	●
Row 2	●	●	●	●	●	●	●	●
Row 3	●	●	●	●	●	●	●	●
Row 4	●	●	●	●	●	●	●	●
Row 5	●	●	●	●	●	●	●	●
Row 6	●	●	●	●	●	●	●	●
Row 7	●	●	●	●	●	●	●	●

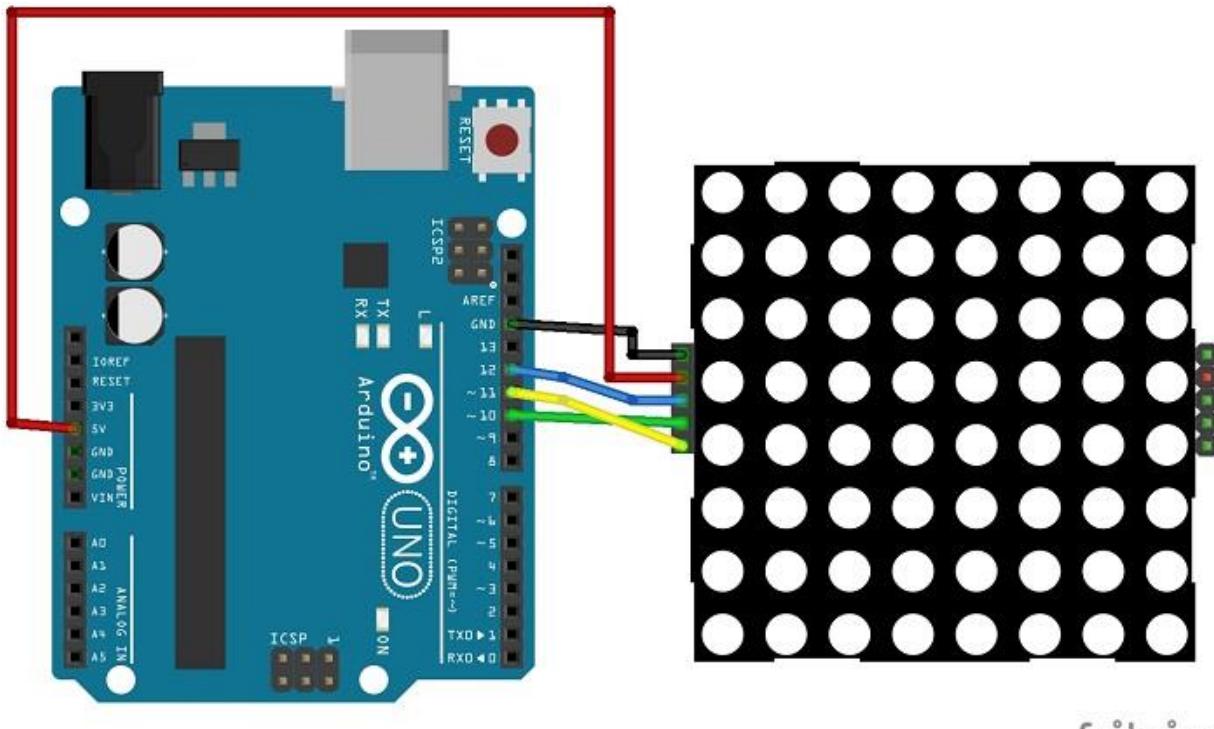
Si desea mostrar algo en la matriz, sólo tiene que saber si en una fila o columna determinada, los LEDs que están encendidos o apagados.

Por ejemplo, si desea mostrar una cara feliz, esto es lo que hay que hacer:



Circuito

Conectar la matriz de puntos como en el diagrama a continuación.



fritzing

He aquí un bosquejo simple que muestra tres tipos de caras: una cara triste, una cara neutra y una cara feliz.

Sube el código siguiente para su tablero:

```

/*
Created by Rui Santos

All the resources for this project:
http://randomnerdtutorials.com/
*/

#include "LedControl.h"
#include "binary.h"

/*
DIN connects to pin 12
CLK connects to pin 11
CS connects to pin 10
*/
LedControl lc=LedControl(12,11,10,1);

// delay time between faces
unsigned long delaytime=1000;

// happy face
byte hf[8]=
{B00111100,B01000010,B10100101,B10000001,B10100101,B10011001,B01000010,B00111100};

// neutral face
byte nf[8]={B00111100,
B01000010,B10100101,B10000001,B10111101,B10000001,B01000010,B00111100};

// sad face
byte sf[8]=
{B00111100,B01000010,B10100101,B10000001,B10011001,B10100101,B01000010,B00111100};

void setup() {
lc.shutdown(0,false);
// Set brightness to a medium value
lc.setIntensity(0,8);
// Clear the display
lc.clearDisplay(0);
}

void drawFaces(){
// Display sad face
lc.setRow(0,0,sf[0]);
lc.setRow(0,1,sf[1]);
lc.setRow(0,2,sf[2]);
lc.setRow(0,3,sf[3]);
lc.setRow(0,4,sf[4]);
lc.setRow(0,5,sf[5]);
lc.setRow(0,6,sf[6]);
lc.setRow(0,7,sf[7]);
delay(delaytime);

// Display neutral face
}

```

```

lc.setRow(0,0,nf[0]);
lc.setRow(0,1,nf[1]);
lc.setRow(0,2,nf[2]);
lc.setRow(0,3,nf[3]);
lc.setRow(0,4,nf[4]);
lc.setRow(0,5,nf[5]);
lc.setRow(0,6,nf[6]);
lc.setRow(0,7,nf[7]);
delay(delaytime);

// Display happy face
lc.setRow(0,0,hf[0]);
lc.setRow(0,1,hf[1]);
lc.setRow(0,2,hf[2]);
lc.setRow(0,3,hf[3]);
lc.setRow(0,4,hf[4]);
lc.setRow(0,5,hf[5]);
lc.setRow(0,6,hf[6]);
lc.setRow(0,7,hf[7]);
delay(delaytime);
}

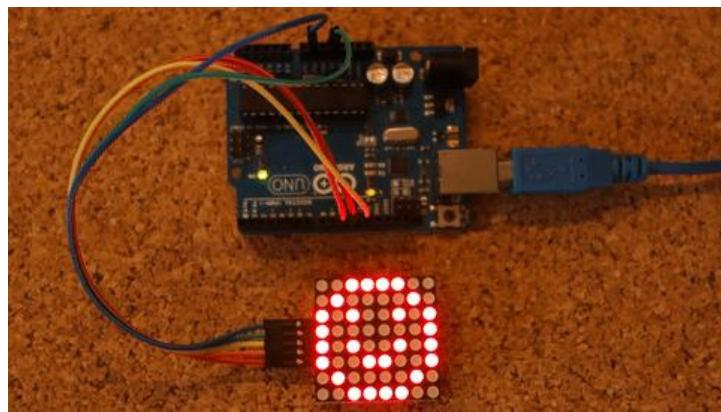
void loop() {
  drawFaces();
}

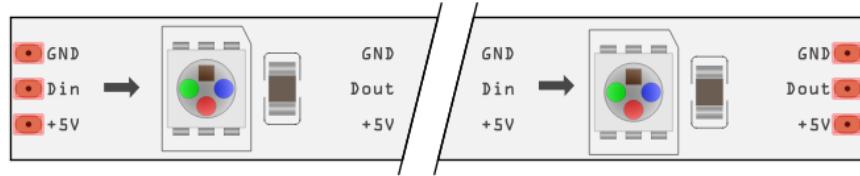
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/dot_matrix_faces.ino

Al final, usted tendrá algo como esto:





WS2812B Tira de Led RGB Addressable

La tira del LED WS2812B es una tira de LED RGB direccionable . La información de esta guía también trabaja con otras tiras de LED similares , tales como tiras de la familia WS28XX, tiras Neopixel y otros.

La tira WS2812 B direccionable LED viene en varias variedades que difieren en tamaño, de sellado o densidad LED. Elija el que mejor se adapte a sus propósitos.



¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [WS2812B addressable RGB LED Strip](#)

WS2812B RGB direccionable tira de LED

En la figura anterior se puede ver mi tira de LED WS2812B. Se encuentra a 5 metros de largo y los LEDs están encerrados en una silicona resistente a la intemperie.

En mi opinión, este es el tipo más cool de tiras de LED. Puede controlar el brillo y el color de cada LED individualmente, lo que le permite producir efectos sorprendentes y complejos de una manera sencilla.

Esta tira del LED es hecho por LEDs WS2812B conectados en serie. Estos LEDs tienen un IC integrada en el LED. Esto permite una comunicación a través de una interfaz de un solo alambre. Esto significa que se puede controlar una gran cantidad de LEDs utilizando sólo un pin digital de la placa Arduino.

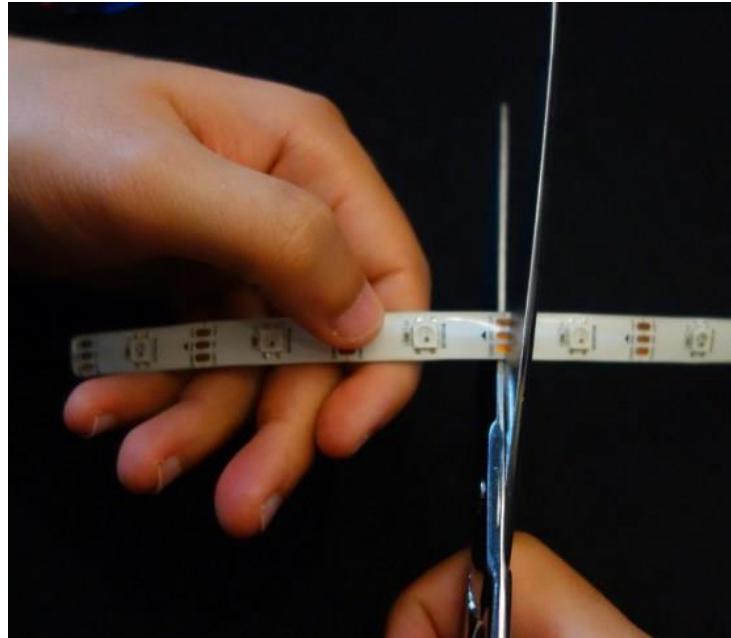
En la siguiente figura se puede ver el chip en el interior del LED. El LED es un LED RGB y funciona como tal.



Este tipo de bandas son muy flexibles y se puede cortar a cualquier longitud que desee. Como se puede ver, la banda se divide en segmentos, y cada segmento contiene un LED RGB.



Puede ajustar su tamaño mediante la reducción de la tira con unas tijeras en el lugar correcto (los lugares adecuados para cortar la tira están marcados).



Estas tiras vienen con conectores en cada extremo . He decidido cortar los conectores y pines del cabezal de soldadura. Es más práctico si desea conectar la tira a un Arduino o para una placa.



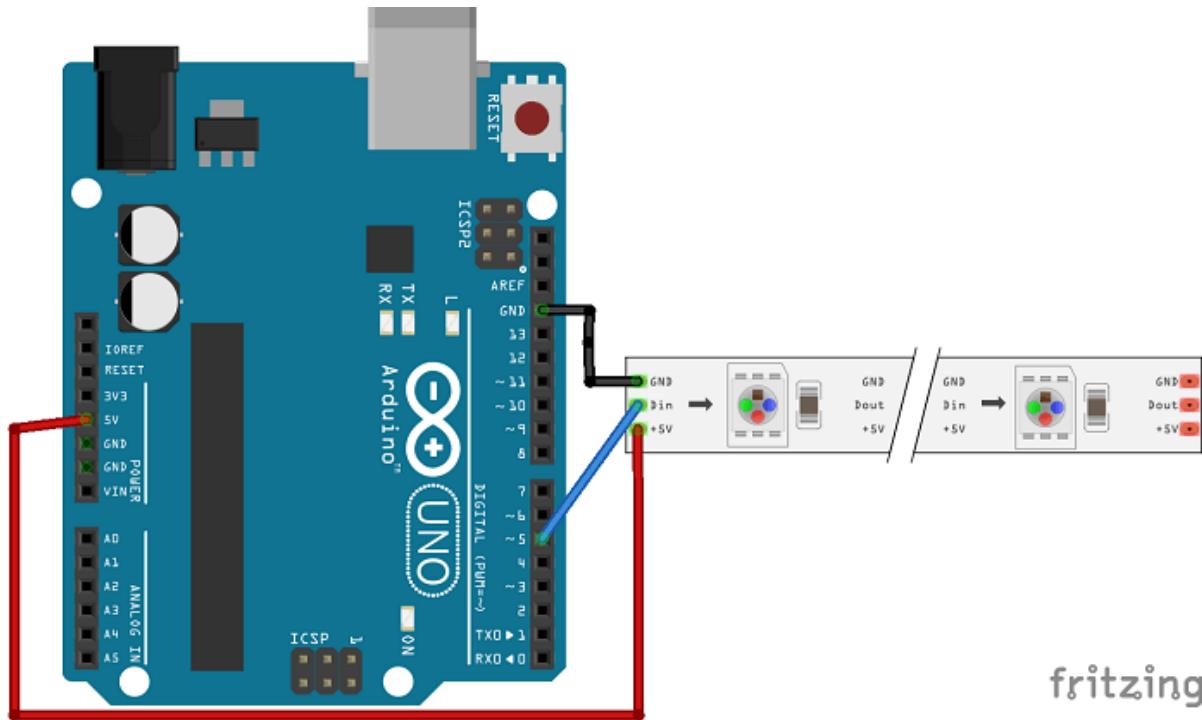
Alimentación de la Strip WS2812B LED

La tira del LED debe ser alimentado mediante una fuente de alimentación de 5V. A 5V, cada LED consume alrededor de 50 mA, cuando se establece en todo su brillo. Esto significa que por cada 30 LED, la tira puede extraer hasta 1,5 A. Asegúrese de que se selecciona una fuente de alimentación compatible con las necesidades de la tira.

Si al final el uso de una fuente externa , no se olvide de la conexión a tierra de alimentación al suelo Arduino.

Círcuito

En este ejemplo, la tira de LED WS2812B será accionado mediante el pasador de 5V Arduino. En mi caso, estoy controlando 14 LEDs. Tenga en cuenta que si usted quiere controlar muchos LEDs, tendrá que utilizar una fuente de alimentación externa.



Consejos útiles:

- Conectar un condensador con una capacitancia entre 100uF y 1000uF de potencia a tierra para suavizar la fuente de alimentación.
- Añadir un 220 o 470 Ohm resistor entre la clavija de salida digital Arduino y la clavija de entrada de datos de banda para reducir el ruido en esa línea.
- Hacer que sus cables entre el Arduino, fuente de alimentación y la tira lo más corto posible para minimizar la pérdida de tensión.
- Si la barra se daña y no funciona, compruebe si el primer LED está roto. Si es así, se corta, Resolder los pasadores de cabecera, y se debe trabajar de nuevo.

Código

Para controlar la tira de LED WS2812B, tendrá que descargar el FastLED biblioteca.

Installing the FastLED library

1. [Haga clic aquí para descargar la biblioteca FastLED ..](#) Usted debe tener una carpeta .zip en la carpeta de Descargas
2. Descomprimir la carpeta .zip y usted debe obtener FastLED-master carpeta
3. Cambiar el nombre de la carpeta de FastLED-maestro para FastLED
4. Mover la carpeta FastLED a su carpeta de bibliotecas de instalación Arduino IDE
5. Por último, vuelva a abrir el IDE de Arduino

Después de instalar la biblioteca es necesario, cargar el código siguiente a la placa Arduino (esto es un sketch ejemplo proporcionado en la carpeta de ejemplos de la biblioteca).

Ir **Archivo• Ejemplos • FastLED • Paleta de color** o copiar el código de abajo.

Nota: tienes que cambiar **# define NUM_LEDS** con el número de LEDs que su tira tiene actualmente.

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Arduino_WS2812B_Color_Palette.ino

Demostración

Al final, esto es lo que tendrá. efectos sorprendentes como ésta:



O este:



Y así (...)

El uso de una caja de tira de LED

Estas tiras vienen generalmente con una cinta extraíble, para que pueda pegarlas donde quieras . El problema es que no se adhieren muy bien , así que lo más probable es que usted encontrará su tira en el suelo al día siguiente.

La solución : me encontré con este caso franja que difunde la luz y así se puede atornillar a un estante, por ejemplo, si desea una solución permanente.





Teclado de membrana

Un teclado le permite interactuar con un microcontrolador . Puede salvar estos teclados de los teléfonos viejos o puede comprarlos en la mayoría de tiendas de electrónica . Vienen en gran variedad de formas y tamaños . Los tamaños más comunes son de 3×4 y 4×4 y se puede obtener con teclados palabras , letras y números escritos en las teclas.



Estos teclados muy populares entre los makers Arduino. Son muy baratos, y usted puede utilizar con cualquier microcontrolador (MCU). Usted puede incluso crear su propio teclado a partir de cero.

¿Donde comprar?

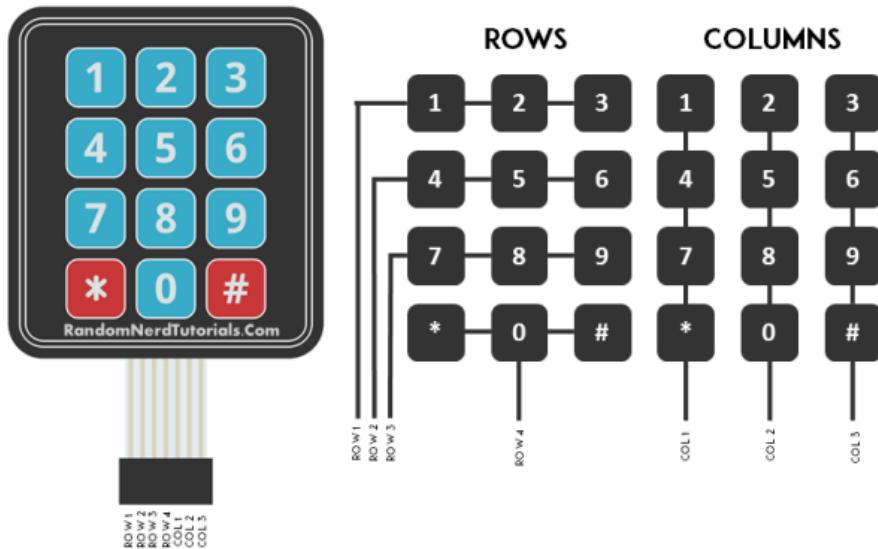
Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Membrane keypad](#)

¿Cómo funciona?

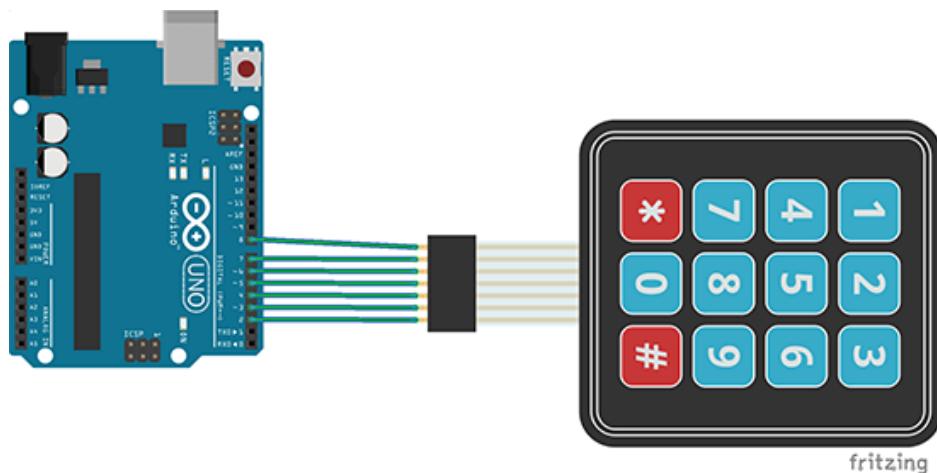
Un teclado de membrana es una matriz constituida por filas y columnas. Cada tecla está asignada a una determinada fila y columna (véase el cuadro abajo).

En un teclado de 12 botones que tiene 4 filas y 3 columnas. La primera clave sería establecer un vínculo entre la fila 1 y la columna 1 (R1C1). 2 habría R1C2, R1C3 3, * R4C1, R3C3 9 y así sucesivamente.



Circuito

Siga los siguientes esquemas. Si su teclado es diferente de la de abajo, la búsqueda de la ficha técnica en línea.



Descarga biblioteca

1. [Haga clic aquí para descargar la biblioteca del teclado](#). Usted debe tener una carpeta zip en la carpeta de Descargas
2. Descomprimir la carpeta .zip y usted debe obtener la carpeta teclado
3. Mover la carpeta del teclado a su carpeta de bibliotecas de instalación Arduino IDE
4. Finally, re-open your Arduino IDE

Código

Si el teclado no funciona con código de abajo puede que tenga que cambiar las conexiones de los esquemasanteriores . Usted tiene que asegurarse de que siga ficha técnica de su teclado.

Nota: Si su teclado tiene más teclas puede cambiar las líneas 3 y 4 para agregar el número correcto de filas y columna. Luego, en la línea 5 se puede cambiar la matriz para que coincida con sus teclas del teclado.

```
#include "Keypad.h"

const byte ROWS = 4; // number of rows
const byte COLS = 3; // number of columns
char keys[ROWS][COLS] = {
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'#','0','*'}

};

byte rowPins[ROWS] = {8, 7, 6, 5}; // row pinouts of the keypad R1 = D8, R2 = D7, R3 = D6, R4 = D5
byte colPins[COLS] = {4, 3, 2}; // column pinouts of the keypad C1 = D4, C2 = D3, C3 = D2
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  Serial.begin(9600);
}
void loop(){
  char key = keypad.getKey();
  if (key != NO_KEY)
    Serial.println(key);
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/Arduino_with_Keypad.ino

Demostración

En este proyecto cuando se pulsa una tecla, aparece su valor en el monitor serie de Arduino. Esto es lo que debe ver en su monitor serie Arduino IDE cuando se inicia pulsando las teclas del teclado .



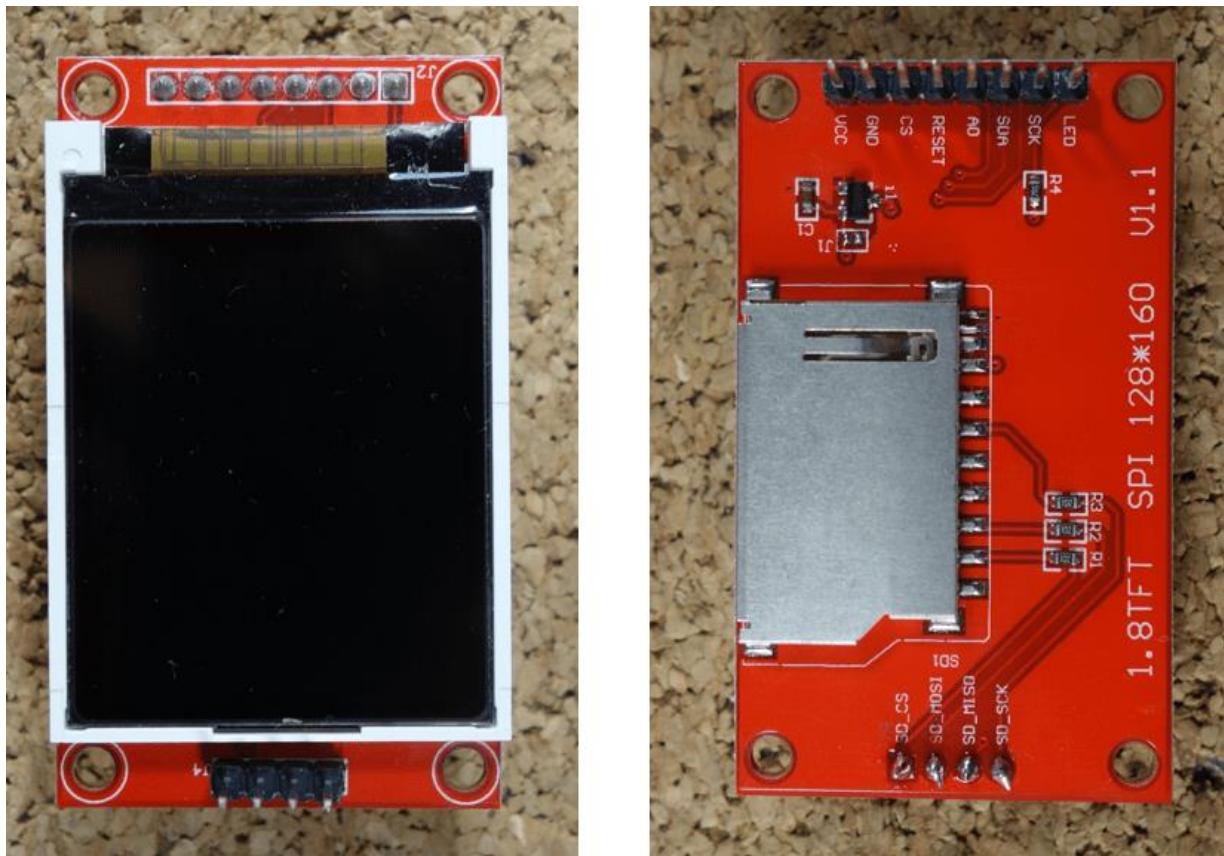
Terminando

Ahora usted puede crear una interfaz para la placa Arduino usando un teclado. Tú también puedes añadir una pantalla LCD a este proyecto .



1.8 TFT Display

El TFT 1.8 es una pantalla de colores con 128 x 160 píxeles de color. La pantalla puede cargar imágenes desde una tarjeta SD - que tiene una ranura para tarjetas SD en la parte posterior. La siguiente figura muestra la parte frontal de la pantalla y vista posterior.



This module uses SPI communication – see the wiring below. To control the display we'll use the TFT library, which is already included with Arduino IDE 1.0.5 and later.

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [1.8 TFT display](#)

Cableado

La tabla a continuación muestra el cableado 1.8 TFT a Arduino UNO.

1.8 TFT Display Pin	Wiring to Arduino Uno
LED	3.3 V
SCK	13
SDA	11
A0 or DC	9
RESET	8
CS	10
GND	GND

Nota: diferentes placas Arduino tienen diferentes pines SPI. Si está utilizando otra placa Arduino, comprobar la documentación oficial de Arduino.

La inicialización de la pantalla

La pantalla TFT se comunica con la comunicación Arduino a través de SPI, por lo que es necesario incluir la librería SPI en su código. También usamos la biblioteca TFT para escribir y dibujar en la pantalla.

```
#include <TFT.h>
#include <SPI.h>
```

Entonces, es necesario definir el CS, A0 (o CC) y los pasadores de RST:

```
#define cs 10
#define dc 9
#define rst 8
```

Crear una instancia de la biblioteca llamada TFTscreen :

```
TFT TFTscreen = TFT(cs, dc, rst);
```

Por último, en el `setup()`, que necesita para inicializar la biblioteca:

```
TFTscreen.begin();
```

Mostrar Texto

Para escribir texto en la pantalla, puede personalizar el color de fondo de pantalla, tamaño de fuente y color.

Para establecer el color de fondo, utilice:

```
TFTscreen.background(r, g, b);
```

En el cual, **r**, **g** **b** son los valores RGB para un color dado.

Para elegir el uso color de la fuente:

```
TFTscreen.stroke(r, g, b);
```

Para establecer el tamaño de la fuente:

```
TFTscreen.setTextSize(2);
```

Puede aumentar o disminuir el número dado como argumento, para aumentar o disminuir el tamaño de la fuente.

Por último, para dibujar texto en la pantalla que utilice la siguiente línea:

```
TFTscreen.text("Hello, World!", x, y);
```

En el cual "¡Hola Mundo!" es el texto que desea mostrar y el (**x**, **y**) coordenada es la ubicación en la que desea iniciar el texto visualizado en la pantalla.

Código

El siguiente ejemplo muestra "Hola, mundo!" En el centro de la pantalla y cambia el color de fuente cada 200 milisegundos.

Copia el siguiente código a su IDE Arduino y subirlo a tu placa Arduino. /*

* Rui Santos

* Complete Project Details <http://randomnerdtutorials.com>

*/

```
// include TFT and SPI libraries
```

```

#include <TFT.h>
#include <SPI.h>

// pin definition for Arduino UNO
#define cs 10
#define dc 9
#define rst 8

// create an instance of the library
TFT TFTscreen = TFT(cs, dc, rst);

void setup() {

    //initialize the library
    TFTscreen.begin();

    // clear the screen with a black background
    TFTscreen.background(0, 0, 0);
    //set the text size
    TFTscreen.setTextSize(2);
}

void loop() {

    //generate a random color
    int redRandom = random(0, 255);
    int greenRandom = random (0, 255);
    int blueRandom = random (0, 255);

    // set a random font color
    TFTscreen.stroke(redRandom, greenRandom, blueRandom);

    // print Hello, World! in the middle of the screen
    TFTscreen.text("Hello, World!", 6, 57);

    // wait 200 miliseconds until change to next color
    delay(200);
}

```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/tft/write_text.ino

Aquí está el texto "Hola, mundo!" En la pantalla TFT de 1,8 pulgadas.



```

#include <SPI.h>

// pin definition for Arduino UNO
#define cs    10
#define dc    9
#define rst   8

// create an instance of the library
TFT TFTscreen = TFT(cs, dc, rst);

void setup() {

    //initialize the library
    TFTscreen.begin();

    // clear the screen with a black background
    TFTscreen.background(0, 0, 0);
}

void loop() {

    //generate a random color
    int redRandom = random(0, 255);
    int greenRandom = random (0, 255);
    int blueRandom = random (0, 255);

    // set the color for the figures
    TFTscreen.stroke(redRandom, greenRandom, blueRandom);

    // light up a single point
    TFTscreen.point(80,64);
    // wait 200 miliseconds until change to next figure
    delay(500);

    // draw a line
    TFTscreen.line(0,64,160,64);
    delay(500);

    //draw a square
    TFTscreen.rect(50,34,60,60);
    delay(500);

    //draw a circle
    TFTscreen.circle(80,64,30);
    delay(500);
}

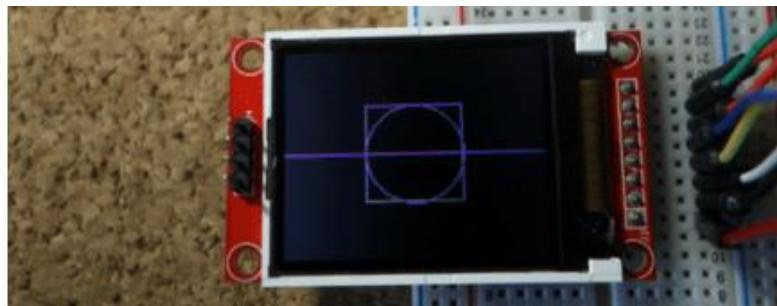
```

```
//erase all figures  
TFTscreen.background(0,0,0);  
}
```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/tft/draw_shapes.ino

He aquí las formas en la pantalla:



Mostrar imágenes

La pantalla TFT de 1,8 pulgadas puede cargar imágenes desde la tarjeta SD. Para leer desde la tarjeta SD se utiliza la biblioteca South Dakota , ya incluido en el software de Arduino IDE. Siga los siguientes pasos para mostrar una imagen en la pantalla:

- 1) Los contactos macho de soldadura para la tarjeta SD. Hay cuatro pasadores opuestos a los pasadores de la pantalla, como se muestra en la figura a continuación.



2) La pantalla puede cargar imágenes más grande o más pequeño que el tamaño de la pantalla (160 x 128 píxeles), pero para obtener mejores resultados, editar el tamaño de la imagen de 160 x 128 píxeles.

3) La imagen debe estar en formato bmp. Para ello, puede utilizar un software de edición de fotos y guardar la imagen como bmp.

4) Copiar la imagen en la tarjeta SD e insertarla en la ranura para tarjetas SD en la parte posterior de la pantalla.

5) Conectar los contactos de la tarjeta SD a la Arduino siguiendo la siguiente tabla:

SD card pins on TFT display	Wiring to Arduino Uno
CS	4
MOSI	11
MISO	13
SCK	12

Tanto la pantalla y el trabajo de la tarjeta SD con la comunicación SPI, por lo que tendrá pines en la Arduino con dos conexiones.

6) En el IDE Arduino ir a **Archivo • Ejemplos • TFT • Arduino • TFTBitmaLogo**.

7) Editar el código, por lo que busca para su imagen. Reemplace “Arduino.bmp” con el nombre de la imagen:

```
// now that the SD card can be access, try to load the image file  
logo = TFTscreen.loadImage ("arduino.bmp");
```

8) Sube el código a su Arduino.

Nota: algunas personas encuentran problemas con esta pantalla al intentar leer desde la tarjeta SD. No sabemos por qué sucede esto. De hecho, hemos probado un par de veces y funcionó bien, y luego, cuando estábamos a punto de grabar que le muestre el resultado final, la pantalla no reconoce la tarjeta SD más - no estamos seguros de si se trata de un problema con el titular de la tarjeta SD que no establezca una conexión adecuada con la tarjeta SD. Sin embargo, estamos seguros de estas instrucciones de trabajo, debido a que los hemos probado antes.

Terminando

En esta guía hemos demostrado cómo utilizar la pantalla de 1.8 TFT con el Arduino: texto de la pantalla, dibujar formas y visualizar las imágenes. Usted puede agregar fácilmente una interfaz visual agradable para sus proyectos utilizando esta pantalla

Shield SIM900 GSM GPRS

El Shield GSM GPRS SIM900 se muestra en la siguiente figura.



GSM significa Sistema Global para Comunicaciones Móviles y es el estándar global para las comunicaciones móviles.

GPRS siglas de General Packet Radio Service. GPRS es un servicio móvil en la comunicación celular 2G y 3G.

Aplicaciones:

El shield GSM GPRS es particularmente útil, ya que permite:

- Conectarse a Internet a través de red GPRS
- Enviar y recibir SMS
- Realizar y recibir llamadas de teléfonos
- Sus capacidades hacen que sea ideal para proyectos con Arduino como:
 - El control remoto de aparatos electrónicos - enviando un SMS al convertir algo en;
 - Recibir notificaciones - enviar SMS a su teléfono celular si se detecta movimiento en su casa;
 - Recibir datos del sensor - enviar SMS periódica a su teléfono celular con los datos meteorológicos diarios.

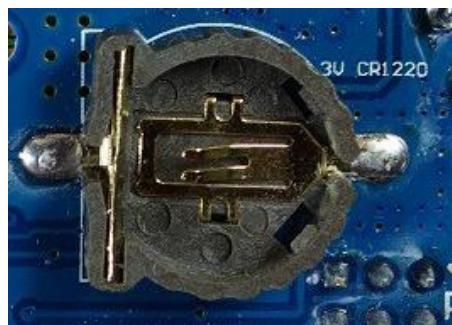
Características

He aquí algunas de las características más importantes del escudo:

- Compatible con Arduino y clones
- Sobre la base de módulo SIM900 de SIMCOM
- Le permite enviar SMS, MMS, GPRS y audio a través de UART usando los comandos AT.
- Cuenta con 12 GPIO, 2 PWM y ADC incorporado del módulo SIM900
- Banda cuádruple: 850; 900; 1800 y 1900 MHz, así que debería funcionar en todos los países con redes GSM

(2G)

- Control a través de los comandos AT
- Soporta RTC (reloj en tiempo real) - tiene un soporte para una batería de 3V CR1220 en la parte posterior



- Tiene micrófono y auriculares para llamadas telefónicas



¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [SIM900 GSM/GPRS Shield](#)

Pasos preliminares

Antes de empezar a utilizar su módulo SIM900 GSM GPRS, es necesario tener en cuenta algunos aspectos acerca de la tarjeta SIM y el poder protector de suministro.

Cobertura GSM

Asegúrese de que tiene la cobertura de una red GSM de 850 MHz, 900 MHz GSM, DCS 1800 MHz o 1900 MHz PCS red. Por GSM 2G que queremos decir.

Prepago tarjeta SIM

Le recomendamos que utilice un plan de prepago o con un plan ilimitado de SMS para realizar pruebas. De lo contrario, si algo va mal, puede que tenga que pagar una enorme factura de cientos de mensajes de texto SMS enviados por error.

En este tutorial que estamos usando un plan de prepago con SMS ilimitados.

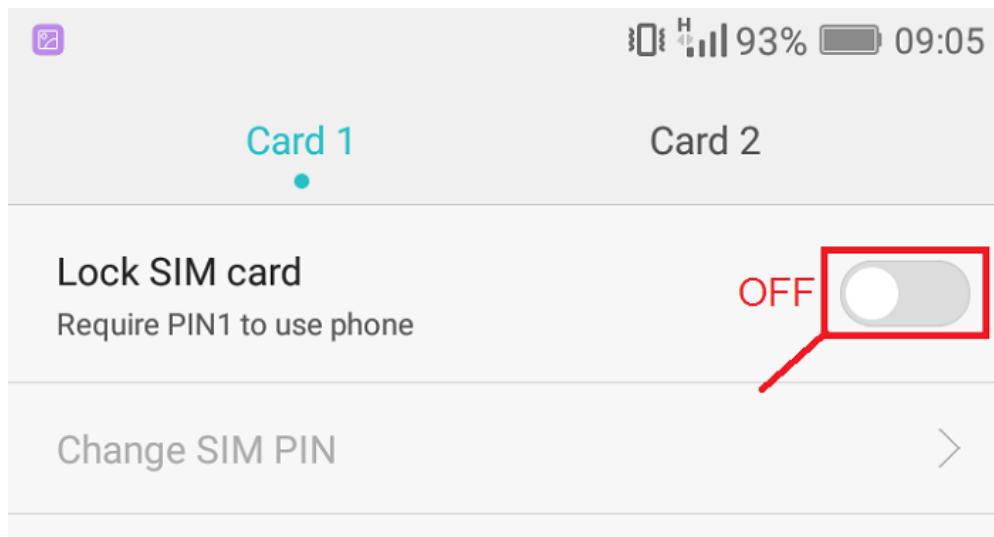


El shield utiliza el tamaño de la tarjeta SIM original, no micro o nano. Si tiene micro o nano puede considerar la obtención de un adaptador de tamaño de tarjeta SIM

Desactivar el bloqueo del PIN

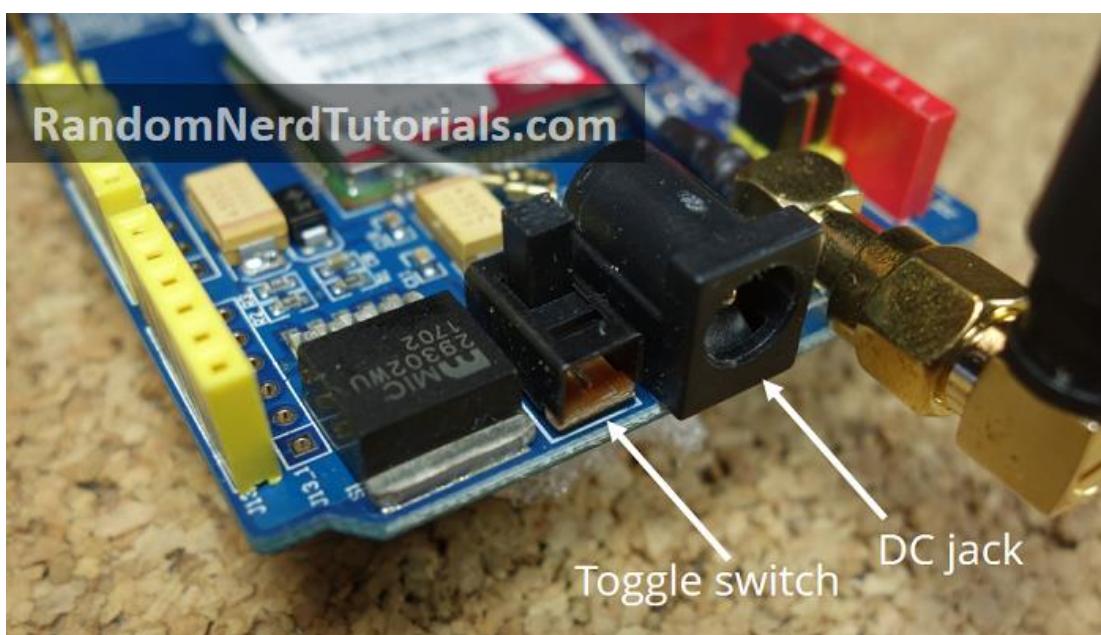
Para utilizar la tarjeta SIM con el escudo, es necesario desactivar el bloqueo del pasador. La forma más sencilla de hacer esto, es insertar la tarjeta SIM en el teléfono inteligente y desactivar el bloqueo del pasador en la configuración de seguridad del teléfono.

En mi caso, tenía que ir a través de: **Ajustes • Ajustes avanzados • Seguridad • bloqueo de SIM** y apague el Bloquear tarjeta SIM con el pasador.



Obtener la fuente de alimentación correcta

El shield tiene una toma de CC de alimentación como se muestra en la figura a continuación.



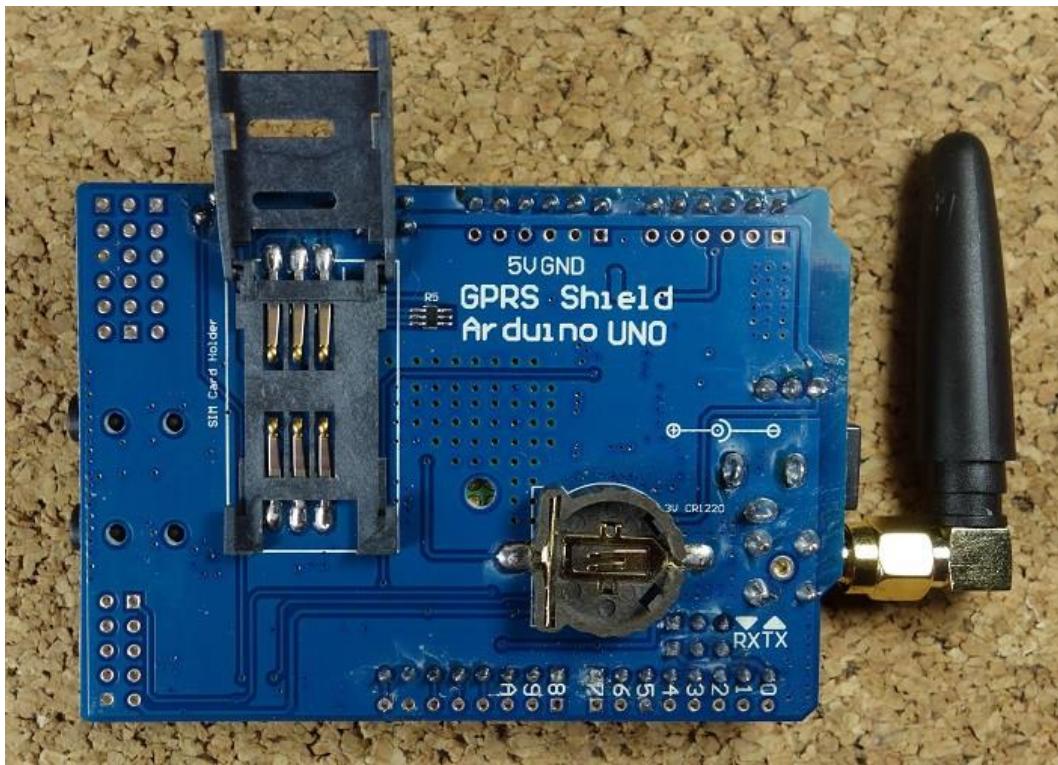
Al lado del conector de alimentación hay un interruptor de palanca para seleccionar la fuente de alimentación. Junto al interruptor de palanca en el tablero, hay una flecha que indica la posición de palanca de usar una fuente de alimentación externa - mover el interruptor de palanca para utilizar la fuente de alimentación externa como se muestra arriba.

Para encender el escudo, es aconsejable utilizar una fuente de alimentación de 5V que puede proporcionar 2A como la que se muestra a continuación . También puede ser alimentado con 9V 1A, o 12V 1A.

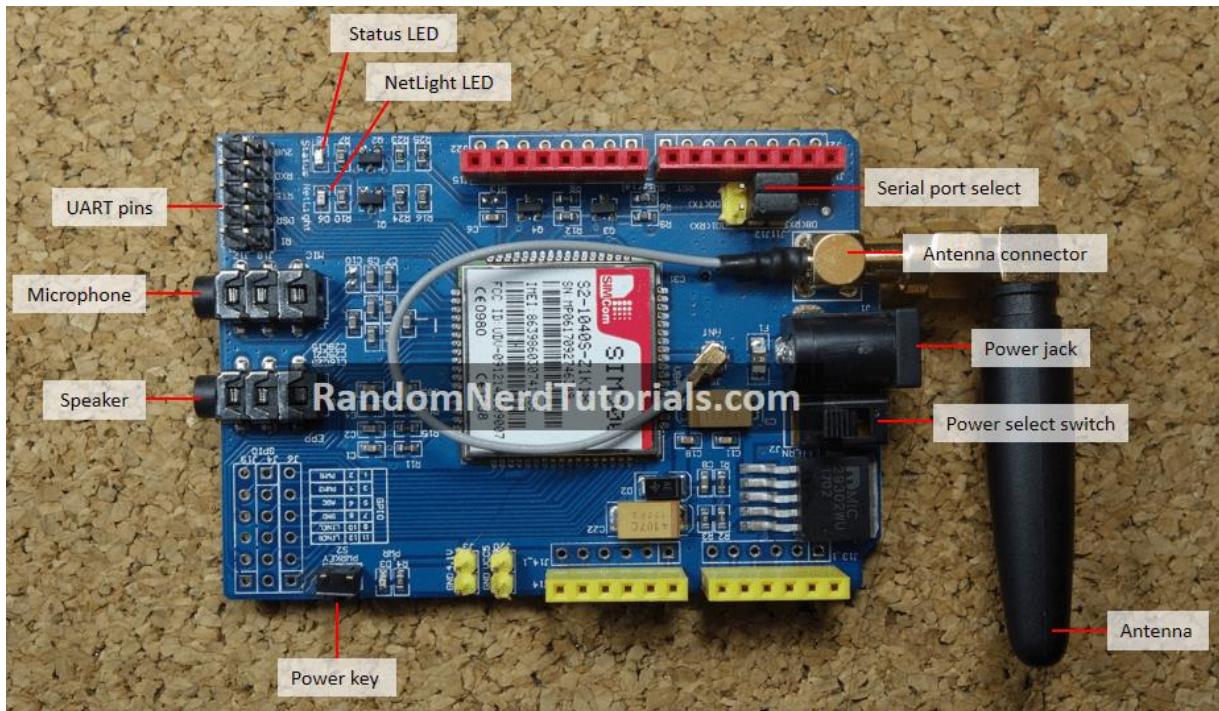


Hardware del shield SIM900 GSM GPRS

La siguiente figura muestra la parte posterior de la pantalla. Cuenta con un soporte para la tarjeta SIM y una batería de 3V CR1220 para el RTC (reloj en tiempo real).



La siguiente figura muestra el escudo de los componentes más importantes de la junta que es necesario prestar atención a.



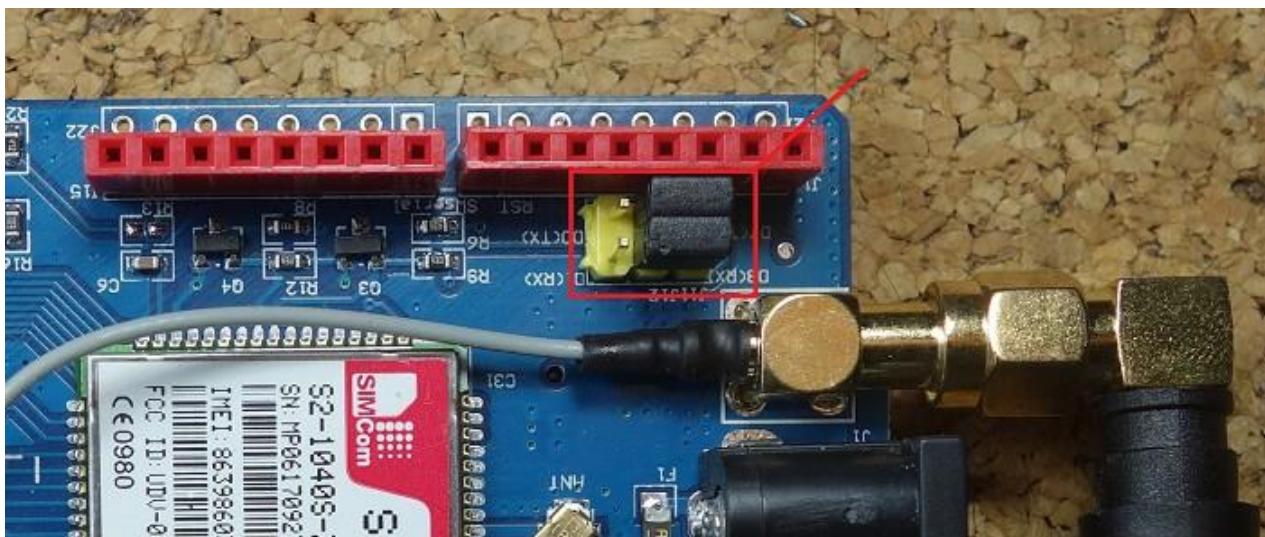
Empezando

1) Inserte la tarjeta SIM en el soporte de la tarjeta SIM - asegúrese de que ha leído los pasos preliminares en la sección anterior.

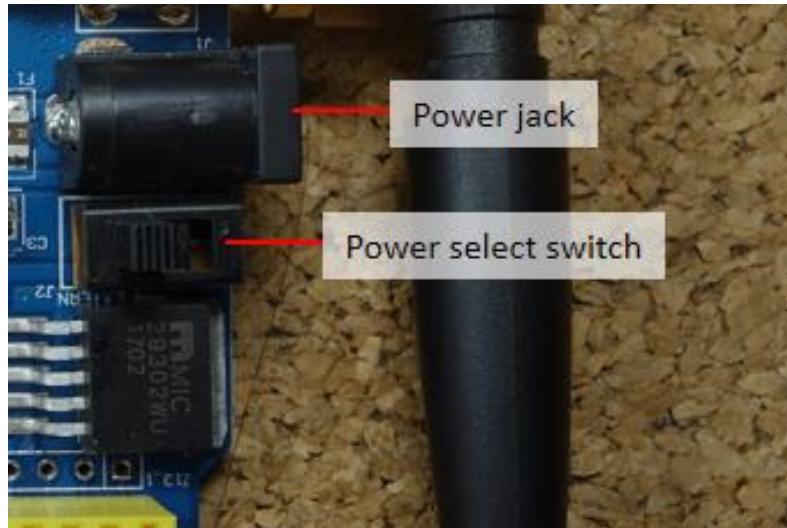


2) Asegúrese de que la antena está bien conectada.

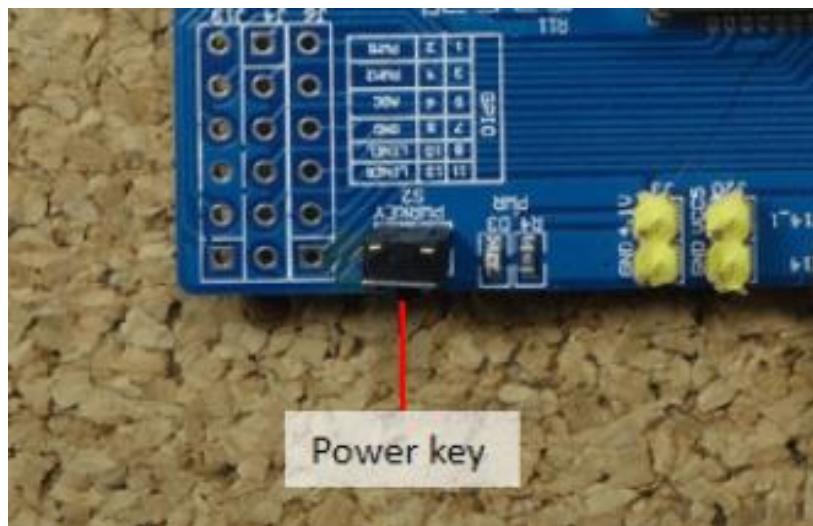
3) En la selección de puerto serie , asegúrese de que la tapa del puente está conectado como se muestra en la figura a continuación para utilizar el software de serie.



4) Alimentar el shield utilizando una fuente de 5V de alimentación externa.
Asegúrese de seleccionar la fuente de alimentación externa con el interruptor de palanca al lado del conector de CC.



5) Para Encender / Apagar presione la tecla de encendido durante unos 2 segundos.



6) A continuación, el LED de estado se iluminará y el LED parpadeará NetLight cada 800 ms hasta que encuentra la red. Cuando encuentra la red, el LED comienza a parpadear NetLight cada tres segundos.

Nota: puede activar de forma automática el shield a través del software. Ver cómo hacerlo en el activar automáticamente el escudo sección, después de los ejemplos de código.

7) Puede probar si el escudo está funcionando correctamente mediante el envío de comandos AT desde el IDE Arduino usando un programador FTDI - como veremos muestra más adelante en esta guía.

SIM900 Comandos AT

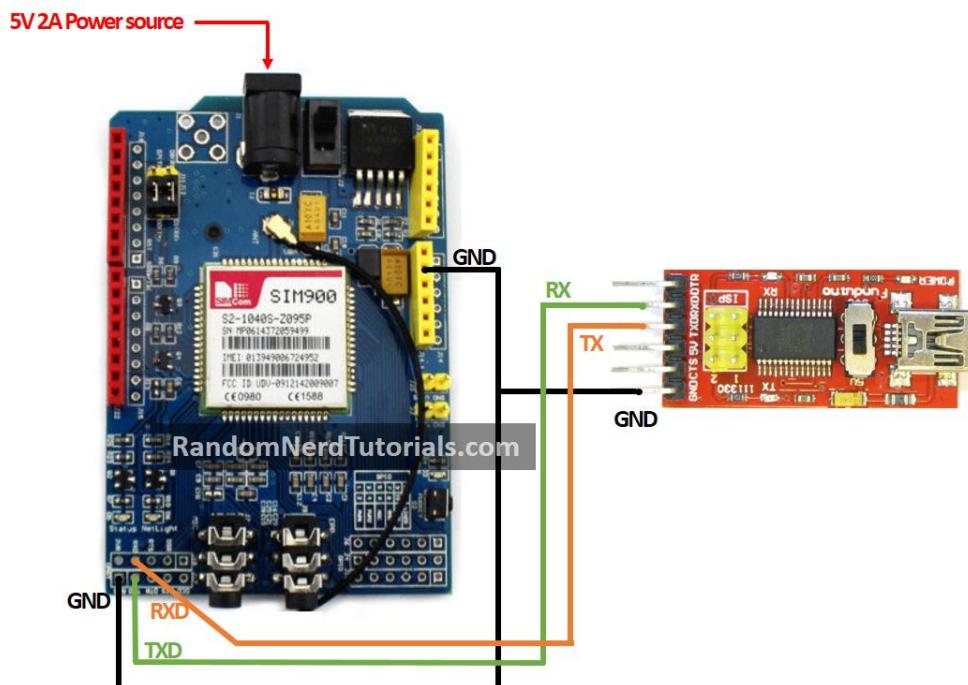
- establecer el SIM900 a modo de texto: AT + CMGF = 1 \ r
- enviar SMS a un número: Un T + CMGS = PHONE_NUMBER (en formato internacional)
- leer el primer SMS desde la bandeja de entrada: AT + CMGR = 1 \ r
- leer la segunda SMS desde la bandeja de entrada: AT + CMGR = 2 \ r
- leer todos los SMS de la bandeja de entrada: AT + CMGR = all \ r
- llamar a un número: ATDP + PHONE_NUMBER (en formato internacional)
- colgar una llamada: ATH
- recibir una llamada entrante: ATA
- Para obtener más información, se puede comprobar el SIM900 manual de comandos AT aquí .

Probando el shield con el programador FTDI

Para probar si todo funciona correctamente , puede probar el escudo mediante el envío de comandos AT del monitor serie Arduino IDE. Por eso, es necesario un programador FTDI como el que se muestra en la siguiente figura. Usted puede conseguir un programador FTDI como este [aqui](#).



1) Conectar el programador FTDI al escudo GSM, como se muestra en la figura a continuaci n.

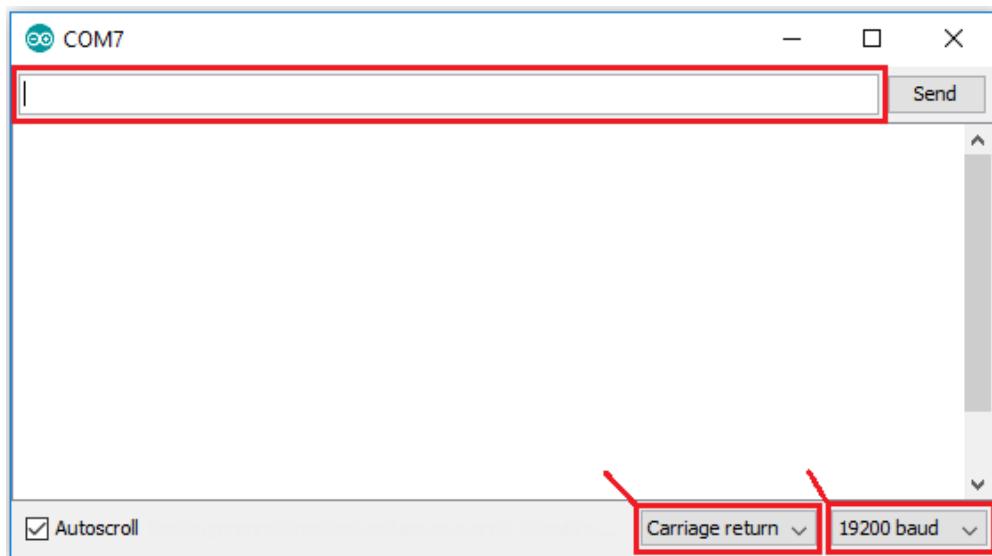


2) Abrir el IDE de Arduino y seleccione el puerto COM correcto.

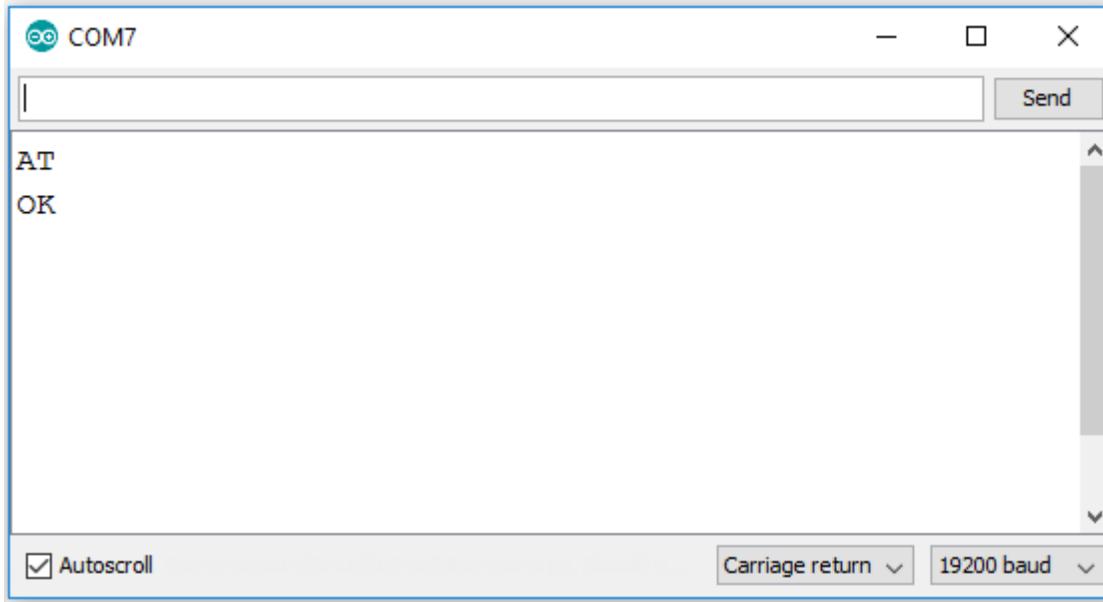
3) Abra el monitor de serie.



4) Seleccione la velocidad de transmisi n 19200 - el ajuste por defecto es 19200 escudo - y retorno de carro. Escribe por el cuadro resaltado en rojo y luego presione intro. Vea la siguiente figura.

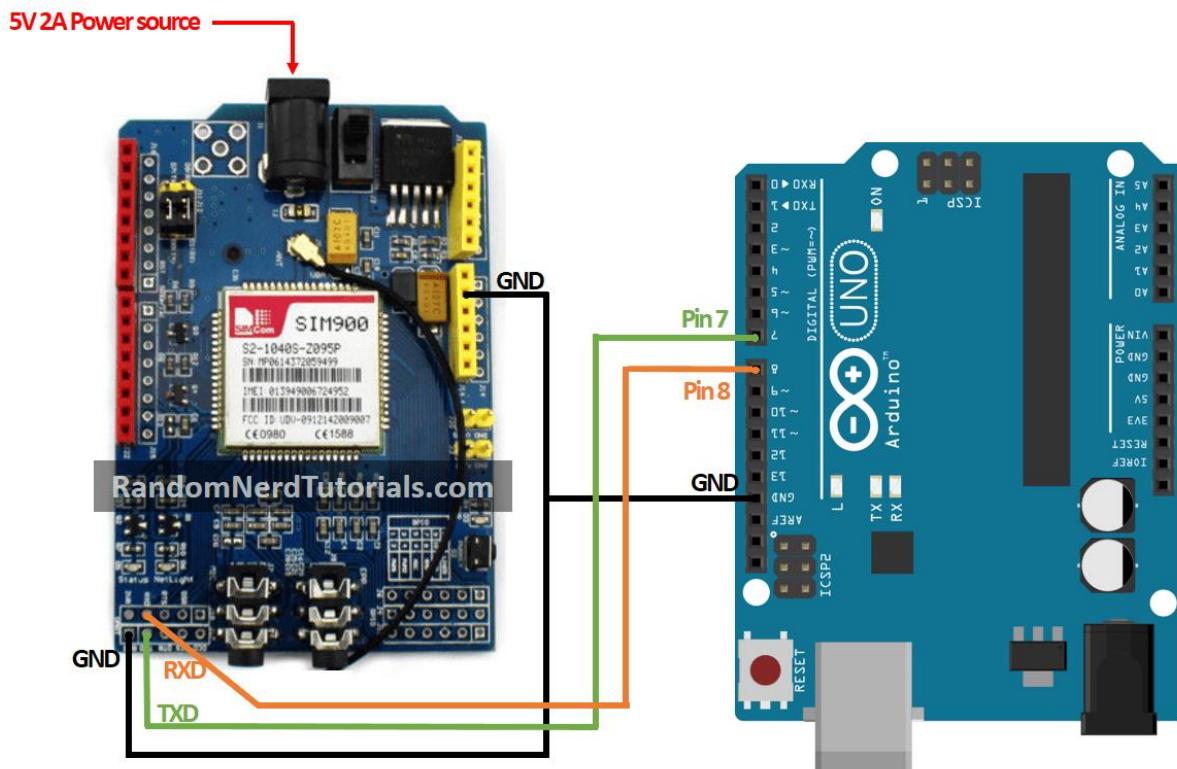


5) El escudo responderá con OK, Si todo funciona correctamente.



Coneectar el SHIELD a Arduino

Coneectar el shield a la Arduino como se muestra en el diagrama esquemático siguiente.



El envío de un SMS

Para enviar un SMS, cargar el código de abajo a la placa Arduino.

```
*****  
Complete project details at http://randomnerdtutorials.com  
*****/  
  
#include <SoftwareSerial.h>  
  
// Configure software serial port  
SoftwareSerial SIM900(7, 8);  
  
void setup() {  
    // Arduino communicates with SIM900 GSM shield at a baud rate of 19200  
    // Make sure that corresponds to the baud rate of your module  
    SIM900.begin(19200);  
    // Give time to your GSM shield log on to network  
    delay(20000);  
  
    // Send the SMS  
    sendSMS();  
}  
  
void loop() {  
}  
  
void sendSMS() {  
    // AT command to set SIM900 to SMS mode  
    SIM900.print("AT+CMGF=1\r");  
    delay(100);  
  
    // REPLACE THE X's WITH THE RECIPIENT'S MOBILE NUMBER  
    // USE INTERNATIONAL FORMAT CODE FOR MOBILE NUMBERS  
    SIM900.println("AT + CMGS = \'"+XXXXXXXXXXXX\'");  
    delay(100);  
  
    // REPLACE WITH YOUR OWN SMS MESSAGE CONTENT  
    SIM900.println("Message example from Arduino Uno.");  
    delay(100);  
  
    // End AT command with a ^Z, ASCII code 26  
    SIM900.println((char)26);  
    delay(100);  
    SIM900.println();  
    // Give module time to send SMS  
    delay(5000);  
}
```

{;} CÓDIGO FUENTE

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/GSM/sendSMS.ino>

En este código, se inicia mediante la inclusión de la SoftwareSerial.h biblioteca y crear un puerto de serie del software en los pines 7 y 8 (pin 7 se está estableciendo como RX y 8 como TX).

```
#include <SoftwareSerial.h>

SoftwareSerial SIM900(7, 8);
```

La función `sendSMS()` es lo que realmente envía el SMS. Esta función utiliza los comandos **AT: AT + CMGF = 1 \r y AT + CMGS**.

Es necesario cambiar el número de móvil del destinatario en: (sustituir las X con el número de teléfono del destinatario)

```
SIM900.println("AT + CMGS = \"+++++++\r");
```

El número de teléfono móvil del destinatario debe estar en formato internacional.

A continuación, en la siguiente línea se puede editar el texto que desea enviar.

```
// REPLACE WITH YOUR OWN SMS MESSAGE CONTENT

SIM900.println("Message example from Arduino Uno.");
```

Leer SMS Recibido

Para leer los SMS entrantes , cargar el código de abajo a la placa Arduino . Tras la descarga , esperar 20 segundos para que el shield haga la comunicación . A continuación, probar el script mediante el envío de un SMS al número del shield de la tarjeta SIM . El SMS se muestra en el monitor serie Arduino - velocidad de transmisión: 19200.

```
*****
Complete project details at http://randomnerdtutorials.com
*****


#include <SoftwareSerial.h>

// Configure software serial port
SoftwareSerial SIM900(7, 8);
//Variable to save incoming SMS characters
char incoming_char=0;
```

```

void setup() {
    // Arduino communicates with SIM900 GSM shield at a baud rate of 19200
    // Make sure that corresponds to the baud rate of your module
    SIM900.begin(19200);
    // For serial monitor
    Serial.begin(19200);
    // Give time to your GSM shield log on to network
    delay(20000);

    // AT command to set SIM900 to SMS mode
    SIM900.print("AT+CMGF=1\r");
    delay(100);
    // Set module to send SMS data to serial out upon receipt
    SIM900.print("AT+CNMI=2,2,0,0,0\r");
    delay(100);
}

void loop() {
    // Display any text that the GSM shield sends out on the serial monitor
    if(SIM900.available() >0) {
        //Get the character from the cellular serial port
        incoming_char=SIM900.read();
        //Print the incoming character to the terminal
        Serial.print(incoming_char);
    }
}

```

{;} CÓDIGO FUENTE

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/GSM/receiveSMS.ino>

En este código, se establece el módulo para enviar los datos de SMS a la salida serie:

```
SIM900.print("AT+CNMI=2,2,0,0,0\r");
```

Almacenar los caracteres entrantes desde el mensaje SMS en la variable **incoming_char**. Uno lee los caracteres utilizando la función SIM900.read ()

Haciendo una llamada telefónica

Para hacer una llamada telefónica, subir el siguiente código a su Arduino.

No se olvide de editar el código con el número de teléfono al que desea llamar.

```
*****  
Complete project details at http://randomnerdtutorials.com  
*****  
  
#include <SoftwareSerial.h>  
  
// Configure software serial port  
SoftwareSerial SIM900(7, 8);  
  
void setup() {  
    // Arduino communicates with SIM900 GSM shield at a baud rate of 19200  
    // Make sure that corresponds to the baud rate of your module  
    SIM900.begin(19200);  
    // Give time to your GSM shield log on to network  
    delay(20000);  
  
    // Make the phone call  
    callSomeone();  
}  
  
void loop() {  
  
}  
  
void callSomeone() {  
    // REPLACE THE X's WITH THE NUMBER YOU WANT TO DIAL  
    // USE INTERNATIONAL FORMAT CODE FOR MOBILE NUMBERS  
    SIM900.println("ATD + +XXXXXXXXXX;");  
    delay(100);  
    SIM900.println();  
  
    // In this example, the call only last 30 seconds  
    // You can edit the phone call duration in the delay time  
    delay(30000);  
    // AT command to hang up  
    SIM900.println("ATH"); // hang up  
}
```

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/GSM/makePhoneCall.ino>

Para realizar la llamada, se utiliza el llamá a alguien() función que utiliza la ATD mando.

```
SIM900.println ("ATD + +XXXXXXXXXX; ");
```

Es necesario sustituir el X 'S (resaltado en rojo) con el número de teléfono al que desea llamar.

No se olvide de conectar un micrófono y auriculares para realizar la llamada.

En este ejemplo de código, la llamada es colgar después de 30 segundos, utilizando el ATH mando:

```
SIM900.println ("ATH");
```

Colgar después de 30 segundos no es muy útil, pero funciona bien para un ejemplo. La idea es que se utiliza el comando ATH cuando se activa un evento. Por ejemplo , conectar un pulsador para el Arduino , que cuando se pulsa envía el comando ATH a colgar el teléfono.

Responder a las llamadas telefónicas entrantes

El siguiente código responde a las llamadas entrantes.

```
*****  
Complete project details at http://randomnerdtutorials.com  
*****  
  
#include <SoftwareSerial.h>  
  
// Configure software serial port  
SoftwareSerial SIM900(7, 8);  
char incoming_char=0;  
  
void setup() {  
  // Arduino communicates with SIM900 GSM shield at a baud rate of 19200 // Make sure  
  // that corresponds to the baud rate of your module  
  SIM900.begin(19200); // for GSM shield  
  // For serial monitor
```

```
Serial.begin(19200);
// Give time to log on to network.
delay(20000);

SIM900.print("AT+CLIP=1\r"); // turn on caller ID notification
delay(100);
}

void loop() {
// Display any text that the GSM shield sends out on the serial monitor
if(SIM900.available() >0) {
    // Get the character from the cellular serial port
    // With an incoming call, a "RING" message is sent out
    incoming_char=SIM900.read();
    // Check if the shield is sending a "RING" message
    if (incoming_char=='R') {
        delay(10);
        Serial.print(incoming_char);
        incoming_char=SIM900.read();
        if (incoming_char =='I') {
            delay(10);
            Serial.print(incoming_char);
            incoming_char=SIM900.read();
            if (incoming_char=='N') {
                delay(10);
                Serial.print(incoming_char);
                incoming_char=SIM900.read();
                if (incoming_char=='G') {
                    delay(10);
                    Serial.print(incoming_char);
                    // If the message received from the shield is RING
                    // Send ATA commands to answer the phone
                    SIM900.print("ATA\r");
                }
            }
        }
    }
}
```

Cuando alguien llama al número SIM900, envía un mensaje diciendo "anillo". Para saber si alguien le está llamando, se puede esperar a que los caracteres entrantes desde el SIM900 y luego, comparar si era un mensaje RING. Eso es lo que se hace en este código . Cuando se recibe un mensaje RING , se envía el ATA mando a contestar el teléfono.

<https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/GSM/receiveCall.ino>

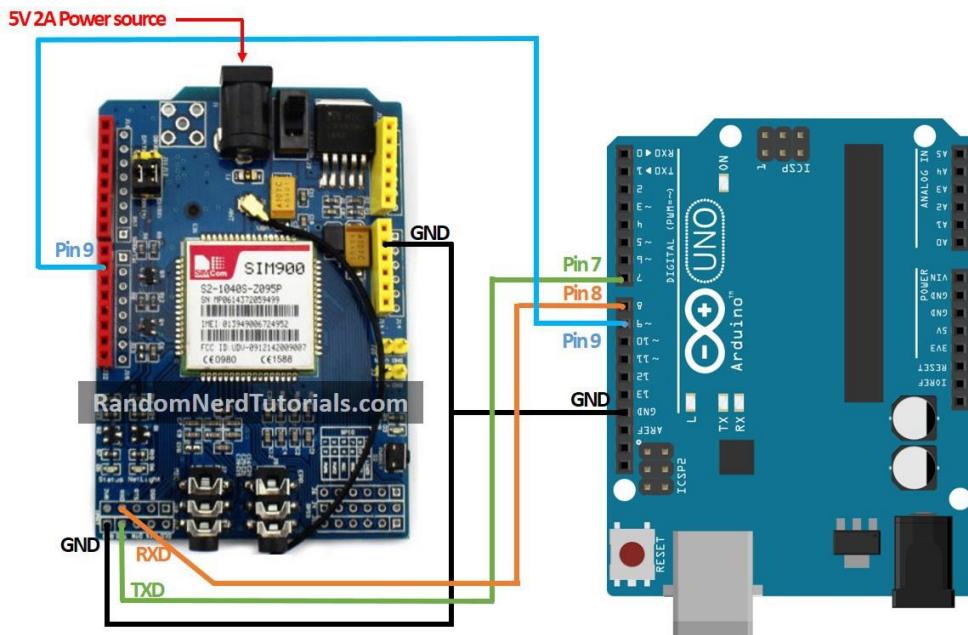
Activar automáticamente el shield

En lugar de pulsar manualmente el botón “Power” para encender en el escudo, puede activar de forma automática en el blindaje a través del software.

- 1) En primer lugar, es necesario soldar R13 conexiones en el escudo, como se muestra en la figura siguiente - resaltados en rojo



- 2) Conecte D9 en el pantallas al terminal D9 Arduino como se muestra en el esquema siguiente.



3) Añadir el siguiente fragmento de código en el preparar() función. Esto es equivalente a presionar el botón del shield “power”.

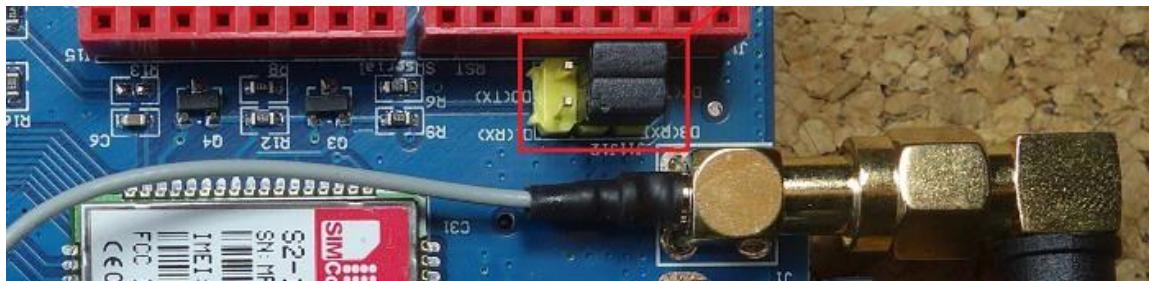
```
digitalWrite(9, HIGH);  
delay(1000);  
digitalWrite(9, LOW);  
delay(5000);
```

Solución de problemas

El shield no responde con OK

Compruebe las conexiones de TX y RX a la Arduino. Trate de repetir el proceso cambiando el TX con los pines RX.

También puedes ver si se ha seleccionado la serie de software mediante la colocación de la tapa del puente en el lugar apropiado en el selector de serie.



No se pueden ver los mensajes en el monitor serie

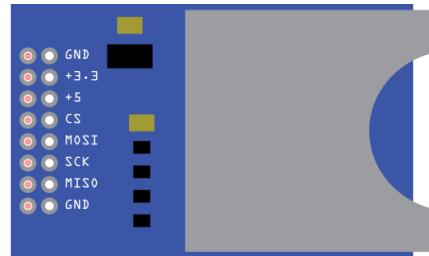
Para ver los mensajes en el monitor serie, el escudo y la velocidad de transmisión del puerto serie del Arduino deben ser los mismos. La velocidad de transmisión por defecto shield SIM900 GSM GPRS 19200. Por lo tanto, seleccione la velocidad de transmisión del Arduino a 19.200.

Sin embargo , si necesita cambiar la velocidad de transmisión escudo , puede enviar el siguiente comando AT para cambiarlo a 19200 u otra velocidad de transmisión adecuada.

```
AT+IPR=19200
```

Terminando

Este tutorial muestra cómo enviar y recibir SMS y hacer y recibir llamadas telefónicas con el Arduino. Puede aplicar los conceptos aprendidos en este tutorial para construir sus propios proyectos para comunicarse a través de una red celular.



Módulo SD Card

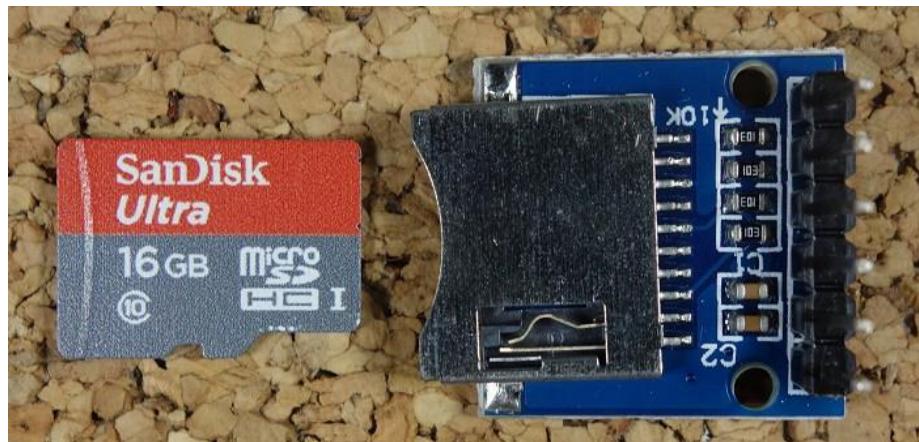
El módulo de tarjeta SD es especialmente útil para proyectos que requieren el registro de datos.

El Arduino se puede crear un archivo en una tarjeta SD para escribir y guardar los datos utilizando el Dakota del Sur biblioteca.

Existen diferentes modelos de diferentes proveedores, pero todos ellos funcionan de manera similar, utilizando el protocolo de comunicación SPI. El módulo utilizado en este tutorial es el mostrado en la figura de abajo (vista frontal y posterior).



Este módulo funciona con la tarjeta micro SD, pero hay otros que funcionan con tarjeta SD.



¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Micro SD card module](#)

Cableado

SD card module	Wiring to Arduino Uno	Wiring to Arduino Mega
VCC	3.3V or 5V (check module's datasheet)	3.3V or 5V (check module's datasheet)
CS	4	53
MOSI	11	51
CLK	13	52
MISO	12	50
GND	GND	GND

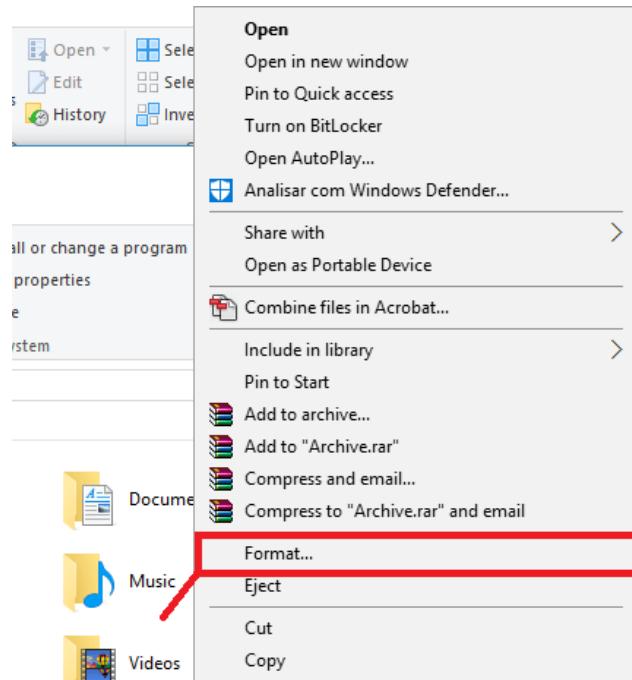
Nota: diferentes placas Arduino tienen diferentes pines SPI. Si está utilizando otra placa Arduino, comprobar la documentación.

Preparación de la tarjeta SD

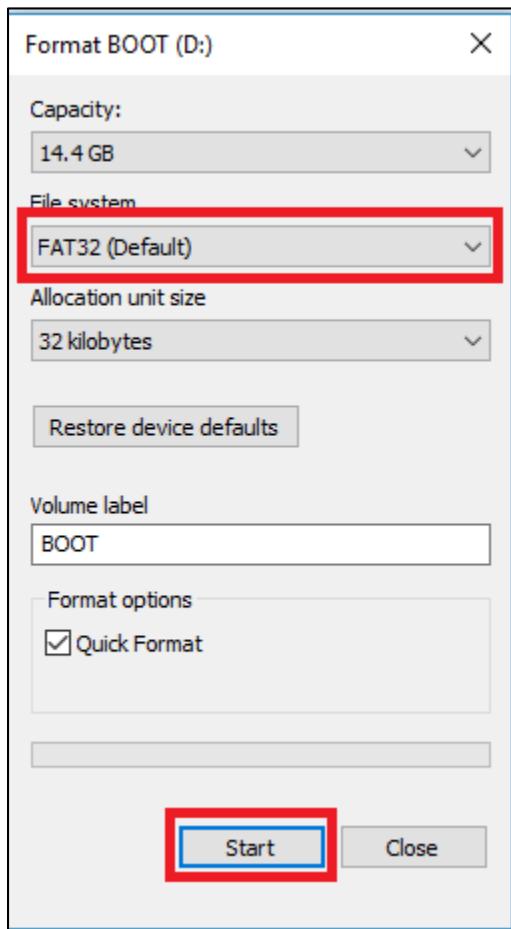
El primer paso cuando se utiliza el módulo de tarjeta SD con Arduino está formateando la tarjeta SD como FAT16 o FAT32.

Sigue las instrucciones de abajo.

1) Para formatear la tarjeta SD, insertarlo en su ordenador. Ir Mi computadora y haga clic derecho en la tarjeta SD. Seleccionar Formato como se muestra en la figura a continuación.



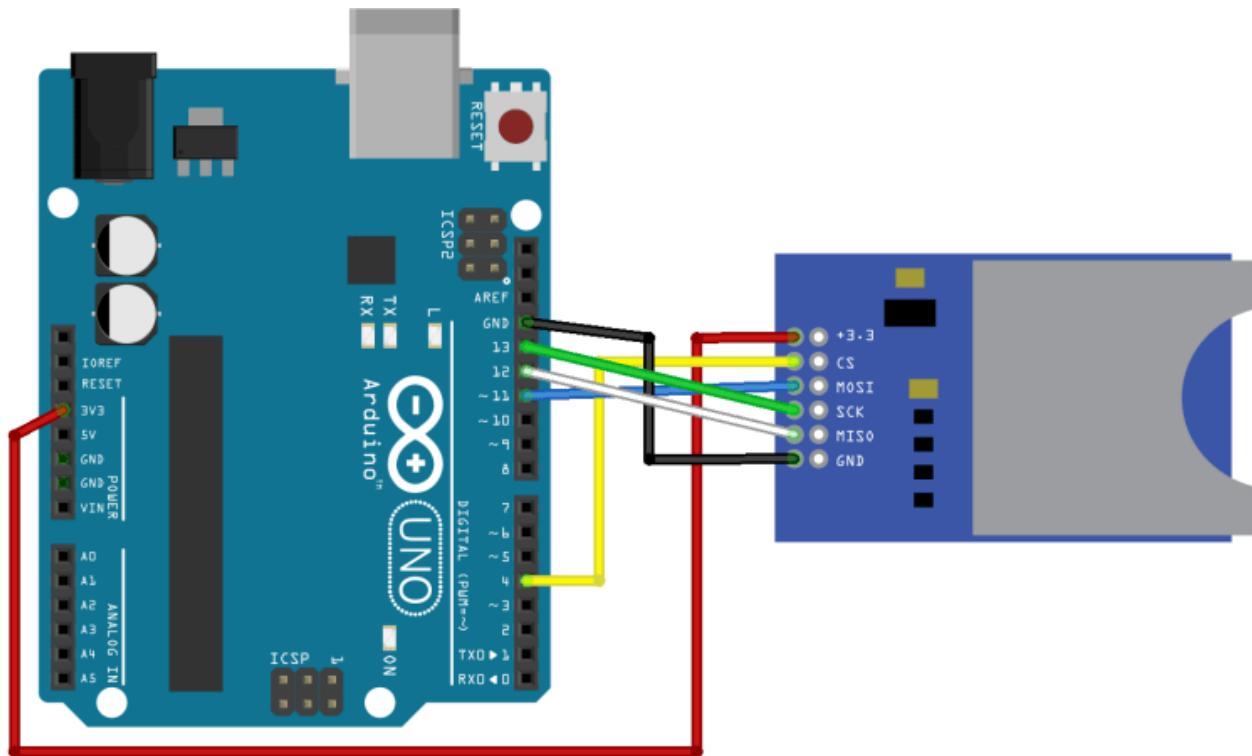
2) Una nueva ventana se abrirá. **Seleccione FAT32**, presiona **comienzo** para iniciar el proceso de formateo y siga las instrucciones en pantalla.



Probando el módulo de tarjeta SD

Inserte la tarjeta SD formateada en el módulo de tarjeta SD.

Conectar el módulo de tarjeta SD para el Arduino como se muestra en el esquema siguiente circuito o detectar cableado de las puntas en la sección anterior.



fritzing

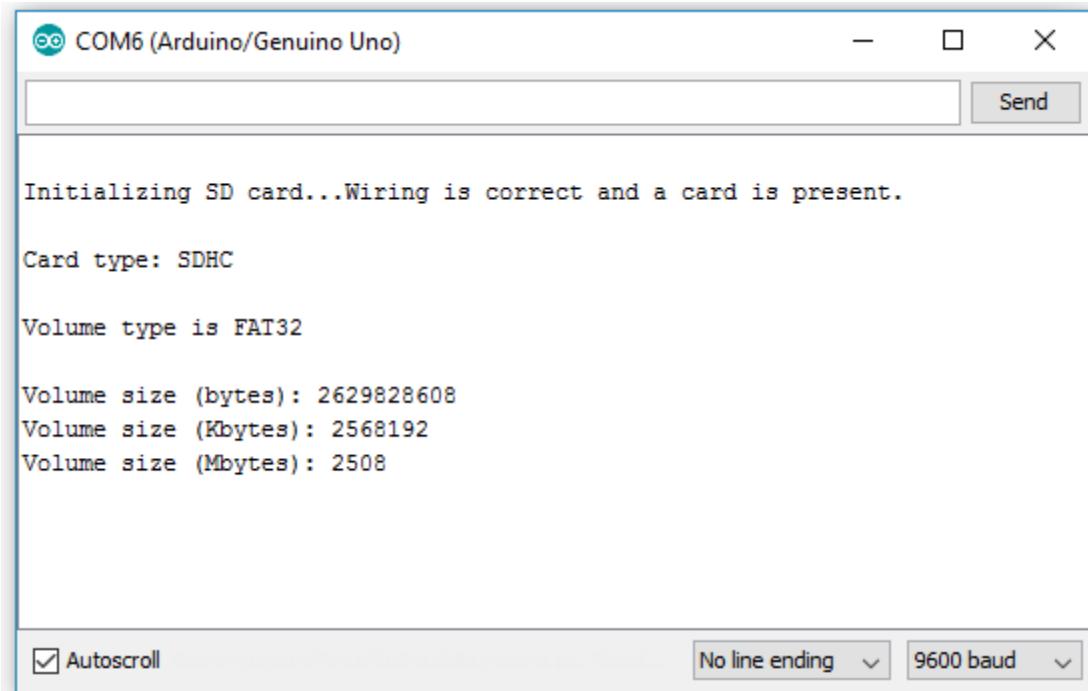
Nota: dependiendo del módulo que está utilizando, los pasadores pueden estar en un orden diferente

Código - cardinfo

Para asegurarse de que todo está conectado correctamente, y la tarjeta SD está funcionando correctamente, en la ventana de Arduino IDE ir a **Archivo • Ejemplos • SD • Cardinfo.**

Sube el código para la placa Arduino. Asegúrate de que tienes el panel a la derecha y el puerto COM seleccionado. Abrir el Monitor de serie a una velocidad de transmisión de 9600 y debería ver la información de su tarjeta SD.

Si todo funciona correctamente verá un mensaje similar en el monitor de serie.



The screenshot shows the Arduino Serial Monitor window titled "COM6 (Arduino/Genuino Uno)". The text output is as follows:

```
Initializing SD card...Wiring is correct and a card is present.  
Card type: SDHC  
Volume type is FAT32  
Volume size (bytes): 2629828608  
Volume size (Kbytes): 2568192  
Volume size (Mbytes): 2508
```

At the bottom, there are checkboxes for "Autoscroll" and "No line ending", and a dropdown for "9600 baud".

Leer y escribir en la tarjeta SD

La biblioteca SD proporciona funciones útiles para escribir fácilmente en y leer desde la tarjeta SD. Para escribir y leer desde la tarjeta SD, primero es necesario incluir la SPI y SD bibliotecas:

```
#include <SPI.h>  
  
#include <SD.h>
```

Usted también tiene que inicializar el módulo de tarjeta SD en la Selección de Chip (CS) pin - en nuestro caso, el pin 4.

```
SD.begin(4);
```

Para abrir un nuevo archivo en la tarjeta SD, es necesario crear un objeto de archivo que hace referencia al archivo de datos. Por ejemplo:

```
dataFile = SD.open("data.txt", FILE_WRITE);
```

El primer parámetro de esta función es el nombre del archivo, datos.txt , y el FILE_WRITE argumento que permite leer y escribir en el archivo.

Esta línea de código crea un archivo llamado datos.txt en su tarjeta SD. Si el datos.txt archivo ya existe, el Arduino se abrirá el archivo en lugar de crear otro.

Para escribir datos en el archivo abierto, se utiliza:

```
dataFile.write(data);
```

En el que la archivo de datos es el objeto de archivo creado con anterioridad y la datos es lo que se desea escribir en el archivo.

También puede utilizar el impresión() o println () funciones para imprimir datos en el archivo:

```
dataFile.print(data);  
dataFile.println(data); // followed by a new line
```

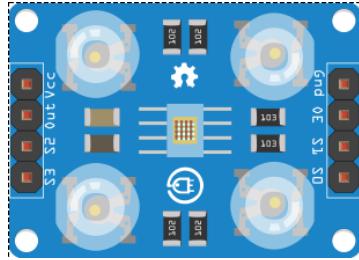
Para leer los datos guardados en el archivo:

```
dataFile.read();
```

Sólo se puede escribir dentro de un archivo a la vez, por lo que necesita cerrar un archivo antes de pasar a la siguiente. Para cerrar el archivos datos.txt que acabamos de crear:

```
SD.close("data.txt");
```

El argumento de esta función es el archivo que desea cerrar, en este caso datos.txt . Para un boceto completa sobre cómo leer y escribir, en su IDE Arduino ir a **Archivo • Ejemplos • SD • Leer • escribir.**



Sensor de color TCS3200

El sensor de color TCS3200 puede detectar una amplia variedad de colores en función de su longitud de onda . Este sensor es especialmente útil para los proyectos de reconocimiento de color, tales como la coincidencia de colores , la clasificación por color, la lectura de tiras reactivas y mucho más.

El sensor de color TCS 3200 - se muestra en la figura de abajo - utiliza un chip sensor TAOS TCS 3200 RGB para detectar color . También contiene cuatro LEDs blancos que iluminan el objeto delante de él.



Presupuesto

- Power: 2.7V to 5.5V
- Size: 28.4 x 28.4mm (1.12 x 1.12")
- Interface: digital TTL
- High-resolution conversion of light intensity to frequency
- Programmable color and full-scale output frequency
- Communicates directly to microcontroller

¿Donde comprar?

Haga clic en el enlace de abajo para encontrar el mejor precio :

- [Color Sensor TCS3200](#)

¿Cómo funciona el sensor de TCS3200?

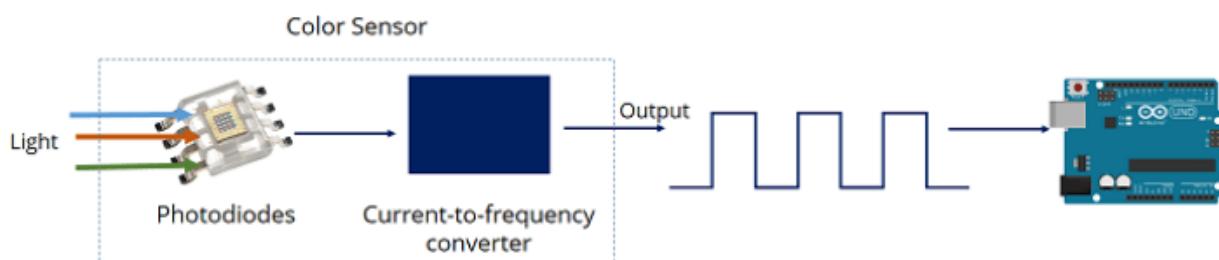
El TCS3200 tiene una serie de fotodiodos con 4 filtros diferentes. Un fotodiodo es simplemente un dispositivo semiconductor que convierte la luz en corriente. El sensor tiene:

- 16 fotodiodos con filtro rojo - sensibles a la longitud de onda roja
- 16 fotodiodos con filtro verde - sensibles a la longitud de onda verde
- 16 fotodiodos con filtro azul - sensibles a la longitud de onda azul
- 16 fotodiodos sin filtro

Si se echa un vistazo más de cerca el chip TCS3200 se puede ver los diferentes filtros.

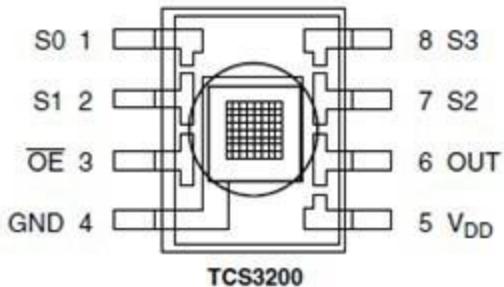


Al elegir selectivamente las lecturas del filtro fotodiodo, que es capaz de detectar la intensidad de los diferentes colores. El sensor tiene un convertidor de corriente a frecuencia que convierte las lecturas de los fotodiodos en una onda cuadrada con una frecuencia que es proporcional a la intensidad de la luz del color elegido. Esta frecuencia es entonces, leído por el Arduino - esto se muestra en la figura siguiente.



Cableado

Aquí está el pinout del sensor:



Sensor Pin	I/O	Description
GND (4)		Power supply ground
OE (3)	I	Enable for output frequency (active low)
OUT (6)	O	Output frequency
S0, S1 (1, 2)	I	Output frequency scaling selection inputs
S2, S3 (7, 8)	I	Photodiode type selection inputs
VDD (5)		Voltage supply

Selección de filtros

Para seleccionar el color leído por el fotodiodo, utiliza los pines de control S2 y S3. A medida que los fotodiodos están conectados en paralelo, el establecimiento de la S2 y S3 LOW y HIGH en diferentes combinaciones le permite seleccionar diferentes fotodiodos. Mira la siguiente tabla:

Photodiode type	S2	S3
Red	LOW	LOW
Blue	LOW	HIGH
No filter (clear)	HIGH	LOW
Green	HIGH	HIGH

Escalado de frecuencia

Pasadores S0 y S1 se utilizan para escalar la frecuencia de salida. Se puede escalar a los siguientes valores predefinidos : 100%, 20% o 2%. Escala de la frecuencia de salida es útil para optimizar las lecturas de los sensores para varios contadores de frecuencia o microcontroladores. Utilice la siguiente tabla:

Output frequency scaling	S0	S1
Power down	L	L
2%	L	H
20%	H	L
100%	H	H

Para el Arduino, es común el uso de una escala de frecuencia de 20%. Así, se establece el pasador de S0 a HIGH y el pasador de S1 a LOW.

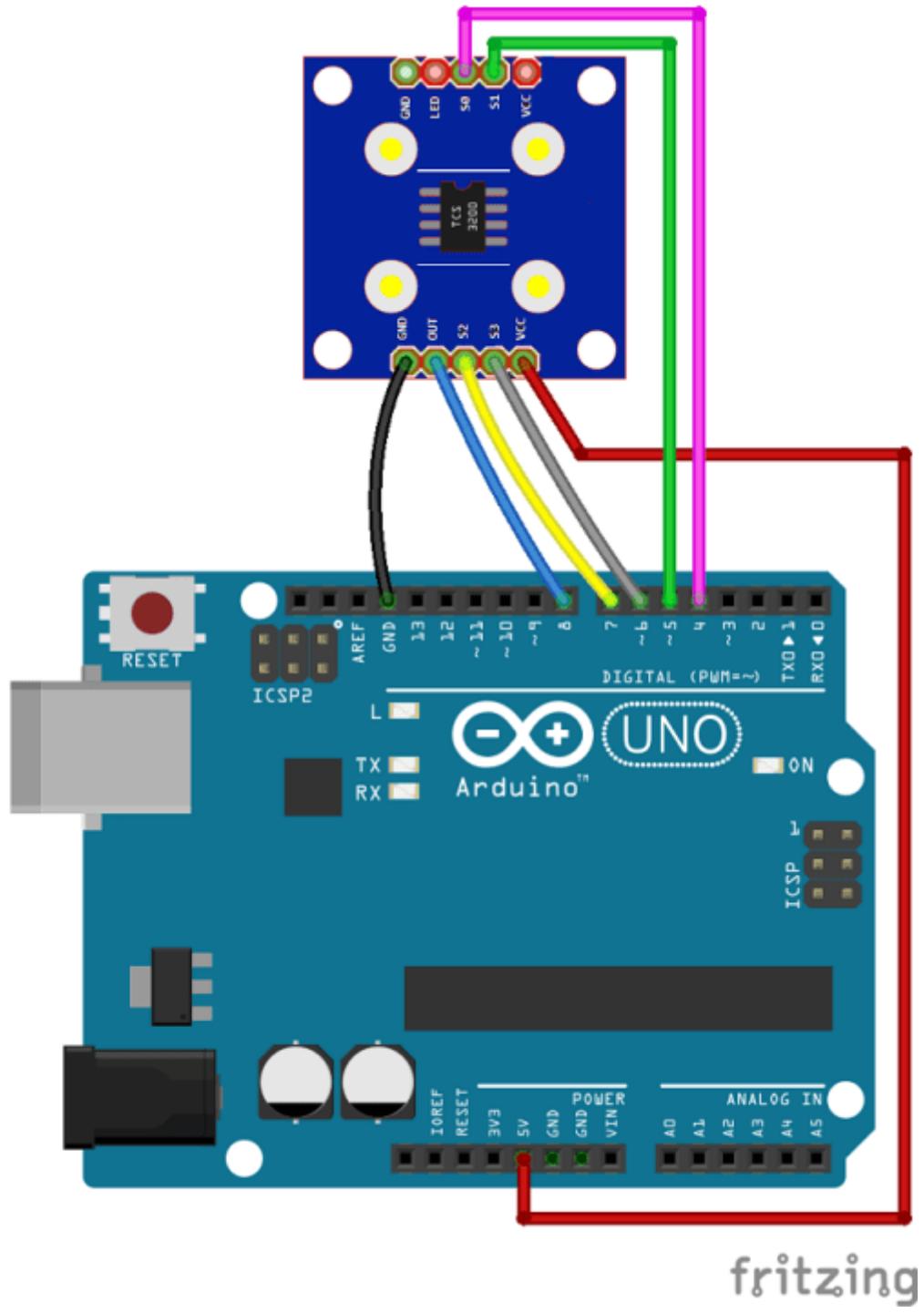
Detección de color con Arduino y TCSP3200

En este ejemplo se va a detectar colores con el Arduino y el sensor de color TCSP 3200 . Este sensor no es muy precisa , pero funciona bien para la detección de colores en proyectos simples. Para este ejemplo, tendrá las siguientes partes:

Figure	Name	Check Price
	Arduino UNO	Find best price on Maker Advisor
	TCS3200 Color Sensor	Find best price on Maker Advisor
	Jumper Wires	Find best price on Maker Advisor

Circuito

Cableado del sensor de TCSP3200 a tu Arduino es sencillo. Sólo tiene que seguir el siguiente diagrama esquemático.



Cableado

He aquí las conexiones entre el TCSP3200 y el Arduino:

Sensor Pin	Wiring to Arduino Uno
S0	Digital pin 4

S1	Digital pin 5
VCC	5V
S3	Digital pin 6
S4	Digital pin 7
OUT	Digital pin 8

Código

Se necesitan dos bocetos para este proyecto:

1. La lectura y la visualización de la frecuencia de salida en el monitor de serie. En esto pieza que necesita para anotar los valores de frecuencia cuando se coloca diferentes colores en frente del sensor.
2. Distinguir entre diferentes colores. En esta sección se va a insertar los valores de frecuencia recogidos previamente en su código, por lo que su sensor puede distinguir entre diferentes colores. Detectaremos los colores rojo, verde y azul.

1. Lectura de la frecuencia de salida

Sube el siguiente código a la placa Arduino.

```
*****
Rui Santos
Complete project details at http://randomnerdtutorials.com
*****/

// TCS230 or TCS3200 pins wiring to Arduino
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;

void setup() {
    // Setting the outputs
```

```

pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);

// Setting the sensorOut as an input
pinMode(sensorOut, INPUT);

// Setting frequency scaling to 20%
digitalWrite(S0,HIGH);
digitalWrite(S1,LOW);

// Begins serial communication
Serial.begin(9600);
}

void loop() {
    // Setting RED (R) filtered photodiodes to be read
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);

    // Reading the output frequency
    redFrequency = pulseIn(sensorOut, LOW);

    // Printing the RED (R) value
    Serial.print("R = ");
    Serial.print(redFrequency);
    delay(100);

    // Setting GREEN (G) filtered photodiodes to be read
    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);

    // Reading the output frequency
    greenFrequency = pulseIn(sensorOut, LOW);

    // Printing the GREEN (G) value
    Serial.print(" G = ");
    Serial.print(greenFrequency);
    delay(100);

    // Setting BLUE (B) filtered photodiodes to be read
    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);

    // Reading the output frequency
    blueFrequency = pulseIn(sensorOut, LOW);
}

```

```
// Printing the BLUE (B) value  
Serial.print(" B = ");  
Serial.println(blueFrequency);  
delay(100);  
}
```

{;}

CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/color/Color_Sensor_TCS230_TCS3200_1.ino

Abrir el monitor de puerto serie a una velocidad de transmisión de 9600.

Coloque un objeto azul delante del sensor a diferentes distancias. Debe guardar dos mediciones: cuando el objeto se coloca lejos del sensor y cuando el objeto está cerca de él.



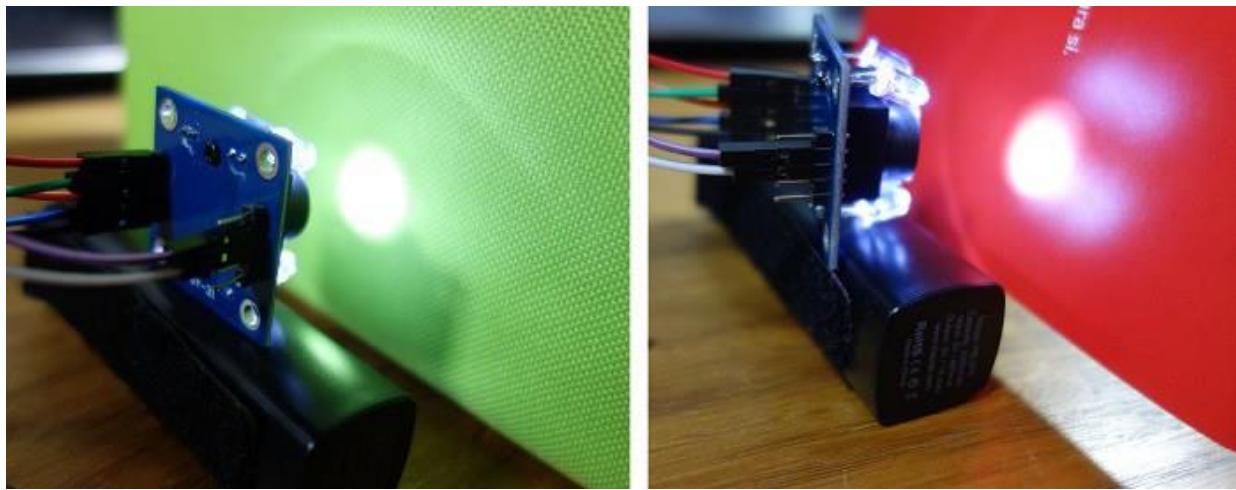
Compruebe los valores que se muestran en el monitor de serie. La frecuencia de azul (B) debe ser el más bajo en comparación con el rojo (R) y las lecturas de frecuencia verde (G) - véase la figura a continuación.

```
R = 1141 G = 590 B = 178
R = 1182 G = 626 B = 191
R = 1289 G = 645 B = 197
R = 1315 G = 664 B = 209
R = 1381 G = 719 B = 223
R = 1401 G = 707 B = 219
R = 1425 G = 733 B = 220
R = 1395 G = 722 B = 209
R = 1275 G = 627 B = 193
R = 1221 G = 619 B = 177
R = 1018 G = 486 B = 141
R = 842 G = 449 B = 145
R = 932 G = 473 B = 148
R = 904 G = 482 B = 146
R = 431 G = 235 B = 73
R = 432 G = 263 B = 75
R = 393 G = 220 B = 63
R = 367 G = 207 B = 65
R = 369 G = 210 B = 62
R = 322 G = 195 B = 59
R = 329 G = 207 B = 63
```

Cuando ponemos el objeto azul delante del sensor, los valores de frecuencia azul (B) oscilan entre 59 y 223 (ver los valores resaltados).

Nota: no puede utilizar estos valores de frecuencia (59 y 223) en su código, usted debe medir los colores para su objeto específico con su propio sensor de color. A continuación, guardar los límites de frecuencia superior e inferior para el color azul, ya que necesitará más adelante.

Repita este proceso con un verde y rojo objetos y anotar los límites de frecuencia superior e inferior para cada color.



2. Distinguir entre diferentes colores

Este próximo boceto mapea los valores de frecuencia a valores RGB (que se encuentran entre 0 y 255).

En el paso anterior la frecuencia máxima de azul como 233 y el mínimo 59. Por lo tanto, 59 en frecuencia corresponde a 255 (en RGB) y 233 en frecuencia a 0 (en RGB). Haremos esto con el Arduino `map()` función. En el `map()` función que necesita para reemplazar los parámetros XX con sus propios valores.

```
*****
Rui Santos
Complete project details at http://randomnerdtutorials.com
*****


// TCS230 or TCS3200 pins wiring to Arduino
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
// Stores the red, green and blue colors
int redColor = 0;
int greenColor = 0;
int blueColor = 0;

void setup() {
    // Setting the outputs
    pinMode(S0, OUTPUT);
    pinMode(S1, OUTPUT);
    pinMode(S2, OUTPUT);
    pinMode(S3, OUTPUT);

    // Setting the sensorOut as an input
    pinMode(sensorOut, INPUT);

    // Setting frequency scaling to 20%
    digitalWrite(S0, HIGH);
    digitalWrite(S1, LOW);

    // Begins serial communication
    Serial.begin(9600);
```

```

}

void loop() {
    // Setting RED (R) filtered photodiodes to be read
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);

    // Reading the output frequency
    redFrequency = pulseIn(sensorOut, LOW);
    // Remaping the value of the RED (R) frequency from 0 to 255
    // You must replace with your own values. Here's an example:
    // redColor = map(redFrequency, 70, 120, 255,0);
    redColor = map(redFrequency, XX, XX, 255,0);

    // Printing the RED (R) value
    Serial.print("R = ");
    Serial.print(redColor);
    delay(100);

    // Setting GREEN (G) filtered photodiodes to be read
    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);

    // Reading the output frequency
    greenFrequency = pulseIn(sensorOut, LOW);
    // Remaping the value of the GREEN (G) frequency from 0 to 255
    // You must replace with your own values. Here's an example:
    // greenColor = map(greenFrequency, 100, 199, 255, 0);
    greenColor = map(greenFrequency, XX, XX, 255, 0);

    // Printing the GREEN (G) value
    Serial.print(" G = ");
    Serial.print(greenColor);
    delay(100);

    // Setting BLUE (B) filtered photodiodes to be read
    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);

    // Reading the output frequency
    blueFrequency = pulseIn(sensorOut, LOW);
    // Remaping the value of the BLUE (B) frequency from 0 to 255
    // You must replace with your own values. Here's an example:
    // blueColor = map(blueFrequency, 38, 84, 255, 0);
    blueColor = map(blueFrequency, XX, XX, 255, 0);

    // Printing the BLUE (B) value
}

```

```

Serial.print(" B = ");
Serial.print(blueColor);
delay(100);
// Checks the current detected color and prints
// a message in the serial monitor
if(redColor > greenColor && redColor > blueColor){
    Serial.println(" - RED detected!");
}
if(greenColor > redColor && greenColor > blueColor){
    Serial.println(" - GREEN detected!");
}
if(blueColor > redColor && blueColor > greenColor){
    Serial.println(" - BLUE detected!");
}
}

```

{;} CÓDIGO FUENTE

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/color/Color_Sensor_TCS230_TCS3200_2.ino

Para distinguir entre diferentes colores que tenemos tres condiciones:

- Cuando el R es el valor máximo (en parámetros RGB) sabemos que tenemos un objeto rojo
- Cuando G es el valor máximo, sabemos que tenemos un objeto verde
- Cuando B es el valor máximo, sabemos que tenemos un objeto azul

Ahora, colocar algo delante del sensor. Se debe imprimir en su monitor serie detecta el color: rojo, verde o azul. Su sensor también puede detectar otros colores con más sentencias if.

Terminando

En este post que ha aprendido cómo detectar los colores con el sensor de color TCSP3200. Usted puede construir fácilmente una máquina de clasificación por color , simplemente añadiendo un servomotor.