

PRACTICA 4

Material:

NodeMCU con cable USB
Sensor LDR
Resistencia de 1K Ohmios
Resistencia 330 Ohmios

SENSORES ANALÓGICOS:

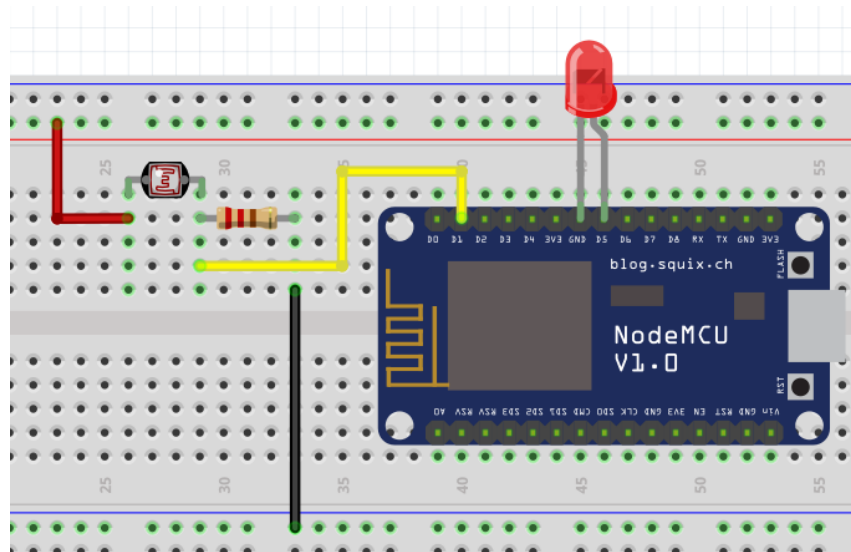
¿QUÉ ES UN FOTORESISTOR LDR?

Un fotoresistor, o LDR (light-dependent resistor) es un dispositivo cuya resistencia varía en función de la luz recibida. Podemos usar esta variación para medir, a través de las entradas analógicas (A0), una estimación del nivel de la luz.

Un fotoresistor disminuye su resistencia a medida que aumenta la luz sobre él. Los valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante.

Por otro lado, la variación de la resistencia es relativamente lenta, de 20 a 100 ms en función del modelo. Esta lentitud hace que no sea posible registrar variaciones rápidas, como las producidas en fuentes de luz artificiales alimentadas por corriente alterna. Este comportamiento puede ser beneficioso, ya que dota al sensor de una gran estabilidad. Por tanto, un LDR es un sensor que resulta adecuado para proporcionar medidas cuantitativas sobre el nivel de luz, tanto en interiores como en exteriores, y reaccionar, por ejemplo, encendiendo una luz, subiendo una persiana, u orientando un robot.

Ejercicio 1: Vamos a utilizar una resistencia de 1Kohmio y un fotoresistencia o LDR conectada en el pin GPIO05 para medir la luz y haremos un circuito para que al disminuir el nivel de luz medida por el LDR, se encienda un LED conectado en el puerto GPIO14.



El valor de resistencia varía dependiendo del nivel que quieras medir pero puedes empezar con una resistencia de 1Kohmio. Realice el código correspondiente utilice el monitor serial del IDE de arduino para verificar el valor en el puerto A0 correspondiente a la medida del LDR. Establezca un valor en la medida de A0 para que se encienda el LED y varíe el valor de la resistencia de ser necesario.

EL CICLO WHILE

Habrán ocasiones en las que querremos que realice acciones mientras se cumpla una condición sin importar el número de veces repita: para ello contamos con el comando while.

```
while (condición) {  
  // comandos  
}
```

```
int pulsador = 12;  
int led = 4;  
  
void setup()  
{  
  pinMode(pulsador, INPUT);  
  pinMode(led, OUTPUT);  
}  
  
void loop()  
{  
  while (digitalRead(pulsador) == HIGH)  
  {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
  }  
}
```

Ejercicio 2: Realizar el ejercicio 1 con el comando while. Utilizando una fotorresistencia o LDR para medir la luz, encienda un LED a cierto nivel.

SENSORES DIGITALES:

Todo proyecto de IoT se compone principalmente de dos partes. Por un lado, la parte de Internet, su conexión con la base de datos la API (interfaz de programación de aplicaciones). Y por otro, la parte de las cosas, es decir, los sensores que nos darán la información que subir a Internet. Tenemos infinitos tipos distintos de sensores, entre ellos:

- **Sensores de temperatura:** DS18B2, LM35, DHT11...
- **Sensores de humedad ambiental:** DHT11, DHT22
- **Sensores de luminosidad:** BH1750, LDR
- **Detectores de movimiento:** HC-SR501
- **Medidores de distancia:** HC-SR04

Y muchos otros más...

Encontraremos sensores que nos dan la información en digital para poder tratarla directamente con un microcontrolador y sensores que nos darán las lecturas en formato analógico (como la práctica anterior), por lo que para tratarlas con un micro requeriremos de un convertidor de tipo ADC (Analog To Digital).

SENSOR DHT11

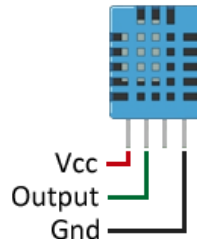
El DHT11 y el DHT22 son dos modelos de una misma familia de sensores, que permiten realizar la medición simultánea de temperatura y humedad.

Estos sensores disponen de un procesador interno que realiza el proceso de medición, proporcionando la medición mediante una señal digital, por lo que resulta muy sencillo obtener la medición desde un microprocesador como el que tiene la placaNodeMCU.

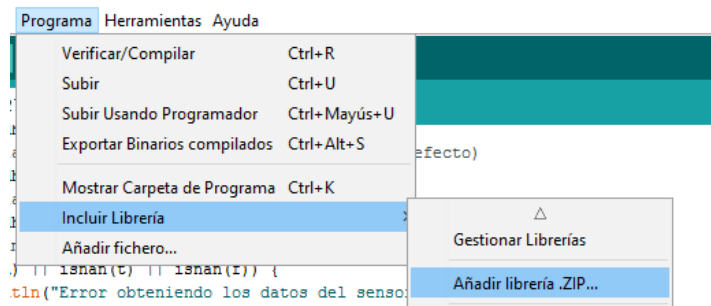
De ambos modelos, el DHT11 es el hermano pequeño de la familia, y cuenta peores características técnicas. El DHT22 es el modelo superior pero, por contra, tiene un precio superior. Las características del DHT11 son realmente escasas, especialmente en rango de medición y precisión.

- Medición de temperatura entre 0 a 50, con una precisión de 2°C
- Medición de humedad entre 20 a 80%, con precisión del 5%.
- Frecuencia de muestreo de 1 muestras por segundo (1 Hz)
- El DHT11 es un sensor muy limitado que podemos usar con fines de formación, pruebas, o en proyectos que realmente no requieran una medición precisa.

Al ser digital y funcionar con el protocolo OneWire, solo necesitamos un solo cable para realizar las medidas tanto de temperatura como de humedad relativa del aire.



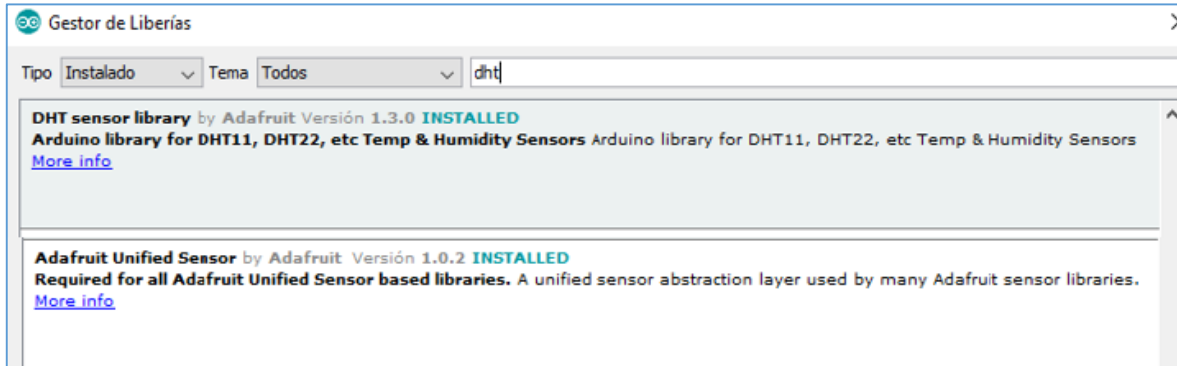
Para facilitar la lectura de variables que nos da el sensor existen varias librerías de Arduino que nos serán de gran ayuda. Estas librerías también son compatibles con el ESP8266.



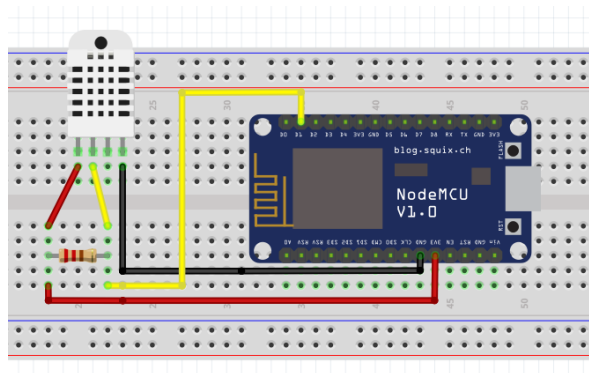
<https://github.com/winlinvip/SimpleDHT>

https://github.com/adafruit/Adafruit_Sensor

Se deberán descargar en zip e instalar estas dos librerías. Puedes revisar desde el gestor de librerías del IDE de arduino para ver si ya las tenemos instaladas.



Ejercicio 3: Implementar un sistema de medida de temperatura y humedad a través del NodeMCU, una resistencia de 330 ohmios y un sensor de temperatura y humedad dht11.



Existen sensores de temperatura DHT11 que traen solo 3 pines y ya tienen la resistencia pull-up incluida de 330 ohmios. Si tienes un sensor con la resistencia ya incluida no necesitas la resistencia del diagrama. En cualquiera de los casos solo se utilizan tres pines, el primero debe ir conectado a la salida de nuestra palca, el segundo lleva los datos de humedad y temperatura y el último va conectado a tierra o GND.

sketch_jun27a

```
#include <DHT.h>           // Incluimos librería
#define DHTPIN 5           // Definimos el pin digital donde se conecta el sensor
#define DHTTYPE DHT11      // Dependiendo del tipo de sensor

DHT dht(DHTPIN, DHTTYPE);  // Inicializamos el sensor DHT11
void setup() {

  Serial.begin(115200);      // Inicializamos comunicación serie
  dht.begin();              // Comenzamos el sensor DHT
}

void loop() {

  delay(2000);              // Esperamos 5 segundos entre medidas
  float h = dht.readHumidity(); // Leemos la humedad relativa
  float t = dht.readTemperature(); // Leemos la temperatura en grados centígrados (por defecto)
  if (isnan(h) || isnan(t)) { // Comprobamos si ha habido algún error en la lectura
    Serial.println("Error obteniendo los datos del sensor DHT11");
    return;
  }
  Serial.print("Humedad: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.println(" *C ");
}
```

COM3

Humedad: 43.00 %	Temperatura: 28.00 *C
Humedad: 43.00 %	Temperatura: 28.00 *C
Humedad: 47.00 %	Temperatura: 28.00 *C
Humedad: 55.00 %	Temperatura: 28.00 *C
Humedad: 43.00 %	Temperatura: 28.00 *C
Humedad: 42.00 %	Temperatura: 26.00 *C
Humedad: 43.00 %	Temperatura: 28.00 *C
Humedad: 48.00 %	Temperatura: 28.00 *C
Humedad: 43.00 %	Temperatura: 28.00 *C
Humedad: 42.00 %	Temperatura: 28.00 *C
Humedad: 40.00 %	Temperatura: 32.00 *C
Humedad: 44.00 %	Temperatura: 28.00 *C
Humedad: 51.00 %	Temperatura: 28.00 *C

☒ Autoscroll

El código utilizado en este caso me permite visualizar a través del monitor serie el valor de humedad y temperatura registrado por el DHT11.

Si utiliza el NodeMCU v1 puedes conectar el sensor a los 3.3v de la placa y si tienes la v3, puedes conectar el sensor a los 3.3 o los 5Vdc de la placa sin problema ya que el rango del sensor lo permite.

TRABAJO PARA CASA