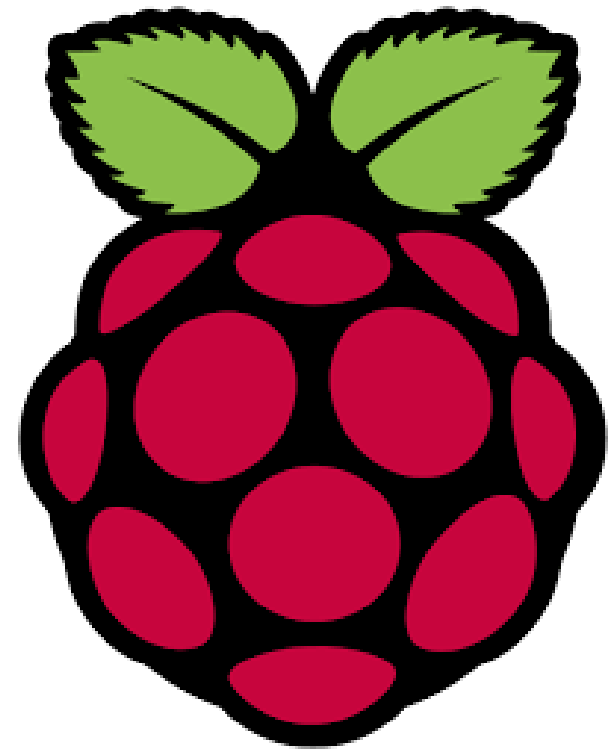
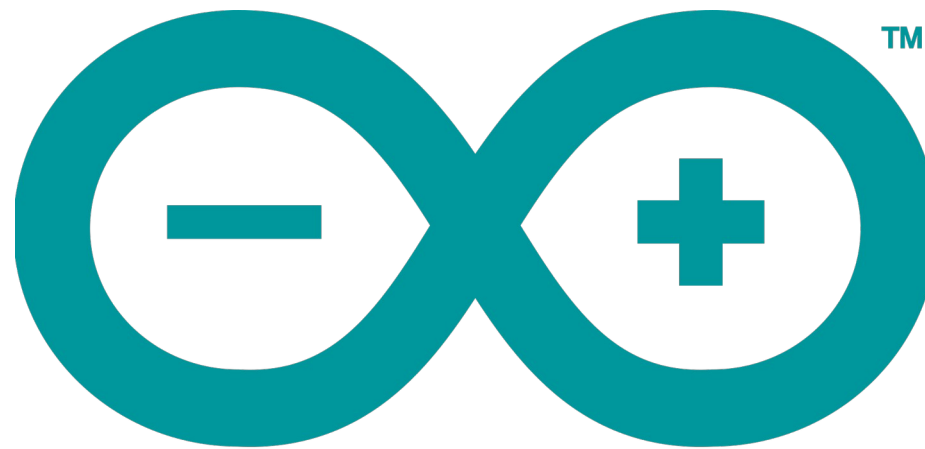


Aplicacións de Arduino e Raspberry pi en domótica

Setembro 2019

Manuel Guimarey Vila
Rocío Martínez Martínez



CFR
VIGO

centro de
formación e recursos

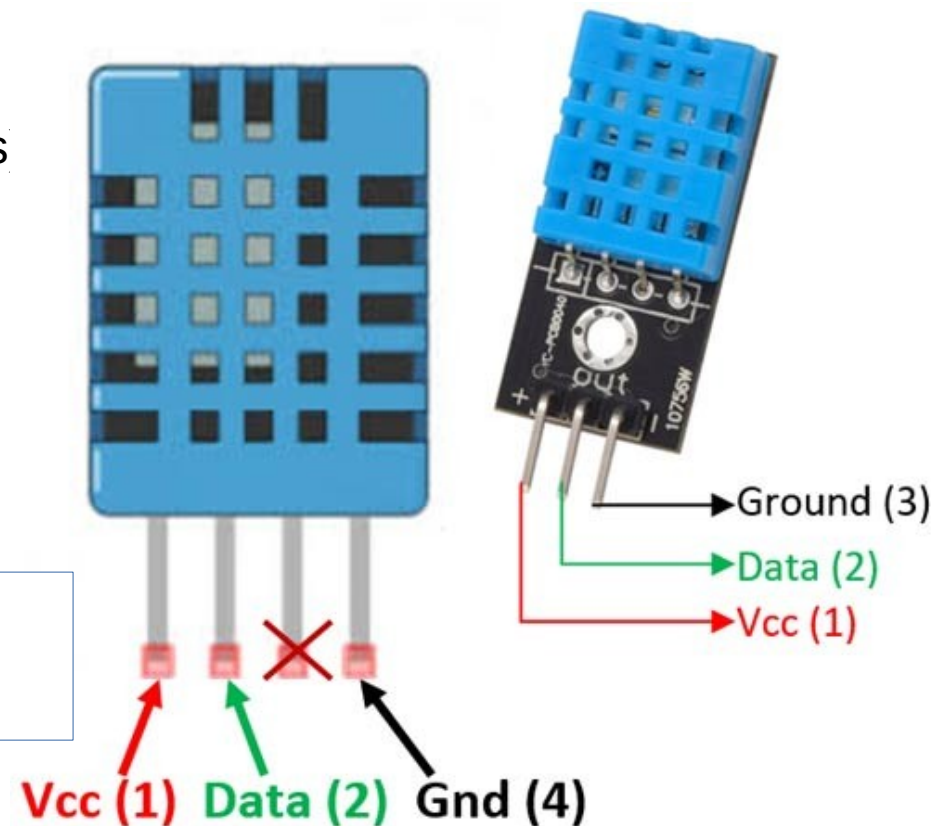
Sensores e actuadores

- Sensor DHT11
- Sensor Ultrasóns HC-SR04
- Sensor Magnético (reed)
- Bucina.
- Lector tarxetas RFID.
- Sensor de barreira IR.
- Sensor PIR HC-SR501
- Sensor 230 V a través de relé bobina 230 V
- Sensor Luminosidade BH1750
- Relé bobina 5 V e contactos 230 V
- Motor paso a paso
- Sensores de gases MQ-2 e MQ-9
- Sensor choiva
- Sensor humidade do chan



Sensor de humidade e temperatura DHT11

- Na maqueta da vivenda executada en Ponteareas empregamos 4 sensores de temperatura diferentes: Os DHT11, os DHT22, os TMP36 e os LM35DZ. Neste curso empregaremos o DHT11. Comentamos diferenzas básicas
- As características do DHT11 son:
 - Bastante barato, sobre 3€
 - Funciona con 3,3 e 5V de alimentación
 - Rango de temperatura: de 0° a 50° con 5% de precisión (pero só mide por grados, non fraccións)
 - Rango de humidade: do 20% ó 80% con 5% de precisión
 - 1 Mostra por segundo
 - Baixo consumo
 - Devolve a medida en °C

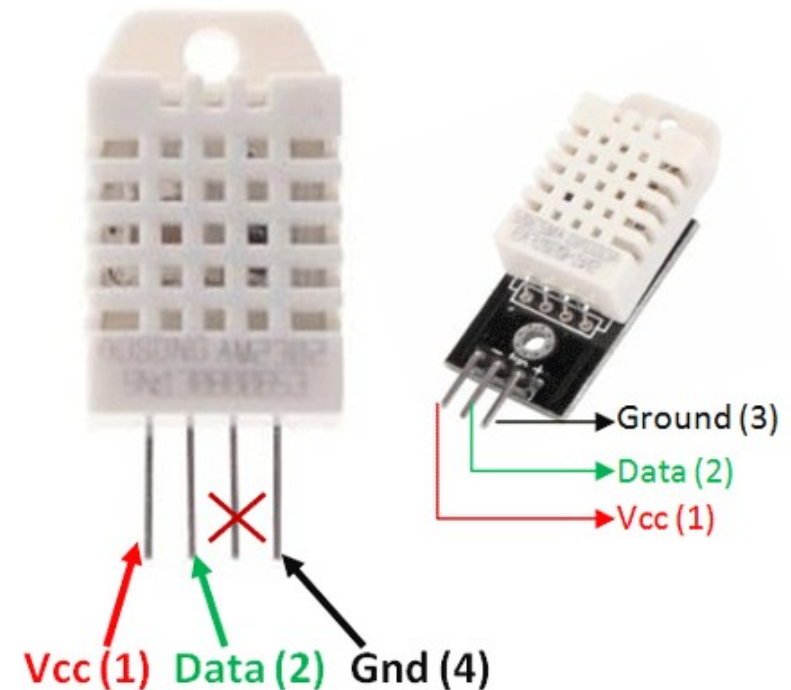


CFR
VIGO

centro de
formación e recursos

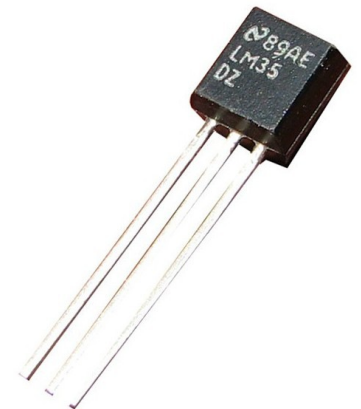
Sensor de humidade e temperatura DHT11

- En canto ó DHT22:
- Barato, entre 6 e 7 €
- Funciona con 3,3 e 5V de alimentación
- Rango de temperatura:
de -40° a $125^{\circ} \pm 0.5^{\circ}\text{C}$
- Rango de humidade: do 0% ó 100%
con 5% de precisión.
- Le 2 veces por segundo.
- Baixo consumo.
- Devolve a medida en $^{\circ}\text{C}$



Sensor de humidade e temperatura DHT11

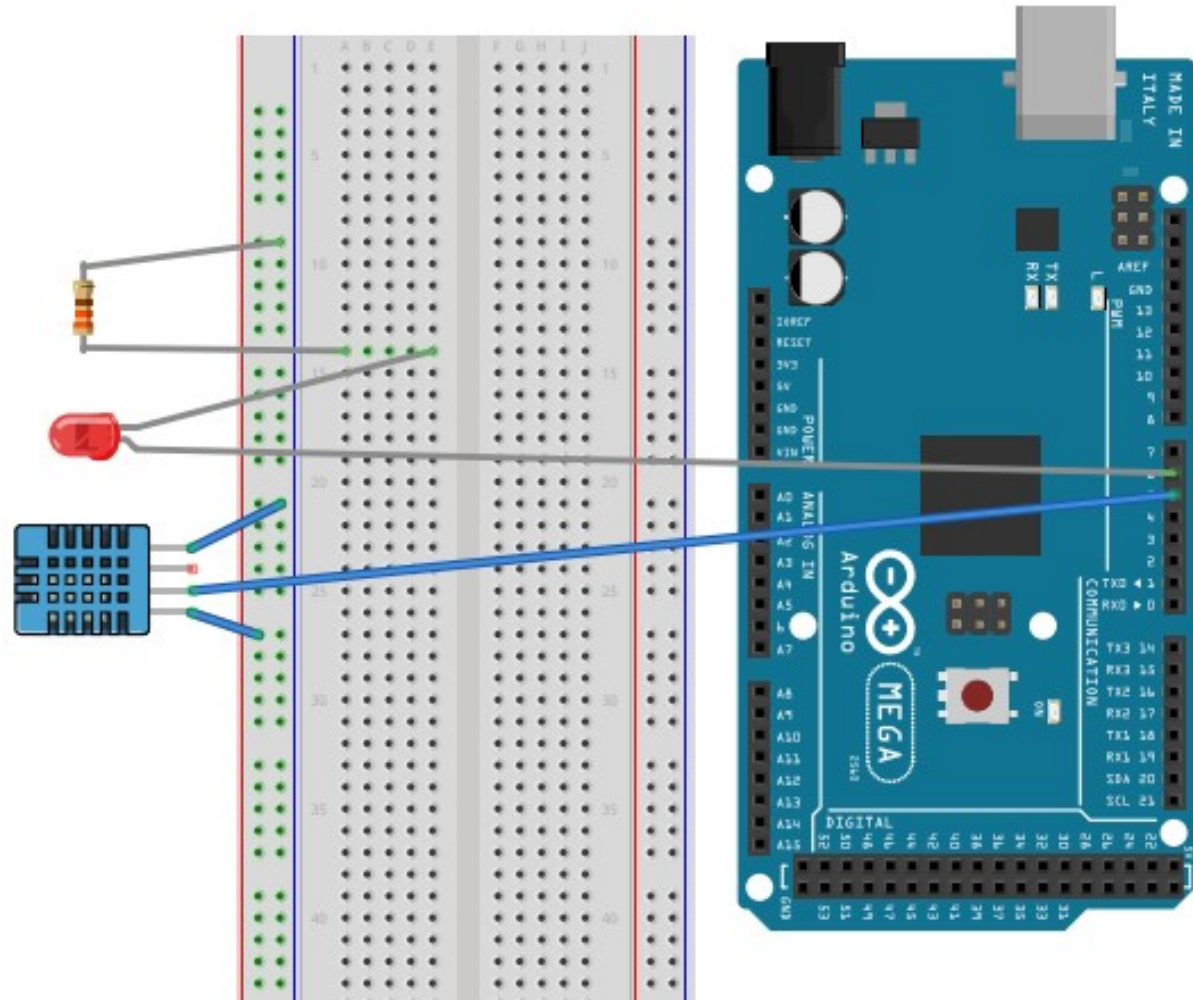
- Os sensores TMP36 e LM35DZ:
- Miden a temperatura en graos centígrados.
- Funcionan entre -50°C e 125°C para o TMP36.
- Funciona entre 0°C y 100°C para o LM35DZ .
- Non son especialmente precisos, xa que teñen $\pm 1^{\circ}\text{C}$ de incertidume, pero normalmente sobranos para proxectos sinxelos e son os máis económicos.
- O encapsulado é semellante ó dun transistor e tamén ten tres patas, así que moito ollo non confundirlos. Intentade ler as letras que leva serigrafiadas.



Sensor de humidade e temperatura

DHT11. Montaxe a executar nº 1

- Empregaremos un arduino mega, pero poderíamos facelo co arduino uno tamén
- Conectaremos o sinal do DHT11 no pin 5 por exemplo, obtendo a temperatura da estancia e a humidade relativa
- Mostraremos os valores no monitor serie de arduino
- Tomaremos man dun led indicador de alarma para avisos de superación da temperatura umbral, que conectaremos no pin 6 por exemplo



CFR
VIGO

centro de
formación e recursos

Sensor de humedad e temperatura

DHT11. Programa nº 1

- ¿Quen se atreve?
- Cousas que debemos lembrar ou retomar, que nos poden axudar:
- Programación. Estructura básica

```
//Declaración de variables
```

```
void setup() {
```

```
//Configuración inicial
```

```
}
```

```
void loop() {
```

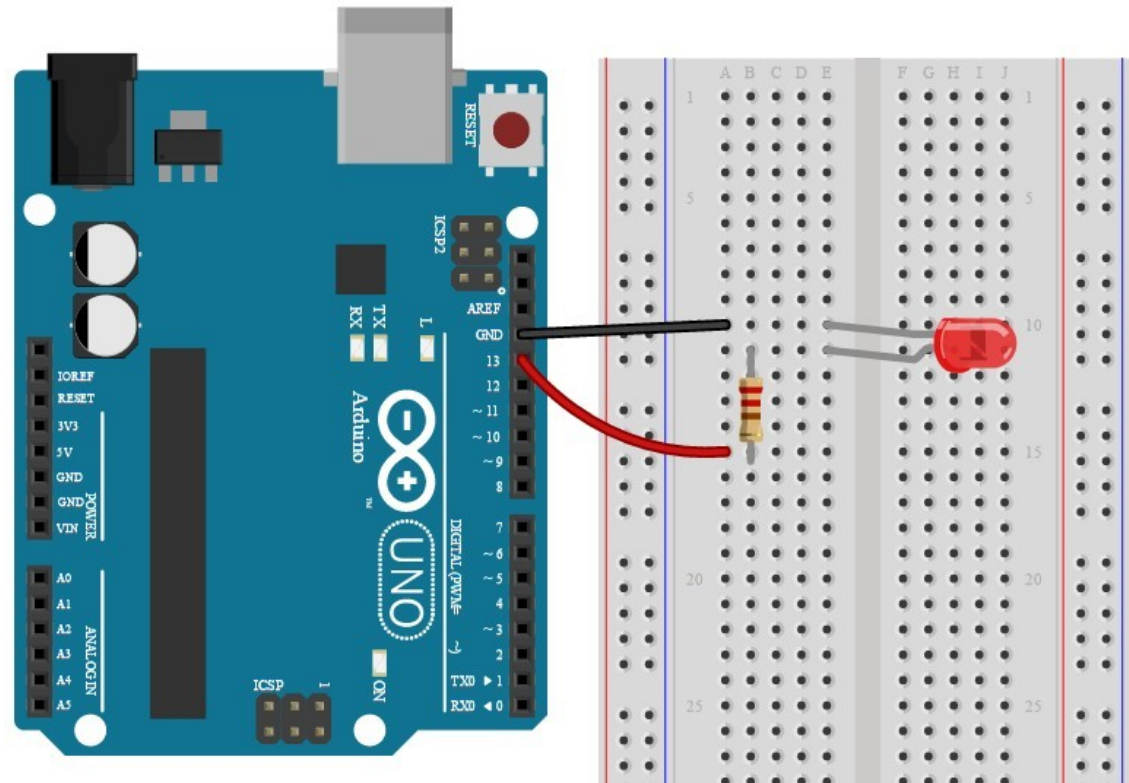
```
//Bucle do programa
```

```
}
```

Sensor de humidade e temperatura DHT11. Programa nº 1. Lembramos o parpadeo básico

- Exemplo parpadeo ou blink

```
void setup() {  
  pinMode(12, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(12, HIGH);  
  delay(300);  
  digitalWrite(12, LOW);  
  delay(300);  
}
```



Sensor de humedad e temperatura DHT11.

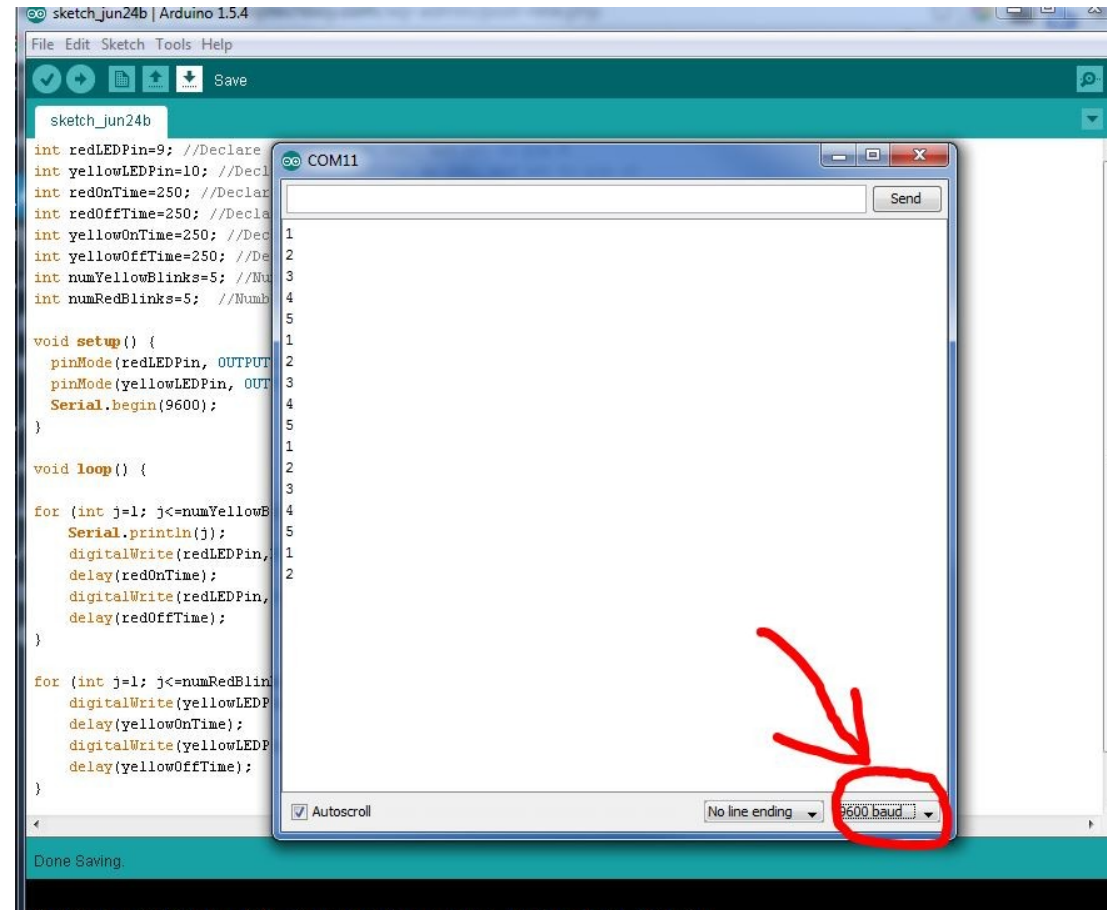
Programa nº 1. Lembramos o “Ola mundo” no monitor serie

- Exemplo “Ola mundo”

```
const int led=12;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600); // (velocidade do porto serie)
  Serial.println("¡Hola Mundo!");
}

void loop() {
  digitalWrite(led, HIGH);
  Serial.println("Enciende");
  delay(300);
  digitalWrite(led, LOW);
  Serial.println("Apaga");
  delay(300);
}
```



Sensor de humedad e temperatura DHT11.

Programa nº 1. Lembramos as funcións básicas de control

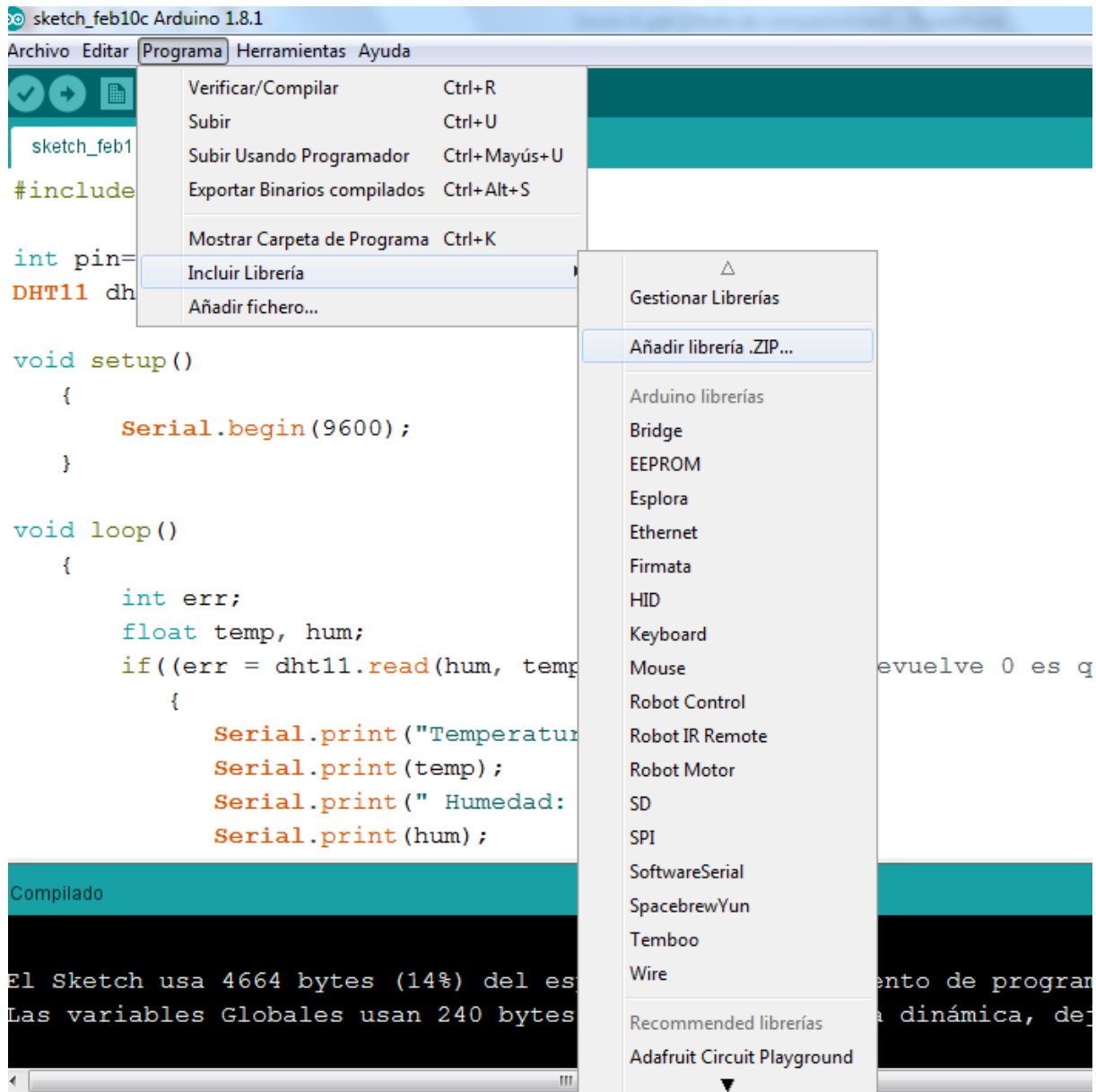
- if

```
if (comparación) {  
    ...  
}
```
- If...else

```
if (comparación) {  
    ...  
}  
else {  
    ...  
}
```
- Operadores de comparación
==, !=, <, >, <=, >=
- Operadores booleanos
&&(AND), ||(OR), !(NOT)

Sensor de humedad e temperatura DHT11.

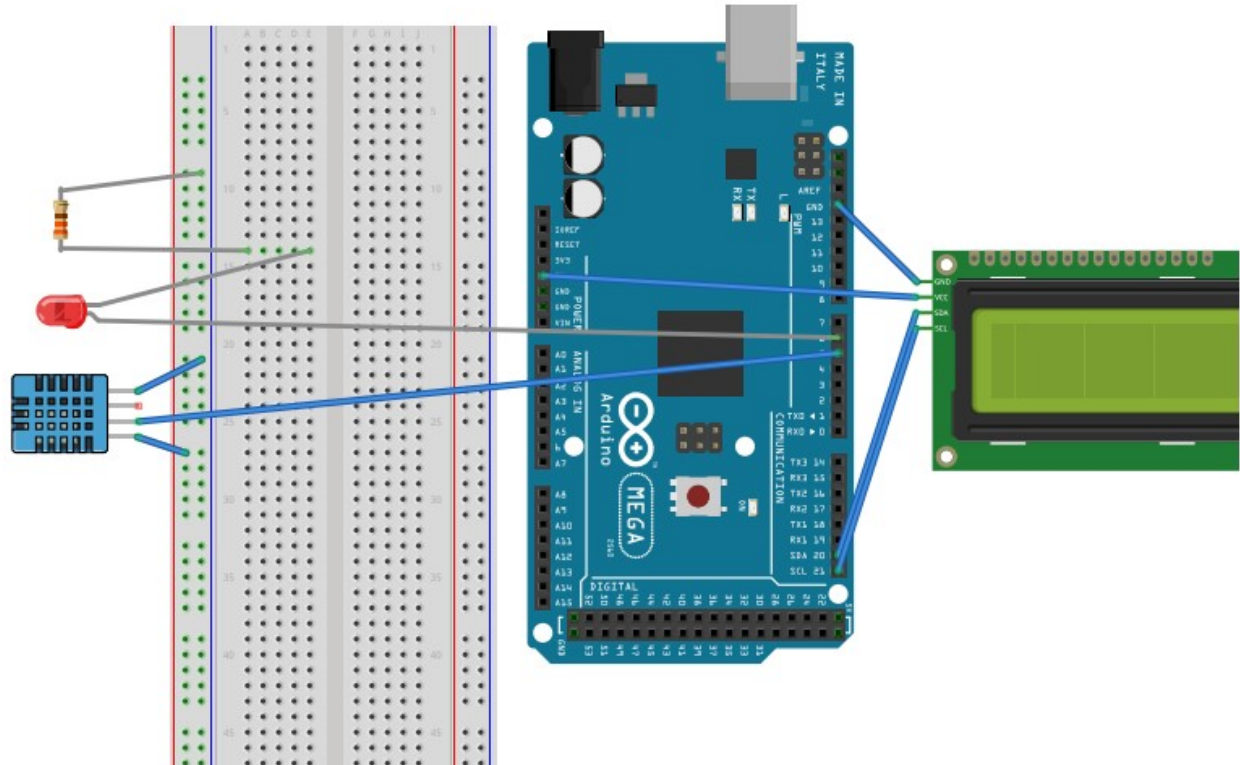
Programa nº 1. Lembramos inserir librerías



Sensor de humidade e temperatura DHT11.

Montaxe a executar nº 2

- Conectaremos o sinal do DHT11 no pin 5 por exemplo, obtendo a temperatura da estancia e a humidade relativa
- Tomaremos man dun led indicador de alarma para avisos de superación da temperatura umbral, que conectaremos no pin 6 por exemplo
- Mostraremos os valores no display I2C, pero que dirección hexadecimal emprega o noso display?

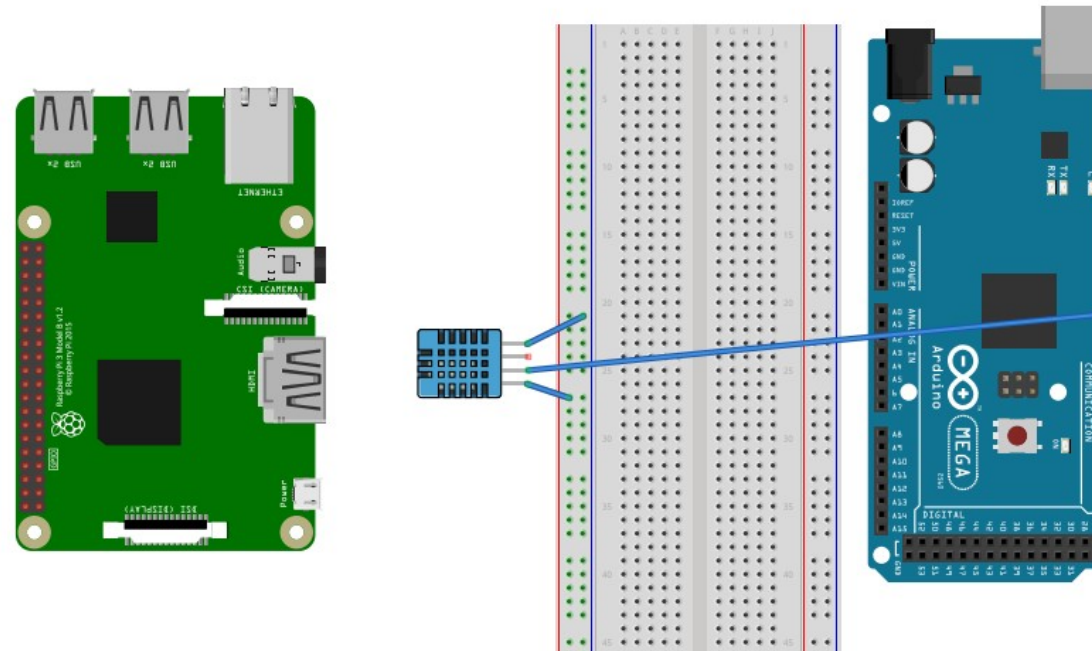


<https://www.prometec.net/bus-i2c/>

Sensor de humidade e temperatura DHT11.

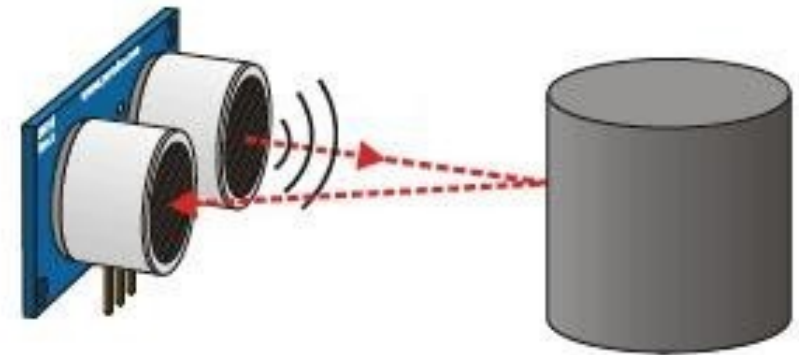
Montaxe a executar nº 3

- Conectaremos o sinal do DHT11 no pin 5 por exemplo, obtendo a temperatura da estancia e a humidade relativa
- Mostraremos os valores a traves do protocolo de comunicación mqtt, empregando a nosa raspberry como servidor mqtt



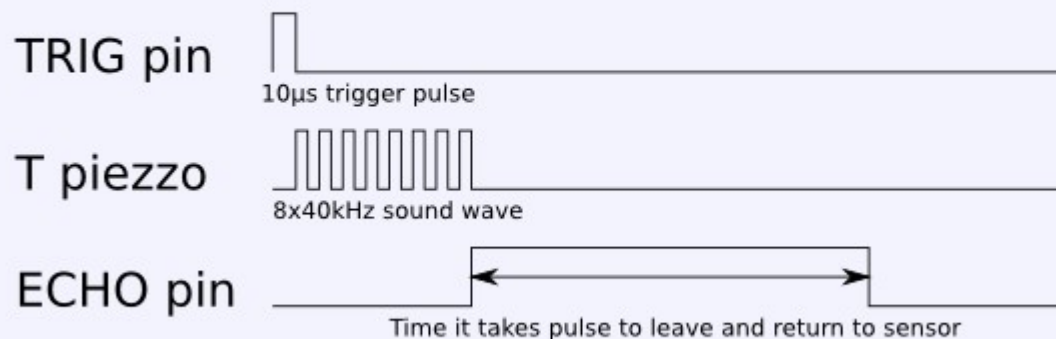
Sensor de ultrasonos HC-SR04. Montaxe a executar nº 4

- Con el podemos medir distancias entre: 2 e 400 cm
- Utilizando a velocidade do son = 340m/s
- Distancia en cm (centímetros) = Tempo medido en μs x 0.017



$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$
$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

HC-SR04 Timing Chart



CFR
VIGO

centro de
formación e recursos

Sensor de ultrasóns HC-SR04. Montaxe a executar nº 4

- Agora temos dous pins a controlar:

```
const int EchoPin = 5;
```

```
const int TriggerPin = 6;
```

- E ademais dos bucles de inicio e traballo, teremos que crear unha función coas fórmulas de calculo que precisa este sensor

```
int mide(int TriggerPin, int EchoPin)
```

```
{
```

```
long duracion, distanciaCm;
```

```
digitalWrite(TriggerPin, LOW);
```

```
delayMicroseconds(4);
```

```
digitalWrite(TriggerPin, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(TriggerPin, LOW);
```

```
duracion = pulseIn(EchoPin, HIGH)/2;
```

```
distanciaCm = duracion * 10 / 292;
```

```
return distanciaCm;
```

```
}
```

