Part 2

lesson 11

DHT11 Temperature and Humidity Sensor

Overview

In this tutorial we will learn how to use a DHT11 Temperature and Humidity Sensor.

It' s accurate enough for most projects that need to keep track of humidity and temperature readings.

Again we will be using a Library specifically designed for these sensors that will make our code short and easy to write.

Component Required:

(1) x Elegoo Uno R3

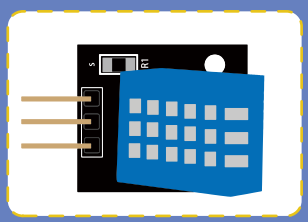
(1) x DHT11 Temperature and Humidity module

(4) x F-M wires (Female to Male DuPont wires)

Component Introduction

Temp and humidity sensor:

DHT11 digital temperature and humidity sensor is a composite Sensor which contains a calibrated digital signal output of the temperature and humidity. The dedicated digital modules collection technology and the temperature and humidity sensing technology are applied to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive moisture sensor and a NTC temperature measurement devices, and connects with a high-performance 8-bit microcontroller.



Applications: HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

Product parameters

Relative humidity:
Resolution: 8Bit
Repeatability: ±1% RH
Accuracy: At 25°C ±5% RH
Interchangeability: fully interchangeable
Response time: 1 / e (63%) of 25°C 6s

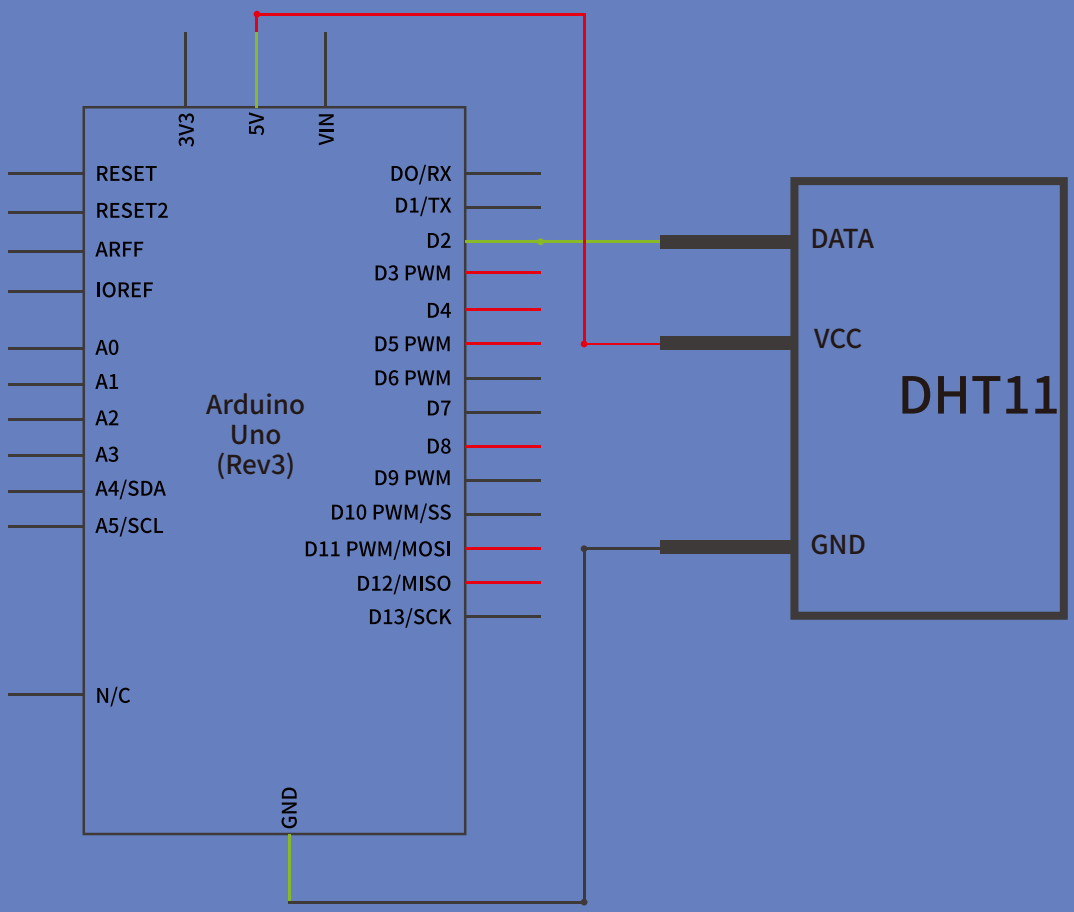
1m / s air 6s Hysteresis:
<± 0.3% RH
Long-term stability: <± 0.5% RH / yr in

Temperature:
Resolution: 8Bit
Repeatability: ±0.2°C
Range: At 0°C ±50°C

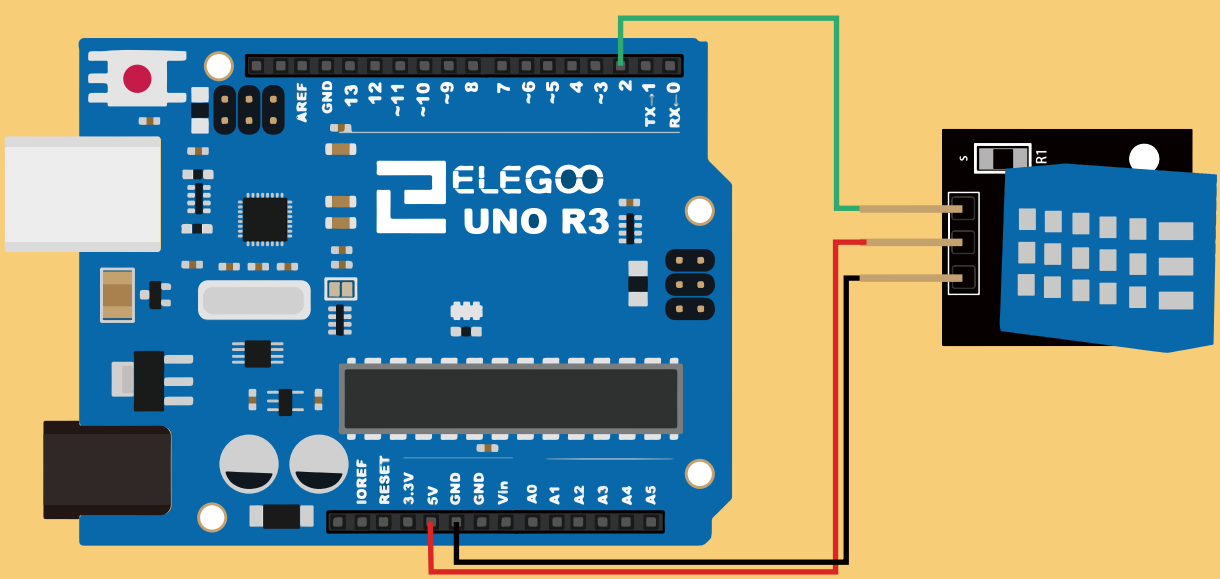
Response time: 1 / e (63%) 10S
Electrical Characteristics
Power supply: DC 3.5～5.5V
Supply Current: measurement 0.3mA standby 60μA
Sampling period: more than 2 seconds

Pin Description:

1. the VDD power supply 3.5～5.5V DC.
2. DATA serial data, a single bus.
3. GND ground, the negative power.



Connection Schematic



The connections are: Signal, Voltage and Ground which can be connected to any Pin on our UNO.

Wiring diagram

Code

- After wiring, please open the program in the code folder- **DHT11_Example** and click UPLOAD to upload the program. See Lesson 5 in part 1 for details about program uploading if there are any errors.
- Before you can run this, make sure that you have installed the **<DHT>** library or re-install it, if necessary. Otherwise, your code won't work.

```
static const int DHT_SENSOR_PIN = 2;
```

- **static**

[Variable Scope & Qualifiers]

Description

The static keyword is used to create variables that are visible to only one function. However unlike local variables that get created and destroyed every time a function is called, static variables persist beyond the function call, preserving their data between function calls.
Variables declared as static will only be created and initialized the first time a function is called.

- **float**

[Data Types]

Description

```
float temperature;
float humidity;
```

Datatype for floating-point numbers, a number that has a decimal point. Floating-point numbers are often used to approximate analog and continuous values because they have greater resolution than integers. Floating-point numbers can be as large as 3.4028235E+38 and as low as -3.4028235E+38. They are stored as 32 bits (4 bytes) of information.

Syntax

- float var = val;

Parameters

- **var**: variable name.
- **val**: the value you assign to that variable.

- **Bool**

[Data Types]

Description

Range:0~2^32.

- **Syntax**

bool var = val;

- **Parameters**

var: variable name.
val: the value to assign to that variable.

- **unsigned long**

[Conversion]

Description

Converts a value to the unsigned long data type.

- **Parameters**

x: a value of any type

- **Returns**

unsigned long

```
static bool measure_environment( float *temperature, float *humidity )
{
 static unsigned long measurement_timestamp = millis( );

 /* Measure once every four seconds. */
 if( millis( ) - measurement_timestamp > 3000ul )
 {
  if( dht_sensor.measure( temperature, humidity ) == true )
  {
   measurement_timestamp = millis( );
   return( true );
  }
 }

 return( false );
}
```

- Upload the program then open the monitor, we can see the data as below:
  (It shows the temperature of the environment, we can see it is 27 to 25 degrees with 45.0% humidity)
- Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in part 2 Lesson 4.