Al-Mustaqbal University

College Of Engineering & Technology

Department of Computer Engineering Techniques

(Stage: 3)

Digital Control

Lecture 10

Arduino programming

Dr.: Fanar Ali Joda

DHT 11 Humidity & Temperature Sensor

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

DHT11 vs DHT22

There are two versions of the DHT sensor, which look a bit similar and have the same pinout, but have different characteristics. Here are the specifications:

DHT11

• Ultra low cost

• 3 to 5V power and I/O

• 2.5mA max current use during conversion (while requesting data)

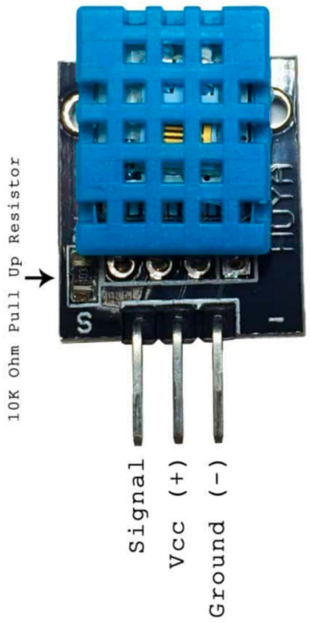
• Good for 20-80% humidity readings with 5% accuracy

• Good for 0-50°C temperature readings ±2°C accuracy

• No more than 1 Hz sampling rate (once every second)

• Body size 15.5mm x 12mm x 5.5mm

• 4 pins with 0.1" spacing



DHT22 / AM2302 (Wired version)

• Low cost

• 3 to 5V power and I/O

• 2.5mA max current use during conversion (while requesting data)

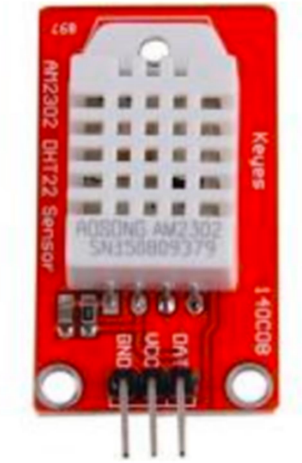
• Good for 0-100% humidity readings with 2-5% accuracy

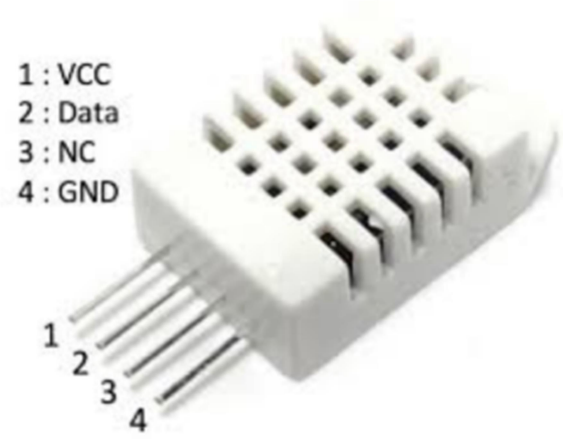
• Good for -40 to 80°C temperature readings ±0.5°C accuracy

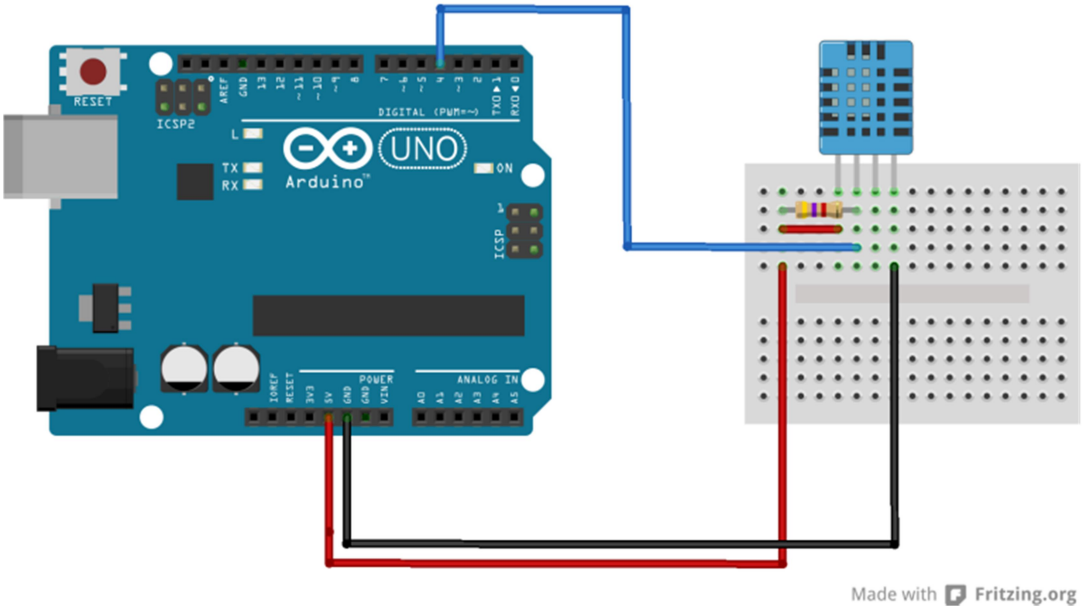
• No more than 0.5 Hz sampling rate (once every 2 seconds)

• Body size 15.1mm x 25mm x 7.7mm

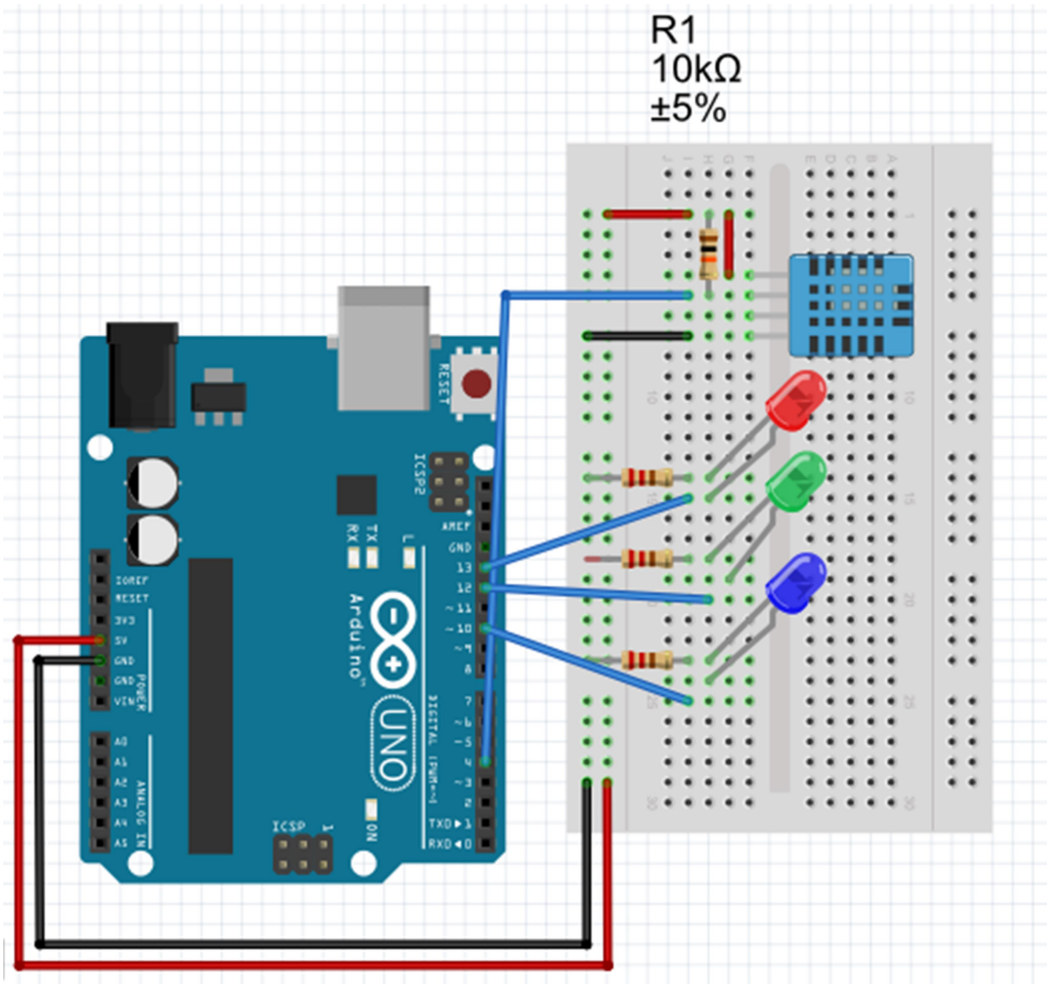
• 4 pins with 0.1" spacing







```
#include <dht11.h>
#define DHT11PIN 4
dht11 DHT11;
void  setup()
{
  Serial.begin(9600);
  DHT11.begin();
}
void loop()
{
  Serial.println();
  int chk = DHT11.read(DHT11PIN);
  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);
  Serial.print("Temperature  (C): ");
  Serial.println((float)DHT11.temperature, 2);
  delay(2000);
}
```




```
#include "DHT.h"
#define DHTPIN 4     // what pin we're connected to
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
int redLED = 13;
int greenLED = 12;
int blueLED = 11;
void setup() {
  Serial.begin(9600);
```

```
  Serial.println("DHTxx test!");
  dht.begin();
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}
void loop()
{
  delay(2000);
  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)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float hi = dht.computeHeatIndex(f, h);
  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(hi);
Serial.println(" *F");
if (t > 28)
{
  digitalWrite(redLED, HIGH);
  digitalWrite(greenLED, LOW);
  digitalWrite(blueLED, LOW);
}
else if (t > 18)
{
  digitalWrite(redLED, LOW);
  digitalWrite(greenLED, HIGH);
  digitalWrite(blueLED, LOW);
```

```
  }
  else
  {
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, HIGH);
  }
}
```

How the Code Works

You start by including the DHT library:

```
#include "DHT.h"
```

Then, you define the pin that the DHT sensor is connected to. In this case it is connected to digital pin 4.

```
#define DHTPIN 4 // what digital pin we're connected to
```

Then, you need to define the DHT sensor type you're using. In our example we're using the DHT11.

```
#define DHTTYPE DHT11 // DHT 11
```

If you're using another DHT sensor, you need to comment the previous line and uncomment one of the following:

```
//#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
```

Then, initialize a DHT object called dht with the pin and type you've defined previously:

```
DHT dht(DHTPIN, DHTTYPE);
```

In the setup(), initialize the Serial Monitor at a baud rate of 9600 for debugging purposes.

```
Serial.begin(9600);
Serial.println("DHTxx test!");
```

Initialize the DHT sensor with the .begin() method.

```
dht.begin();
```

In the loop(), at the beginning, there's a delay of 2 seconds. This delay is needed to give enough time for the sensor to take readings. The maximum sampling rate is two seconds for the DHT22 and one second for the DHT11.

```
delay(2000);
```

Reading temperature and humidity is very simple. To get humidity, you just need to use the readHumidity() method on the dht object. In this case, we're saving the humidity in the h variable. Note that the readHumidity() method returns a value of type float.

```
float h = dht.readHumidity();
```

Similarly, to read temperature use the readTemperature() method.

```
float t = dht.readTemperature();
```

To get temperature in Fahrenheit degrees, just pass true to the readTemperature() method as follows:

```
float f = dht.readTemperature(true);
```

This library also comes with methods to compute the heat index in Fahrenheit and Celsius:

```
// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
```