Lesson 14 DHT11 Temperature and Humidity Sensor

Introduction

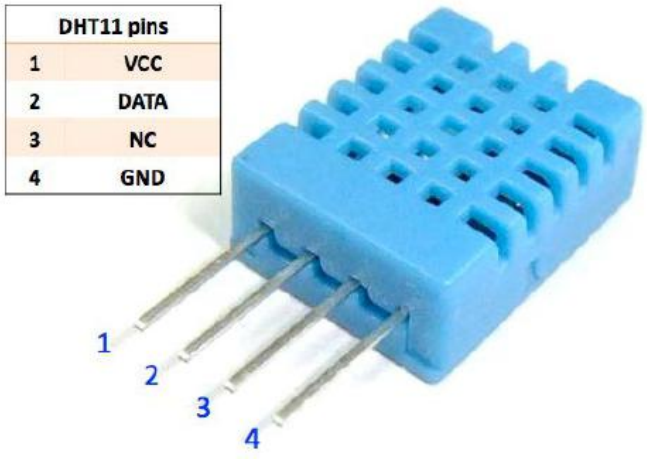
In this lesson,you will learn how to use a DHT11 Temperature and Humidity Sensor.

Hardware Required

- ✓ 1 * RexQualis Mega 2560
- ✓ 1 * DHT11 Temperature and Humidity module
- ✓ 3 * F-M Jumper Wires

Principle

DHT11 Temperature and Humidity Sensor



DHT11 output calibrated digital signal. It applys exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of pro

gramme in OTP memory, when the sensor is detecting, it will cite coefficient fr om memory.

Small size & low consumption & long transmission distance(100m) enable DH T11 to be suited in all kinds ofharsh application occasions. Single-row packag ed with four pins, making the connection very convenient.

Supply voltage: DC 3.3 to 5.5V

Measuring range (T) : -20 to +60 Celsius(-4 to +140 Fahrenheit)

Measuring range (RH): 5 to 95% relative humidity

Typ. Temperature accuracy: ±2 Celsius

Typ. Humidity accuracy: ±5%RH at 25 Celsius

Long term drift(T): <1 Celsius/year

Long term drift(RH) : <1%RH/year

Resolution(T): 0.1 Celsius

Resolution(RH): 1%RH

Sensor Type: Capacitive sensor

Interface: One line digital

Housing material: ABS

Net weight: 1g

Code interpretation

```
#define#define DHT11_PIN 0 // pin A0

byte read_dht11_dat()

{

   byte i = 0;
```

```
    byte result = 0;
    for (i = 0; i < 8; i++) {
        while (!(PINF & _BV(DHT11_PIN)));  // wait for 50us
        delayMicroseconds(30);
        if (PINF & _BV(DHT11_PIN))
            result |= (1 << (7 - i));
        while ((PINF & _BV(DHT11_PIN))); // wait '1' finish
    }
    return result;
}
void setup()
{
    DDRF |= _BV(DHT11_PIN);
    PORTF |= _BV(DHT11_PIN);
    Serial.begin(19200);
    Serial.println("Ready");
}
#if 0
void loop()
{
    PORTF &= ~_BV(DHT11_PIN); //pull-down i/o pin from 18ms
    delay(18);
```

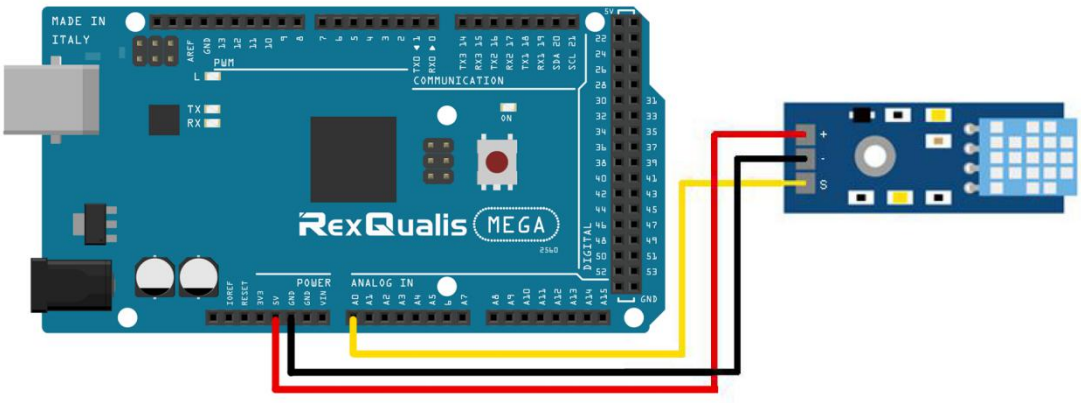
```
    PORTF |= _BV(DHT11_PIN);
    delay(40);
}
#endif

#if 1
void loop()
{
    byte dht11_dat[5];
    byte dht11_in;
    byte i;
    // start condition
    // 1. pull-down i/o pin from 18ms
    PORTF &= ~_BV(DHT11_PIN); // pull-down i/o pin from 18ms
    delay(18);
    PORTF |= _BV(DHT11_PIN);
    delayMicroseconds(40);
    DDRF &= ~_BV(DHT11_PIN);    //Set to input
    delayMicroseconds(40);
    dht11_in = PINF & _BV(DHT11_PIN); //Read data line is high voltage
    if (dht11_in) {
        Serial.println("dht11 start condition 1 not met");
```

```
    return;
  }
  delayMicroseconds(80);
  dht11_in = PINF & _BV(DHT11_PIN);
  if (!dht11_in) {
    Serial.println("dht11 start condition 2 not met");
    return;
  }
  delayMicroseconds(80);
  // now ready for data reception
  for (i = 0; i < 5; i++)
    dht11_dat[i] = read_dht11_dat();
  DDRF |= _BV(DHT11_PIN);
  PORTF |= _BV(DHT11_PIN);
  byte dht11_check_sum = dht11_dat[0] + dht11_dat[1] + dht11_dat[2] +
dht11_dat[3];
  // check check_sum
  if (dht11_dat[4] != dht11_check_sum)
  {
    Serial.println("DHT11 checksum error");
  }
  Serial.print("Current humdity = ");
  Serial.print(dht11_dat[0], DEC);
```

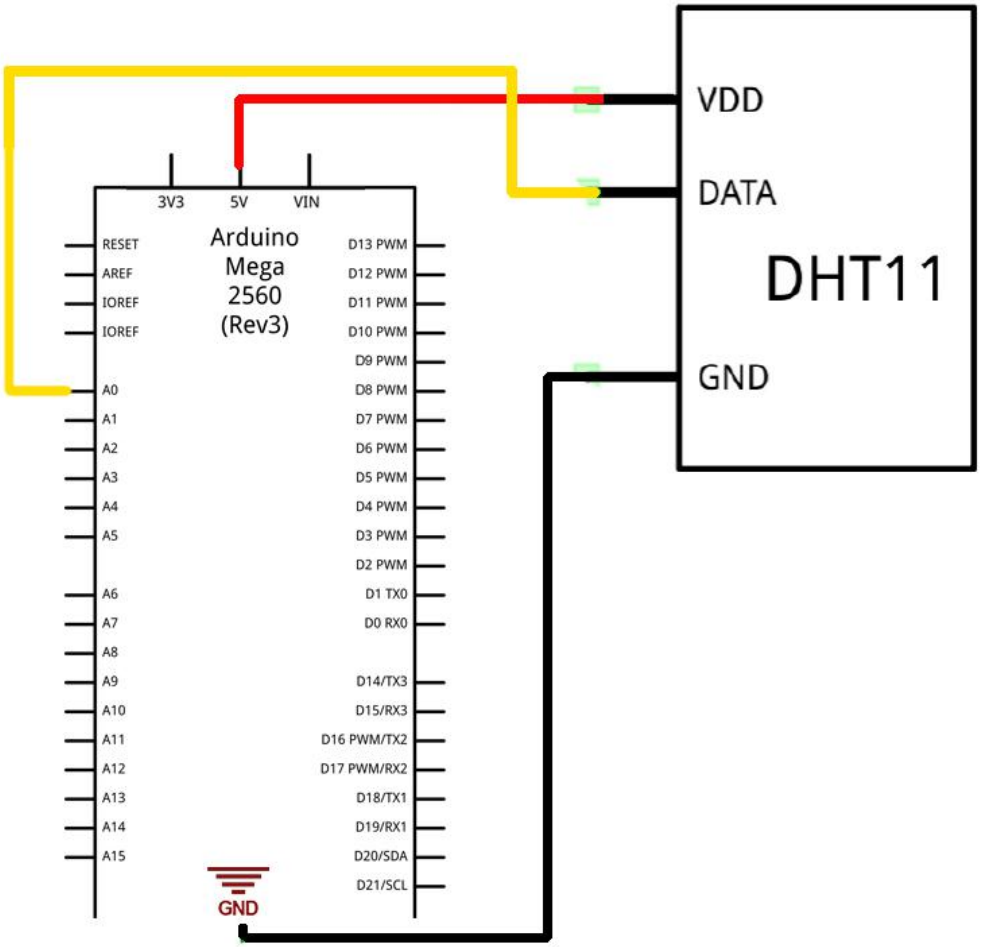
```
    Serial.print(".");
    Serial.print(dht11_dat[1], DEC);
    Serial.print("% ");
    Serial.print("temperature = ");
    Serial.print(dht11_dat[2], DEC);
    Serial.print(".");
    Serial.print(dht11_dat[3], DEC);
    Serial.println("C ");
    delay(2000);
}
#endif
```

Experimental Procedures

Step 1:Build the circui

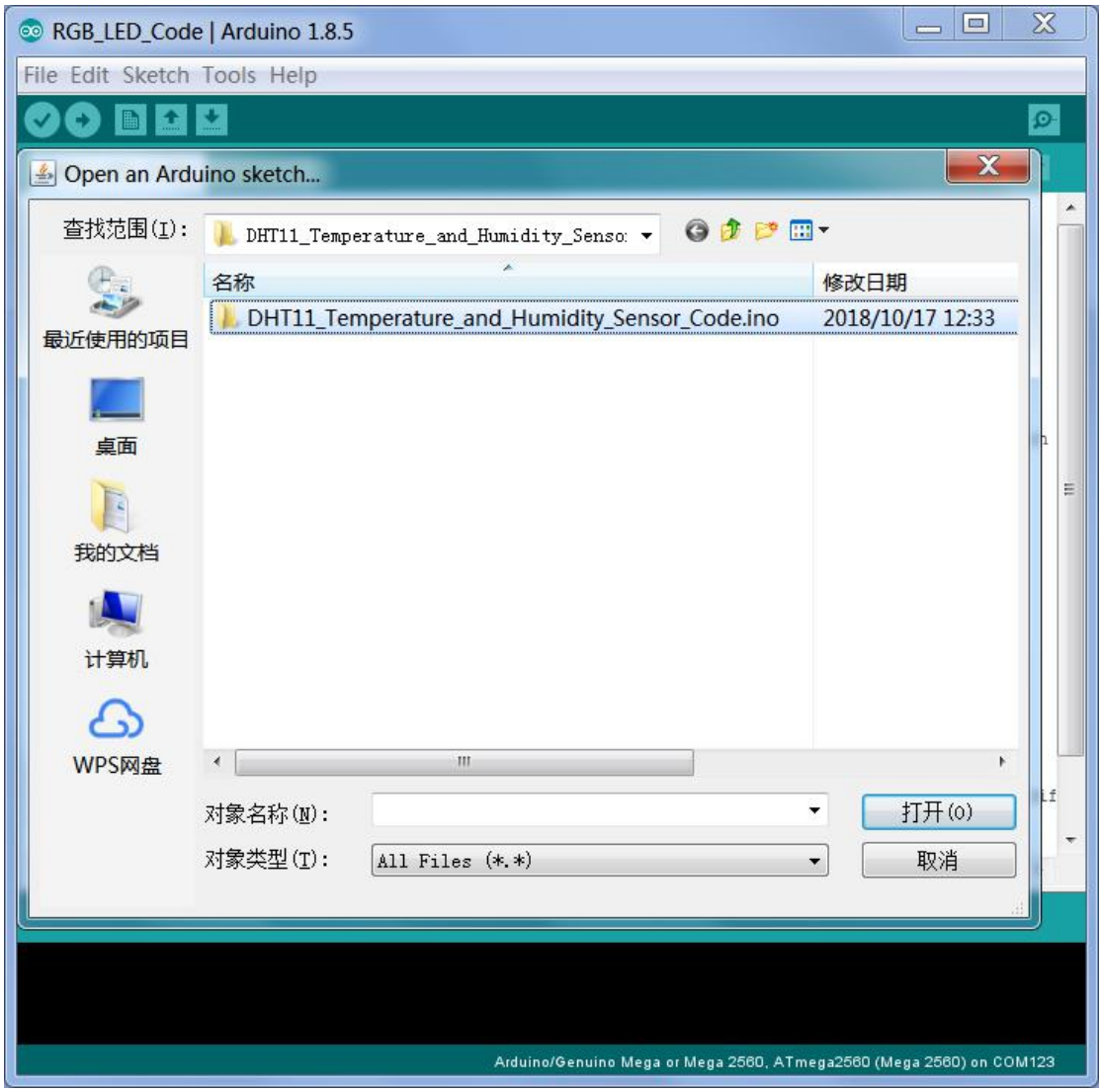


Schematic Diagram


VDD
DATA
DHT11
GND
3V3
5V
VIN
Arduino
Mega
2560
(Rev3)
RESET
AREF
IOREF
IOREF
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
D13 PWM
D12 PWM
D11 PWM
D10 PWM
D9 PWM
D8 PWM
D7 PWM
D6 PWM
D5 PWM
D4 PWM
D3 PWM
D2 PWM
D1 TX0
D0 RX0
D14/TX3
D15/RX3
D16 PWM/TX2
D17 PWM/RX2
D18/TX1
D19/RX1
D20/SDA
D21/SCL
GND

Step 2:Open the code:

DHT11_Temperature_and_Humidity_Sensor_Code

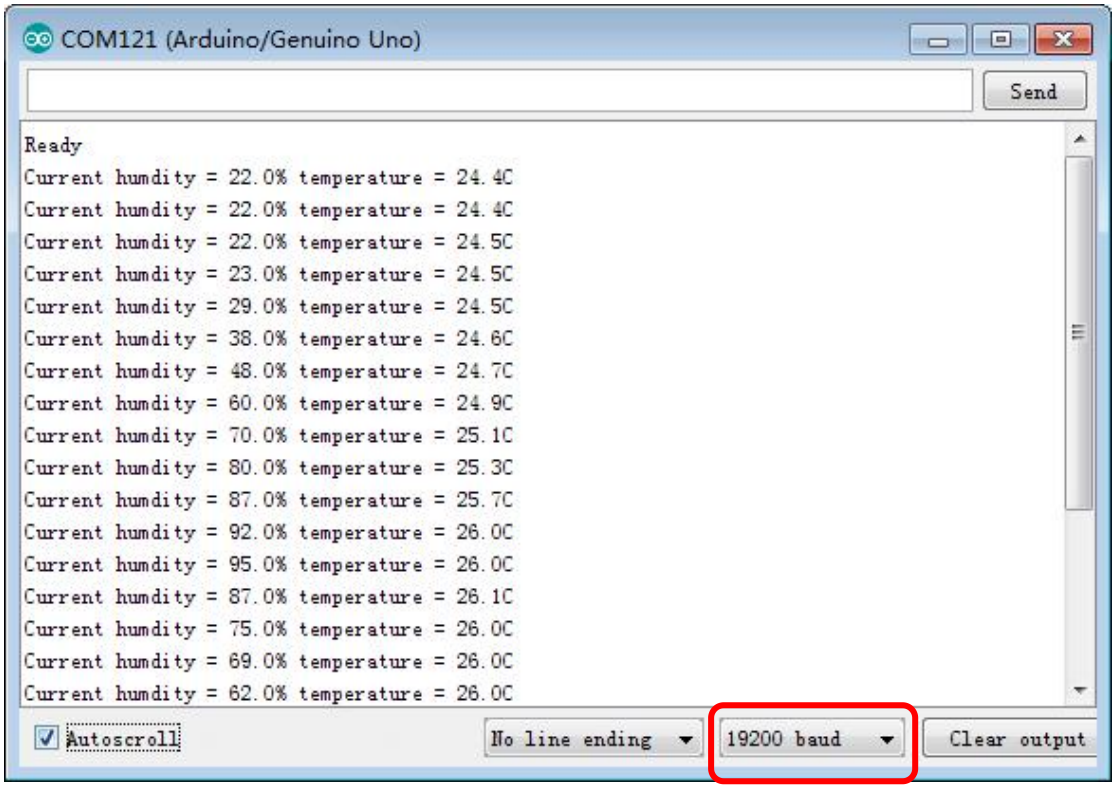


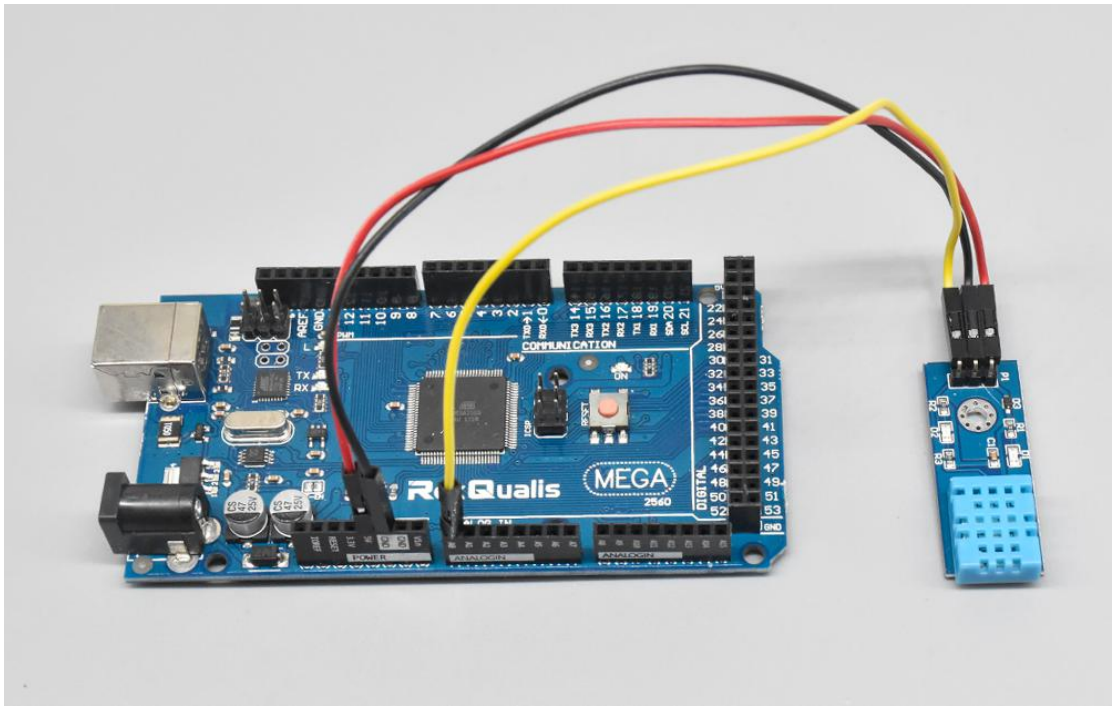
Step 3:Attach Arduino Mega 2560 board to your computer via USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.

Step 4:Upload the code to the RexQualis Mega 2560 board.

Step 5:Open the Serial Monitor,alter the baud rate to 19200,then you can see the data as blow:

(How to use the Serial Monitor is introduced in details in Lesson 0 Preface)





You can see the video of the experiment results on YouTube:

https://youtu.be/yAN5E1eqcbw

If it isn’t working, make sure you have assembled the circuit

correctly, verified and uploaded the code to your board.For how to upload the code and install the library, check Lesson 0 Preface.