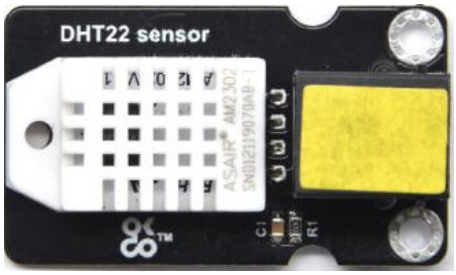
DHT22 Temperature and Humidity Sensor (000x0000 Article Number) (TS2176)

DESIGN THE WORLD

Product Details



This DHT22 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 temperature and humidity sensing technology are applied to ensure that the product has high reliability and excellent long-term stability.
Qualities of ultra-fast response, strong anti-interference, and high cost performance make it a wide applied application or even the most demanding one.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Wide range and high precision temperature and humidity sensor with digital output and pre-calibrated sensor.

Technical Specifications

Sensor type	Digital input
Working voltage	3.3V-5V
Humidity measurement range	0----100%RH
Humidity measurement accuracy	±2%RH
Temperature measurement range	- 40℃ to 80℃
Dimensions	43mm * 26mm * 18mm
Weight	9.3g

Applications

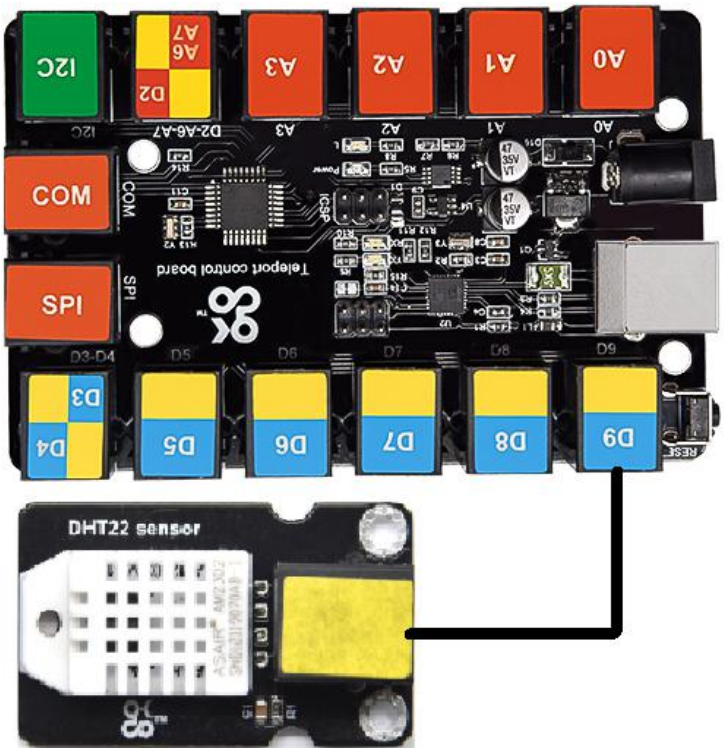
- Weather station
- Automatic control
- Testing and inspection equipment
- Soil moisture detector

Comparison

Project	DHT11	DHT22
Moisture		
Range	5 ~ 95%RH	0 ~ 100%RH
Accuracy	±5% (Typical)	±2% (Typical)
Temperature		
Range	-20 ~ 60℃	-40 ~ 80℃
Accuracy	±2% (Typical)	±0.5% (Typical)

This module is compatible with the TS2180-Raspberry Pi shield, the TS2179-Micro:bit shield and the TS2178-TelePort main board.

➢ **Arduino Application**



This module is compatible with the TS2178 TelePort control board.

Test Code

Unzip the library files, that is, copy the unzipped **DHT** folder into the libraries of Arduino IDE.

After pasting it, then reboot the compiler.

For instance: C:\Program Files\Arduino\libraries

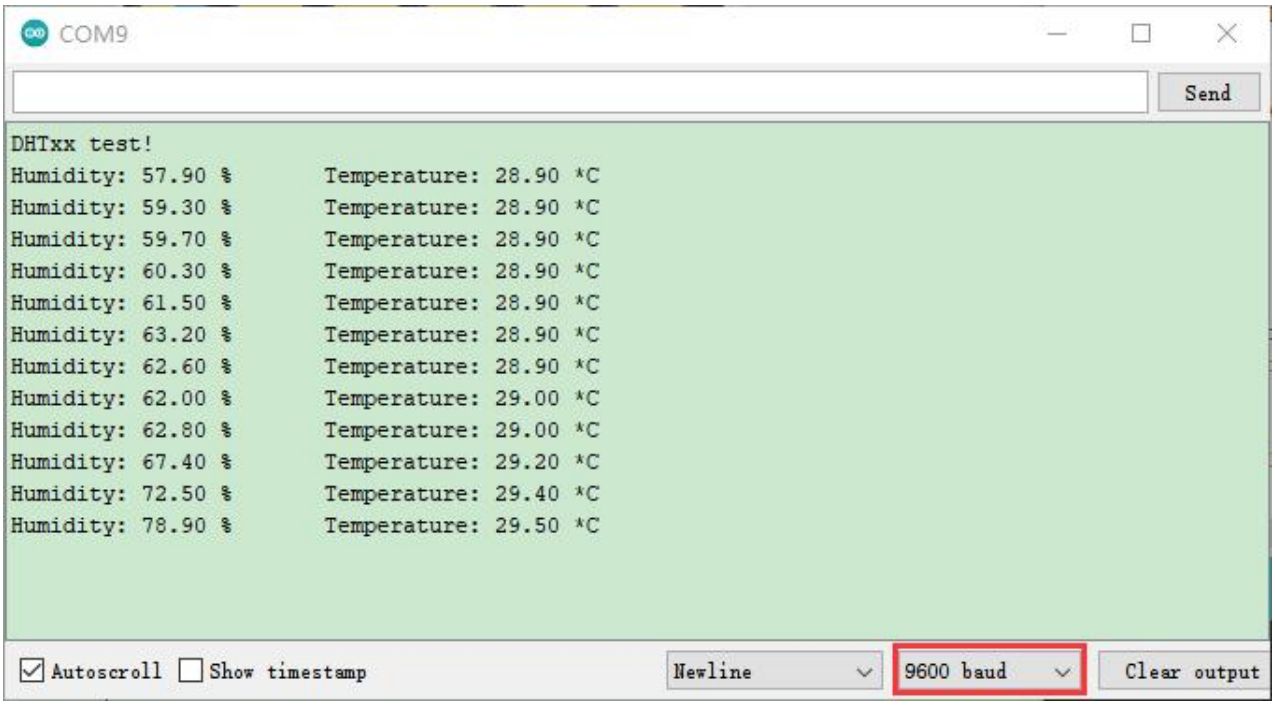
```
#include "DHT.h"
#define DHTPIN 9     // define the connection pin
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
  Serial.begin(9600); //set the baud rate
  Serial.println("DHTxx test!");//print the character and line wrap

  dht.begin();
}

void loop() {
  float h = dht.readHumidity(); // calculate the humidity value
  float t = dht.readTemperature(); //calculate the temp.value
  if (isnan(t) || isnan(h))
{
    Serial.println("Failed to read from DHT");//show the contents and line wrap
  }
else
{
    Serial.print("Humidity: "); //print the humidity
    Serial.print(h); //print the humidity value
    Serial.print(" %\t");//print the content
    Serial.print("Temperature: "); //print the temperature
Serial.print(t);//print the temperature value
Serial.println(" *C");//print the temperature unit and line wrap
  }
}
```

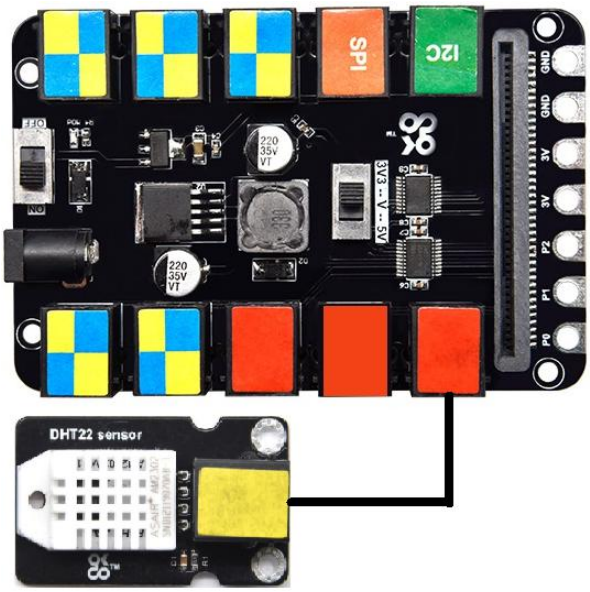
Test Result

Wire up, upload code, power it up, open serial monitor and set baud rate to 9600. Then the serial monitor will show the current temperature and humidity value.



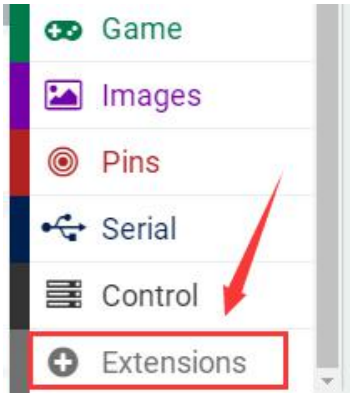
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➢ Micro:bit Application

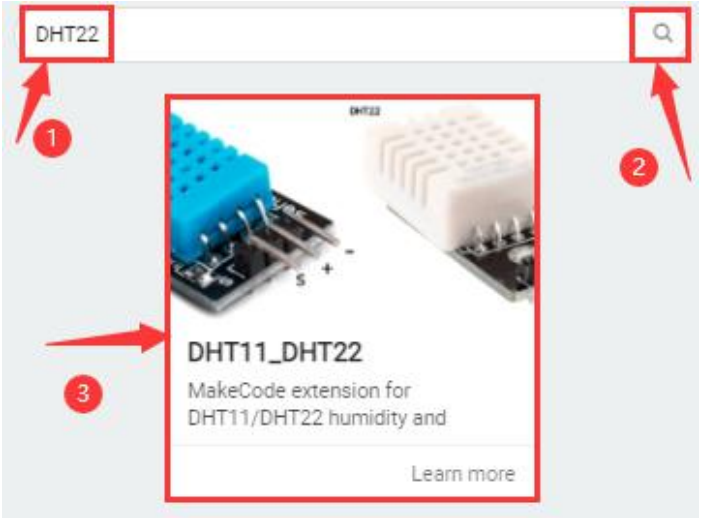


It is compatible with the Micro:bit board and the TS2179 Micro:bit expansion board.

Add the library of the DHT22 temperature and humidity sensor, as shown below;
Use the library file to set code, click“Extensions”



Enter DHT22 to search, as shown below, click the library file and download it automatically.



After the library of the DHT22 temperature and humidity sensor is installed, then you can view the corresponding block in the blocks list.

Test Code

.....①Run the "on start" block to boot the program
.....②Open the LED matrix of the Micro:bit
.....③The program is run circularly under the command of "forever" block

.....④check the data from P0 of the DHT22 sensor. If you use a 4-pin/no pcb version, you need to unplug the data pin. It is recommended to wait 1 (DHT11) or (DHT22) 2 seconds between each check

.....⑤The Micro:bit shows the temperature

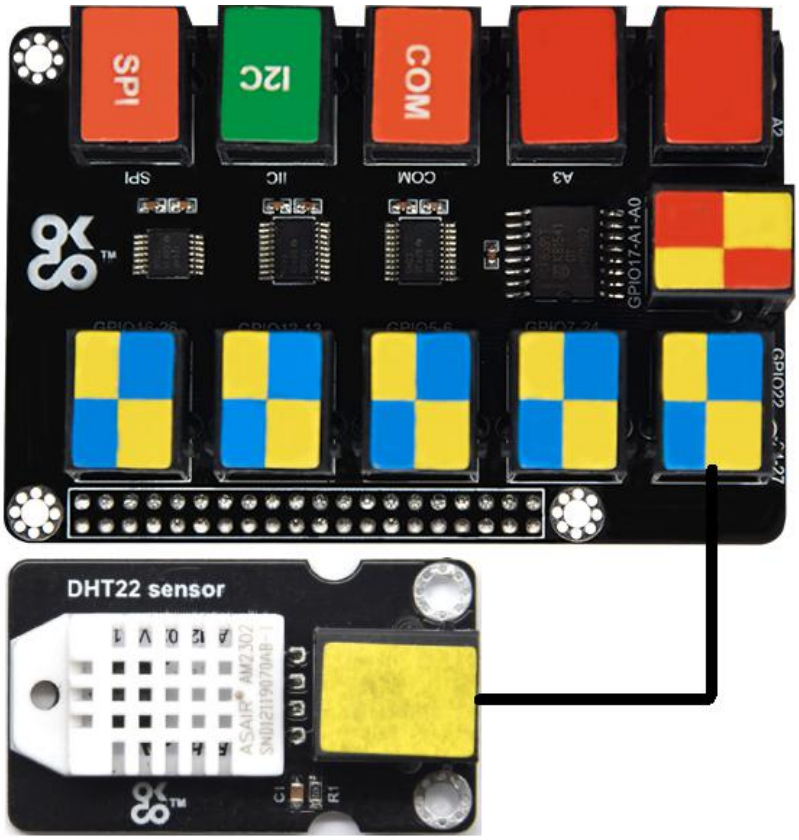
.....⑥The Micro:bit shows the humidity

Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn DIP switch to 3V3, upload test code and power it up. Then the Micro:bit will display the ambient temperature and humidity.

If you want to know more details about the Micro:bit board and Micro:bit shield, you can refer to TS2179.

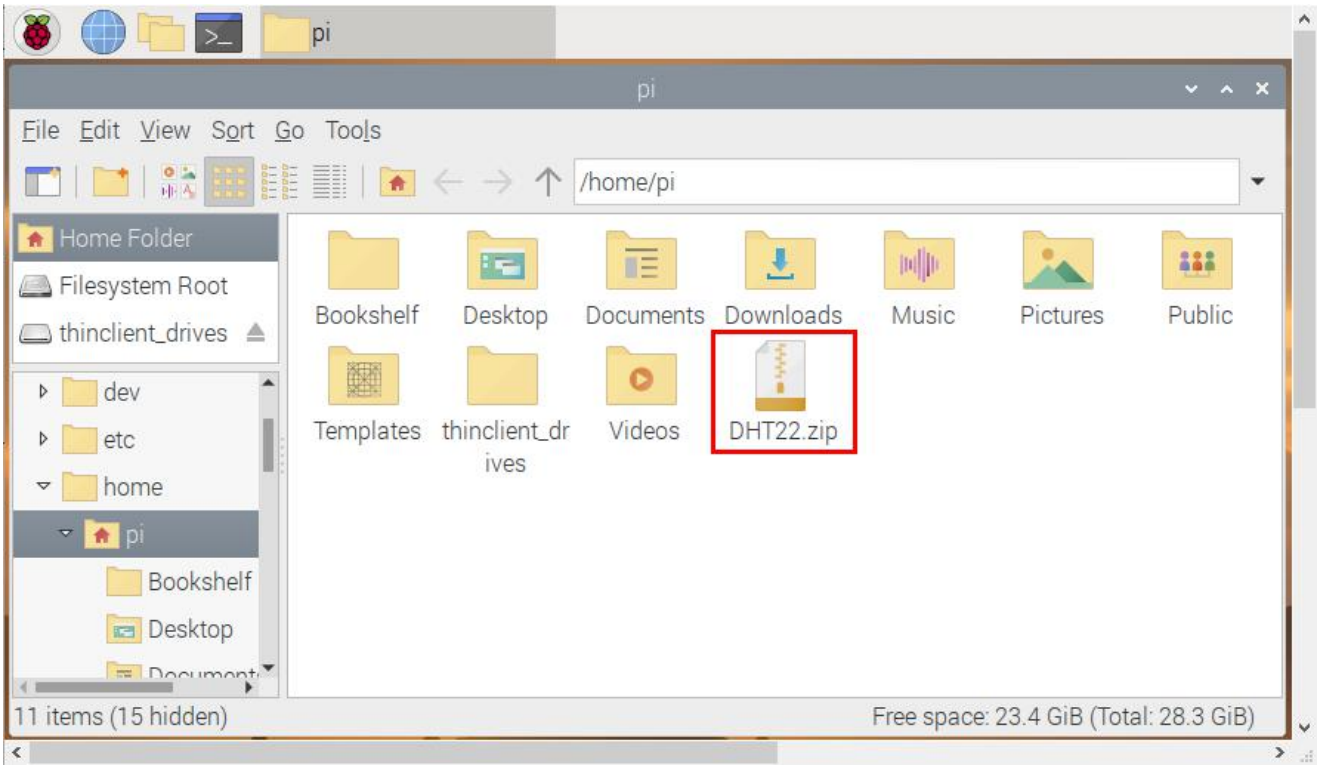
➢ Raspberry Pi Application

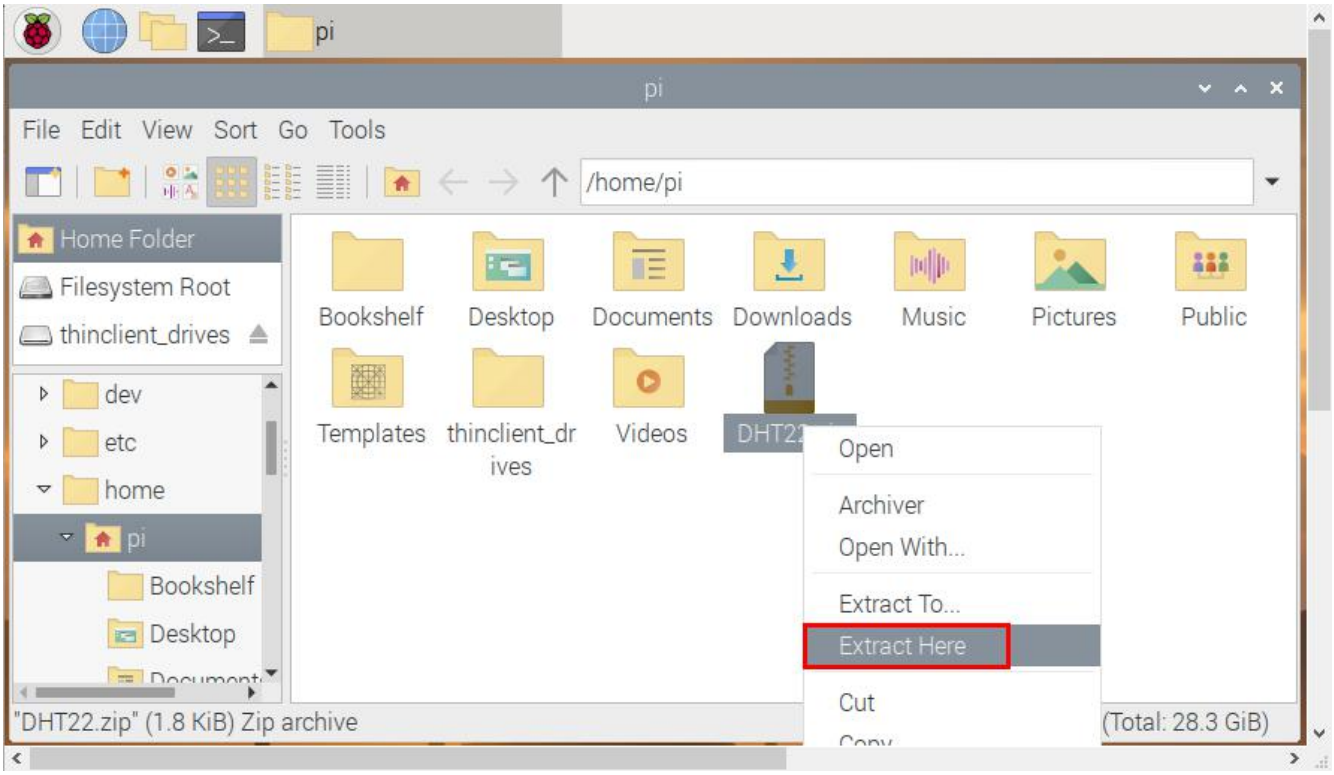


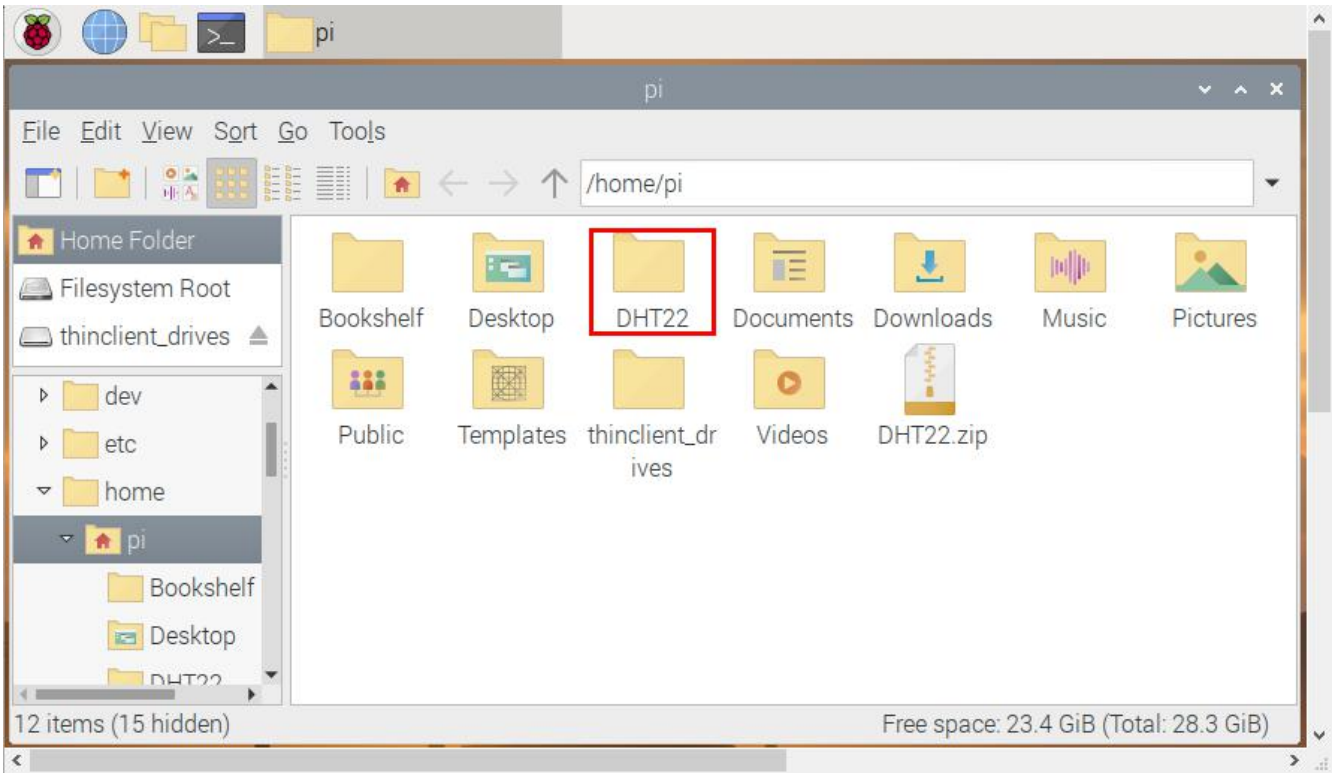
This module is compatible with the Raspberry Pi board and the TS2180 Raspberry Pi shield.

Copy the test code to Raspberry Pi system to run it

(1) Save the test code in the **pi** folder of Raspberry Pi system. Then place the DHT22.zip file we provide in the **pi** folder, right-click and click **Extract Here.** As shown below:







(2) Compile and run test code:

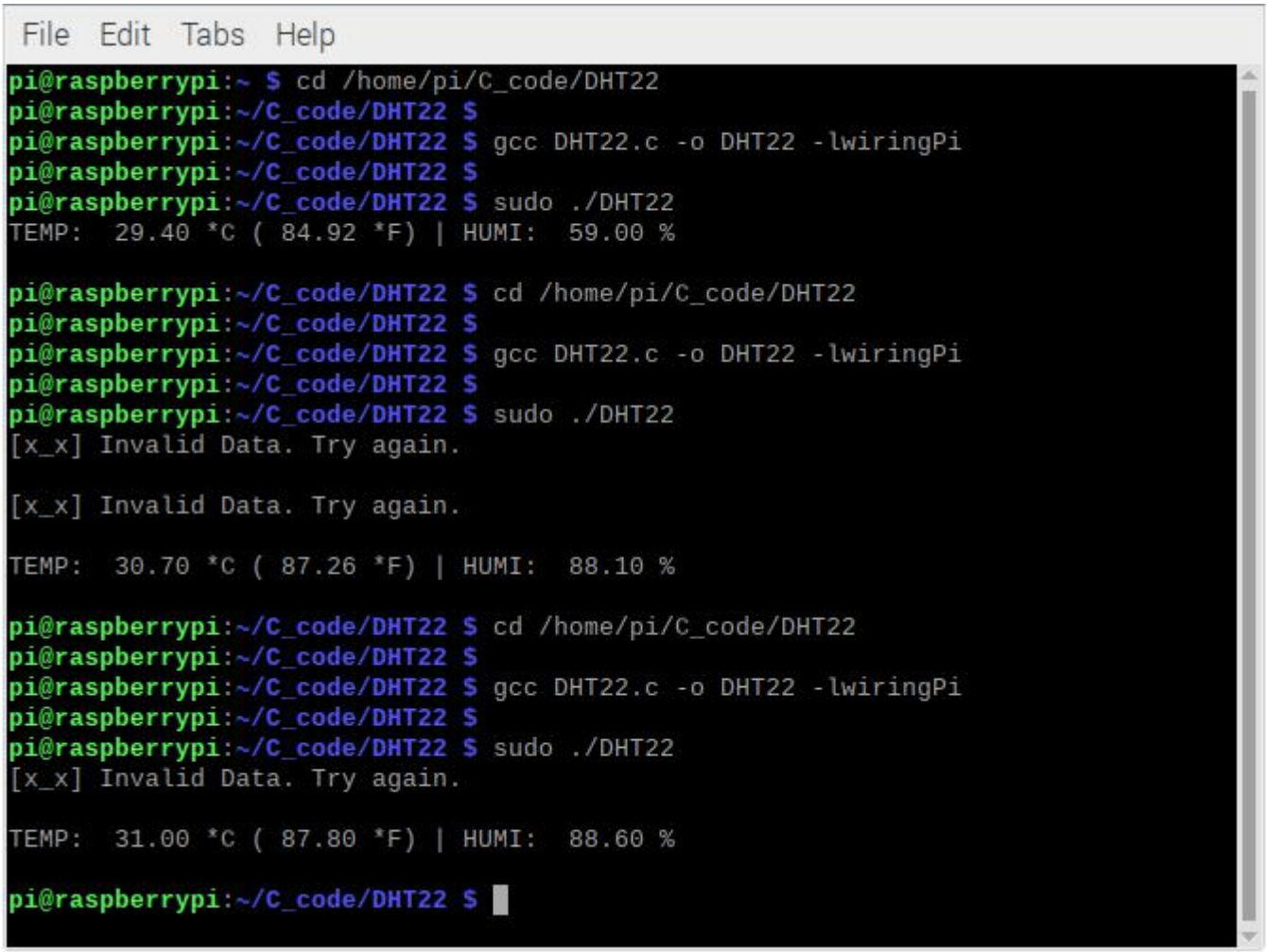
Input the following code and press"Enter"

```
cd /home/pi/DHT22
gcc DHT22.c -o DHT22 -lwiringPi
sudo ./DHT22
```

(3) Test Result:

Insert the shield into the Raspberry Pi board. After programming finishes, then the terminal will show the temperature and humidity value detected by the DHT22 temperature and humidity sensor.

Note: press Ctrl + C to exit code running



Test Code

File name: DHT22.c

```
#include <stdio.h>
#include <wiringPi.h>

static const unsigned short signal = 22;
unsigned short data[5] = {0, 0, 0, 0, 0};


short readData()
{
	unsigned short val = 0x00;
	unsigned short signal_length = 0;
	unsigned short val_counter = 0;
	unsigned short loop_counter = 0;

	while (1)
	{
		// Count only HIGH signal
		while (digitalRead(signal) == HIGH)
		{
			signal_length++;

			// When sending data ends, high signal occur infinite.
			// So we have to end this infinite loop.
			if (signal_length >= 200)
			{
				return -1;
			}

			delayMicroseconds(1);
		}

		// If signal is HIGH
		if (signal_length > 0)
		{
			loop_counter++;		// HIGH signal counting

			// The DHT22 sends a lot of unstable signals.
			// So extended the counting range.
			if (signal_length < 10)
			{
				// Unstable signal
				val <<= 1;			// 0 bit. Just shift left
			}
```

```
            else if (signal_length < 30)
            {
                // 26~28us means 0 bit
                val <<= 1;              // 0 bit. Just shift left
            }

            else if (signal_length < 85)
            {
                // 70us means 1 bit
                // Shift left and input 0x01 using OR operator
                val <<= 1;
                val |= 1;
            }

            else
            {
                // Unstable signal
                return -1;
            }

            signal_length = 0;      // Initialize signal length for next signal
            val_counter++;          // Count for 8 bit data
        }

        // The first and second signal is DHT22's start signal.
        // So ignore these data.
        if (loop_counter < 3)
        {
            val = 0x00;
            val_counter = 0;
        }

        // If 8 bit data input complete
        if (val_counter >= 8)
        {
            // 8 bit data input to the data array
            data[(loop_counter / 8) - 1] = val;

            val = 0x00;
            val_counter = 0;
        }
    }
```

```
}


int main(void)
{
	float humidity;
	float celsius;
	float fahrenheit;
	short checksum;

	// GPIO Initialization
	if (wiringPiSetupGpio() == -1)
	{
		printf("[x_x] GPIO Initialization FAILED.\n");
		return -1;
	}

	for (unsigned char i = 0; i < 10; i++)
	{
		pinMode(signal, OUTPUT);

		// Send out start signal
		digitalWrite(signal, LOW);
		delay(20);					// Stay LOW for 5~30 milliseconds
		pinMode(signal, INPUT);		// 'INPUT' equals 'HIGH' level. And signal read mode

		readData();		// Read DHT22 signal

		// The sum is maybe over 8 bit like this: '0001 0101 1010'.
		// Remove the '9 bit' data using AND operator.
		checksum = (data[0] + data[1] + data[2] + data[3]) & 0xFF;

		// If Check-sum data is correct (NOT 0x00), display humidity and temperature
		if (data[4] == checksum && checksum != 0x00)
		{
			// * 256 is the same thing '<< 8' (shift).
			humidity = ((data[0] * 256) + data[1]) / 10.0;

			// found that with the original code at temperatures > 25.4 degrees celsius
			// the temperature would print 0.0 and increase further from there.
			// Eventually when the actual temperature drops below 25.4 again
			// it would print the temperature as expected.
			// Some research and comparisin with other C implementation suggest a
```

```
                        // different calculation of celsius.
                        //celsius = data[3] / 10.0; //original
                        celsius = (((data[2] & 0x7F)*256) + data[3]) / 10.0; //Juergen Wolf-Hofer

                        // If 'data[2]' data like 1000 0000, It means minus temperature
                        if (data[2] == 0x80)
                        {
                                celsius *= -1;
                        }

                        fahrenheit = ((celsius * 9) / 5) + 32;

                        // Display all data
                        printf("TEMP: %6.2f *C (%6.2f *F) | HUMI: %6.2f %\n\n", celsius, fahrenheit, humidity);
                        return 0;
                }

                else
                {
                        printf("[x_x] Invalid Data. Try again.\n\n");
                }

                // Initialize data array for next loop
                for (unsigned char i = 0; i < 5; i++)
                {
                        data[i] = 0;
                }

                delay(2000);    // DHT22 average sensing period is 2 seconds
        }

        return 0;
}
```

If you want to know how to utilize Raspberry Pi and the Raspberry Pi shield, you can refer to TS2180.

END