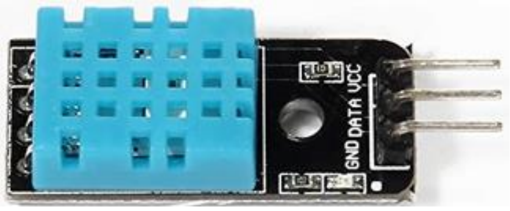
DHT11 Experiment

Overview



This lesson will teach you how to use DHT11 module, which is simple and easy to use.

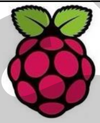
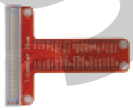
Specification

Please view DHT11-datasheet.pdf.
Path: \Public_materials\Datasheet\ DHT11-datasheet.pdf

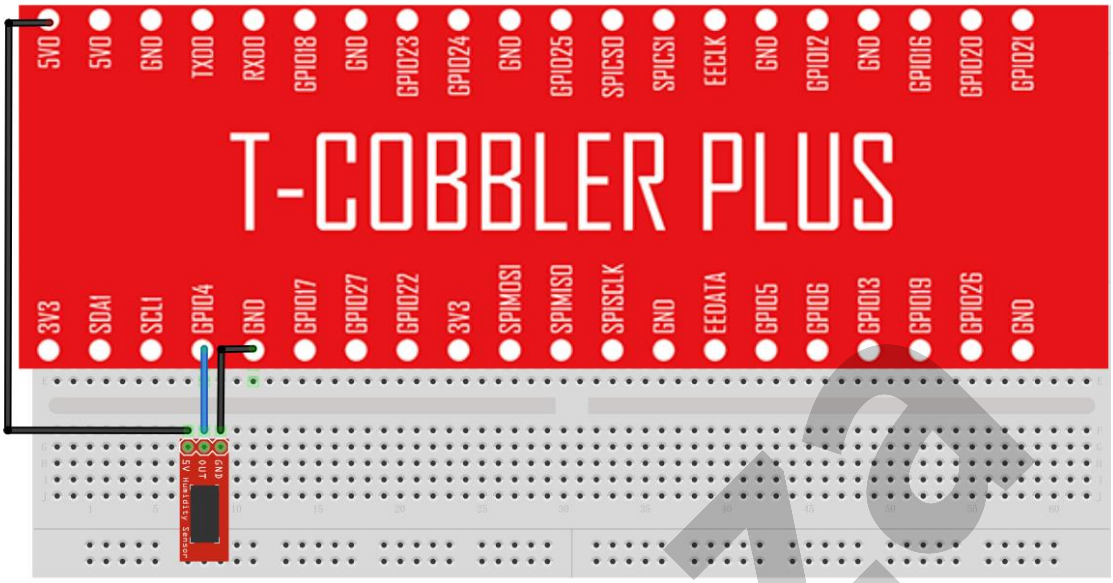
Pin definition

RPI		DHT11
GND	->	GND/’-’
GPIO4	->	DATA/’out’
5V0	->	VCC/’+’

Hardware required

Material diagram	Material name	Number
	DHT11 Module	1
	Raspberry Pi Board	1
	T-Cobbler Plus	1
	40P GPIO Cable	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Connection

RPI	DHT11
GND	GND/'-'
GPIO4	DATA/'out'
5V0	VCC/'+'

Sample code

Note: sample code under the **Sample code** folder

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#define MAXTIMINGS    85
#define DHTPIN            7
int dht11_dat[5] = { 0, 0, 0, 0, 0 };

void read_dht11_dat()
{
        uint8_t laststate       = HIGH;
        uint8_t counter         = 0;
        uint8_t j               = 0, i;
        float       f; /* fahrenheit */

        dht11_dat[0] = dht11_dat[1] = dht11_dat[2] = dht11_dat[3] = dht11_dat[4] = 0;

        /* pull pin down for 18 milliseconds */
        pinMode( DHTPIN, OUTPUT );
```

```
    digitalWrite( DHTPIN, LOW );
    delay( 18 );
    /* then pull it up for 40 microseconds */
    digitalWrite( DHTPIN, HIGH );
    delayMicroseconds( 40 );
    /* prepare to read the pin */
    pinMode( DHTPIN, INPUT );

    /* detect change and read data */
    for ( i = 0; i < MAXTIMINGS; i++ )
    {
        counter = 0;
        while ( digitalRead( DHTPIN ) == laststate )
        {
            counter++;
            delayMicroseconds( 1 );
            if ( counter == 255 )
            {
                break;
            }
        }
        laststate = digitalRead( DHTPIN );

        if ( counter == 255 )
            break;

        /* ignore first 3 transitions */
        if ( (i >= 4) && (i % 2 == 0) )
        {
            /* shove each bit into the storage bytes */
            dht11_dat[j / 8] <<= 1;
            if ( counter > 16 )
                dht11_dat[j / 8] |= 1;
            j++;
        }
    }

    /*
     * check we read 40 bits (8bit x 5 ) + verify checksum in the last byte
     * print it out if data is good
     */
    if ( (j >= 40) &&
             (dht11_dat[4] == ( (dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) &
0xFF) ) )
```

```
    {
        f = dht11_dat[2] * 9. / 5. + 32;
        printf( "Humidity = %d.%d %% Temperature = %d.%d *C (%.1f *F)\n",
            dht11_dat[0], dht11_dat[1], dht11_dat[2], dht11_dat[3], f );
    }else   {
        printf( "Data not good, skip\n" );
    }
}

int main( void )
{
    printf( "Welcome to Smraza\n");
    printf( "Raspberry Pi wiringPi DHT11 Temperature test program\n" );
    printf( "Press Ctrl+C to exit\n" );
    if ( wiringPiSetup() == -1 )
        exit( 1 );

    while ( 1 )
    {
        read_dht11_dat();
        delay( 1000 ); /* wait 1sec to refresh */
    }

    return(0);
}
```

Compiling: gcc -Wall -o DHT11 DHT11.c -lwiringPi

Run: sudo ./DHT11

Tips: Press “Ctrl+C” to exit

Application effect

When you are running program, you will see the parameters returned by the DHT11 module.