



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Waterproof DS18B20 Digital Temperature Sensor (SKU:DFR0198)



Contents

- [1 Introduction](#)
 - [1.1 Specification](#)
- [2 Sensor Connection](#)
- [3 Sample Code](#)
- [4 Additional documentation](#)

Introduction

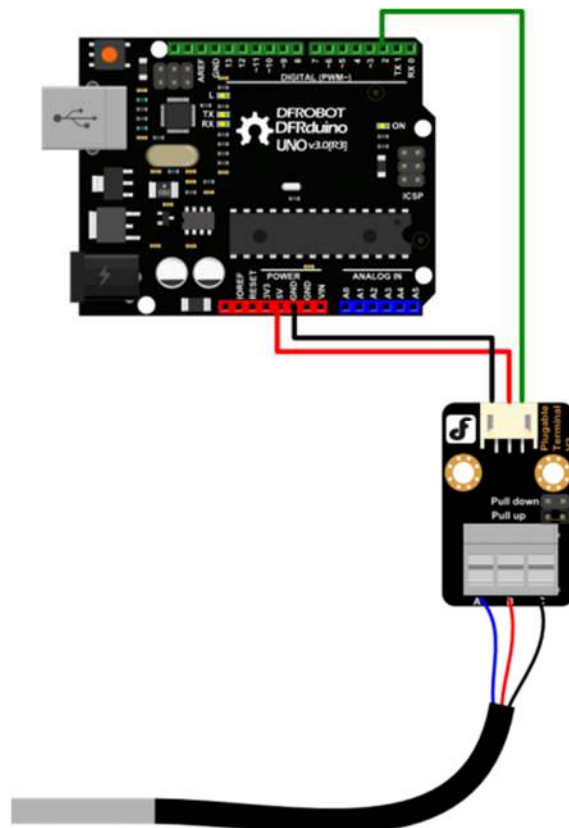
This is a waterproofed version of the **DS18B20 Arduino Temperature sensor**. Handy for when you need to measure something far away, or in wet conditions. While the sensor is good up to 125°C the cable is jacketed in PVC so we suggest keeping it under 100°C. Because they are digital, you don't get any signal degradation even over long distances! The DS18B20 provides 9 to 12-bit (configurable) temperature readings over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor. Usable with 3.0-5.5V systems. Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

Specification

- Usable with 3.0V to 5.5V power/data
- $\pm 0.5^{\circ}\text{C}$ Accuracy from -10°C to $+85^{\circ}\text{C}$
- Usable temperature range: -55 to 125°C (-67°F to $+257^{\circ}\text{F}$)
- 9 to 12 bit selectable resolution
- Uses 1-Wire interface- requires only one digital pin for communication
- Unique 64 bit ID burned into chip
- Multiple sensors can share one pin
- Temperature-limit alarm system
- Query time is less than 750ms
- 3 wires interface:
 - Red wire - VCC
 - Black wire - GND
 - Yellow wire - DATA
- Stainless steel tube 6mm diameter by 35mm long
- Cable diameter: 4mm
- Length: 90cm

Sensor Connection

This sensor requires a 4.7K Ohm resistor between the voltage and Signal pin. as seen in the picture below. Optionally you can use a Plugable Terminal sensor adapter to help in making this connection secure.



Sample Code

Sample code for Arduino 1.0 and above.

```
#include <OneWire.h>

int DS18S20_Pin = 2; //DS18S20 Signal pin on digital 2

//Temperature chip i/o
OneWire ds(DS18S20_Pin); // on digital pin 2

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  float temperature = getTemp();
  Serial.println(temperature);

  delay(100); //just here to slow down the output so it is easier to read
}

float getTemp(){
  //returns the temperature from one DS18S20 in DEG Celsius

  byte data[12];
  byte addr[8];

  if ( !ds.search(addr)) {
    //no more sensors on chain, reset search
    ds.reset_search();
    return -1000;
  }
```

```
if ( OneWire::crc8( addr, 7) != addr[7]) {
    Serial.println("CRC is not valid!");
    return -1000;
}

if ( addr[0] != 0x10 && addr[0] != 0x28) {
    Serial.print("Device is not recognized");
    return -1000;
}

ds.reset();
ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at the end

byte present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad

for (int i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read();
}

ds.reset_search();

byte MSB = data[1];
byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB); //using two's compliment
float TemperatureSum = tempRead / 16;

return TemperatureSum;
```

```
}
```

Additional documentation

ZIP file With sample codes, datasheet, and required libraries.

<http://www.dfrobot.com/image/data/DFR0198/DFRobot%20DFR0198.zip>