SYNACORP TECHNOLOGIES SDN. BHD. (1310487-K)
No.25 Lorong 1/SS3, Bandar Tasek Mutiara,
14120 Simpang Ampat, Penang, Malaysia.
T: +604.586.0026 F: +604.586.0026
WEBSITE: www.synacorp.my EMAIL: sales@synacorp.my

REF: **FR1-18B20**

DS18B20 Temperature Sensor 18B20 TO-92

Description

The DS18B20 is a 1-Wire® temperature sensor manufactured by Dallas Semiconductor (acquired by Maxim Integrated). Because it is a 1-wire device, it only needs one digital pin to communicate with the microcontroller. The sensor is typically available in two form factors. One comes in a TO-92 package, which resembles a simple transistor. The other comes in the form of a waterproof probe, which is more useful when measuring something far away, underwater, or beneath the ground.

Specifications

- IC output type: digital
- Sensing accuracy range: ± 0.5°C
- Temperature detection range: -55°C to +125°C
- Supply current: 1mA
- Supply voltage range: 3V to 5.5V
- Resolution, bit: 12bit
- Package Type:TO-92
- Number of pins: 3

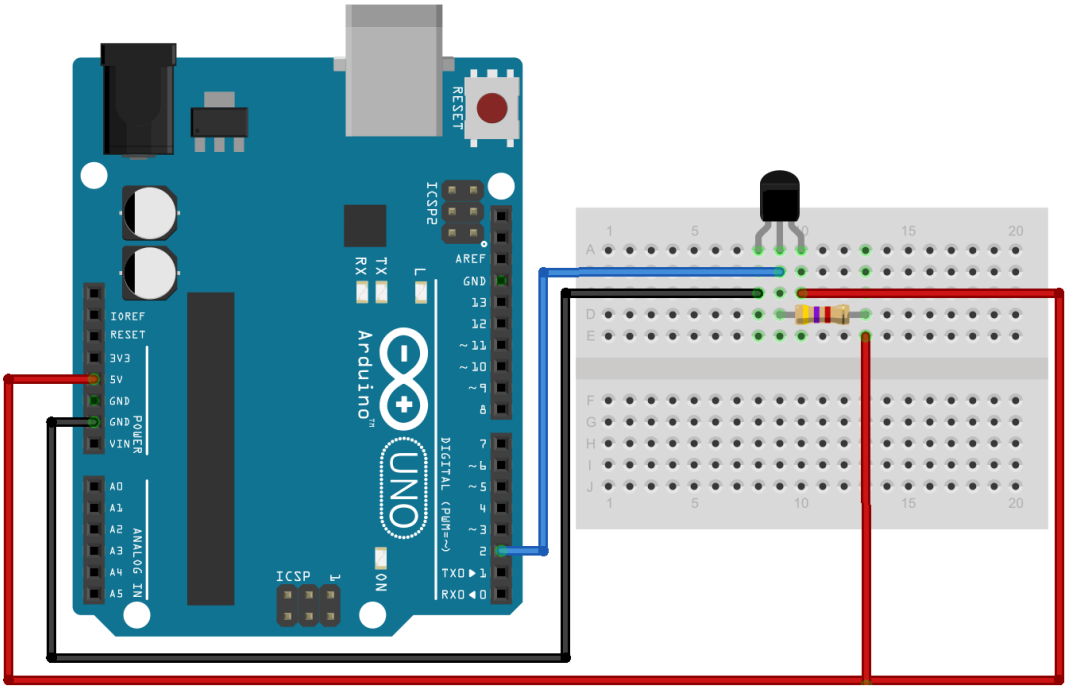
Pin connection

The connections are straightforward. Begin by connecting VDD to the Arduino's 5V pin and GND to ground. Connect the signal pin DQ to Arduino's digital pin 2. To keep the data transfer stable, you'll also need to connect the 4.7k pull-up resistor between the signal and power pins (Note: internal pull-ups on the arduino do not work here). To avoid overheating and damage, make sure the DS18B20 is connected properly.

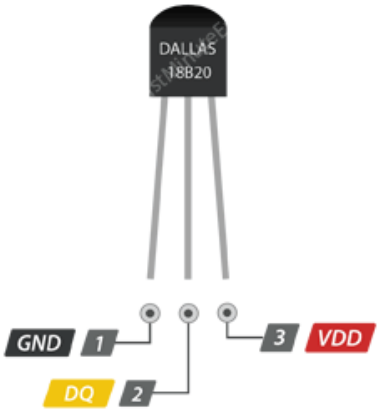
Adruino uno (PIN)	DS18B20 (PIN)
5V	VDD
GND	GND
D2	DQ
5V	4.7K pullup resistor

Circuit diagram

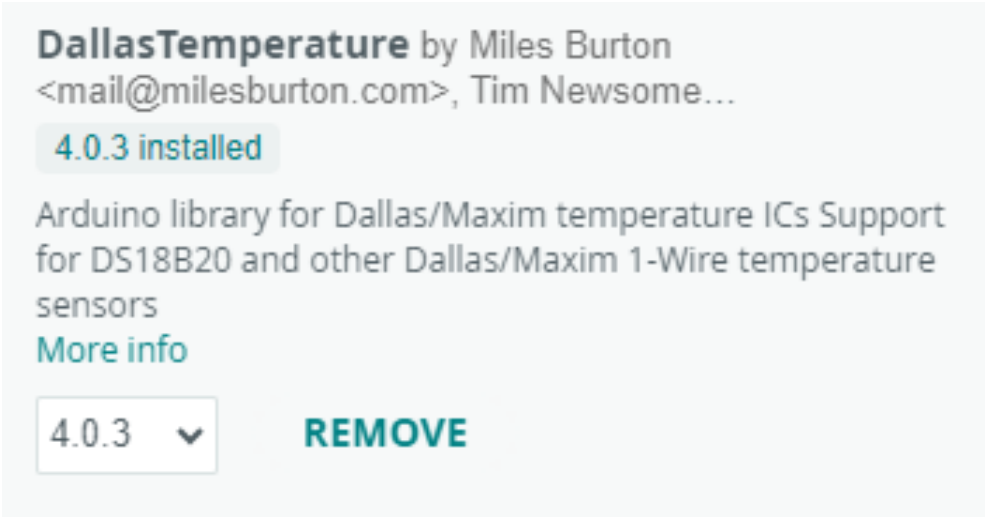
Connect the signal pin DQ to Arduino's digital pin 2. To keep the data transfer stable, you'll also need to connect the 4.7k pull-up resistor between the signal and power pins (Note: internal pull-ups on the arduino do not work here).

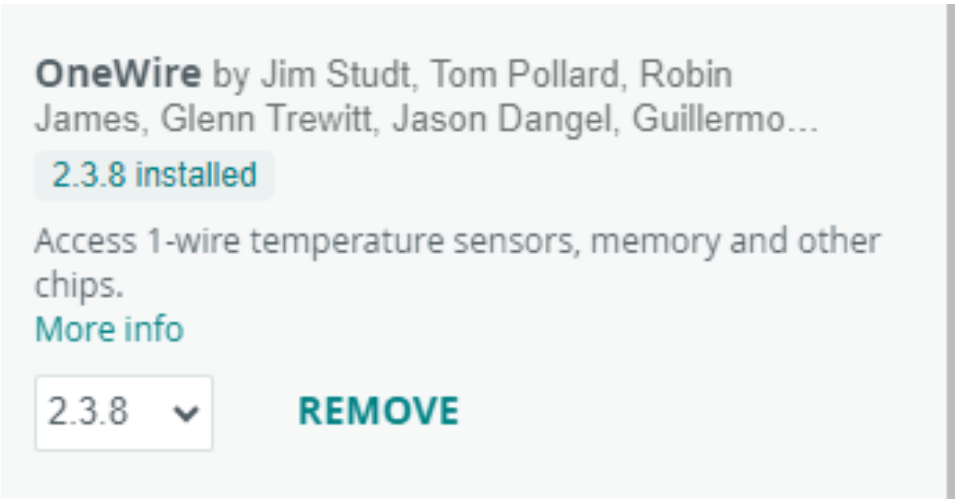


Pin diagram



Library





CODING

The sketch below will provide you with a thorough understanding of how to read temperature readings from a DS18B20 Temperature Sensor and can serve as the foundation for more practical experiments and projects.

Tester.ino

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into digital pin 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire device
OneWire oneWire(ONE_WIRE_BUS);

// Pass oneWire reference to DallasTemperature library
DallasTemperature sensors(&oneWire);

void setup(void)
{
  sensors.begin();  // Start up the library
  Serial.begin(9600);
}

void loop(void)
{
  // Send the command to get temperatures
  sensors.requestTemperatures();

  //print the temperature in Celsius
  Serial.print("Temperature: ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print((char)176);//shows degrees character
  Serial.print("C  |  ");

  //print the temperature in Fahrenheit
  Serial.print((sensors.getTempCByIndex(0) * 9.0) / 5.0 + 32.0);
  Serial.print((char)176);//shows degrees character
  Serial.println("F");

  delay(500);
}
```

Result

Here's what the output looks like on the serial monitor

```
11:15:44.329 -> Temperature: 29.25°C  |  84.65°F
11:15:44.968 -> Temperature: 29.31°C  |  84.76°F
11:15:45.493 -> Temperature: 29.31°C  |  84.76°F
11:15:46.138 -> Temperature: 29.25°C  |  84.65°F
11:15:46.721 -> Temperature: 29.25°C  |  84.65°F
11:15:47.307 -> Temperature: 29.25°C  |  84.65°F
11:15:47.848 -> Temperature: 29.25°C  |  84.65°F
11:15:48.578 -> Temperature: 29.31°C  |  84.76°F
11:15:49.040 -> Temperature: 29.31°C  |  84.76°F
11:15:49.641 -> Temperature: 29.31°C  |  84.76°F
11:15:50.230 -> Temperature: 29.31°C  |  84.76°F
11:15:50.840 -> Temperature: 29.31°C  |  84.76°F
11:15:51.381 -> Temperature: 29.31°C  |  84.76°F
11:15:52.000 -> Temperature: 29.31°C  |  84.76°F
11:15:52.586 -> Temperature: 29.31°C  |  84.76°F
11:15:53.189 -> Temperature: 29.25°C  |  84.65°F
```