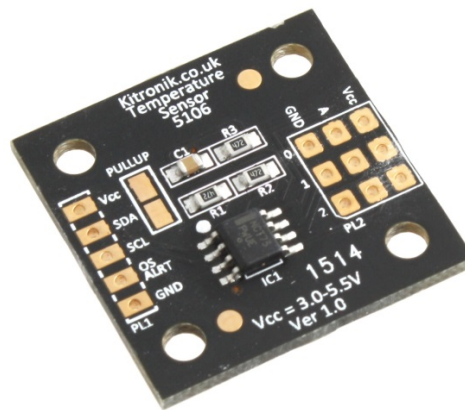


## Temperature Sensor Breakout - [5106](#)

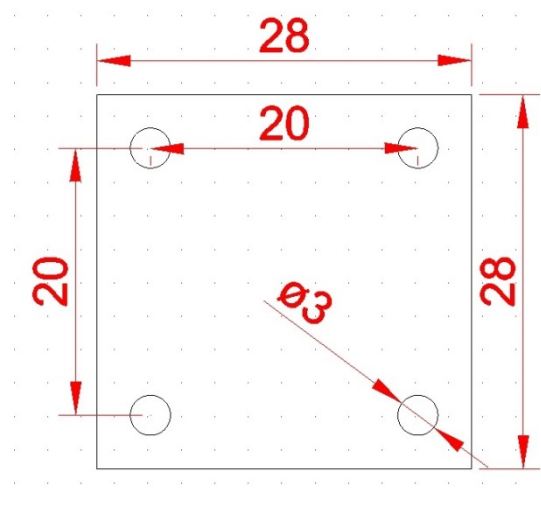
### Circuit Overview

This board uses the NCT75-D temperature sensor which has a temperature accuracy of up to 1°C and temperature resolution of 0.0625°C.

The device also features an over-temperature alert pin which is highly configurable along with 3 address pins allowing the user to select between 8 device addresses, meaning you can have up to 8 of these devices on the same bus, a useful feature for comparing temperatures at several locations at once.



### Mechanical Information



(All dimensions in mm)

## Electrical Information

Operating Supply Voltage	3.0 to 5.5V
Operating Ambient Temperature Range	-55 to 125 °C
ADC Resolution	12 Bit
Temperature Resolution	0.0625 °C
Accuracy at VDD = 3.0 V to 5.5 V	+/-1°C@ 0°C to +70°C +/-2°C@ -25°C to +100°C +/-3°C@ -55°C to +125°C
Update Rate	80ms
Supply Current	0.44mA to 0.8mA

## Pinout

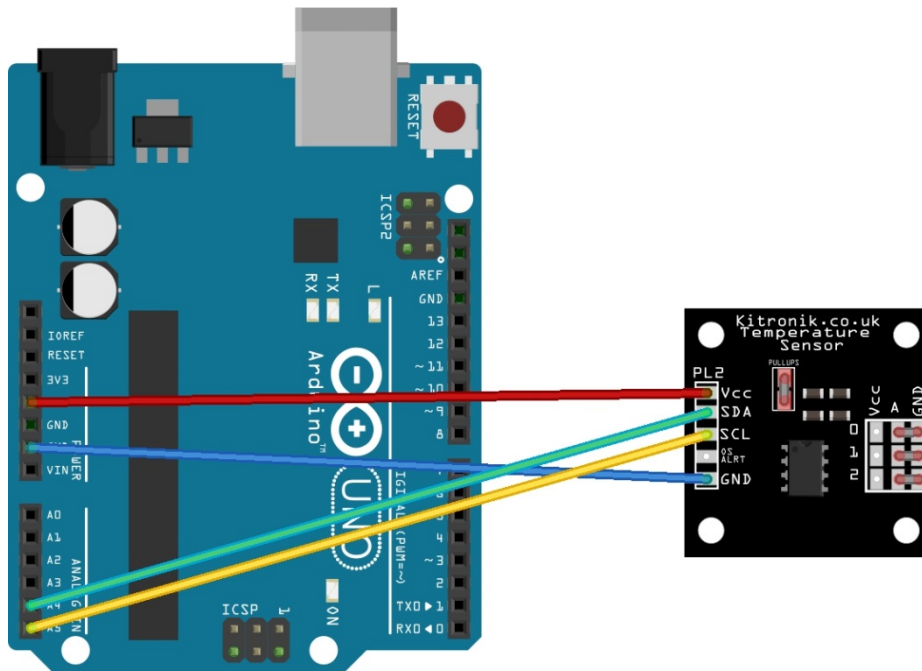
Vcc	Supply Voltage
A0	Address Pin 1
A1	Address Pin 2
A2	Address Pin 3
SDA	I2C Data
SCL	I2C Clock
OS/ALRT	Configurable over-temperature alarm
GND	Ground

## Configurable Address Pin

A2	A1	A0	Address
0	0	0	0x48
0	0	1	0x49
0	1	0	0x4A
0	1	1	0x4B
1	0	0	0x4C
1	0	1	0x4D
1	1	0	0x4E
1	1	1	0x4F

By linking the pins A1, A2 and A3 on the grid to either Vcc (represented in the truth table as 1) or GND (represented in the truth table as 0) the address can be set at different values as demonstrated in the truth table above. The example shown in the next section where all three address lines are connected to 0V would give an address of 0x48.

## Using the board with an Arduino



Made with  Fritzing.org

To begin measuring temperature with the temperature sensor connect the pins on the breakout board to your Arduino as described in the table below.

Temperature Sensor Board	Arduino
Vcc	5V
A0	GND
A1	GND
A2	GND
SDA	A4
SCL	A5
OS/ALRT	Not used
GND	GND

The code below assumes the device is set to its default address **0x48** (all address pins pulled low). Remember solder across the 'PULLUP' pads to activate the pull-up resistors if you don't already have any on the SDA/SCL lines!

The sketch can be copied and pasted directly into the Arduino IDE, it is also available as a download from the website. The sketch defines the function `getTemperature()` then calls the function at one second intervals and prints the temperature in Celsius to the serial monitor.

## Arduino Sketch

```
#include <Wire.h> //we include the Wire library as it deals with I2C
communication
int tempsensorAddress = 0x48; //here we tell the Arduino where the sensor can be
found on the I2C bus

void setup(){
  Serial.begin(9600); //in the setup we start serial communication and set the
baud rate
  Wire.begin();
}

void loop(){//this section prints the temperature to the serial monitor

  float celsius = getTemperature();
  Serial.print("Celsius: ");
  Serial.println(celsius);

  delay(1000); //here we choose how often the temperature is printed, here it is
set to once per second (1000 milliseconds)
}

float getTemperature(){ //this function deals with requesting data from the
sensor and converting it to a readable format
  Wire.requestFrom(tempsensorAddress,2); //here we request 2 bytes from the
temperature sensor

  byte MSB = Wire.read();//the Most Significant Byte is received first
  byte LSB = Wire.read();//the Least Significant Byte is received second

  //it's a 12bit integer, using two's compliment for negative
  int TemperatureSum = (MSB << 8) | LSB >> 4;

  float celsius = TemperatureSum*0.0625; //the sensor returns the temperature as
a 12 bit number, the sensor has a resolution of 0.0625 Celsius

  return celsius;
}
```