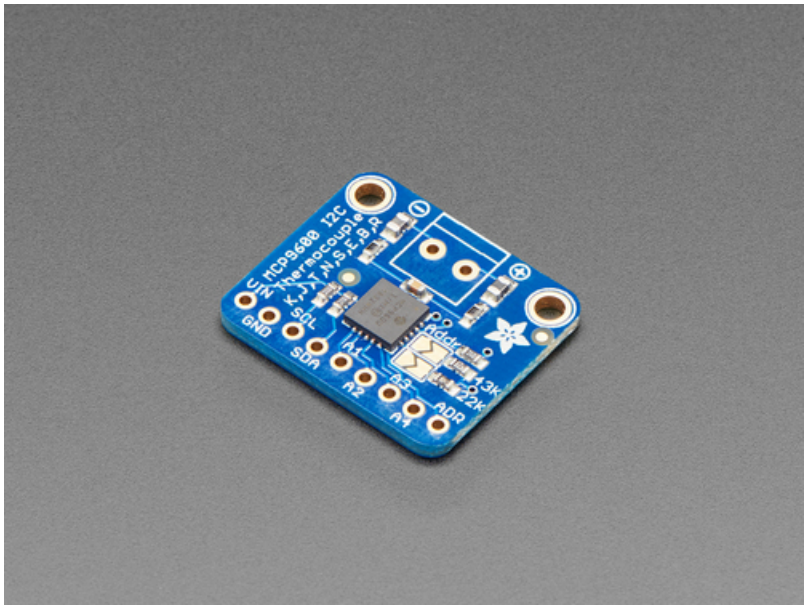
adafruit learning system

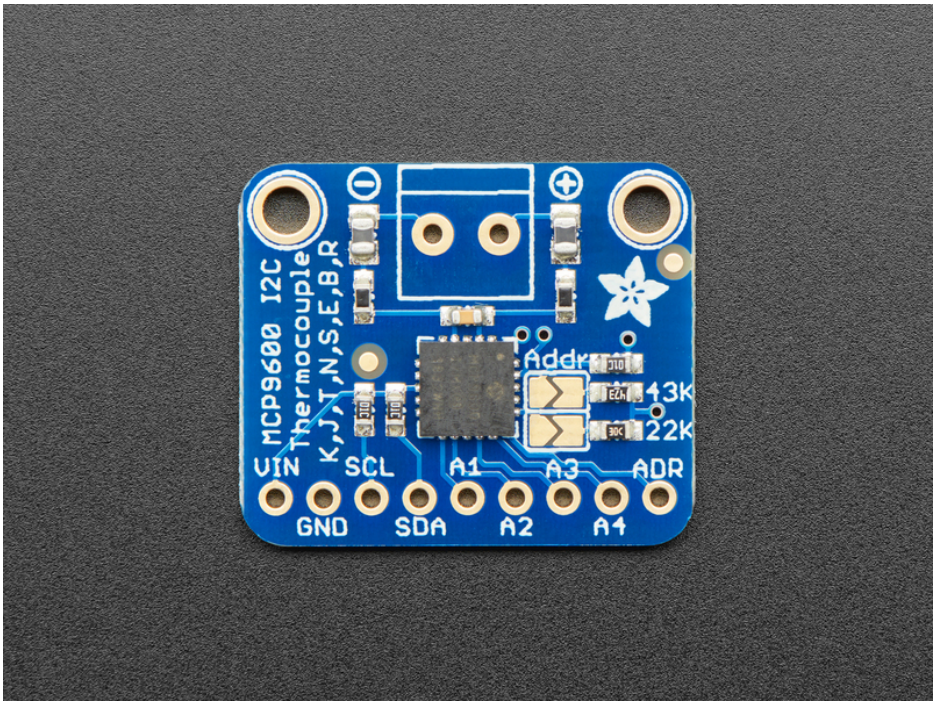
Adafruit MCP9600 I2C Thermocouple Amplifier

Created by Kattni Rembor



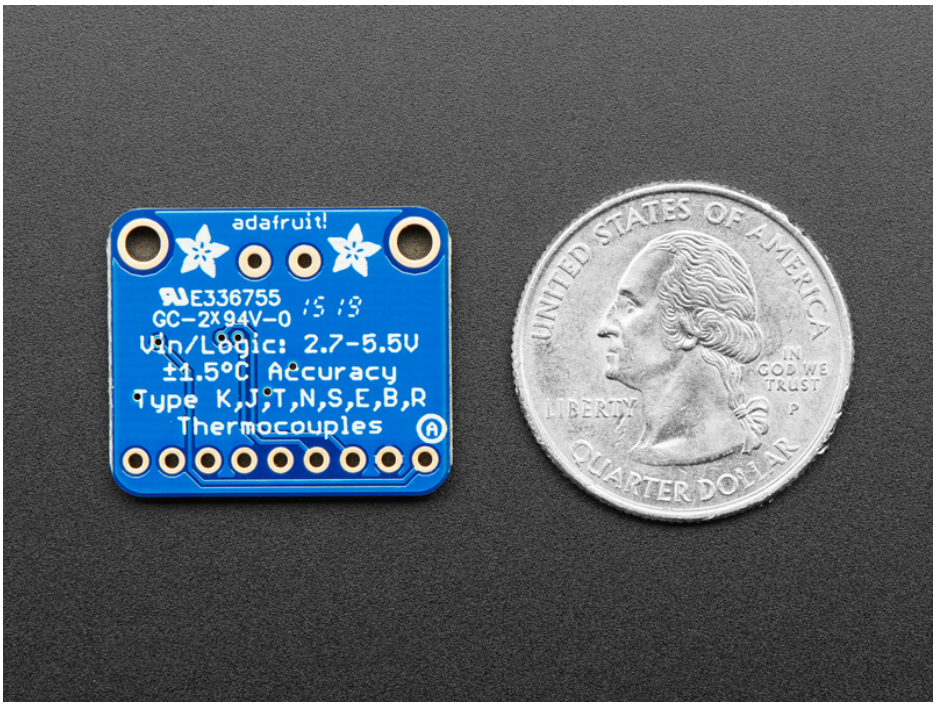
Last updated on 2019-07-01 01:57:47 PM UTC

Overview



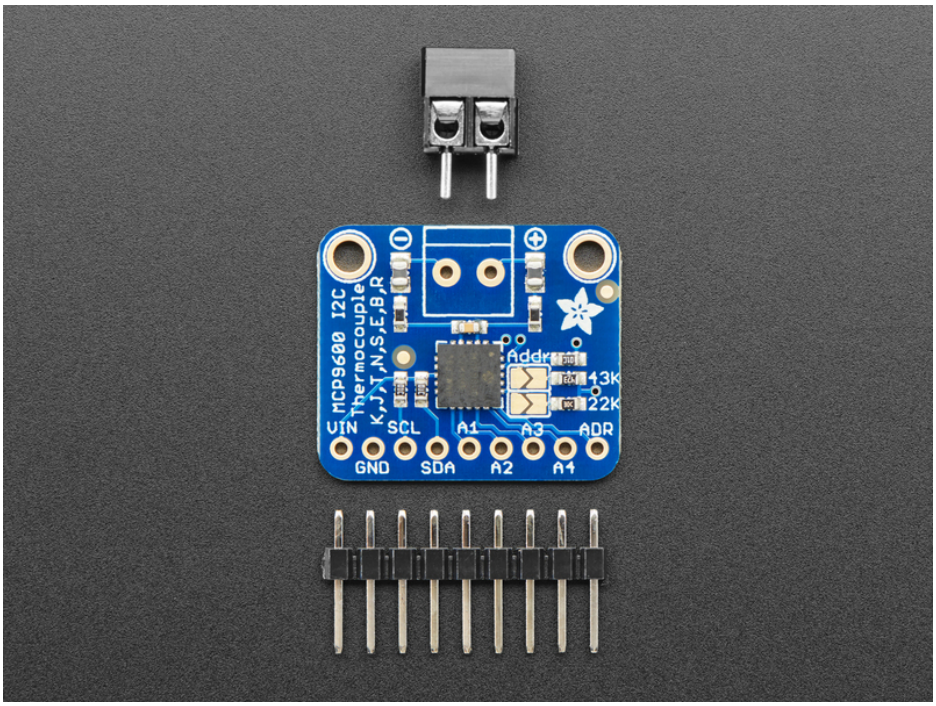
Thermocouples are very sensitive, requiring a good amplifier with a cold-compensation reference. The **Adafruit MCP9600** does all that for you, and can be easily interfaced with any microcontroller or single-board-computer with I2C. Inside, the chip handles all the analog stuff for you, and can interface with just about any thermocouple type: K, J, T, N, S, E, B and R type are all supported! You can also set various alerts for over/under temperature, and read the thermocouple (hot) temperature and the chip (cold) temperature. All this over common I2C.

This breakout board has the chip itself, a 3.3V regulator and level shifting circuitry, all assembled and tested. Works great with 3.3V *or* 5V logic. Comes with a 2 pin terminal block (for connecting to the thermocouple) and pin header (to plug into any breadboard or perfboard).

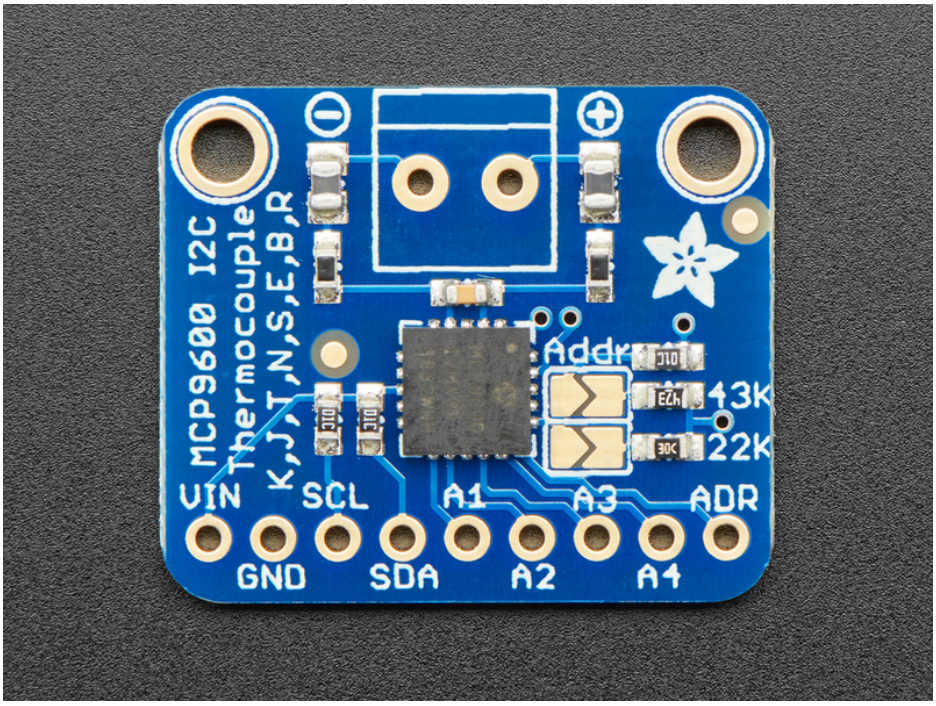


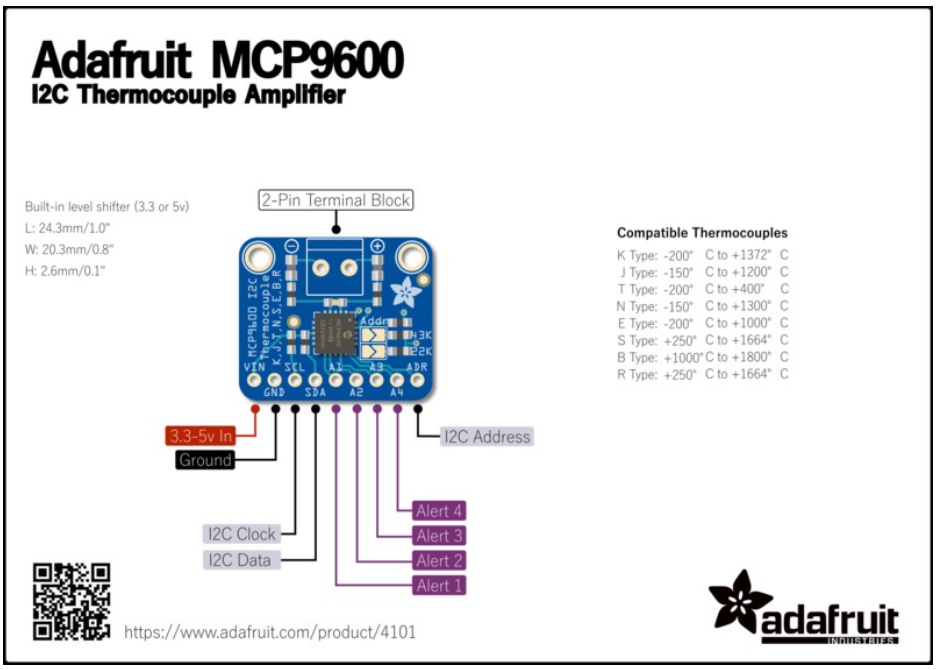
The Adafruit MCP9600 features:

- Works with any K, J, T, N, S, E, B and R type thermocouple
- Datasheet rated for:
  K Type: -200°C to +1372°C
  J Type: -150°C to +1200°C
  T Type: -200°C to +400°C
  N Type: -150°C to +1300°C
  E Type: -200°C to +1000°C
  S Type: +250°C to +1664°C
  B Type: +1000°C to +1800°C
  R Type: +250°C to +1664°C
- Resolution of ±0.0625 °C - note that K thermocouples have about ±2°C to ±6°C accuracy
- Internal temperature reading
- 3.3 to 5v power supply and logic level compliant
- I2C data connection


MCP9600 I2C
Thermocouple
K,J,T,N,S,E,B,R
Addr
4.3K
22K
VIN
SCL
A1
A3
ADR
GND
SDA
A2
A4

Pinouts





Power Pins

- **Vin** - this is the power pin. Since the sensor chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

I2C Logic Pins

- **SCL** - this is the I2C clock pin, connect to your microcontroller's I2C clock line.
- **SDA** - this is the I2C data pin, connect to your microcontroller's I2C data line.

Alert Pins

- **A1 - A4** - Alert 1 - Alert 4 output pins

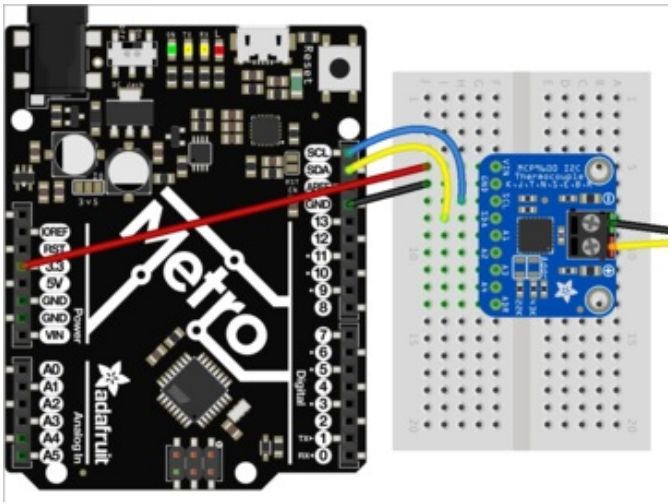
Address Pin

- **ADR** - Allows for setting I2C address

Arduino

Wiring

Connecting the MCP9600 to your Feather or Arduino is easy:



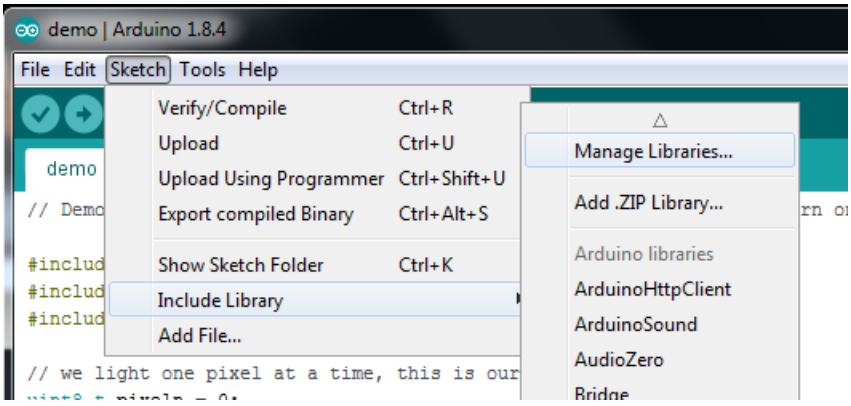
- If you are running a Feather (3.3V), connect **Feather 3V** to **board VIN**
- If you are running a 5V Arduino (Uno, etc.), connect **Arduino 5V** to **board VIN**
- Connect **Feather or Arduino GND** to **board GND**
- Connect **Feather or Arduino SCL** to **board SCL**
- Connect **Feather or Arduino SDA** to **board SDA**
- Connect **thermocouple** + to **board screw terminal +**
- Connect **thermocouple** - to **board screw terminal -**

The final results should resemble the illustration above, showing an Adafruit Metro development board.

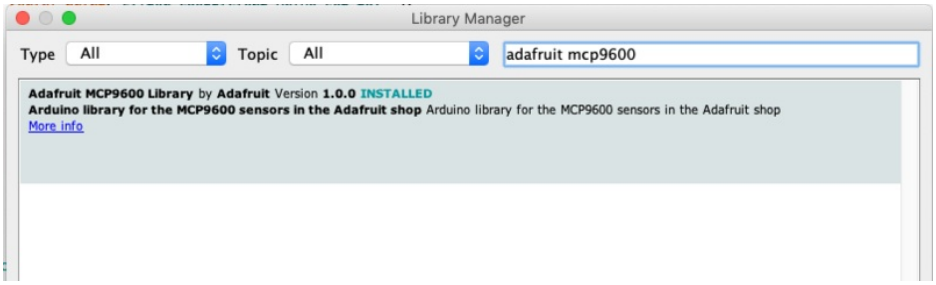
The MCP9600 will return a temperature for the hot junction even if there is no thermocouple connected. There will not be an error!

Installation

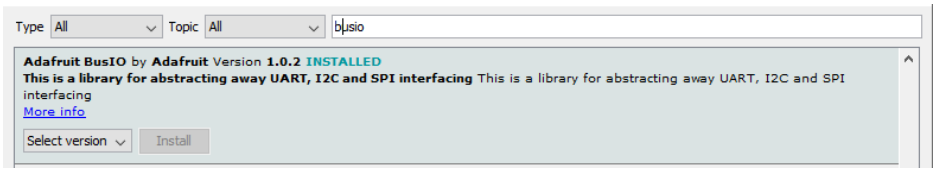
You can install the **Adafruit MCP9600 Library** for Arduino using the Library Manager in the Arduino IDE:



Click the **Manage Libraries ...** menu item, search for **Adafruit MCP9600**, and select the **Adafruit MCP9600** library:



Also get the **Adafruit BusIO** library



Load Example

Open up **File -> Examples -> Adafruit MCP9600 -> mcp9600_test** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools->Serial Monitor**). You should see the the values for hot junction, cold junction and ADC.

Example Code

The following example code is part of the standard library, but illustrates how you can retrieve sensor data from the MCP9600 for the hot junction, cold junction and ADC values:

```
#include <Wire.h>
#include <Adafruit_I2CDevice.h>
#include <Adafruit_I2CRegister.h>
#include "Adafruit_MCP9600.h"

#define I2C_ADDRESS (0x67)

Adafruit_MCP9600 mcp;
Adafruit_I2CDevice i2c_dev = Adafruit_I2CDevice(I2C_ADDRESS);

void setup()
{
    Serial.begin(115200);
    while (!Serial) {
      delay(10);
    }
    Serial.println("MCP9600 HW test");

    /* Initialise the driver with I2C_ADDRESS and the default I2C bus. */
    if (! mcp.begin()) {
        Serial.println("Sensor not found. Check wiring!");
        while (1);
    }

  Serial.println("Found MCP9600!");

  mcp.setADCresolution(MCP9600_ADCRESOLUTION_18);
  Serial.print("ADC resolution set to ");
```

```
  Serial.print("ADC resolution set to ");
  switch (mcp.getADCresolution()) {
    case MCP9600_ADCRESOLUTION_18:   Serial.print("18"); break;
    case MCP9600_ADCRESOLUTION_16:   Serial.print("16"); break;
    case MCP9600_ADCRESOLUTION_14:   Serial.print("14"); break;
    case MCP9600_ADCRESOLUTION_12:   Serial.print("12"); break;
  }
  Serial.println(" bits");

  mcp.setThermocoupleType(MCP9600_TYPE_K);
  Serial.print("Thermocouple type set to ");
  switch (mcp.getThermocoupleType()) {
    case MCP9600_TYPE_K:  Serial.print("K"); break;
    case MCP9600_TYPE_J:  Serial.print("J"); break;
    case MCP9600_TYPE_T:  Serial.print("T"); break;
    case MCP9600_TYPE_N:  Serial.print("N"); break;
    case MCP9600_TYPE_S:  Serial.print("S"); break;
    case MCP9600_TYPE_E:  Serial.print("E"); break;
    case MCP9600_TYPE_B:  Serial.print("B"); break;
    case MCP9600_TYPE_R:  Serial.print("R"); break;
  }
  Serial.println(" type");

  mcp.setFilterCoefficient(3);
  Serial.print("Filter coefficient value set to: ");
  Serial.println(mcp.getFilterCoefficient());

  mcp.setAlertTemperature(1, 30);
  Serial.print("Alert #1 temperature set to ");
  Serial.println(mcp.getAlertTemperature(1));
  mcp.configureAlert(1, true, true);  // alert 1 enabled, rising temp

  mcp.enable(true);

  Serial.println(F("------------------------------"));
}

void loop()
{
  Serial.print("Hot Junction: "); Serial.println(mcp.readThermocouple());
  Serial.print("Cold Junction: "); Serial.println(mcp.readAmbient());
  Serial.print("ADC: "); Serial.print(mcp.readADC() * 2); Serial.println(" uV");
  delay(1000);
}
```

You should get something resembling the following output when you open the Serial Monitor at 9600 baud:


/dev/cu.usbmodem144444231 (Adafruit Feather M4 Express (SAMD51))
Send
MCP9600 HW test
Found MCP9600!
ADC resolution set to 18 bits
Thermocouple type set to K type
Filter coefficient value set to: 3
Alert #1 temperature set to 30.00
Hot Junction: 21.44
Cold Junction: 21.37
ADC: -2 uV
Hot Junction: 21.37
Cold Junction: 21.44
ADC: -2 uV
Hot Junction: 21.37
Cold Junction: 21.37
ADC: -2 uV
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

Arduino Docs

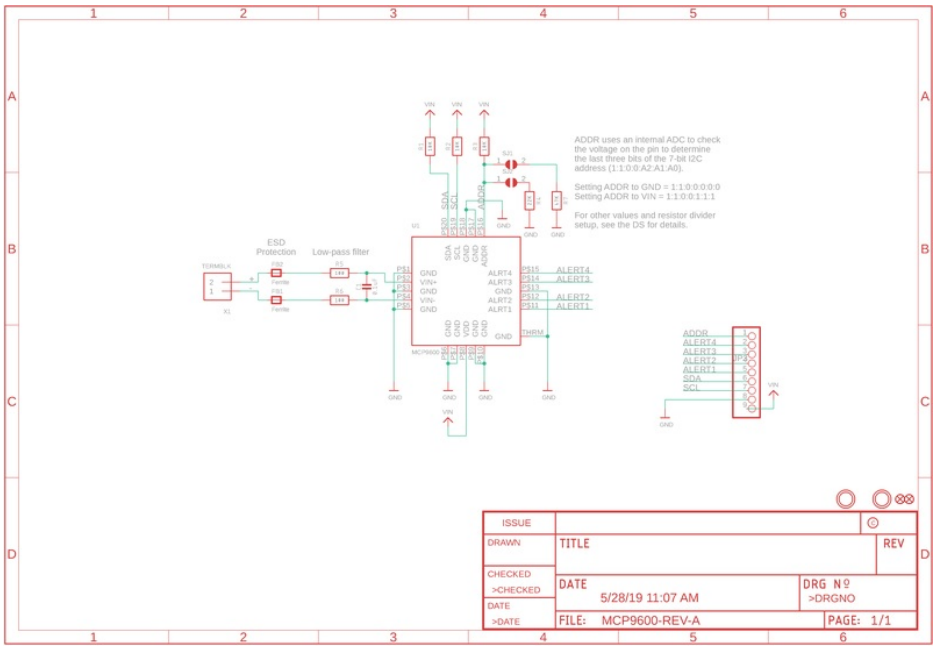
Arduino Docs (https://adafru.it/EZR)

Downloads

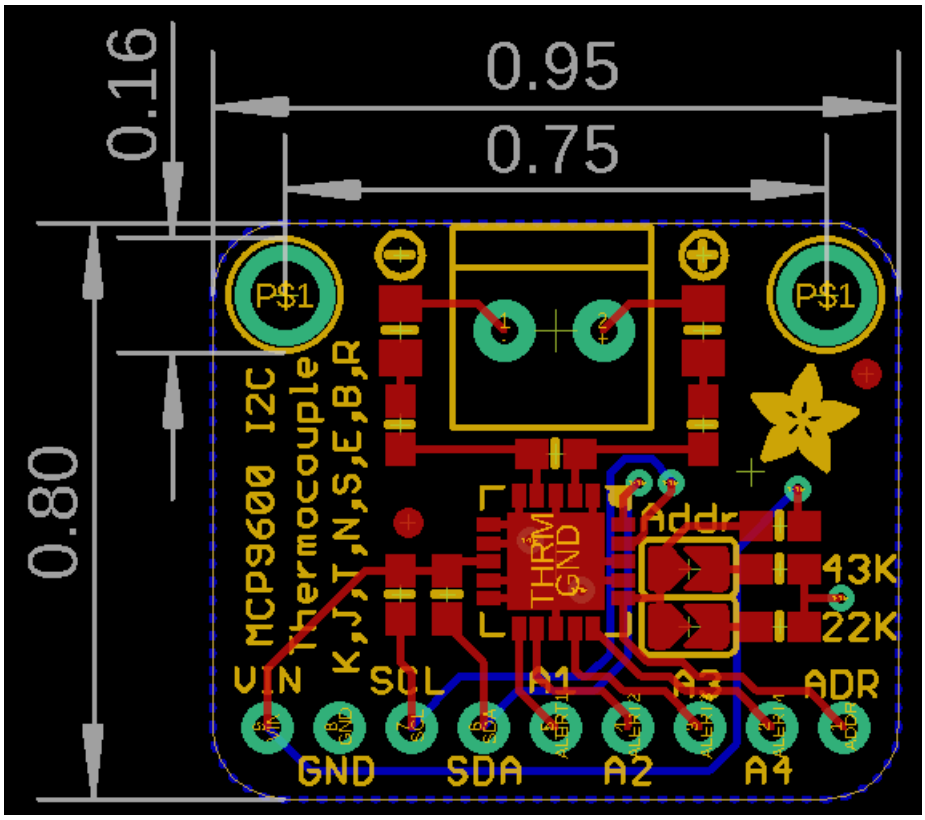
Files

- MCP9600 Datasheet (https://adafru.it/EYI)
- EagleCAD files on GitHub (https://adafru.it/EYJ)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/EYK)

Schematic



Fab Print


0.16
0.95
0.75
0.80
P$1
P$1
MCP9600 I2C
Thermocouple
K,J,T,N,S,E,B,R
VIN
SCL
A1
A3
ADR
GND
SDA
A2
A4
THRM
GND
Addr
43K
22K