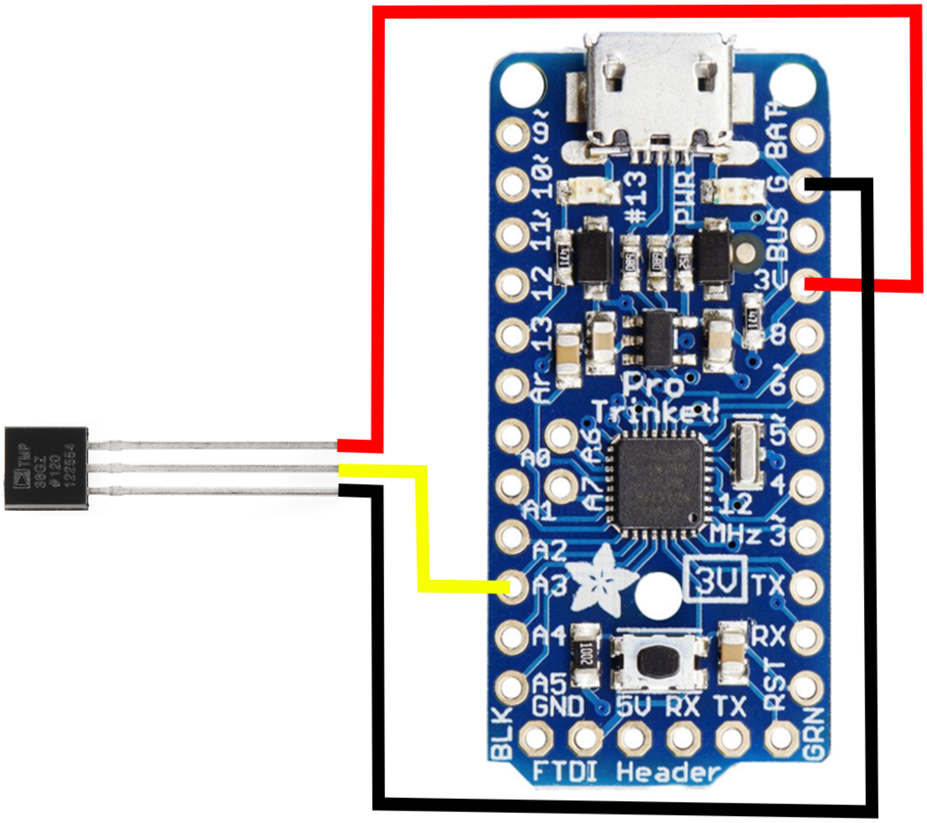
TMP-36 Integration

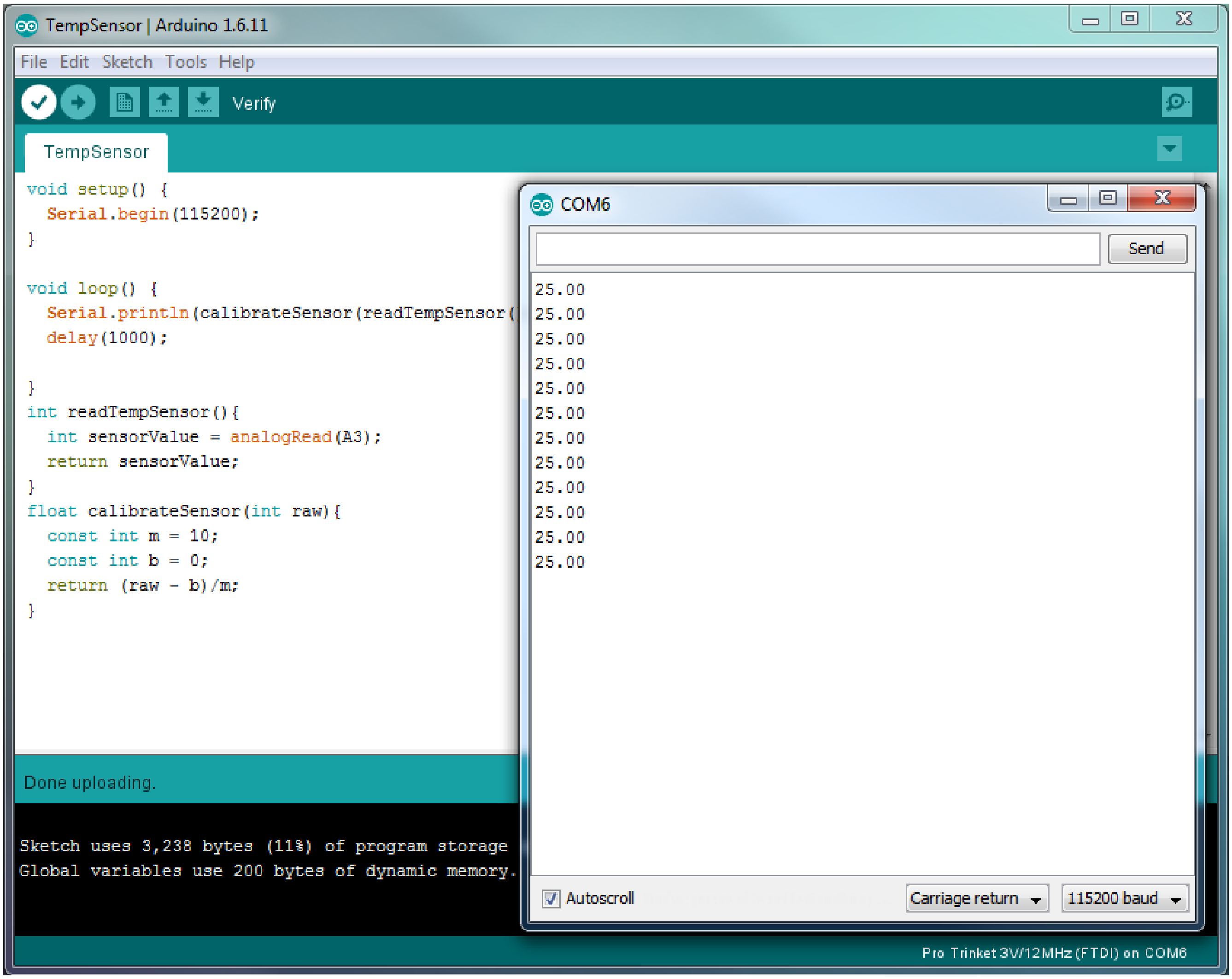
In this document, we will connect the TMP-36 temperature sensor to our Arduino by means of a standard analog connection. We will then read some raw measurements from the device. We will then use a calibration function to turn those into real units. While we specifically work with the TMP-36 in this example, the principals here could easily be applied to any 3 pin analog sensor.

Part 1 Wiring
The TMP-36 has only 3 pins: power, ground and data. Wire the TMP-36 to the Arduino as shown in the schematic below. Remember that your prototyping boards provide power and ground rails, so you don't need to bring power and ground all the way back to the pins on the Arduino to connect power and ground.



Part 2 TMP-36 Software

ext, open the sketch *RisingData/src/TempSensor/TempSensor.ino*. You can upload the sketch as written and then open the serial monitor to see the output of the TMP-36 device.



```
void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println(calibrateSensor(readTempSensor(
  delay(1000);

}
int readTempSensor(){
  int sensorValue = analogRead(A3);
  return sensorValue;
}
float calibrateSensor(int raw){
  const int m = 10;
  const int b = 0;
  return (raw - b)/m;
}
```

```
25.00
25.00
25.00
25.00
25.00
25.00
25.00
25.00
25.00
25.00
25.00
25.00
```

☑ Autoscroll   Carriage return   115200 baud

Done uploading.

```
Sketch uses 3,238 bytes (11%) of program storage
Global variables use 200 bytes of dynamic memory.
```

Pro Trinket 3V/12MHz (FTDI) on COM6

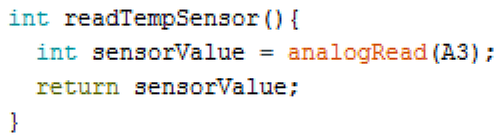
Appendix: Code Tour

```
void setup() {
  Serial.begin(115200);
}
```

The only thing we do in the setup is to start up our serial connection for debugging.

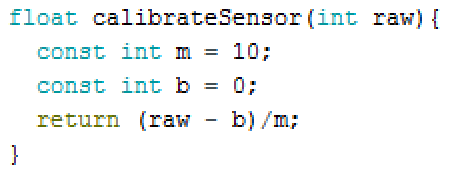
```
void loop() {
  Serial.println(calibrateSensor(readTempSensor()));
  delay(1000);

}
```

Our main program loop reads the raw value from the temp sensor, calibrates it, and prints it. This is accomplished by using the following functions.



```
int readTempSensor(){
  int sensorValue = analogRead(A3);
  return sensorValue;
}
```

This function polls analog sensor 3, to which we’ve attached our temperature sensor, and returns that value as an integer.



```
float calibrateSensor(int raw){
  const int m = 10;
  const int b = 0;
  return (raw - b)/m;
}
```

This function expects an integer as input and uses a linear interpolation to calculate the appropriate value:

NOTE: I did have time to actually do a calibration before writing this draft. We need to replace the m and b values with real numbers.