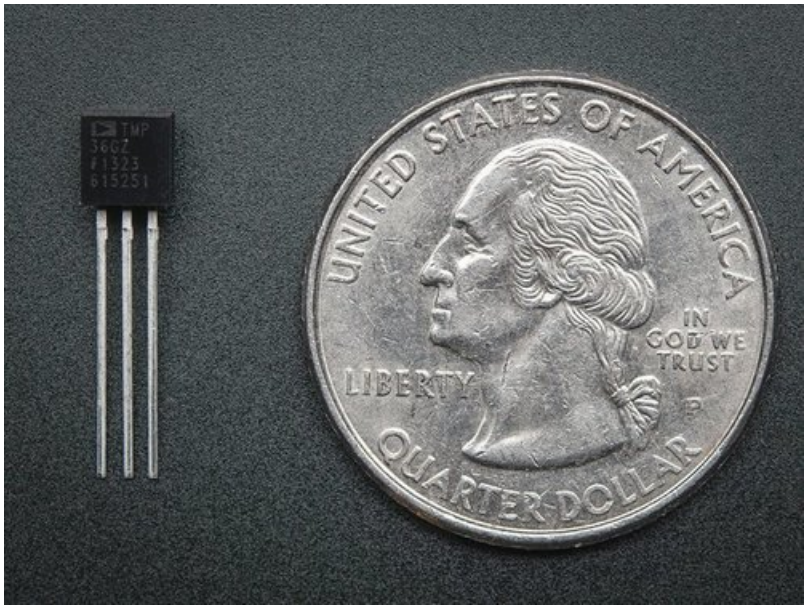
adafruit learning system

TMP36 Temperature Sensor

Created by lady ada

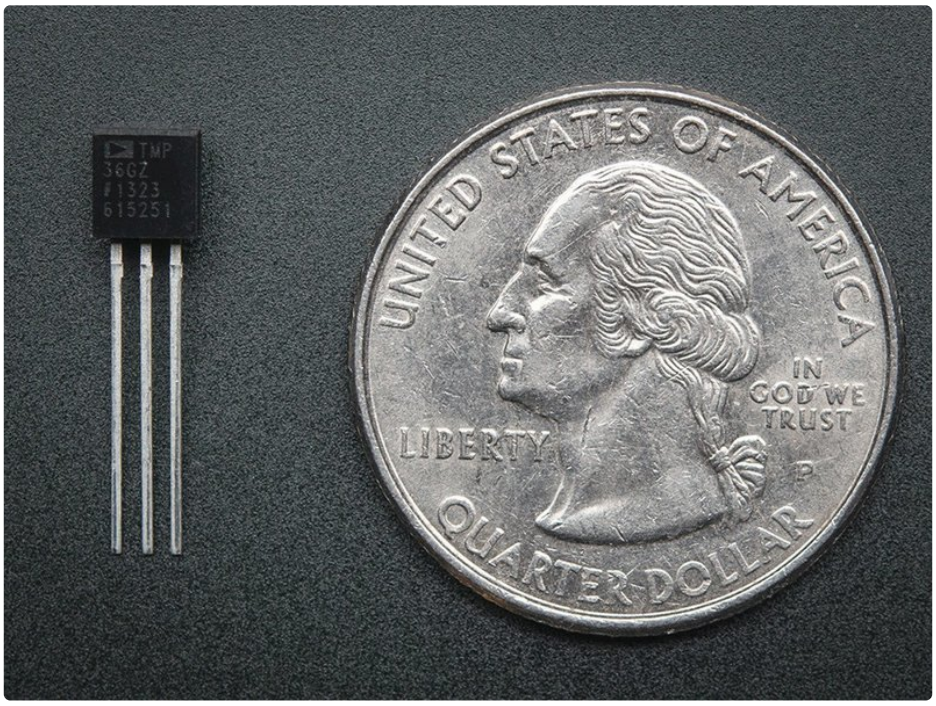


Last updated on 2021-10-22 10:55:10 AM EDT

Guide Contents

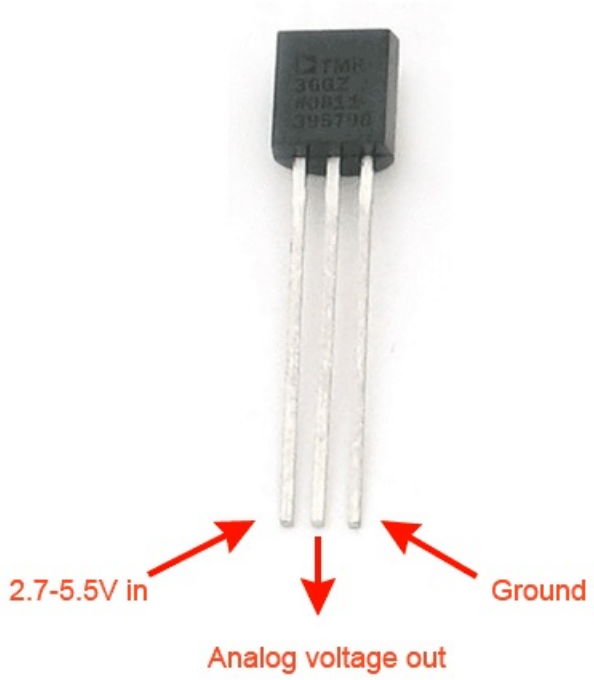
Overview

An analog temperature sensor is pretty easy to explain, its a chip that tells you what the ambient temperature is!



These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetalic strips (https://adafru.it/aKJ) (like in some home thermometers or stoves), nor do they use thermistors (https://adafru.it/aK6)(temperature sensitive resistors). Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor.) By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.

The good news is all that complex calculation is done *inside* the chip - it just spits out the temperature, ready for you to use!



Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistant between sensors and readings. Moreover they are very inexpensive and quite easy to use.

Some Basic Stats

These stats are for the temperature sensor in the Adafruit shop, the Analog Devices TMP36 (https://adafru.it/clW) (-40 to 150C). Its very similar to the LM35/TMP35 (Celsius output) and LM34/TMP34 (Farenheit output). The reason we went with the '36 instead of the '35 or '34 is that this sensor has a very wide range and doesn't require a negative voltage to read sub-zero temperatures. Otherwise, the functionality is basically the same.

- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** $1.50 at the Adafruit shop (https://adafru.it/aIH)
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw
- Datasheet **(https://adafru.it/clW)**

How to Measure Temperature

Using the TMP36 is easy, simply connect the left pin to power (2.7-5.5V) and the right pin to ground. Then the middle pin will have an analog voltage that is directly proportional (linear) to the temperature. The

analog voltage is independent of the power supply.

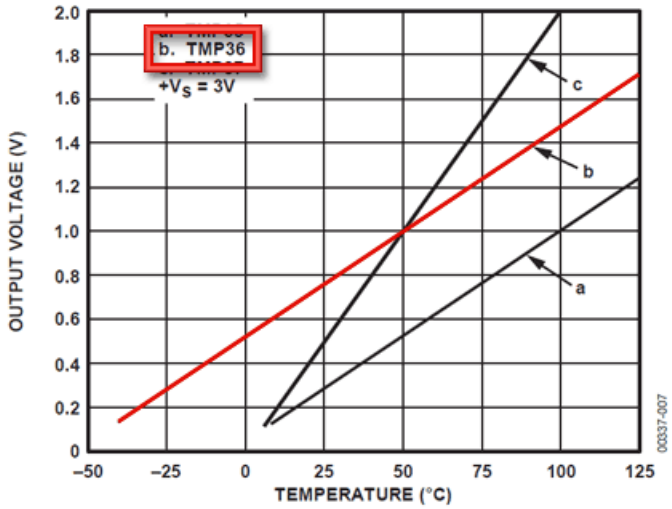


Figure 6. Output Voltage vs. Temperature

To convert the voltage to temperature, simply use the basic formula:

Temp in °C = [**(Vout in mV) - 500**] / **10**

So for example, if the voltage out is 1V that means that the temperature is **((1000 mV - 500) / 10) = 50 °C**

If you're using a LM35 or similar, use line 'a' in the image above and the formula: **Temp in °C** = **(Vout in mV) / 10**

Problems you may encounter with multiple sensors:

If, when adding more sensors, you find that the temperature is inconsistant, this indicates that the sensors are interfering with each other when switching the analog reading circuit from one pin to the other. You can fix this by doing two delayed readings and tossing out the first one

[See this post for more information](https://adafru.it/aKL) (https://adafru.it/aKL)

Testing a Temp Sensor

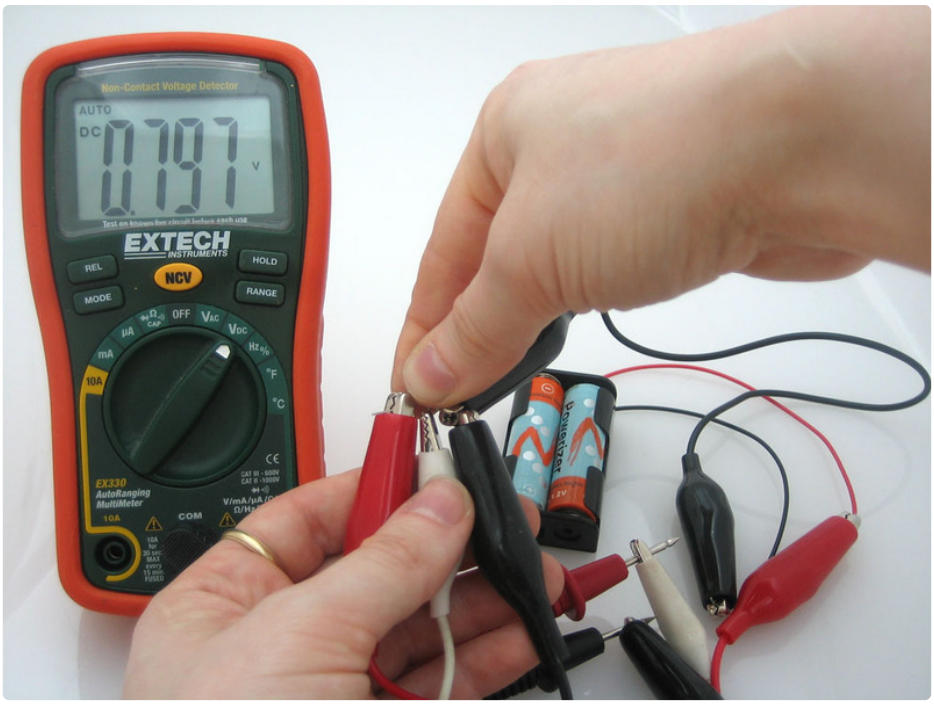
Testing these sensors is pretty easy but you'll need a battery pack or power supply.

Connect a 2.7-5.5V power supply (2-4 AA batteries work fantastic) so that ground is connected to pin 3 (right pin), and power is connected to pin 1 (left pin)

Then connect your multimeter in DC voltage mode to ground and the remaining pin 2 (middle). If you've got a TMP36 and its about room temperature (25°C), the voltage should be about 0.75V. Note that if you're using a LM35, the voltage will be 0.25V

The sensor is indicating that the temperature is 26.3°C also known as 79.3°F

You can change the voltage range by pressing the plastic case of the sensor with your fingers, you will see the temperature/voltage rise.



With my fingers on the sensor, heating it up a little, the temperature reading is now 29.7°C / 85.5°F

Or you can touch the sensor with an ice cube, perferrably in a plastic bag so it doesn't get water on your circuit, and see the temperature/voltage drop.

I pressed an ice-cube against the sensor, to bring the temperature down to 18.6°C / 65.5°F

Using a Temp Sensor

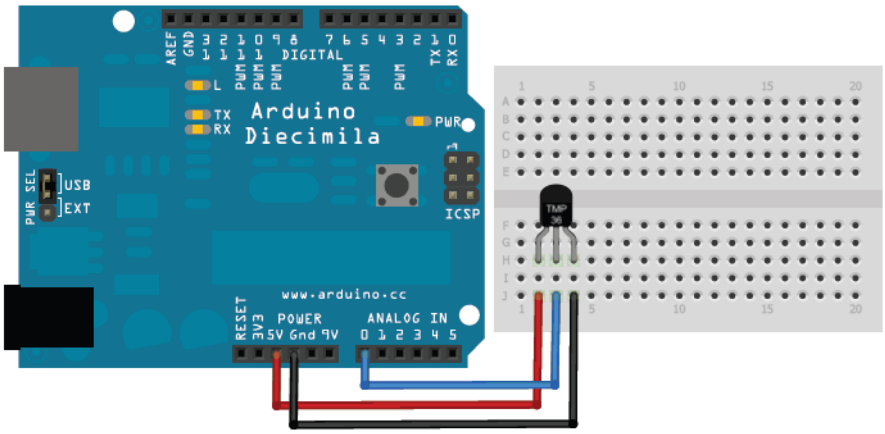
Connecting to a Temperature Sensor

These sensors have little chips in them and while they're not that delicate, they do need to be handled properly. Be careful of static electricity when handling them and make sure the power supply is connected up correctly and is between 2.7 and 5.5V DC - so don't try to use a 9V battery!

They come in a "TO-92" package which means the chip is housed in a plastic hemi-cylinder with three legs. The legs can be bent easily to allow the sensor to be plugged into a breadboard. You can also solder to the pins to connect long wires. [If you need to waterproof the sensor, you can see below for an Instructable for how to make an excellent case.](https://adafru.it/Oa8) (https://adafru.it/Oa8)

Reading the Analog Temperature Data

Unlike the FSR or photocell sensors we have looked at, the TMP36 and friends doesn't act like a resistor. Because of that, there is really only one way to read the temperature value from the sensor, and that is plugging the output pin directly into an Analog (ADC) input.



Remember that you can use anywhere between 2.7V and 5.5V as the power supply. For this example I'm showing it with a 5V supply but note that you can use this with a 3.3v supply just as easily. No matter what supply you use, the analog voltage reading will range from about 0V (ground) to about 1.75V.

If you're using a 5V Arduino, and connecting the sensor directly into an Analog pin, you can use these formulas to turn the 10-bit analog reading into a temperature:

Voltage at pin in milliVolts = (*reading from ADC*) * **(5000/1024)**
This formula converts the number 0-1023 from the ADC into 0-5000mV (= 5V)

If you're using a 3.3V Arduino, you'll want to use this:

Voltage at pin in milliVolts = **(** ***reading from ADC*****) * (3300/1024)**
This formula converts the number 0-1023 from the ADC into 0-3300mV (= 3.3V)

Then, to convert millivolts into temperature, use this formula:

Centigrade temperature = **[(analog voltage in mV) - 500] / 10**

Arduino Sketch - Simple Thermometer

This example code for Arduino shows a quick way to create a temperature sensor, it simply prints to the serial port what the current temperature is in both Celsius and Fahrenheit.

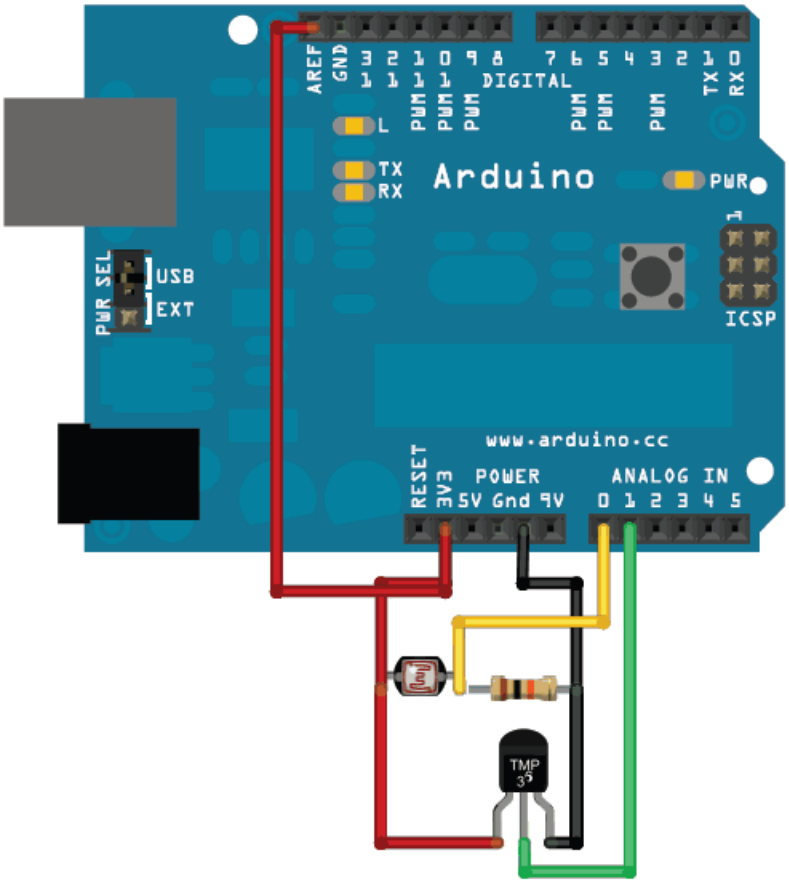
```
//TMP36 Pin Variables
int sensorPin = 0; //the analog pin the TMP36's Vout (sense) pin is connected to
                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures

/*
 * setup() - this function runs once when you turn your Arduino on
 * We initialize the serial connection with the computer
 */
void setup()
{
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
}

void loop()                     // run over and over again
{
 //getting the voltage reading from the temperature sensor
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
 float voltage = reading * 5.0;
 voltage /= 1024.0;

 // print out the voltage
 Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV
offset
                                               //to degrees ((voltage - 500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degrees C");

 // now convert to Fahrenheit
 float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
 Serial.print(temperatureF); Serial.println(" degrees F");

 delay(1000);                                     //waiting a second
}
```

Getting Better Precision

For better results, using the 3.3v reference voltage as ARef instead of the 5V will be more precise and less noisy

This example from the light&temp datalogging tutorial has a photocell but you can ignore it

Note we've changed the TMP36 to A1



To use the 3.3v pin as your analog reference, don't forget to specify "analogReference(EXTERNAL)" in your setup as in the code below:

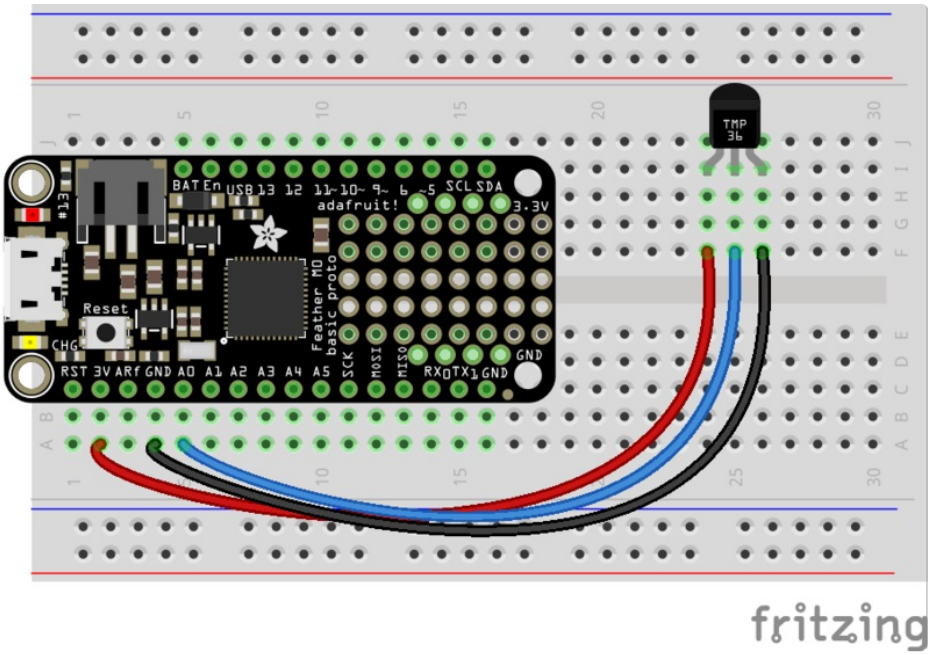
```
/* Sensor test sketch
  for more information see http://www.ladyada.net/make/logshield/lighttemp.html
  */

#define aref_voltage 3.3         // we tie 3.3V to ARef and measure it with a multimeter!




//TMP36 Pin Variables
int tempPin = A1;        //the analog pin the TMP36's Vout (sense) pin is connected to
                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures
int tempReading;        // the analog reading from the sensor

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);

  // If you want to set the aref to something other than 5v
  analogReference(EXTERNAL);
}


void loop(void) {

  tempReading = analogRead(tempPin);

  Serial.print("Temp reading = ");
  Serial.print(tempReading);     // the raw analog reading

  // converting that reading to voltage, which is based off the reference voltage
  float voltage = tempReading * aref_voltage;
  voltage /= 1024.0;

  // print out the voltage
  Serial.print(" - ");
  Serial.print(voltage); Serial.println(" volts");

  // now print out the temperature
  float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV
offset
                                               //to degrees ((volatge - 500mV) times 100)
  Serial.print(temperatureC); Serial.println(" degrees C");

  // now convert to Fahrenheight
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF); Serial.println(" degrees F");

  delay(1000);
}
```

TMP36 with CircuitPython

With CircuitPython it's easy to read the TMP36 sensor using the analog I/O module (https://adafru.it/BBj) and analog to digital converter built-in to your board. You can easily turn the TMP36 output voltage into a precise temperature reading with just a few lines of Python code.

To follow this page make sure to wire up the TMP36 sensor to your CircuitPython board as shown on the previous page. The A0 analog input will be used as the input from the TMP36's temperature output. Here's an example of a Feather M0 wired to the TMP36 on the A0 analog input:

Note: The simple circuit below has been found to give incorrect readings with CircuitPython because of the speed at which CircuitPython reads the analog value. To fix this problem, add a 0.01uF or 0.1uF capacitor and a 47k resistor across the output and ground pins of the TMP36. We will revise the diagram later.



https://adafru.it/A2T

https://adafru.it/A2T

First connect to the board's serial REPL (https://adafru.it/pMf)so you are at the CircuitPython >>> prompt.

Next import the necessary **board** and **analogio** modules:

```
import board
import analogio
```

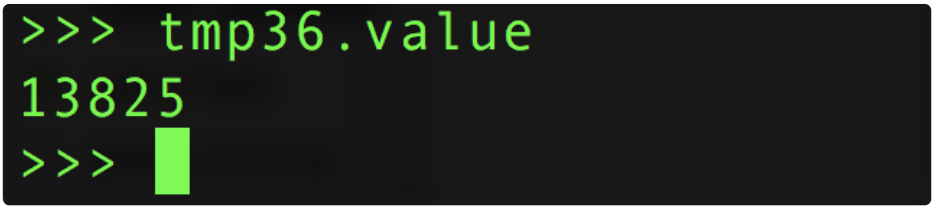
Now create an analog input for the A0 pin on the board:

```
tmp36 = analogio.AnalogIn(board.A0)
```

At this point you can read the raw ADC value of the TMP36 sensor output. Like the analog I/O guide mentions this value will range from 0 to 65535 proportional to the voltage output by the sensor (from 0 to the analog reference voltage of your board, typically 3.3V to 5V).

For example try reading the raw ADC value:

```
tmp36.value
```



You can convert this value into a voltage (in millivolts) using a similar formula mentioned on the previous page. However there's one small change to increase the range of values from 1023 to 65535--this is necessary because CircuitPython uses a wider range of values for ADC inputs. In addition with CircuitPython you can directly access the board's analog reference voltage so one simple equation will work for both 3.3V and 5V references:

```
tmp36.value * (tmp36.reference_voltage * 1000 / 65535)
```

Once you have the analog voltage value output by the TMP36 you can turn it into a temperature in degrees Celsius just like the previous page shows:

```
millivolts = tmp36.value * (tmp36.reference_voltage * 1000 / 65535)
(millivolts - 500) / 10
```

Let's make a function to perform this math for us and return the temperature in degrees Celsius:

```
def tmp36_temperature_C(analogin):
    millivolts = analogin.value * (analogin.reference_voltage * 1000 / 65535)
    return (millivolts - 500) / 10

tmp36_temperature_C(tmp36)
```

```
>>> def tmp36_temperature_C(analogin):
...     millivolts = analogin.value * (analogin.reference_voltage * 1000 / 65535)
...     return (millivolts - 500) / 10
...
```

```
>>> tmp36_temperature_C(tmp36)
20.547
>>>
```

You can turn this into a complete program that reads and prints the temperature every second too. Save this as a **main.py** on your board and check the serial output:

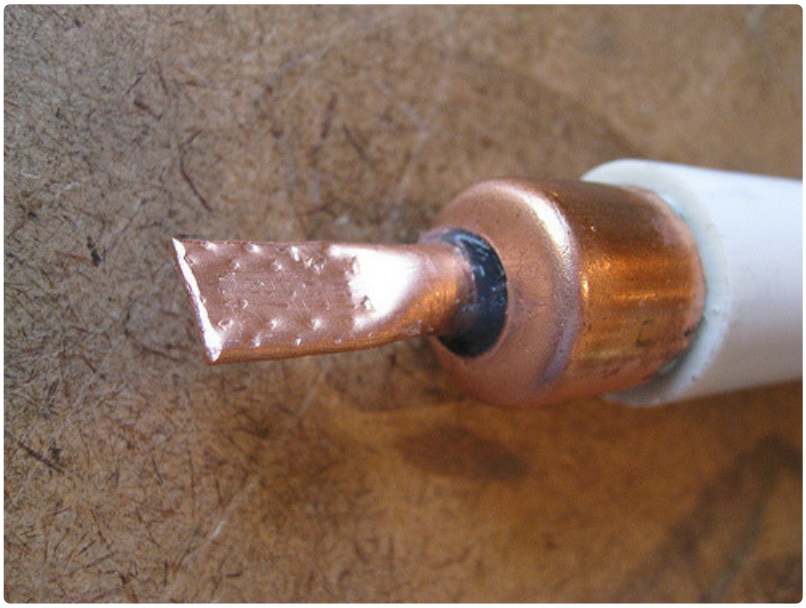
```
import board
import analogio
import time


TMP36_PIN = board.A0  # Analog input connected to TMP36 output.


# Function to simplify the math of reading the temperature.
def tmp36_temperature_C(analogin):
    millivolts = analogin.value * (analogin.reference_voltage * 1000 / 65535)
    return (millivolts - 500) / 10


# Create TMP36 analog input.
tmp36 = analogio.AnalogIn(TMP36_PIN)

# Loop forever.
while True:
    # Read the temperature in Celsius.
    temp_C = tmp36_temperature_C(tmp36)
    # Convert to Fahrenheit.
    temp_F = (temp_C * 9/5) + 32
    # Print out the value and delay a second before looping again.
    print("Temperature: {}C {}F".format(temp_C, temp_F))
    time.sleep(1.0)
```

That's all there is to using the TMP36 with CircuitPython!

Example Projects

Remote temperature sensor

Video editor that uses biofeedback (body temperature)



How to waterproof a LM35 sensor for use in a Remotely Operated Vehicle (robot submarine) (https://adafru.it/aKM)

A "smart coaster" lets you know when your coffee/tea is safe to drink (https://adafru.it/BBk) Some of these projects use thermistors (resistors that change their resistance based on temperature), but can very easily be adapted to to a solid state sensor like the TMP36.