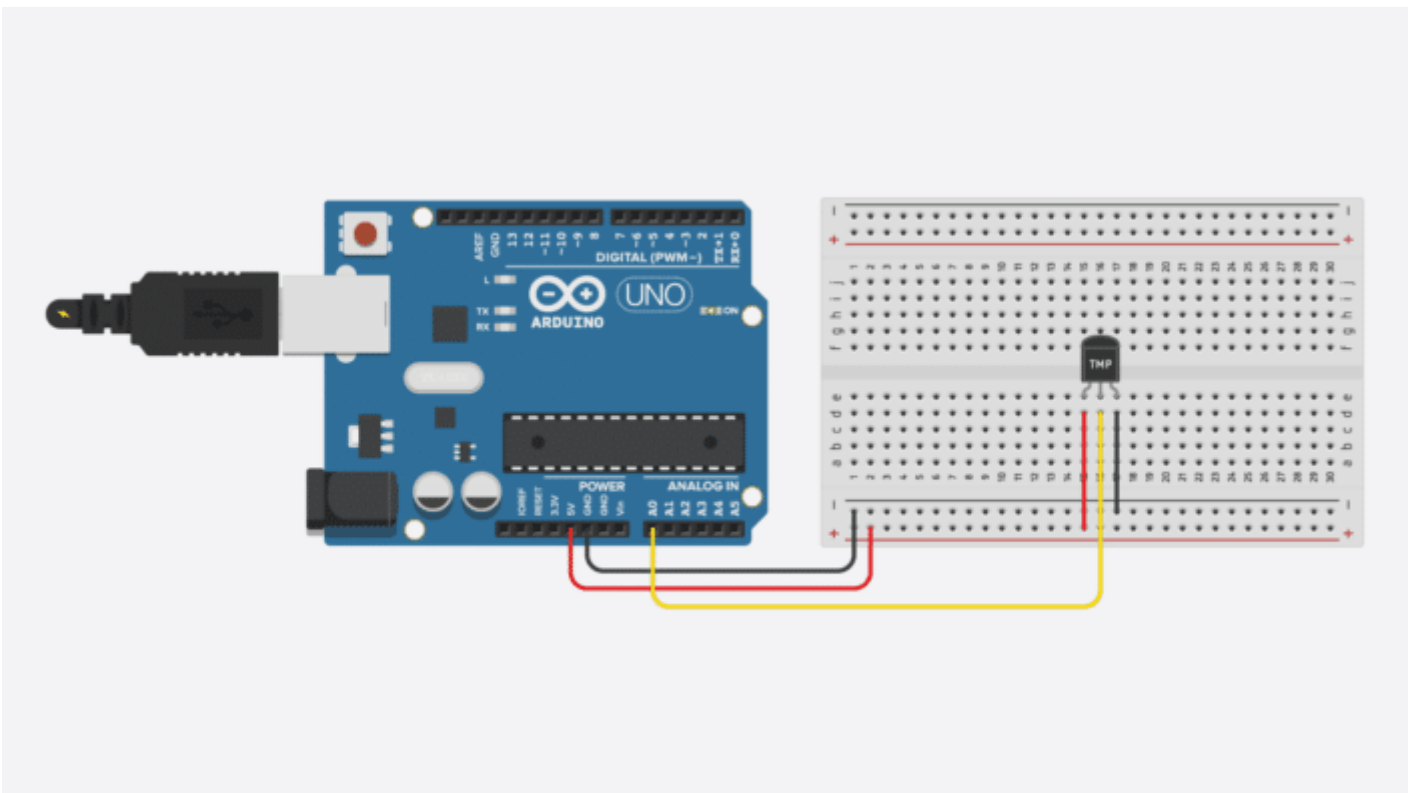


Makerguides.com

TMP36 analog temperature sensor with Arduino tutorial

Written by Benne de Bakker (<https://www.makerguides.com/author/benne-de-bakker/>)



In this tutorial, you will learn how to use a TMP36 analog temperature sensor (<https://amzn.to/3mlgmBX>) with Arduino. I have included a wiring diagram and several example codes to help you get started!

In the first part of this article, you can find the specifications and information about the TMP35, TMP36, and TMP37 sensors. Next, we will look at how to connect the sensor to the Arduino.

The first code example can be used to take temperature readings from the sensor and display the results in the Serial Monitor. In the second example, I will show you how to display the temperature on an I2C LCD (<https://amzn.to/3hNtP3K>) to create a standalone thermometer.

If you would like to learn more about other temperature sensors, check out the articles below.

Recommended articles

- The complete guide for DS18B20 digital temperature sensors with Arduino (<https://www.makerguides.com/ds18b20-arduino-tutorial/>)
 - How to use DHT11 and DHT22 Sensors with Arduino (<https://www.makerguides.com/dht11-dht22-arduino-tutorial/>)
 - LM35 analog temperature sensor with Arduino tutorial (<https://www.makerguides.com/lm35-arduino-tutorial/>)
 - How to control a character I2C LCD with Arduino (<https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>)
 - How to use a 16×2 character LCD with Arduino (<https://www.makerguides.com/character-lcd-arduino-tutorial/>)
-

Supplies

Hardware components

TMP36 analog temperature sensor (https://amzn.to/3mlgmBX) (TO-92)	× 1	Amā
Arduino Uno (https://amzn.to/374aJjX)	× 1	Amā
Breadboard (https://amzn.to/2sZTxNA)	× 1	Amā
Jumper wires (https://amzn.to/2EG9wDc)	~ 10	Amā AliE (http
(https://www.amazon.com/HiLetgo-Display-Backlight-Controller-Character/dp/B00HJ6AFW6/ref=as_li_ss_tl?keywords=lcd+16x2&qid=1562768812&s=gateway&sr=8-3&linkCode=ll1&tag=makerguides-20&linkId=560296518db01cb992dfe0ac9cc8e4d7&language=en_US) 16×2 character I2C LCD (https://s.click.aliexpress.com/e/_d82Tvi7)	× 1	Amā AliE (http
USB cable type A/B (https://amzn.to/34SBuXf)	× 1	Amā

Software

Arduino IDE (<https://www.arduino.cc/en/Main/Software>)

Makerguides.com is a participant in the Amazon Services LLC Associates Program, an affiliate advertising program designed to provide a means for sites to earn advertising fees by advertising and linking to products on Amazon.com.

About the TMP35/TMP36/TMP37 analog temperature sensors

The TMP35/TMP36/TMP37 are low voltage, precision centigrade temperature sensors made by Analog Devices (<https://www.analog.com/>). They provide a voltage output that is linearly proportional to the temperature in degree Celsius ($^{\circ}\text{C}$) and are, therefore, very easy to use with the Arduino. Moreover, they are precise, never wear out, and very inexpensive!

The TMP35/TMP36/TMP37 sensors do not require any external calibration to provide a typical accuracy of $\pm 1^{\circ}\text{C}$ at $+25^{\circ}\text{C}$ and $\pm 2^{\circ}\text{C}$ over the -40°C to $+125^{\circ}\text{C}$ temperature range.

The difference between the TMP35, TMP36, and TMP37 is their temperature operating range and output scale factor. The TMP35 reads temperatures from 10°C to 125°C and provides a 250 mV output at 25°C . This sensor is functionally compatible with the LM35 made by Texas Instruments. You can find a dedicated tutorial for the LM35 here:

- LM35 analog temperature sensor with Arduino tutorial (<https://www.makerguides.com/lm35-arduino-tutorial/>)

The TMP36 reads temperatures from -40°C to 125°C , provides a 750mV output at 25°C , and operates to $+125^{\circ}\text{C}$ from a single 2.7 V supply. This sensor is functionally compatible with the LM50.

The TMP35 and TMP36 have the same output scale factor of 10 mV/ $^{\circ}\text{C}$.

The TMP37 is intended for an operating range of 5°C to 100°C and provides a 500 mV output at 25°C. This sensor provides a slightly higher precision than the other sensors and has an output scale factor of 20 mV/°C.

Output voltage (V) versus temperature (°C). Source: Analog Devices

As you can see in the figure above, the output range of all of the sensors is between 0.1 V and 2 V. Note that the output voltage is independent of the supply voltage you use.

TMP35/TMP36/TMP37 pinout

The TMP35/TMP36/TMP37 sensors come in 3 different form factors, but the most common type is the 3-pin TO-92 package (<https://amzn.to/3mlgmBX>), which looks just like a transistor. The TMP36 version of this sensor has the model number TMP36GT9Z.

([https://www.amazon.com/Bridgoid-TMP36GT9Z-](https://www.amazon.com/Bridgoid-TMP36GT9Z-Temperature-Analog-Voltage/dp/B07LG758H3/ref=as_li_ss_il?dchild=1&keywords=tmp36&qid=1603622318&sr=8-4&linkCode=li3&tag=makerguides-20&linkId=0068f09f1f35185df2a932ff2b3ffaca&language=en_US)

[Temperature-Analog-Voltage/dp/B07LG758H3/ref=as_li_ss_il?
dchild=1&keywords=tmp36&qid=1603622318&sr=8-
4&linkCode=li3&tag=makerguides-
20&linkId=0068f09f1f35185df2a932ff2b3ffaca&language=en_US\)](https://www.amazon.com/Bridgoid-TMP36GT9Z-Temperature-Analog-Voltage/dp/B07LG758H3/ref=as_li_ss_il?dchild=1&keywords=tmp36&qid=1603622318&sr=8-4&linkCode=li3&tag=makerguides-20&linkId=0068f09f1f35185df2a932ff2b3ffaca&language=en_US)

The pinout of the sensor is given in the figure below:

Note that pin 1 (+V_S) is the leftmost pin when the flat side of the sensor (with the text printed on it) is facing towards you.

You can find more specifications of the TMP36 in the table below.

TMP36 analog temperature sensor specifications

Supply voltage	2.7 V to 5.5 V
Quiescent current	50 μ A
Temperature range	-40°C to + 125°C
Accuracy	\pm 1°C at +25°C \pm 2°C from -40°C to +125°C
Output scale factor	10 mV/°C
Output voltage at 25°C	750 mV
Package	3-pin TO-92
Manufacturer	Analog Devices (https://www.analog.com/)
Cost	Check price (https://amzn.to/3mlgmBX)

For more information, you can also check out the datasheet here:

TMP35/TMP36/TMP37 Datasheet 

Wiring – Connecting TMP36 temperature sensor to Arduino

Connecting a TMP36 to the Arduino is very easy as you only need to connect 3 pins. Start by connecting the $+V_S$ pin to the 5 V output of the Arduino and the GND pin to the ground. If you are using a 3.3 V Arduino, simply connect $+V_S$ to 3.3 V instead.

Next, connect the middle pin (V_{OUT}) to any of the analog inputs of the Arduino. In this case, I used the analog input pin A0.

TMP36 analog temperature sensor with Arduino Uno wiring diagram

The connections are also given in the table below:

TMP36 analog temperature sensor connections

TMP36	Arduino
Pin 1 (+V _S)	5 V
Pin 2 (V _{OUT})	Pin A0
PIN 3 (GND)	GND

To improve the stability of the sensor, the datasheet recommends adding a 0.1 μ F ceramic capacitor (<https://amzn.to/3dWkAwN>) between the +V_S pin and GND. When you are using long cables, adding a small resistor (<https://amzn.to/2J46370>) (e.g. 750 Ω) in series with the signal line (V_{OUT}) can also reduce the noise.

TMP36 with optional resistor and capacitor

When I tested the sensor with an Arduino Uno, I got stable readings without the capacitor and resistor, but your results may vary.

Converting the TMP36 output voltage into temperature

To convert the output voltage of the sensor into the temperature in degree Celsius, you can use the following formula:

$$\text{Temperature (}^\circ\text{C)} = (V_{\text{OUT}} - 500) / 10$$

with V_{OUT} in millivolt (mV). So if the output of the sensor is 750 mV, then the temperature is:

$$(750 - 500) / 10 = 25^\circ\text{C}$$

As you can see in the wiring diagram above, the output of the TMP36 is connected to one of the analog inputs of the Arduino. The value of this analog input can be read with the function `analogRead()` (<https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>) as you will see in the code examples below. However, the function `analogRead(pin)` will not actually return the output voltage of the sensor.

Arduino boards contain a multichannel, 10-bit analog to digital converter (ADC), which will map input voltages between 0 and the operating voltage (5 V or 3.3 V) into integer values between 0 and 1023. On an Arduino Uno, for example, this yields a resolution between readings of 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit.

So if you use `analogRead()` to read the voltage at one of the analog inputs of the Arduino, you will get a value between 0 and 1023.

To convert this value back into the output voltage of the sensor, you can use:

$$V_{OUT} = \text{reading from ADC} * (5000 / 1024)$$

And if you are using a 3.3 V Arduino:

$$V_{OUT} = \text{reading from ADC} * (3300 / 1024)$$

We will use these formulas in the code examples below.

TMP36 analog temperature sensor with Arduino example code

With the following example code, you can read the temperature from a TMP36 sensor and display it in the Serial Monitor.

You can upload the example code to your Arduino using the Arduino IDE (<https://www.arduino.cc/en/main/software>).

To copy the code, click on the button in the top right corner of the code field.

```
1.  /* TMP36 analog temperature sensor with Arduino example code. More info:
    https://www.makerguides.com */
2.
3.  // Define to which pin of the Arduino the output of the TMP36 is connected:
4.  #define sensorPin A0
5.
6.  void setup() {
7.    // Begin serial communication at a baud rate of 9600:
8.    Serial.begin(9600);
9.  }
10.
11. void loop() {
12.   // Get a reading from the temperature sensor:
13.   int reading = analogRead(sensorPin);
14.
15.   // Convert the reading into voltage:
16.   float voltage = reading * (5000 / 1024.0);
17.
18.   // Convert the voltage into the temperature in Celsius:
19.   float temperature = (voltage - 500) / 10;
20.
21.   // Print the temperature in the Serial Monitor:
```

```
22.     Serial.print(temperature);
23.     Serial.print(" \xC2\xB0"); // shows degree symbol
24.     Serial.println("C");
25.
26.     delay(1000); // wait a second between readings
27. }
```

You should see the following output in the Serial Monitor (Ctrl + Shift + M).

Serial Monitor output

Make sure that the baud rate of the Serial Monitor is also set to 9600.

How the code works

First, I defined to which pin of the Arduino the V_{OUT} pin of the sensor is connected. In this case, we used the analog pin A0. The statement `#define` (<https://www.arduino.cc/reference/en/language/structure/further-syntax/define/>) can be used to give a name to a constant value. The compiler will replace all references to this constant with the defined value when the program is compiled. So everywhere you mention `sensorPin`, the compiler will replace it with A0 when the program is compiled.

```
3. // Define to which pin of the Arduino the output of the TMP36 is connected:
4. #define sensorPin A0
```

In the setup section of the code, we begin serial communication at a baud rate of 9600.

```
6. void setup() {
7.     // Begin serial communication at a baud rate of 9600:
8.     Serial.begin(9600);
9. }
```

In the loop section of the code, we start by taking a reading from the sensor with the function `analogRead(pin)`.

```
12. // Get a reading from the temperature sensor:
13. int reading = analogRead(sensorPin);
```

Next, we use the formulas that I mentioned earlier in the article to convert the reading into voltage and then into temperature.

```
15. // Convert the reading into voltage:
16. float voltage = reading * (5000 / 1024.0);
17.
18. // Convert the voltage into the temperature in degree Celsius:
19. float temperature = (voltage - 500) / 10;
```

If you are using a 3.3 V Arduino, like the Arduino Due (<https://amzn.to/31ljLmp>) or Arduino Nano 33 BLE (<https://amzn.to/35yf3Ja>), you need to connect the $+V_S$ pin to 3.3 V and replace the highlighted line with:

```
15. // Convert the reading into voltage:
```

```
16. float voltage = reading * (3300 / 1024.0);
```

Lastly, the results are printed in the Serial Monitor:

```
21. // Print the temperature in the Serial Monitor:
22. Serial.print(temperature);
23. Serial.print(" \xC2\xB0"); // shows degree symbol
24. Serial.println("C");
```

Display the TMP36 temperature readings on an I2C LCD

If you want to make a standalone thermometer that doesn't need a computer, it can be nice to know how to display the temperature readings on an LCD display.

With the example code below, you can display the temperature readings on a 16×2 character I2C LCD (<https://amzn.to/2TffbbL>).

Connecting the I2C LCD is fairly easy as you can see in the wiring diagram below. You can check out my detailed tutorial below for more information.

- How to control a character I2C LCD with Arduino (<https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>)

If you want to use a standard non-I2C LCD instead, take a look at this article:

- How to use a 16×2 character LCD with Arduino (<https://www.makerguides.com/character-lcd-arduino-tutorial/>)

TMP36 analog temperature sensor with 16x2 character I2C LCD and Arduino wiring diagram.

The connections are also given in the table below:

I2C LCD Connections

I2C Character LCD	Arduino
GND	GND
VCC	5 V
SDA	A4
SCL	A5

Note that the TMP36 temperature sensor is connected in the same way as before.

Installing the required Arduino libraries

To use an I2C LCD, you need to install the **LiquidCrystal_I2C** Arduino library.

To install this library, go to Tools > Manage Libraries (Ctrl + Shift + I on Windows) in the Arduino IDE (<https://www.arduino.cc/en/main/software>). The Library Manager will open and update the list of installed libraries.

Now search for 'liquidcrystal_i2c' and look for the library by **Frank de Brabander**. Select the latest version and then click Install.

Installing the LiquidCrystal_I2C Arduino library

TMP36 with I2C LCD example code

```
1.  /* TMP36 analog temperature sensor with I2C LCD and Arduino example code. More info:
    https://www.makerguides.com */
2.
3.  // Include the required Arduino libraries:
4.  #include <LiquidCrystal_I2C.h>
5.
6.  // Create a new instance of the LiquidCrystal_I2C class:
7.  LiquidCrystal_I2C lcd(0x27, 16, 2);
8.
9.  // Degree symbol:
10. byte Degree[] = {
11.     B00111,
12.     B00101,
13.     B00111,
14.     B00000,
15.     B00000,
16.     B00000,
17.     B00000,
18.     B00000
19. };
20.
21. // Define to which pin of the Arduino the TMP36 is connected:
22. #define sensorPin A0
```

```
23.
24. void setup() {
25.     // Start the LCD and turn on the backlight:
26.     lcd.init();
27.     lcd.backlight();
28.
29.     // Create a custom character:
30.     lcd.createChar(0, Degree);
31. }
32.
33. void loop() {
34.     // Get a reading from the temperature sensor:
35.     int reading = analogRead(sensorPin);
36.
37.     // Convert the reading into voltage:
38.     float voltage = reading * (5000 / 1024.0);
39.
40.     // Convert the voltage into the temperature in degree Celsius:
41.     float temperature = (voltage - 500) / 10;
42.
43.     // Print the temperature on the LCD;
44.     lcd.setCursor(0, 0);
45.     lcd.print("Temperature:");
46.     lcd.setCursor(0, 1);
47.     lcd.print(temperature);
48.     lcd.write(0); // print the custom character
49.     lcd.print("C");
50.
51.     delay(1000); // wait a second between readings
52. }
```

You should see the following output on the LCD:

Conclusion

In this tutorial, I have shown you how to use a TMP36 analog temperature sensor with Arduino. I hope you found it useful and informative. If you did, please **share this article** with a friend who also likes electronics and making things.

I would love to know what projects you plan on building (or have already built) with this sensor. If you have any questions, suggestions, or if you think that things are missing in this tutorial, **please leave a comment below**.

Note that comments are held for moderation to prevent spam.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Beginner



 1
SHARES

What to read next?

LM35 analog temperature sensor with Arduino tutorial
(<https://www.makerguides.com/lm35-arduino-tutorial/>)

TMP36 analog temperature sensor with Arduino tutorial
(<https://www.makerguides.com/tmp36-arduino-tutorial/>)

Arduino Nano Board Guide (Pinout, Specifications, Comparison)
(<https://www.makerguides.com/arduino-nano/>)

The complete guide for DS18B20 digital temperature sensors with Arduino
(<https://www.makerguides.com/ds18b20-arduino-tutorial/>)

How to use an IR receiver and remote with Arduino
(<https://www.makerguides.com/ir-receiver-remote-arduino-tutorial/>)

(<https://www.makerguides.com/maxbotix-mb7389-arduino-tutorial/>)


MaxBotix MB7389 weather-resistant distance sensor tutorial
(<https://www.makerguides.com/maxbotix-mb7389-arduino-tutorial/>)

(<https://www.makerguides.com/hc-sr04-arduino-tutorial/>)

How to use an HC-SR04 Ultrasonic Distance Sensor with Arduino
(<https://www.makerguides.com/hc-sr04-arduino-tutorial/>)

(<https://www.makerguides.com/28byj-48-stepper-motor-arduino-tutorial/>)

28BYJ-48 Stepper Motor with ULN2003 Driver and Arduino Tutorial
(<https://www.makerguides.com/28byj-48-stepper-motor-arduino-tutorial/>)

 Ezoic (<https://www.ezoic.com/what-is-ezoic/>)

[report this ad](#)

© 2021 Makerguides.com - All Rights Reserved

