Home About Projects Learn Library Blog Store Forums

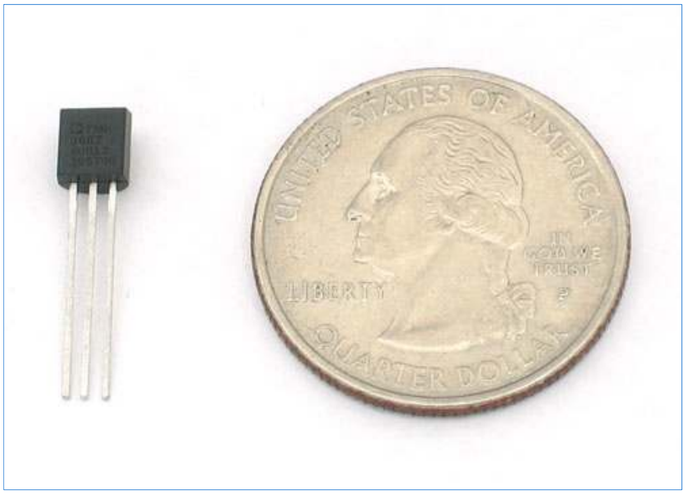
ladyada.net Search

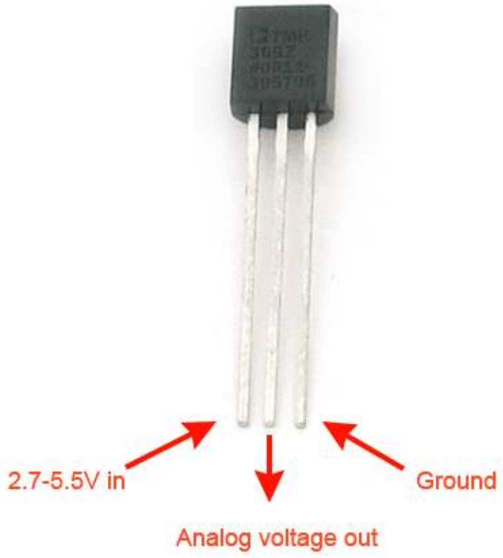
Temperature

October 20, 2009 19:46

Hot or not? Now you will know to 0.1 degree precision!

Forums
Overview
FSR
Photocell
Temp.
Tilt
PIR
Buy sensor





What is a temperature sensor?

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 dont use mercury (like old thermometers), bimetalic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors). Instead, they use the fact as temperature increases, the votage 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 genereate 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.

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 in the Adafruit shop, the Analog Devices TMP36 (-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:** $2.00 at the Adafruit shop
- **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**

How to measure temperature!

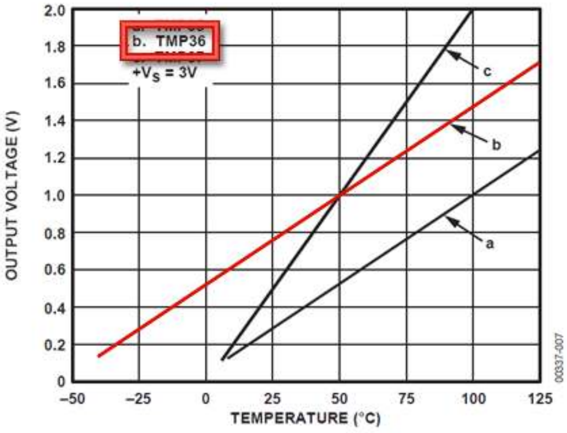


Figure 6. Output Voltage vs. Temperature

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 independant of the power supply.

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**

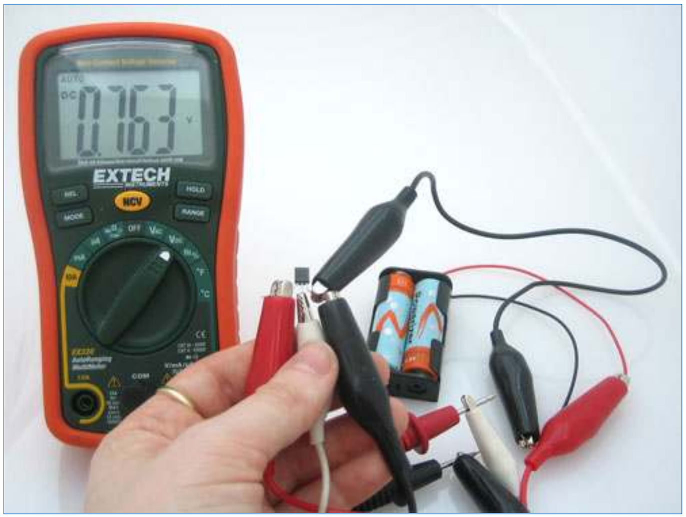
If you're using a LM35 or similar, use line 'a' in the image above and the formula: **Temp in °C = (Vout in mV) / 10**

Testing your temperature sensor

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

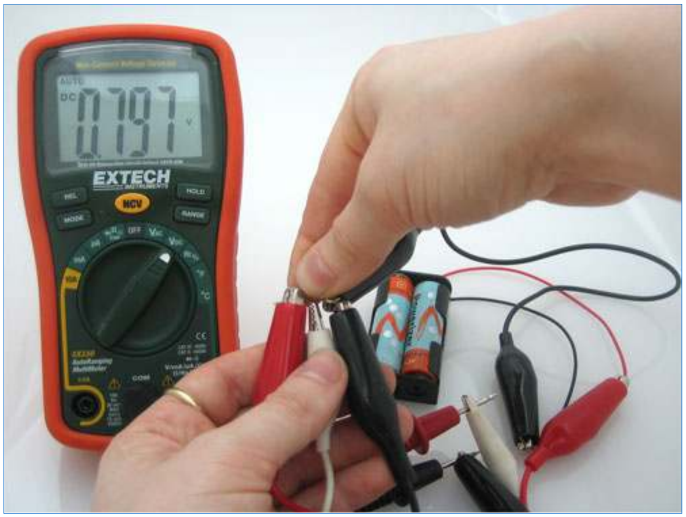
Connect a 2.7-5.5V power supply (2-4 AA batteries work fantastic) so that ground is connected to pin 3 (right pin), and power is connected to pin 1 (left pin)

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



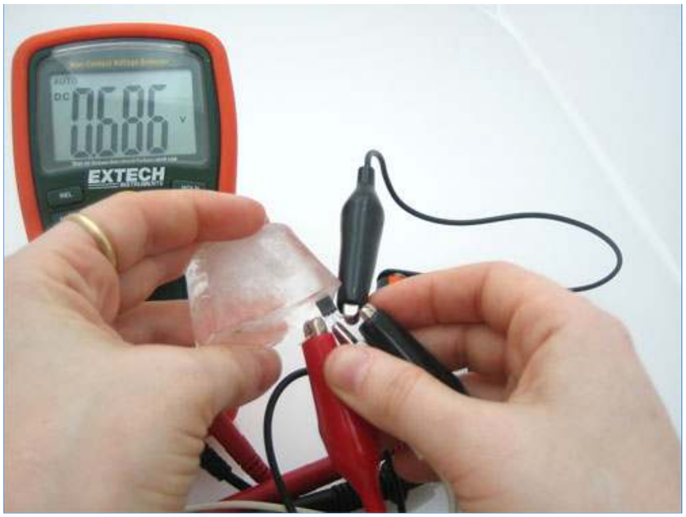
The sensor is indicating that the temperature is 26.3°C also known as 79.3°F

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

Connecting to your 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!

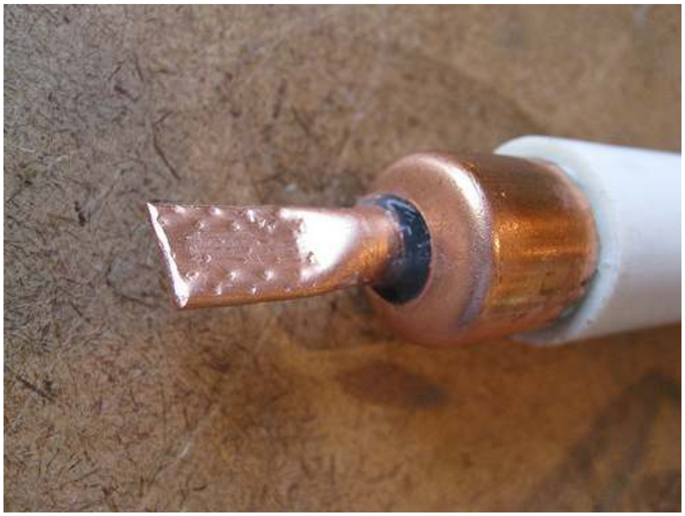
breadboarded to-92

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.

Project examples

Remote temperature sensor

Video editor that uses biofeedback (body temperature)

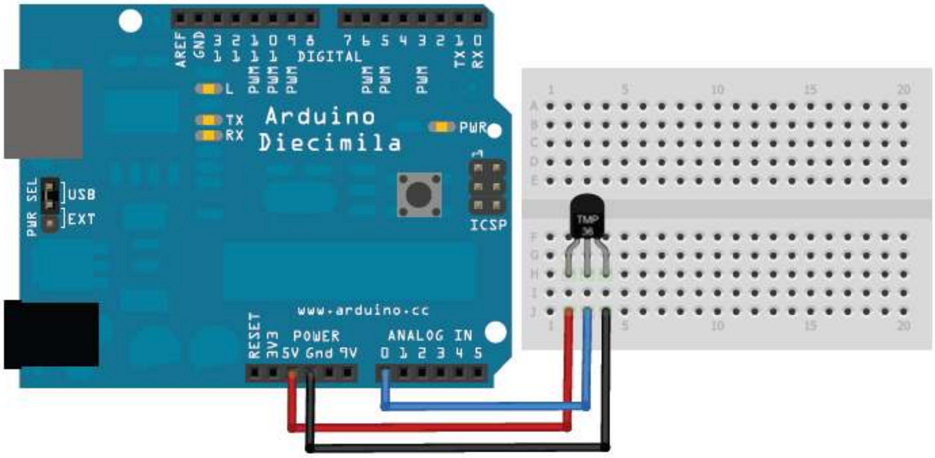

How to waterproof a LM35 sensor for use in a Remotely Operated Vehicle (robot submarine)

A "smart coaster" lets you know when your coffee/tea is safe to drink

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

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)

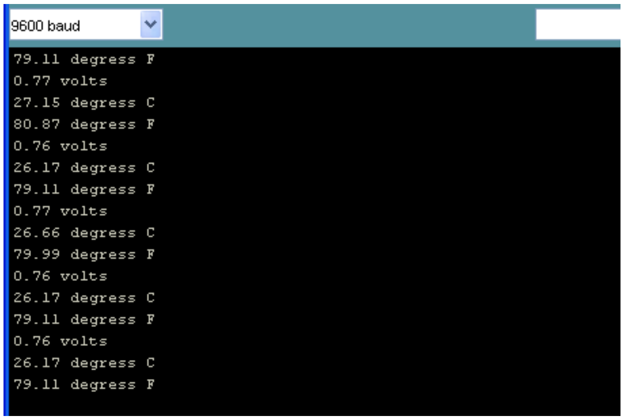
This formula converts the number 0-1023 from the ADC into 0-3300mV (= 3.3V)

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

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

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



```
79.11 degress F
0.77 volts
27.15 degress C
80.87 degress F
0.76 volts
26.17 degress C
79.11 degress F
0.77 volts
26.66 degress C
79.99 degress F
0.76 volts
26.17 degress C
79.11 degress F
0.76 volts
26.17 degress C
79.11 degress F
```

```
//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 / 1024;

 // 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 ((volatge - 500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degress C");

 // now convert to Fahrenheight
 float temperatureF = (temperatureC * 9 / 5) + 32;
 Serial.print(temperatureF); Serial.println(" degress F");

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

Auto-calibrating supply-independent thermometer

This example is similar to the one above except that now we use a special trick where we read the analog value of a fixed reference voltage inside the chip and then use that to make a precise calculation. This also means it will work right no matter what voltage the Arduino is running at!

```
//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

#define BANDGAPREF 14   // special indicator that we want to measure the bandgap

/*
 * 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
  delay(500);
}

void loop()                     // run over and over again
{
  // get voltage reading from the secret internal 1.05V reference
  int refReading = analogRead(BANDGAPREF);
  Serial.println(refReading);

  // now calculate our power supply voltage from the known 1.05 volt reading
  float supplyvoltage = (1.05 * 1024) / refReading;
  Serial.print(supplyvoltage); Serial.println("V power supply");

  //getting the voltage reading from the temperature sensor
  int reading = analogRead(sensorPin);

  // converting that reading to voltage
  float voltage = reading * supplyvoltage / 1024;

  // 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 ((volatge - 500mV) times 100)
  Serial.print(temperatureC); Serial.println(" degress C");

  // now convert to Fahrenheight
  float temperatureF = (temperatureC * 9 / 5) + 32;
  Serial.print(temperatureF); Serial.println(" degress F");

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