Using a Temperature Sensor

Add a temperature sensor to the ATmega Board.

Site: iCODE
Course: Machine Science Guides (Arduino Version)
Book: Using a Temperature Sensor
Printed by: Ivan Rudnicki
Date: Wednesday, July 30, 2014, 03:32 PM

Contents

About Temperature Sensors

Temperature sensors are used in a wide range of electronic devices, including digital thermometers, home thermostats, microwave ovens, and refrigerators. Figure 1 shows two devices with temperature sensors.

Figure 1. Devices with temperature sensors.

The temperature sensor provided in your kit is a precision analog sensor, whose voltage output is linearly proportional to the temperature. Configured as described here, the sensor has an operating range of about 0°C to +150°C.

Adding the Temperature Sensor

The temperature sensor is a three-pin integrated circuit. When the device's flat side is facing towards you, the three pins are labeled 1, 2, and 3, from left to right, as shown in Figure 2.

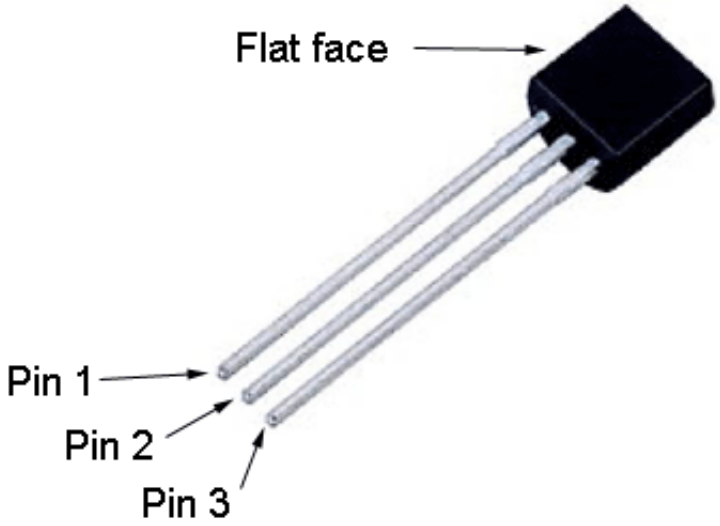

Figure 2. Temperature sensor.

Depending on your kit, you may have either an LM35 temperature sensor or an LM335 temperature sensor. You can identify your sensor by reading the text on the flat face of the device, as shown in Figure 3. The two sensors are wired slightly differently on the breadboard and produce different analog outputs, so it is critical to identify which type you have before moving ahead.

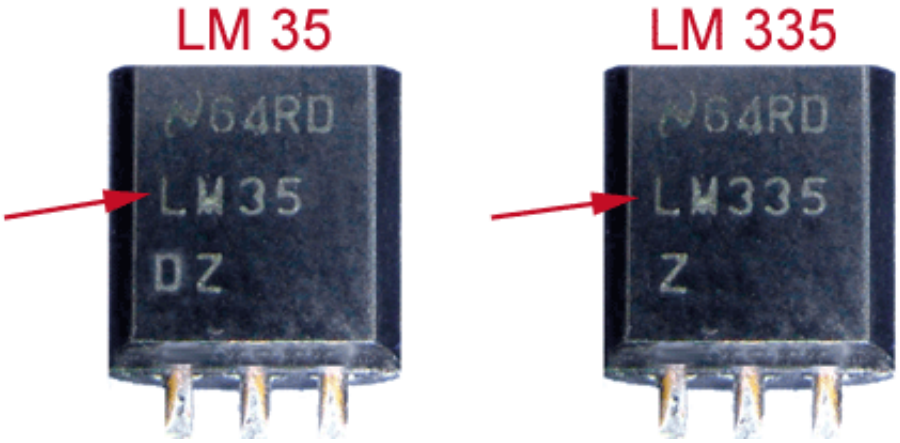

Figure 3. LM35 (left) and LM335 (right).

If you have the LM35, connect the temperature sensor as shown in Figure 4. Pin 1 is connected to power, and pin 3 is connected to ground. Pin 2 connects to ground through a 1.0 uF capacitor, and to Port C4 (pin 27) on the microcontroller through a 10K Ohm resistor.

Figure 4. Circuit schematic for the LM35.

If you have the LM335, connect the temperature sensor as shown in Figure 5. Pin 1 is inserted into the breadboard but not connected to anything. Pin 3 is connected to ground. Pin 2 connects to power through a 1K ohm resistor, and to Port C4 (pin 27) on the microcontroller.

Figure 5. Circuit schematic for LM335.

Reading Analog Values

The following code reads the analog-to-digital converter value on Port C4 and displays it on the LCD. This value is proportional to the ambient Celsius temperature.

1. Rename your code file analogvalue.c.
2. Modify your code file, as follows:

```
#include <mxapi.h>
#include <lcd.h>
#include <adc.h>

int main(void)
{
    int analog_value;     //Declare a variable
    adc_init();           //Initialize analog to digital converter
    lcd_init();           //Initialize the LCD
    while(1==1)
    {
        analog_value=adc_read(4);                  //Read the ADC value
on Port C4
        lcd_decimal(FIRST_LINE, analog_value, 3);  //Display value on LCD
        delay_ms(100);                             //Wait 100
milliseconds
    }
}
```

3. Compile and test your new code.

Note that the value displayed on the LCD is the raw analog value, not the ambient temperature in Celsius. In the next step, you will convert this value to an accurate Celsius temperature.

Converting to Celsius (LM35)

The adc_read function returns values ranging from 0 to 1024, corresponding to voltages ranging from 0mV to 5100mV (5.1V) on the pin being monitored, so each adc_read increment represents a voltage change of about 5mV. The LM35 temperature sensor creates 10mV of output voltage for every degree Celsius. Therefore, the adc_read function value increments by 2 for every degree Celsius. Using the following code, you can convert the analog voltage from the LM35 into a Celsius temperature to display on the LCD.

1. Rename your code file celsiustemp.c.
2. Modify your code file, as follows:

```
#include <mxapi.h>
#include <lcd.h>
#include <adc.h>

int main(void)
{
    int analog_value;
    adc_init();
    lcd_init();
    while(1==1)
    {
        analog_value=adc_read(4);                  //Read value on Port C4
        analog_value=analog_value/2;               //Divide value by 2
        lcd_decimal(FIRST_LINE, analog_value, 3); //Display value on LCD
        lcd_character(FIRST_LINE+4, 223);          //Display degree symbol
        lcd_text(FIRST_LINE+5, "C");               //Display the letter
"C"
        delay_ms(100);                             //Wait 100 milliseconds
    }
}
```

3. Compile and test your new code.

Converting to Celsius (LM335)

The adc_read function returns values ranging from 0 to 1024, corresponding to voltages ranging from 0mV to 5100mV (5.1V) on the pin being monitored, so each adc_read increment represents a voltage change of about 5mV. The LM335 temperature sensor creates 10mV of output voltage for every degree Celsius. Therefore, the adc_read function value increments by 2 for every degree Celsius. (Because the LM335 is designed to work in Kelvin units, the value is offset from Celsius temperature by 273 degrees.) Using the following code, you can convert the analog voltage from the LM35 into a Celsius temperature to display on the LCD.

1. Rename your code file celsiustemp.c.
2. Modify your code file, as follows:

```
#include <mxapi.h>
#include <lcd.h>
#include <adc.h>

int main(void)
{
    int analog_value;
    adc_init();
    lcd_init();
    while(1==1)
    {
        analog_value=adc_read(4);                    //Read value on Port C4
        analog_value=(analog_value/2)-273;           //Divide value by 2 and
subtract 273
        lcd_decimal(FIRST_LINE, analog_value, 3); //Display value on LCD
        lcd_character(FIRST_LINE+4, 223);            //Display degree symbol
        lcd_text(FIRST_LINE+5, "C");                 //Display the letter
"C"
        delay_ms(100);                               //Wait 100 milliseconds
    }
}
```

3. Compile and test your new code.