Please ask about problems and questions regarding this tutorial on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Temperature Sensor

Description: Measuring Temperature using the TMP102

Tutorial Level: BEGINNER

Next Tutorial: Push Button (/rosserial_arduino/Tutorials/Push%20Button)

electric	fuerte	groovy	hydro	indigo	jade	kinetic

Contents

1. Hardware
2. Code
3. Testing

In this tutorial, we will use an Arduino and a TMP102 temperature sensor (http://focus.ti.com/docs/prod/folders/print/tmp102.html) to measure the ambient temperature in a room. The tutorial's goals are two fold. The first is to show an example temperature sensing node. The second is to demonstrate the Arduino as an I2C interface for ROS. The TMP102 is one of many pieces of hardware that use I2C to communicate. You can now interface SPI/I2C to ROS with ease using an Arduino.

1. Hardware

For this tutorial, you will need an Arduino and a TMP102 Breakout board from Sparkfun (http://www.sparkfun.com/products/9418). This board is a great little temperature sensor that can measure temperature with a 0.625 degree C resolution. Connecting it to an Arduino is very straightforward. It is a 3.3V sensor communication over I2C so it connects to the Arduino's 3.3V output, GND, SDA, SCL pins. SDA and SCL are the I2C data line and I2C clock line respectively. They are exposed on the Arduino board as analog pins 4 and 5.

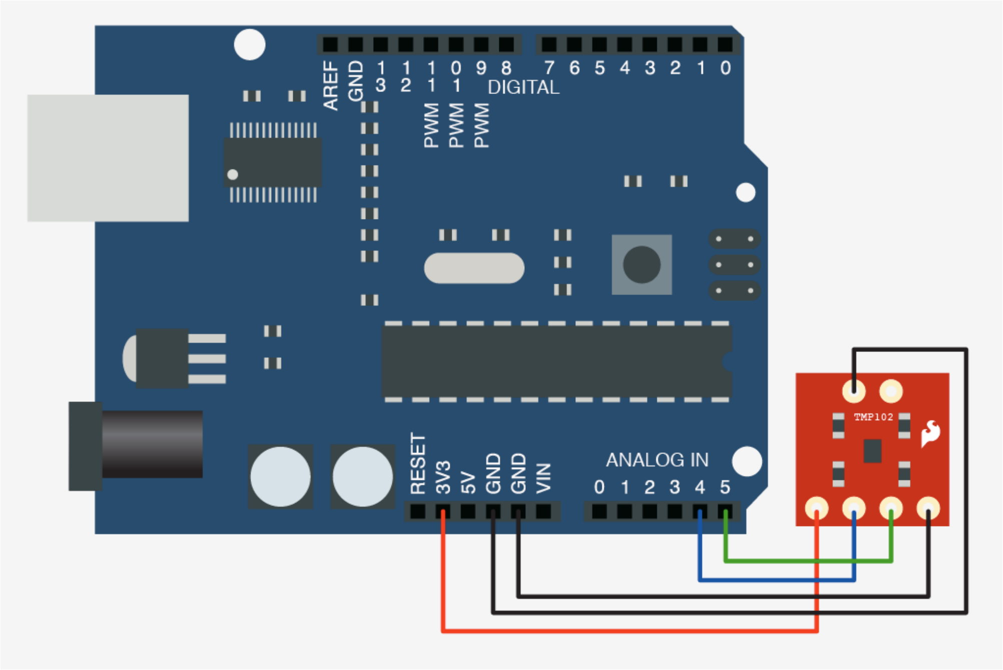

Diagram from here (http://bildr.org/2011/01/tmp102-arduino)

2. Code

Toggle line numbers

```
#include <Wire.h>
#include <ros.h>
#include <std_msgs/Float32.h>


//Set up the ros node and publisher
std_msgs::Float32 temp_msg;
ros::Publisher pub_temp("temperature", &temp_msg);
ros::NodeHandle nh;

int sensorAddress = 0x91 >> 1;  // From datasheet sensor address is 0x91
                                // shift the address 1 bit right,
                                 //the Wire library only needs the 7
                                // most significant bits for the address

void setup()
{
  Wire.begin();        // join i2c bus (address optional for master)

  nh.initNode();
  nh.advertise(pub_temp);

}

long publisher_timer;

void loop()
{

  if (millis() > publisher_timer) {
  // step 1: request reading from sensor
    Wire.requestFrom(sensorAddress,2);
    delay(10);
    if (2 <= Wire.available())  // if two bytes were received
    {
      byte msb;
      byte lsb;
      int temperature;

      msb = Wire.read();  // receive high byte (full degrees)
      lsb = Wire.read();  // receive low byte (fraction degrees)
      temperature = ((msb) << 4);  // MSB
      temperature |= (lsb >> 4);   // LSB

      temp_msg.data = temperature*0.0625;
      pub_temp.publish(&temp_msg);
    }

  publisher_timer = millis() + 1000; //publish once a second
  }

  nh.spinOnce();
}
```

The special bit of code in this example is the use of Arduino's Wire library (http://www.arduino.cc/en/Reference/Wire). Wire is a I2C library that simplifies reading and writing to the I2C bus.

3. Testing

```
roscore
```

```
rostopic echo temperature
```