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.

SRF08 Ultrasonic Ranger

Description: In this tutorial, we will use an Arduino and a SRF08 Ultrasonic Ranger (http://www.acroname.com/robotics/parts/R145-SRF08.html) as a Range Finder.The SRF08 communicates with an Arduino over SPI/I2C.

Tutorial Level: BEGINNER

Next Tutorial: BlinkM (/rosserial_arduino/Tutorials/BlinkM)

electric | fuerte | groovy | hydro | indigo | jade | kinetic

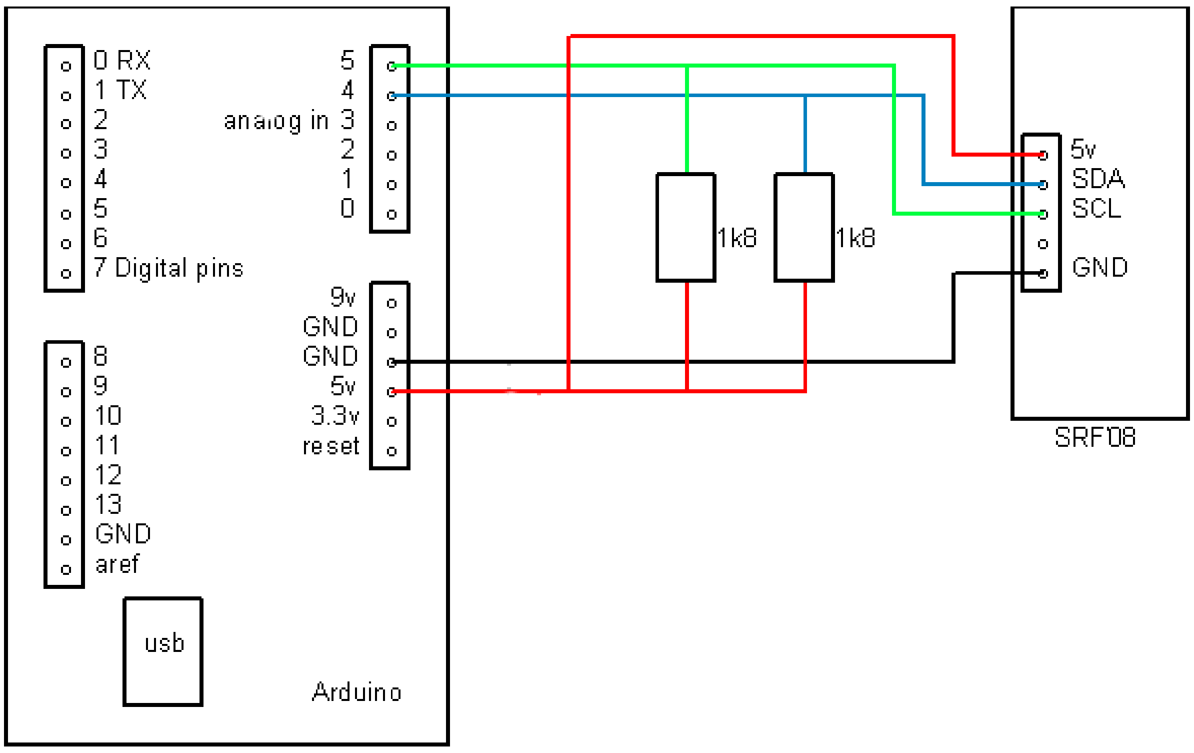
Contents

rosserial (/rosserial) allows you to easily integrate Arduino-based hardware with ROS. This tutorial will explain how to use a SRF08 Ultrasonic ranger with an Arduino.

You will need an Arduino (http://www.arduino.cc) ,a SRF08 Ultrasonic Ranger (http://www.acroname.com/robotics/parts/R145-SRF08.html), and a way to connect your Ranger to your Arduino such as a breadboard or protoboard.The SRF08 Arduino library can be downloaded from Sonar_srf08.zip (/rosserial_arduino/Tutorials/SRF08%20Ultrasonic%20Range%20Finder?action=AttachFile&do=view&target=Sonar_srf08.zip)

1. Hardware Setup

Below is the diagram for setting up the SRF08 sensor for a typical Arduino. The only additional circuitry necessary are two 1.8kOhm pullup resistors as seen below.



2. The Code

Next, open up your Arduino IDE and copy in the code below.

Toggle line numbers

```
/*
 * rosserial SRF08 Ultrasonic Ranger Example
 *
 * This example is calibrated for the SRF08 Ultrasonic Ranger.
 */
#include <Sonar_srf08.h> //SRF08 specific library
#include <WProgram.h>
#include <Wire.h>
#include <ros.h>
#include <std_msgs/Float32.h>


//Set up the ros node and publisher
std_msgs::Float32 sonar_msg;
ros::Publisher pub_sonar("sonar", &sonar_msg);
ros::NodeHandle nh;


Sonar_srf08 MySonar; //create MySonar object

#define CommandRegister 0x00
int New_Address = 248; //0xF8
#define ResultRegister  0x02

float sensorReading =0;

char unit = 'i'; // 'i' for inches , 'c' for centimeters


void setup()
{
  MySonar.connect();
  MySonar.changeAddress(CommandRegister, New_Address);
  New_Address += 4;
  nh.initNode();
  nh.advertise(pub_sonar);

}


long publisher_timer;

void loop()
{

  if (millis() > publisher_timer) {

   // step 1: request reading from sensor
   MySonar.setUnit(CommandRegister, New_Address, unit);

   //pause
  delay(70);

  // set register for reading
  MySonar.setRegister(New_Address, ResultRegister);

  // read data from result register
  sensorReading = MySonar.readData(New_Address, 2);

  sonar_msg.data = sensorReading;
  pub_sonar.publish(&sonar_msg);

  publisher_timer = millis() + 4000; //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. Launching the App

```
roscore
```

```
rostopic echo sonar
```

Wiki: rosserial_arduino/Tutorials/SRF08 Ultrasonic Range Finder (last edited 2014-09-22 16:30:14 by AustinHendrix (/AustinHendrix))

Brought to you by: Open Source Robotics Foundation

(http://www.osrfoundation.org)