Project Title

ROS publisher node development for distance measurement using Ultrasonic sensor and Raspberry pi 3 (Ubuntu 16.04)

Developed by: MSc. Shiva Agrawal

Place: Germany

Date: October 2018

Content

List of Figures

1. Introduction

1.1. Project description

ROS is very powerful and efficient way of implementing and designing the Robotics project. In order to work with different sensors in ROS, their drivers are required to develop and test. As I am working in the field of Robotics with main focus as self-driving cars, this project is part of the series projects which I am going to implement in future.

For the self-driving vehicle, the very first step is the perception and information extraction as shown in figure below [1].

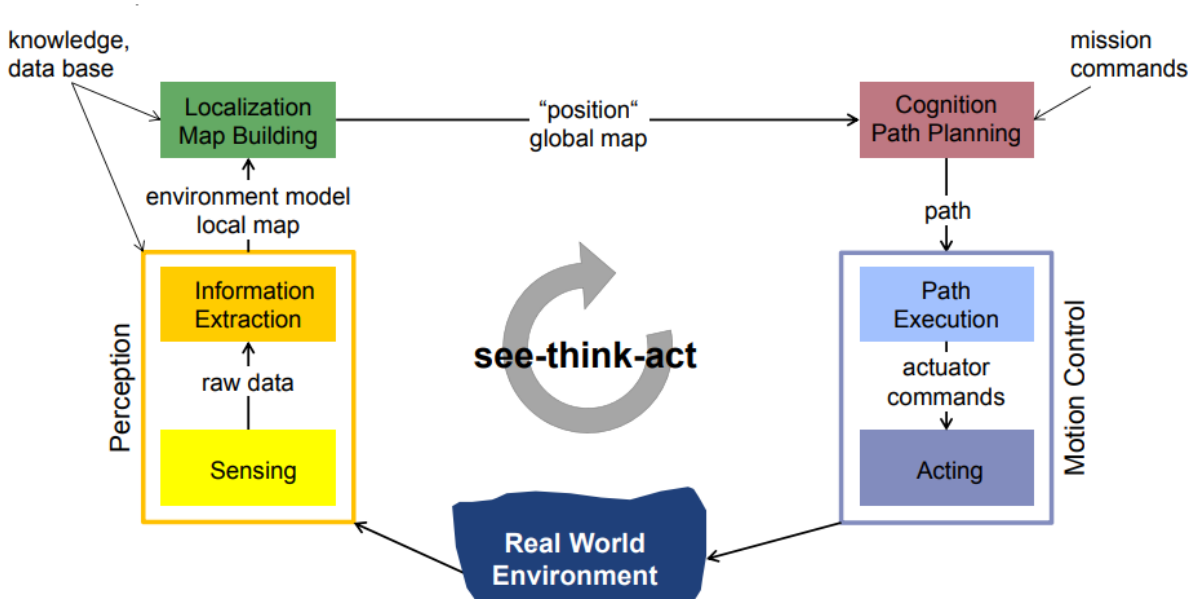

Figure 1 See Think Act cycle | Autonomous mobile robots

Perception includes two main parts:

- Perception of environment
- Perception of own vehicle parameters and behavior

Environmental perception is mainly done by following sensors:

- Laser scanner and Velodyne
- Radar
- Lidar
- Ultrasonic sensor
- Mono / stereo cameras

To work with each sensor in ROS, node is required to develop which can do the perception and then send the data to other processing nodes.

In this project, Ultrasonic sensor node is written which is interfaced with ROS Kinetic through Raspberry pi 3 model B. The node is written using C++ and is tested for measuring distance from the obstacle for various situations.

The ROS is installed on the rasp-pi development board. Other resources used as references are ROS wiki and online documentation of HC-SR04 Ultrasonic sensor.

1.2. Outline

- Chapter 1: Project description and outline of the project report
- Chapter 2: ROS and HC-SR04
- Chapter 3: Implementation
- Chapter 4: results
- Chapter 5: Conclusion
- Chapter 6: References

2. ROS and Ultrasonic sensor

2.1. ROS (Robot Operating System)

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license. The official website of the ROS www.ros.org is full with everything one needs to start with ROS [2].

Different versions of the ROS are available for the user purpose, but I have used ROS Kinetic.

2.2. ROS package

Software are organized in ROS using packages. Each package then further contains one or more nodes (a process that performs computation), dataset, configuration files, 3rd party software piece, robot model files, launch files, service and parameter files, etc. [3]

Common files and directories inside a package directory are:

- include/package_name: C++ include headers
- msg/: Folder containing Message (msg) types
- src/package_name/: Source files (C++ / Python)
- srv/: Folder containing Service (srv) types
- scripts/: executable scripts
- launch/: contain launch files to execute multiple nodes at a time
- urdf/: contains xacro and urdf files of robot model
- CMakeLists.txt: CMake build file
- package.xml: Package catkin/package.xml

NOTE: detail description of package development is out of scope of this document. Readers are requested to check ROS wiki.

2.3. Ultrasonic sensor [HC-SR04] and interfacing with Raspberry-pi

The HC-SR04 Ultrasonic sensor has four pins: ground (GND), Echo Pulse Output (ECHO), Trigger Pulse Input (TRIG), and 5V Supply (Vcc) as shown in fig.2. GND and VCC pins are connected to GND and 5V of raspberry pi. Further raspberry Pi send an input signal to TRIG (10 microseconds), which triggers the sensor to send an ultrasonic pulse. The pulse waves bounce off any nearby objects and some are reflected to the sensor. The sensor detects these return waves and measures the time between the trigger and returned pulse, and then sends a 5V signal on the ECHO pin.

ECHO will be “low” (0V) until the sensor is triggered when it receives the echo pulse. Once a return pulse has been located ECHO is set “high” (5V) for the duration of that pulse. Pulse duration is the full time between the sensor outputting an ultrasonic pulse, and the return pulse

being detected by the sensor receiver. This duration is measured and then distance to the object is calculated in cm as per below formula.

$$34300 = \frac{Distance}{Time/2}$$

$$17150 = \frac{Distance}{Time}$$

$$17150 \times Time = Distance$$

NOTE: The sensor output signal (ECHO) on the HC-SR04 is rated at 5V. However, the input pin on the Raspberry Pi GPIO is rated at 3.3V. Sending a 5V signal into that unprotected 3.3V input port could damage your GPIO pins. Hence voltage divider circuit is used between the echo pin of the ultrasonic sensor and raspberry pi as shown in fig 3.

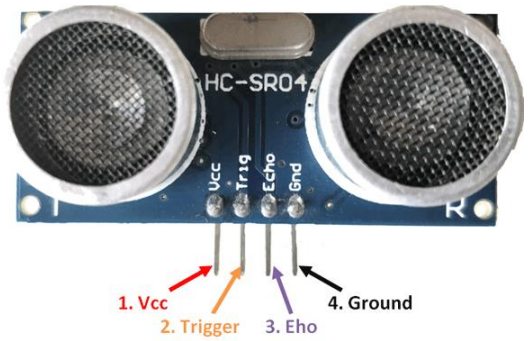

Figure 2 - Ultrasonic sensor

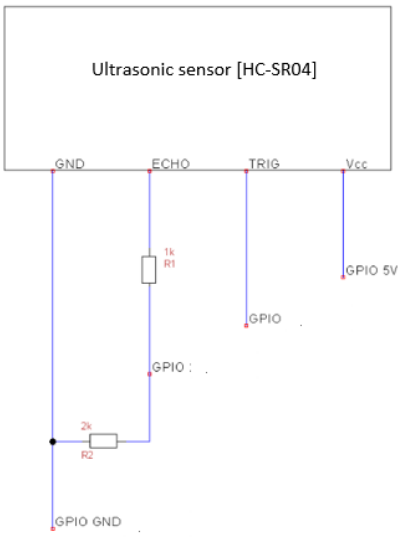

Figure 3- HC-SR04 pin connections

3. ROS node development

ROS node is developed in C++ and is available with ROS package 'sensorRead' in src folder of the project.

The process flow of the node can be explained in below steps

1. ROS publisher node is created.
2. It measured the distance using ultrasonic sensor at 1Hz
3. The distance is stored as Float32 type of std_msgs (inbuilt ROS messages)
4. Then this msg is published on ROS topic

Moreover, this project report and user guide for ultrasonic sensor is available in doc folder of the project repository

4. Test and results

To test the node, at first ROS master is started using the command

$ roscore

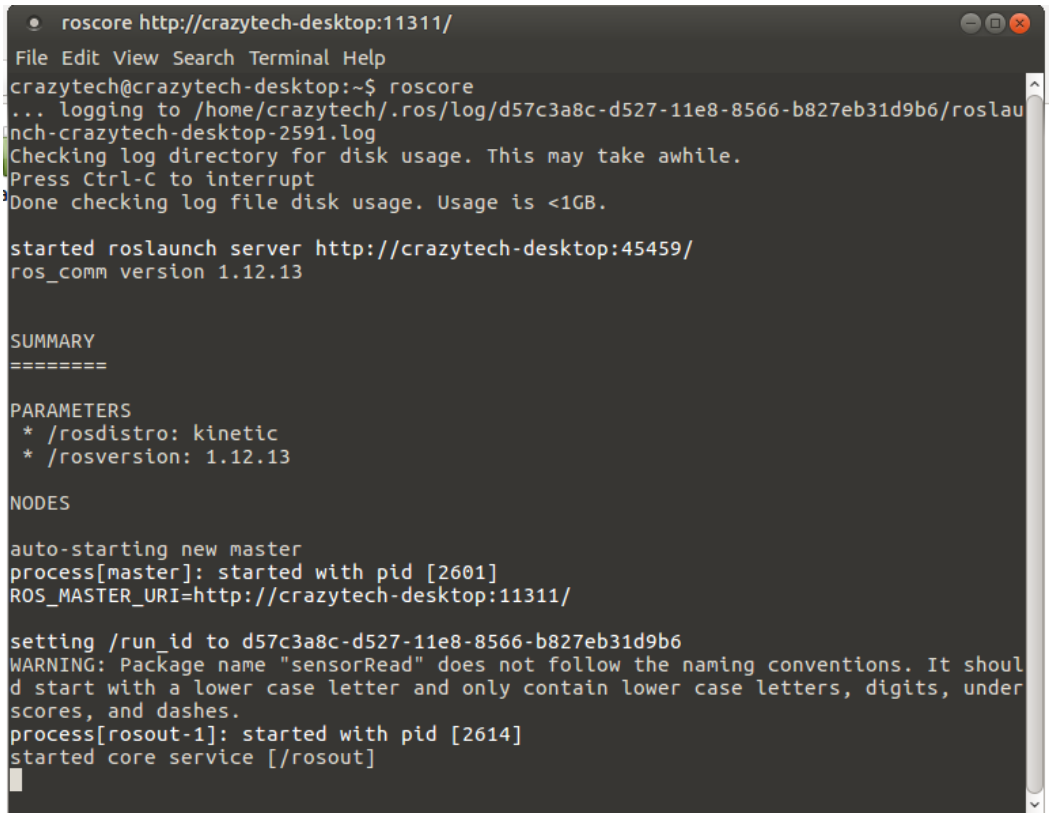

Figure 4 ROS master

Next, difference distance measurements are taken by keeping an obstacle in front of the sensor mounted on the vehicle and then the measurement are seen in the terminal window.

To start the node

$ rosrun sensorRead UltrasonicSensorNode

1. Ultrasonic sensor distance Measurement 01

```
crazytech@crazytech-desktop: ~
File  Edit  View  Search  Terminal  Help

crazytech@crazytech-desktop:~$ rosrun sensorRead UltrasonicSensorNode
[ INFO] [1540123587.075731393]: Ultrasonic sensor measurement
[ INFO] [1540123587.076786162]: Distance from Object :30.372650 cm

[ INFO] [1540123588.075523695]: Ultrasonic sensor measurement
[ INFO] [1540123588.076310345]: Distance from Object :29.961050 cm

[ INFO] [1540123589.075477031]: Ultrasonic sensor measurement
[ INFO] [1540123589.076290764]: Distance from Object :29.532301 cm

[ INFO] [1540123590.075500465]: Ultrasonic sensor measurement
[ INFO] [1540123590.076312635]: Distance from Object :29.961050 cm

[ INFO] [1540123591.075476863]: Ultrasonic sensor measurement
[ INFO] [1540123591.075799929]: Distance from Object :29.943899 cm

[ INFO] [1540123592.075470078]: Ultrasonic sensor measurement
[ INFO] [1540123592.075788561]: Distance from Object :29.961050 cm

[ INFO] [1540123593.075471049]: Ultrasonic sensor measurement
[ INFO] [1540123593.075733543]: Distance from Object :29.961050 cm
```

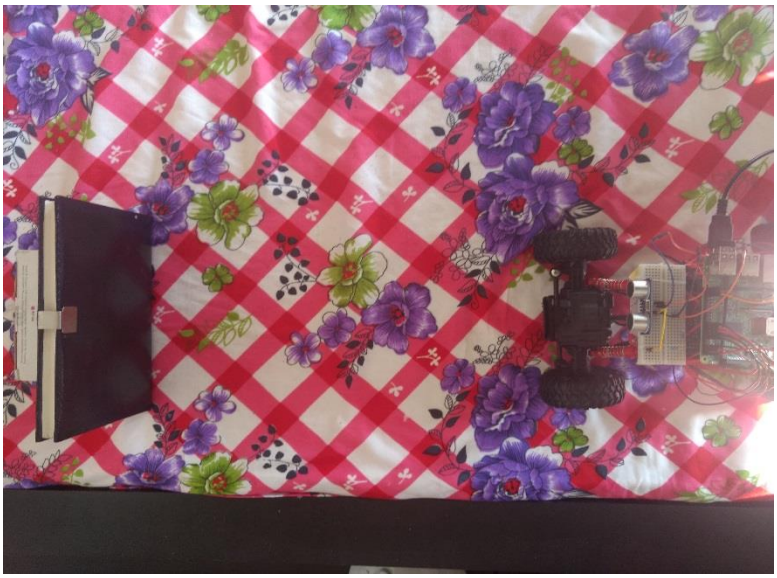

Figure 5 - Measurement 01

2. Ultrasonic sensor distance Measurement 02

```
crazytech@crazytech-desktop: ~
File  Edit  View  Search  Terminal  Help

crazytech@crazytech-desktop:~$ rosrun sensorRead UltrasonicSensorNode
[ INFO] [1540123808.146692242]: Ultrasonic sensor measurement
[ INFO] [1540123808.147507282]: Distance from Object :49.443451 cm

[ INFO] [1540123809.146381604]: Ultrasonic sensor measurement
[ INFO] [1540123809.146798733]: Distance from Object :47.728451 cm

[ INFO] [1540123810.146341191]: Ultrasonic sensor measurement
[ INFO] [1540123810.146739883]: Distance from Object :47.745602 cm

[ INFO] [1540123811.146341421]: Ultrasonic sensor measurement
[ INFO] [1540123811.146747405]: Distance from Object :47.745602 cm

[ INFO] [1540123812.146367506]: Ultrasonic sensor measurement
[ INFO] [1540123812.146764844]: Distance from Object :48.140049 cm

[ INFO] [1540123813.146335068]: Ultrasonic sensor measurement
[ INFO] [1540123813.146739906]: Distance from Object :47.762749 cm

[ INFO] [1540123814.146332546]: Ultrasonic sensor measurement
[ INFO] [1540123814.146732228]: Distance from Object :47.728451 cm
```

Figure 6 Measurement 02

3. Ultrasonic sensor distance Measurement 03

```
crazytech@crazytech-desktop: ~
File  Edit  View  Search  Terminal  Help

crazytech@crazytech-desktop:~$ rosrun sensorRead UltrasonicSensorNode
[ INFO] [1540123914.669002592]: Ultrasonic sensor measurement
[ INFO] [1540123914.669471753]: Distance from Object :76.540451 cm

[ INFO] [1540123915.668954378]: Ultrasonic sensor measurement
[ INFO] [1540123915.669371143]: Distance from Object :76.557602 cm

[ INFO] [1540123916.668964301]: Ultrasonic sensor measurement
[ INFO] [1540123916.669398878]: Distance from Object :76.557602 cm

[ INFO] [1540123917.668938768]: Ultrasonic sensor measurement
[ INFO] [1540123917.669357826]: Distance from Object :76.574753 cm

[ INFO] [1540123918.668938353]: Ultrasonic sensor measurement
[ INFO] [1540123918.669352827]: Distance from Object :76.540451 cm

[ INFO] [1540123919.668938316]: Ultrasonic sensor measurement
[ INFO] [1540123919.669363414]: Distance from Object :76.557602 cm

[ INFO] [1540123920.668953916]: Ultrasonic sensor measurement
[ INFO] [1540123920.669391619]: Distance from Object :76.540451 cm
```

Figure 7 Measurement 03

All the above screenshots are also available in the **test** folder of project repository.

5. Conclusion

ROS node for distance measurement in cm suing ultrasonic sensor HC-SR04 is developed, implemented and tested.

6. References

[1] R. Siegwart, Introduction to Autonomous Mobile Robots 2nd edition.

[2] "ROS wiki," open source Robotics Foundation , [Online]. Available: http://www.ros.org/.

[3] "packages," ROS open source, [Online]. Available: http://wiki.ros.org/Packages.