Table of Contents

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Introduction to ROS
- Development Guide
- Driving Willy
- Manual
- Wiki Manual

Build of Willy

- Design history
- Hardware

Architecture

- Software Architecture
- ROS topic design

Raspberry Pi’s

- Sensor node
- Social Interaction node

Components

Lessons learned

Archive

1. MPU9250

The MPU9250 is a 9-axis Inertia Measurement Unit. This means that it can detect rotation in 9 directions. It does this by utilizing an accelerometer, gyroscope and magnetometer. The IMU is located on the top of the screen, in a 3-d printed box with the words "IMU TOP" printed on the lid.

The magnetometer (AK9863) is separate from the accelerometer and gyroscope, but can be read using the same library. The data sheet is referenced in the technical design document, which also contains a more detailed explanation.

1.1. Arduino

The MPU9250 is connected using I2C to an arduino, which does a large part of the processing. It calculates the quaternion, as well as transforming some of the data into useful units. These are then changed into separate bytes, and sent via Serial communication to the Sensor Node Raspberry Pi.

1.2. Raspberry Pi

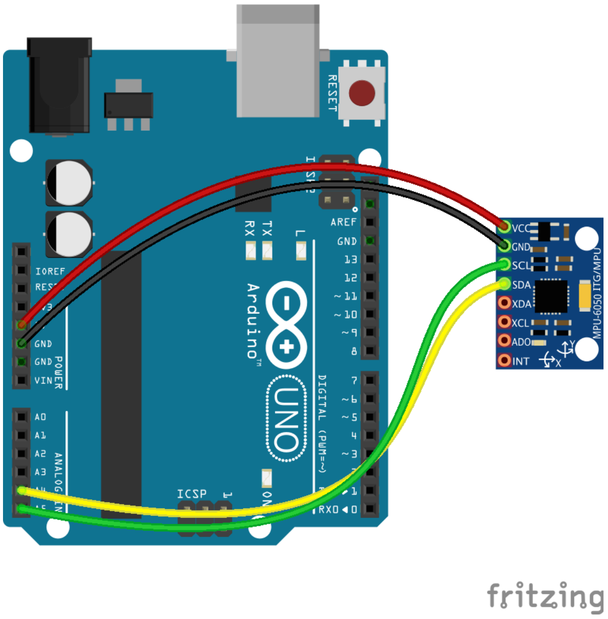
The raspberry pi stores all the data it receives into a string, which it then proceeds to check for a start byte. After this, it uses Unions, an old C technique, to process the separated bytes into the desired data types. These are built into a message, and sent to the master.

If there is an error in the data from the arduino, such as missed data or un-needed data, the entire message is binned and the input is cleared.

1.2.1. Quaternions

The MPU9250 uses a complex mathematical system of tracking orientation called "quaternions". These are a method of storing current rotation in a 3-d space, while avoiding the issue of Gimbal lock found in Euler angles. While these are complex, they are the preferred method of tracking rotation according to ROS rep 103. A comprehensive tutorial on the mathematics behind quaternions can be found here.

1.3. Wire



1.4. set setpoint to 0

Sometimes it is needed to set the "set point" to 0. The command to do this is as follows:

```
rosservice call /imu/set_zero_orientation
```

2. IMU

The IMU sensor (pose tracking) is used to get a higher accuracy of Willy the Robot’s orientation for navigation and localization purposes.

2.1. Repository

Windesheim-Willy/pose_tracking

2.2. Prerequisites

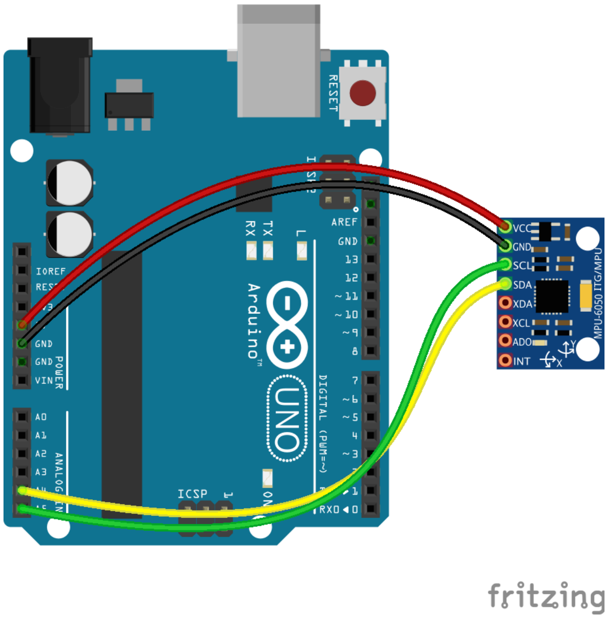
The IMU sensor (Arduino) must be connected to the Sensor Pi via USB.

2.3. How to run?

The pose tracking will start automatic when the PI is running. If it doesn’t, the pose tracking node can be started manually by going into the root of the pose tracking node’s source directory:

```
./START start
```

2.4. Wiring schematic



2.5. Calibration

Sometimes it is needed to set the sensor’s origin to 0. The command to do this is as follows:

```
rosservice call /imu/set_zero_orientation
```

It would be wise to do this each time WTR is started, and after its position/orientation has been set in RViz.

2.6. Background

The IMU data that is published on the topic `/imu/data`, is used by move_base, amcl and/or the laser_scan_matcher node.