SWIMMING TANGO
User Guide

Rishabh Ananthan
Anupam Goli

ECE 4012 L7A

Advisor: Dr. Mick West

1. WHAT YOU'LL NEED

Before you get started, you will need the following items:

- A computer running Ubuntu 14.04, pressure transducer and compass module

- A breadboard and wires, in order to make necessary connections.

- An mbed microcontroller. This acts as an A/D converter for the pressure transducer.

- A serial-to-USB cable, for connecting the compass to the computer.

2. SETTING UP THE DEVELOPMENT ENVIRONMENT

- You will have to install the following software: GTKTerm and OpenCV. Installing GTKTerm will also install all drivers necessary to read RS-232 data on Ubuntu.

- GTKTerm can be installed by running the command:

  ```
  sudo apt-get install gtkterm
  ```

- OpenCV can be downloaded by cloning the following GIT repository: https://github.com/Itseez/opencv

  Instructions on how to install OpenCV after cloning can be found here: http://docs.opencv.org/3.0-beta/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation

3. USING THE PRESSURE TRANSDUCER AND COMPASS

The pressure transducer (Figure 1) is connected to the breadboard as shown in Figure 2.

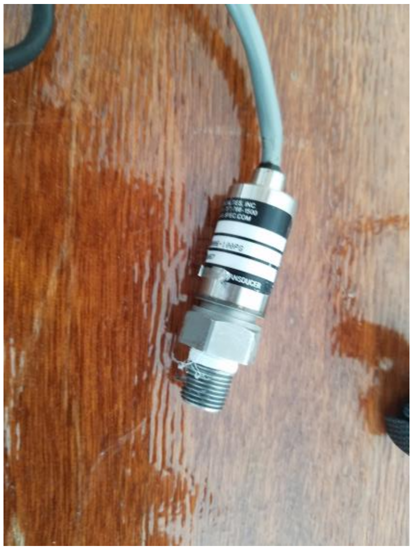


Figure 1. The pressure transducer.

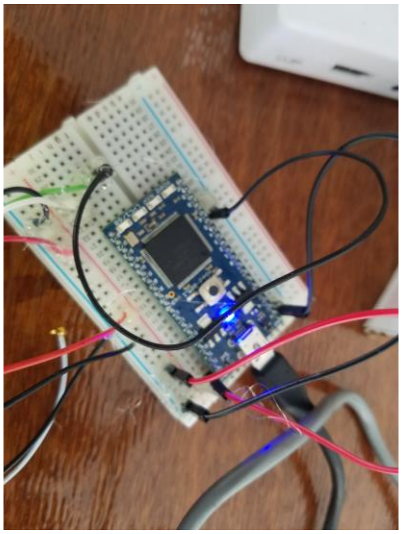


Figure 2. Connecting the pressure transducer to the mbed.

The compass is connected to the serial-to-USB connector as shown in Figure 3. The green wire goes into pin 2 and the blue wire goes into pin 3.

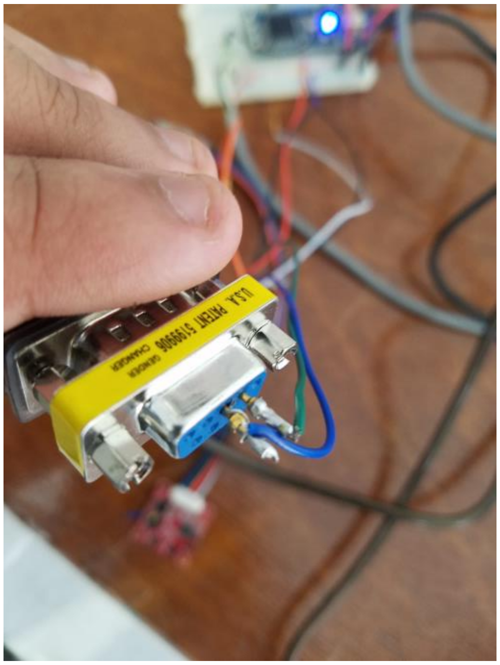
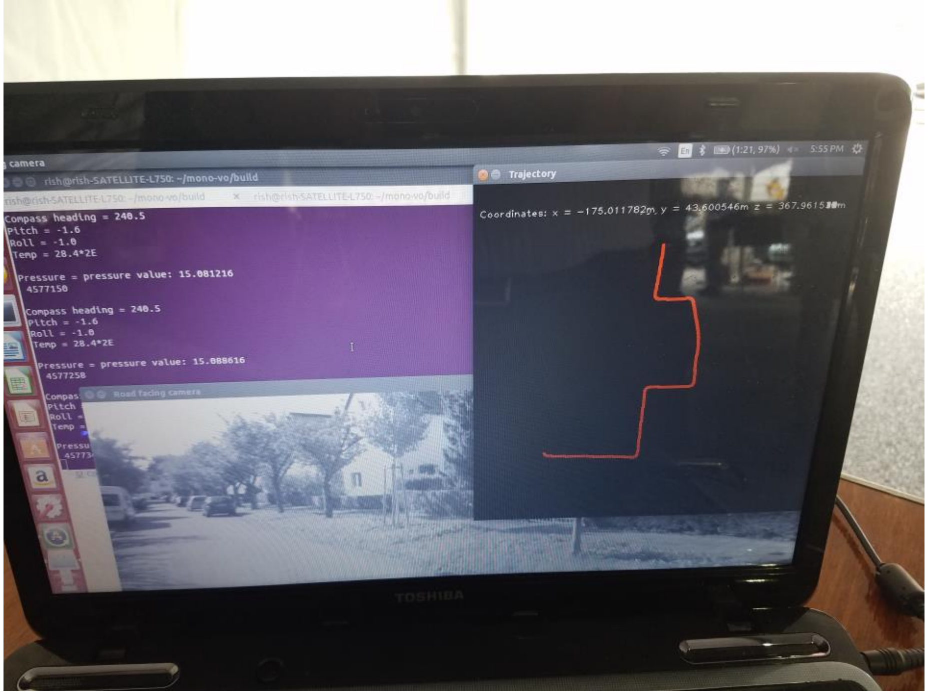


Figure 3. Connecting the compass to the serial-to-USB connector.

Once this is done, download the file ‘serial_read.py’ from the project webpage. In the terminal, navigate to the folder where you’ve saved the file and enter:

```
sudo python serial_read.py
```

This should show output similar to the terminal window in Figure 4.



camera
rish@rish-SATELLITE-L750: ~/mono-vo/build
rish@rish-SATELLITE-L750: ~/mono-vo/build
rish@rish-SATELLITE-L750: ~/mono-vo/build
Road facing camera
Trajectory
(1:21, 97%)
5:55 PM
TOSHIBA

```
Compass heading = 240.5
Pitch = -1.6
Roll = -1.0
Temp = 28.4*2E

Pressure = pressure value: 15.081216
 4577150

Compass heading = 240.5
Pitch = -1.6
Roll = -1.0
Temp = 28.4*2E

Pressure = pressure value: 15.088616
 4577258
```

```
Coordinates: x = -175.011782m, y = 43.600546m z = 367.9615[illegible]m
```

Figure 4. Image showing visual odometry and serial_read.py running.

4. VISUAL ODOMETRY

For visual odometry, you will have to download two items:

- The code, which can be cloned from: https://github.com/avisingh599/mono-vo

- The dataset, which can be downloaded from: http://www.cvlibs.net/datasets/kitti/eval_odometry.php

  Download either the greyscale or the color version. The velodyne dataset is pointcloud information, for which the algorithm will have to modified to support pointcloud data!

Once you have cloned the git repository, navigate into the 'src' directory and open the file visodo.cpp. You will have to either edit the folder structure in the four places where the code looks for images to match wherever you have downloaded the dataset, or download the dataset to the exact same folder as specified in the code. Once this is done, navigate to the build directory and in the terminal, enter:

```
./vo
```

You will see output similar to that of the black window and image feed in Figure 4.