CSSE4011 Prac 3
Localisation and Data Fusion

1 Assessment

- Due in the first hour of your lab session in week 9 .
- Course Marks: 10%
- Electronic Course Profile Pass Hurdle: Must be submitted.

2 Resources

- Argon , Thingy:52 , NRF52840 USB Dongle or B-L475e-IOT01A
- Partner: You will need to be in a group of 2, to demo this prac.
- Grid and $1m$ high, static node stands. (Shared amongst groups)

3 Introduction

This practical introduces you to the basics of ranging and localisation using data fusion with RF signal strength and ultrasonic ranging for wireless sensor networks. Ranging and localisation are commonly used with data fusion algorithms for tracking objects, which is an important application of wireless sensor networks. The goal is to understand the basics of data fusion using RF signal strength proximity, ultrasonic ranging and a dead reckoning motion model.

- Use Zephyr OS
- Re-usable (not copied) for future practicals and projects.
- Allow for Library plugins.
- Serialising Libraries e.g. JSON or Protocol Buffers (nanopb).
- Python (for PC based processing)

An overview of the prac can be seen in Figure 1. This prac will involve tracking a person on a minimum $4m$ by $4m$ grid (this may change depending on the room used), using a mobile node (worn by person) and four static nodes. The base node (laptop) is used to receive messages from the mobile node (via Wifi).

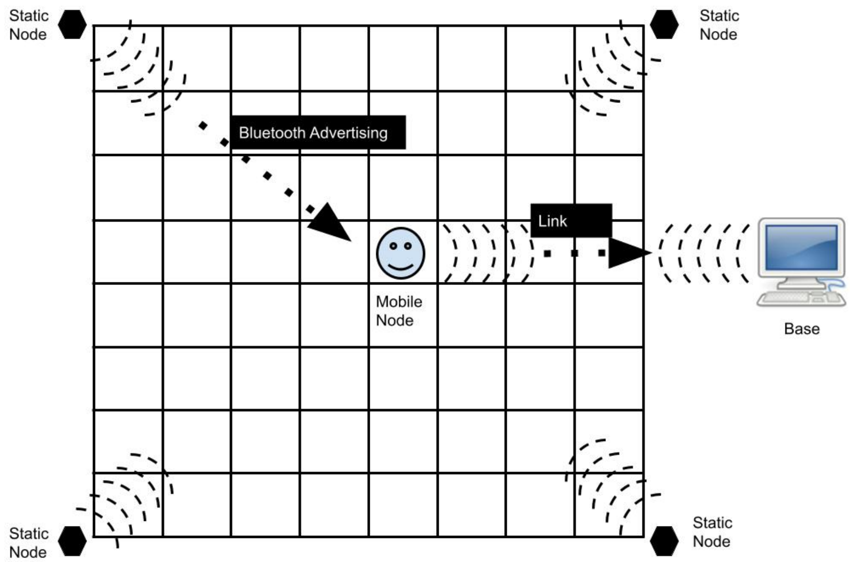

Figure 1: Network Overview

4 Design Tasks

Design tasks are due in the first hour of your week 9 stage session. Refer to Tutorial 8, UQCsse4011 Lab 4 presentation and "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors" by Savvides et al. You can use the python examples that are linked on Black Board.

4.1 MyOsLib Setup

Create your MyOsLib library. Follow the MyOslib guide. Your MyOsLib must be committed to your git repository.Your myoslib code must meet the guidelines specified in the myoslib guide on Blackboard.

4.2 Design Task 1: Network

Implement the network seen in Figure 1. Communicate via Bluetooth advertising from the static to the mobile node. The base node must be based on the NRF52840 USB Dongle connected to a Laptop. The mobile node must be based on the Thingy:52 . The static nodes can be based on the Argon (with SCU), Thingy:52 , NRF52840 USB Dongle or other Bluetooth devices (that you can provide). Two static nodes should transmit ultrasonic ranging information to the mobile node. The remaining static nodes should be used only for RSSI ranging. Use a Bluetooth connection between the mobile and base node. The mobile node should send ranging (ultrasonic and RSSI) and motion information back to the base for processing. The motion information should consist of displacement (e.g. step) and heading information, extracted from the mobile node's Inertial Measurement Unit (IMU) sensors. Use a suitable serialisation protocol such as JSON or protocol buffers (nanopb library), to encode the ranging and motion information (as messages). You must be able to show the timestamp, received ranging and motion information. You must be able to explain your time synchronisation process.

4.3 Design Task 2: Ultrasonic Ranging

The ultrasonic ranger sensor uses ultrasound to detect objects, placed in a direct line of sight. The ultrasonic ranger generates an ultrasound pulse and then measures the returned echo. The period of the returned echo can be used to estimate the distance between the ultrasonic ranger and an object. You will attempt to use the ultrasonic ranger sensor to determine short distance resolution. This involves:

- Writing the myoslib driver to use the ultrasonic ranger on the static nodes. Hint use a delay timer to with microsecond precision.
- Performing ranging tests and deriving a relation between the ultrasonic pulse period and distance between the ultrasonic ranger and object.
- Refer to the following links for more details:
- Ultrasonic Ranger
- Connecting the ultrasonic ranger to the Argon board.

4.4 Design Task 3: Bluetooth Advertising based RSSI Ranging

You will attempt to localise object inside a coverage area using your BLE scanning/advertising Library. This involves:

- Using known static beacon coordinates [x1,y1]. . . [xk,yk], create matrix A and vector b of the linearised multilateration problem
- Use the least squares equation to estimate location of the object. Refer to Tutorial 8, which introduces the Numpy Least Squares Function.
- Display the location coordinates of the object using the python listener script (graphical UI is expected).

4.5 Design Task 4: Data Fusion

- Estimate the error of RF and ultrasound ranging methods based on measuring the ranging error at several different Tag locations.
- Develop a localization system that uses both RF (high coverage, low accuracy) and ultrasound (low coverage, high accuracy) methods. Use a Kalman filter to fuse information from RF and ultrasound localization methods and track a moving object. You can assume a simple motion model (constant velocity motion). You can use the motion

and heading information from the mobile node. You can assume the starting position of the tracked tag is known. You must be able display the location estimates from the Kalman filter output (graphical UI) and demonstrate higher localisation accuracy in areas where ultrasound ranging is available.

- Display the current position on a PC, in realtime.

5 Final Demo

You must be able to combine all design tasks (where possible) into the same `main.c` file and demo all of them without reprogramming. *Note*: You may be asked to make minor modifications to your code by the assessor, during your demo, which must be passed.

5.1 Git Submission

You ar required to submit your code in your `p3` folder. Create the following folders for use with your code: static, mobile and base. Any references should be listed in the README.md.

6 Criterion

The prac demonstrations are marked according to the criterion outlined in the table below. If you fail to demonstrate sufficient understanding and functionality in the specified marking time you will not be allowed to repeat the stage. You must pass the pre-demo checks before you are allowed to demo. **All code assessed for the stage must be your own work.**

6.0.1 Pre Demo Checks

The following criteria **must** be met **before** you are allowed to demo.

Check	**P/F**
Your latest stage and mylib code **must** be in git.	
Your git repository must be up to date.	
Your stage code must build without errors.	
Your myoslib and top comments are correctly filled out.	

Failure to meet pre-demo checks will mean that you are not allowed to demo the stage.

6.0.2 Demo Criteria

Note: You may be asked to make minor modifications to your code by the assessor, during your demo, which must be passed.

Design Task 1: Network	
3	Network Implementation.
2	Serialisation protocol with display output viewer.
Design Task 2: Ultrasonic Ranging	
5	For ultrasonic ranging, you must show that you have derived a distance separation equation
Design Task 3: Bluetooth RSSI Ranging	
5	For RF Localisation, you must demonstrate that you can determine an object's position using multilateration and be able to display the object's position (text).
3	Use weighted multilateration to utilize the extended propagation model (i.e., the ranging error estimates) in the localization
Design Task 4: Data Fusion	
5	• For Data Fusion, you must show both position estimates using Multilateration and Kalman filtering using graphical real-time interface (i.e., dot moving on a screen).