A Scalable long range RF communication platform over water bodies with an Android app

at

Acoustic Research Laboratory

TMSI

National University of Singapore

by

Pritesh Nandgaonkar

Department of Electrical Engineering

Indian Institute of Technology-Bombay

India

pncooldude91@gmail.com

prit91@iitb.ac.in

Guided by

Koay Teong Beng

koay@arl.nus.edu.sg

Index

1. Acknowledgment

2.0 Overview and motive
- 2.1 Primary Objective

3.0 Explanation of the project
- 3.1 Hardware Setup
- 3.2 Explanation of Blocks B1 and B2
  - 3.2.1 Format of the data
  - 3.2.2 Java Program
- 3.3 Explanation of block B3(TTL converter),B4(RF transmitter) and B5(RF receiver)
- 3.4 Explanation of the Block B6(Arduino) and B7(RS232 converter)
- 3.5 Explanation of the Block B8(BT module)

4.0 Android App
- 4.1 Plotting GPS points
- 4.2 GUI of the app

5.0 Steps to Setup the Hardware
- 5.1 Acquainting with the Android program and Java program

6.0 Future Improvements

7.0 References

1.Acknowledgement

I would like to thank Prof. Mandar Chitre for giving me the valuable opportunity to work at the lab.

I would like to sincerely thank Koay Teong Beng for guiding me all through the project and giving me the opportunity to work on this. He has been a patient and supportive mentor who provided me with the necessary input and helped me whenever required. It has been a great learning experience to work on this project under him.

Last but not the least, I thank Shailabh Suman as well for his support as the internship coordinator who has always been a person of great help and support, regardless of the issue and time that is needed to be spent on it.

The laboratory as such is a wonderful place to work, where I had the opportunity to interact with various people from various parts of the world and distinct thought processes. This has been an amazing experience for me and it definitely made my summers a valuable and enjoyable one

2.0 Overview And Motive

The aim of the project is to set up the communication link between the base station and the chase boat which is monitoring the movement of the AUV's. The need of this communication is felt when the AUV observer on the boat needs to verify whether the AUV is following the allotted mission path or not. Also if there is some unlikely event the observer should be equipped with the facility to abort the AUV's mission.

2.1 Primary Objective

- Send data from base station to the mobile through RF link
- Display the data on the android phone with easy to understand GUI
- Setup the communication such that the mission files at the base station can be changed through the communication
- Write a program which monitors the updates and mission files and sends the data through RF link on chase bat or to the handset

3.0 Explanation of the Project

3.1 Hardware Setup:-

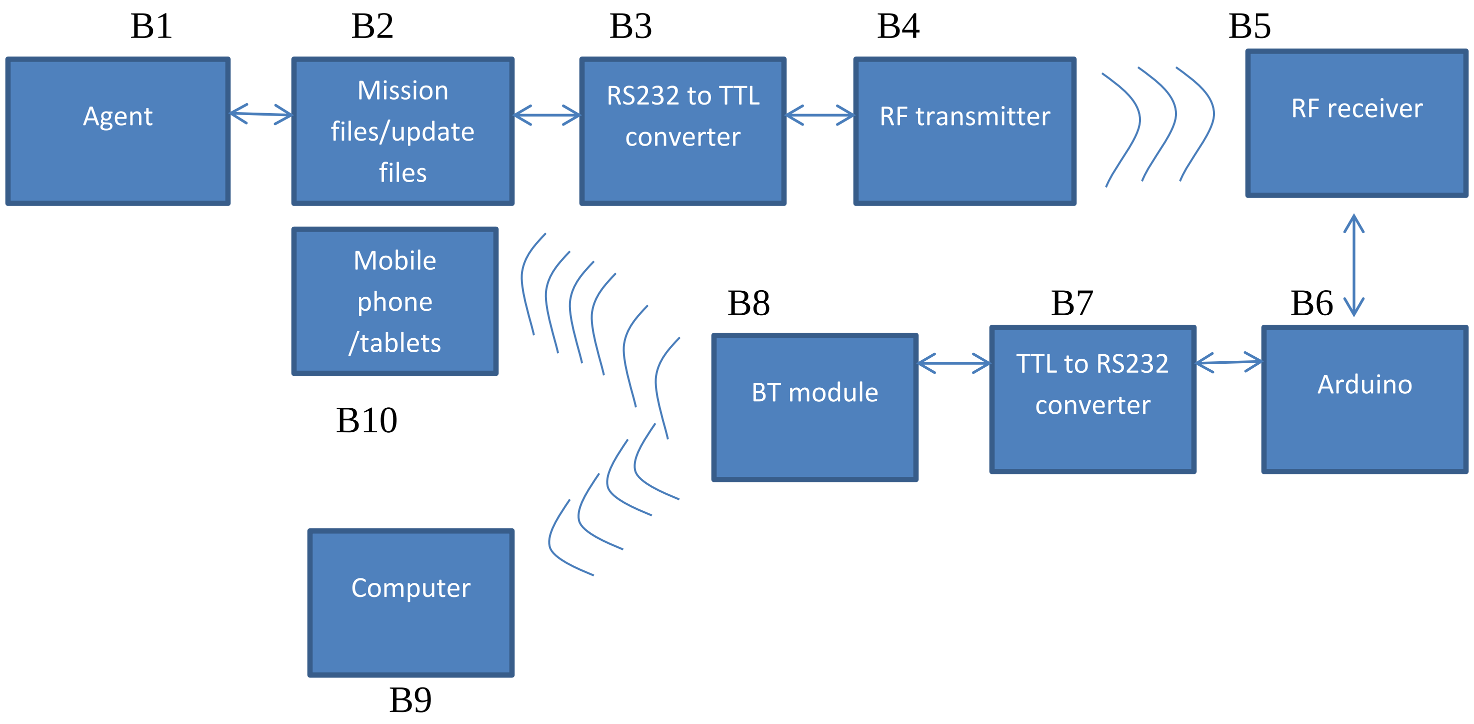

Fig 1

3.2 Explanation of the Blocks B1 and B2:-

The agent at the base station (place where AUV's are deployed) updates the mission files (it's a file which contains all the points of the path, the AUV has to follow) in the pre decided folder once the AUV with a particular id is about to be deployed. Also agent will update the AUV update file (it's a file which stores the current location updates of the AUV received by the agent) of different AUV's distinguished on the basis of the id's assigned to each AUV. There is a java program which continuously runs in background for checking any changes in these mission and AUV update files. And as soon as the new mission file is created related to particular AUV id or the AUV update file is updated with the latest location, these java program sends the updated information of AUV update file and the whole mission file through COM port.

3.2.1 Format of the data:-

The format used to store the data in mission and AUV update files are JSON format. The example of this format is as follows:-

Mission file:-

{"mission":{"id":0,"lat-lon":[1.296,103.7767,1.376,103.8567,1.456,103.9367,1.536,104.0167,1.616,104.0967,1.616,104.1967,1.626,104.1667]}}

Update file:-

{"update":{"id":1,"lat":1.206,"lon":103.707,"head":50}}

{"update":{"id":1,"lat":1.216,"lon":103.717,"head":70}}

Just by seeing the fields of the format we can make out what it means. The field "head " means the angle of the AUV with the assumption that 0^0 points to north direction. And "lat" and "lon" are the latitude and longitudes respectively.

The data stored in “lat-lon” field is an array with the first element as latitude, second one as longitude ,third one as again latitude and this goes on. The format can be made more robust(part of future work),because in this case if there is an error in receiving one lon point then we would mess up all the future points.

The name of the JSON object (e.g “mission” in mission file and “update” in update file) is passed on as command line argument for verification in java program.

The advantage of this format is that it is easy to parse it as you receive the data.

3.2.2 Java Program:-

The java program which I am using for accessing the port and sending the data through the same use RxTx API, the documentation of which can be found easily on internet.These java program continuously checks the files named "mission_1.txt","update_1.txt","mission_2.txt ","update_2.txt ","mission_3.txt ","update_3.txt ","mission_4.txt ","update_4.txt ","mission_5.txt ","update_5.txt " in a folder ,the path of which is passed as a command line argument to the program. These files are not created manually, it is created by the agent once the mission file of the particular id is deployed in the AUV. The format of the mission file is “mission_id”. Once the agent starts receiving update it makes the file named “update_id” and adds the data in it. Here I have assumed the no. of AUV’s deployed simultaneously would be less than or equal to five. The things which are passed as a command line argument are COM port no., the path of the directory/folder with no blank spaces, the JSON object name of the mission files and AUV update file as discussed in previous in section.

The settings of the COM port are hardcoded in the code since I have used the same settings in all the blocks of the hardware. The settings are as follows:-

Baud rate:-57600
Data Bits:-8
Parity:-None
Stop bit:-1
Flow Control:- None

One can see the documentation and steps to use the RxTx API at these link:- http://www.kuligowski.pl/java/rs232-in-java-for-windows,1

The idea behind checking the change or update in the file is to check the “last modified” field associated with the class “File” in Java. There is a method associated with the object of the “File” to get this information.

One can refer the documentation and idea used in the detection of change in the file by looking at this link:-

http://www.rgagnon.com/javadetails/java-0490.html

3.3 Explanation of block B3(TTL converter),B4(RF transmitter) and B5(RF receiver):-

Once the data is transmitted through COM port of the computer through the above explained java program it is required to be converted in TTL format. The data transmitted from the COM port is converted first to RS232 serial protocol with the help of USB to serial cord easily available in the market. We cannot use it directly to feed the data to RF transmitter or Arduino (to check the correctness of the data) because the voltages in these devices are in the range of 0V-5V.And the voltage range in RS232 protocol ranges from +25 to -25,which is incompatible. That's why one has to use RS232 to TTL converter to get the voltage in the range 0V-5V.The Converter used is "ET-Mini 232-TTL3".

Block B4 is the RF transmitter which is set in the communication mode of "40 pin header". This can be done by moving the communication jumper of the RF transmitter on the board to "40 pin header". The jumper of the power source is at "power conn", i.e. power is obtained from the 40-pin header of the board. The baud rate of the board can be adjusted by loading the software accompanied with the RF board in the PC. The tutorials accompanied with the CD helps to configure the baud rate of the RF board. The currently configured baud rate is 57600.

Once the Block B4 is configured with the above settings, one should replicate the same settings for the RF receiver board too. The RF transmitter board starts to send the data, once it receives the data on its pins in 40-pin header. Similarly the data is available on its 40-pin header in RF receiver for displaying on mobile.

3.4 <u>Explanation of the Block B6(Arduino) and B7(RS232 converter):-</u>

Block B6 is used just for checking whether the information or the data received from the RF transmitter is correct or not. One can even use the Arduino before the RF transmitter to check the data received from the COM port of the PC. One can ignore block B6 in the final implementation. The block B7 is TTL to RS232 converter which is used because the data received from the RF receiver is in TTL level and the Bluetooth module requires the data in RS232 level.

3.5 <u>Explanation of the Block B8(BT module):-</u>

The Bluetooth module which is used in this project has following specifications:-

BT module starter kit: - cB−OBS421i−26−A
Feature:-It has a unique feature of getting connected to more than 1 Bluetooth device at the same time. Maximum it can connect to 7 Bluetooth devices. It also has many modes such as

broad cast mode and addressed mode in it. The documentation can be easily on internet. The link is as follows, http://www.connectblue.com/products/wireless-ready-to-embed-modules/bluetooth-dual-mode-serial-port-module-obs421/

For configuring it, one needs to download the Serial Port Adapter Toolbox which helps in easily selecting the port setting and configuring the board. These GUI does nothing but sending the set of the AT commands to configure the BT module. The set of AT commands an again be found in the above link.

The settings of the BT module used by me are as follows:-

- Configure both as a “Client” and “Server”.
- The profile used in “Client” and “Server” is SPP.
- In client profile MAC address of the BT module is specified with the appropriate peer id.
- “Connect on Data” and “Always Connected” are marked. Which means that BT module will try to be always connected to the specified BT device and specifically try to connect to the BT device if it receives the data to transfer to that BT device.
- In “Server” tab “Always Master” is marked. Which means once the connection is initiated and connection is successfully established by some other BT device with the current Bluetooth module the current BT module will then switch its role from slave to master.

- The rs232 settings are filled based on the settings mentioned in the previous Section “3.2.2 Java program”.
- The Data mode option is ticked in the “Basic settings” tab.
- Once all the basic settings are done, which are given in the documentation on the above mentioned link, one should check the “Reset Module Going to Data Mode” and then press “Data mode”. Now the module is ready to use.

The idea behind using the Bluetooth module as a mediator in sending the data to android mobile is, BT module can be used to send data to multiple mobile handsets, so that multiple users can use the app. Also with the usage of the BT module the operator sitting on the chase boat can change the mission file of the AUV and send the data via BT module-RF Transmitter-RF Receiver-Agent. So the communication link can be setup to be both ways. Also via the BT module the data can be obtained on the PC.

4.0 Android app:-

The Android app has two major parts. The first part is related to showing the current location of the user on the Google map using GPS. We need these information because the chase boat operator needs to know how close he/she is to the deployed AUV’s. The second part of the android app is related to showing the information about the AUV obtained by the mobile from BT module.

The android app uses Google map API version 2 to display the Google maps on the mobile. The steps to set up the Google Maps API can be found at this link https://developers.google.com/maps/documentation/android/start

4.1 Plotting GPS points: -

The current android app has the functionality that it will plot the GPS points of the user on the Google map. The app checks the location of the user in every 2sec and checks whether the distance moved by the user form its previous update is more than 20 meters or not. If the distance is more than 20 meters then the app will plot the new GPS points otherwise it will ignore it. The app only shows the last 9 GPS points on the map, this is because then the map will become too cluttered. Also the user has an option to select the number of GPS points to display on the map.

One can change the criteria of the 2secs and 20 meters used above to any value by changing the following line of code,

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 2000, 20, listener);
```

The snapshot of the GPS portion of the app is as follows.

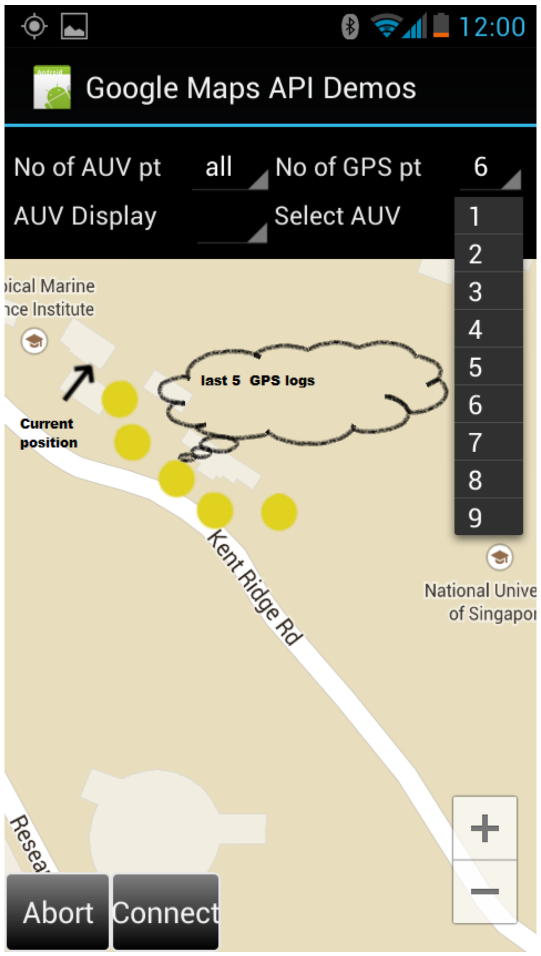

Fig 2:-The snapshot of the working GPS logs in the app.

The arrow seen in Fig.2 rotates as the user moves, i.e. The arrow points in the direction of the user. These feature helps user or chase boat operator to decide in which direction we are with respect to AUV and can move accordingly.

4.2 GUI of the app:-

From the Fig.2 we can see that we have the options to select the no. of GPS points to display on the map. Also we have the option to select the no. of AUV points to display on the map. Also we have the option to see only those AUV’s data which we need. There is an option to send “abort” message if we want to abort the mission of particular AUV. For sending the abort message for a particular AUV id, first one has to select the AUV no. from the drop down menu of “Select AUV” and then by clicking the abort button it will send the message “Abort AUV no. id”. Also if the connection with Bluetooth module is lost there is an option to reconnect with the Bluetooth module.

4.3 Data from BT module:-

Once the app starts it tries to set up the connection with the Bluetooth module. If the Bluetooth of the mobile is turned off the app will prompt the user to turn it on. Once the mobile’s Bluetooth is on there will be a message to pair with the Bluetooth module. No password will be asked, but if the password is asked then enter “1234”.

If the mobile is successfully connected to the module the mobile will transmit the message “Send the data” to BT module and BT module will transmit it to the agent via RF link. Once the java program running on agent side gets this information, all the data of the mission files will be transmitted and the app will receive it.

The BT module is configured such that it is always connected to mobile. Once the any data is updated on the agent side, or any new mission file is created ,the java program will detect it and send the data via RF link to BT module and from BT module to mobile.

The sample snapshot of some dummy AUV data on the map is given in Fig.3.

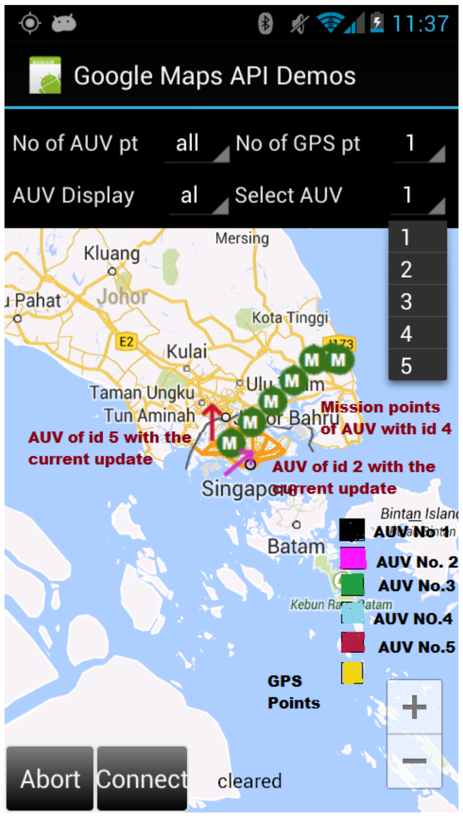

Fig.3. These fig is showing the sample data of AUV no.2, 5 and 4.The details of the type of different data is explained in figure.The color codes used for different AUV no. is given in the above snapshot.

5.0 <u>Steps to Setup the Hardware and Android app</u>:-

- Setting up the RF Transceiver:-
  - First one should install the RF transceiver configure toolbox accompanied with the transceiver.
  - Second one should adjust the jumper of the RF transceiver to USB enable mode, in order to configure the RF transceiver from the pc.
  - Now through the toolbox one should configure the baud rate to be 57600.
  - After it is done change the communication mode from "USB enable" to "40 pin header".
  - Also put the jumper of "power source" to "power conn"
- Install the Arduino programm named "bt_debugger_part3" in the two arduinos. One arduino would be connected to RF transmitter and other to RF receiver.
- Do the following connections.

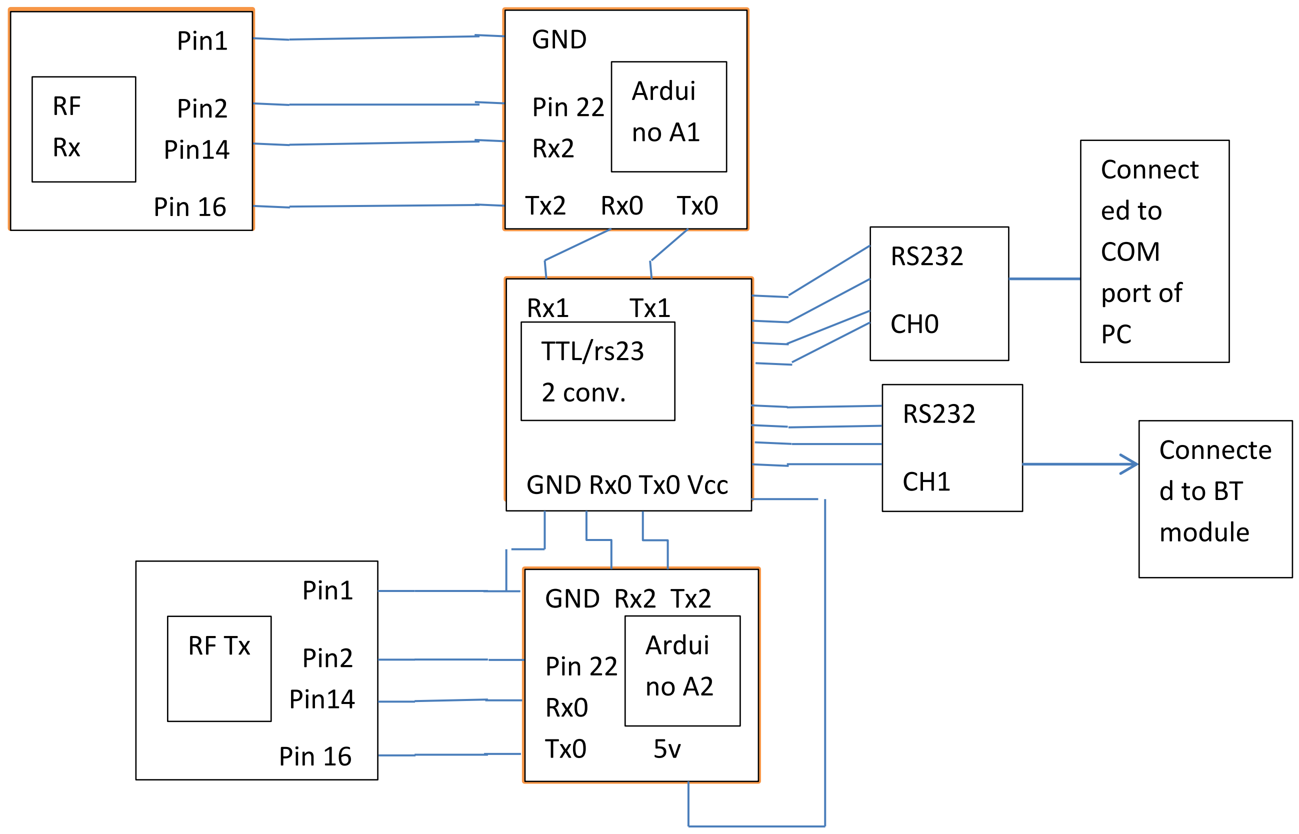

Fig4. The connection diagram of the entire setup

- Configure the BT module as explained in previous sections.
- Install app with “MainActivity.apk” in the android mobile phone.
- Now power up all the arduino’s. This will power up the entire setup
- Now run the java program in the laptop for detecting the change in mission files and AUV_update files.
- Start the app in the mobile and check the results.
- If one wants to change the icons of the android app then following things have to be done.
    - ♦ Replace the following files with the same name(because in the code name is hardcoded) to change the icon for the AUV update points and gps points:-

File	Nature of point
dot.png	AUV No. 1
dot_pink.png	AUV No. 2
dot_green.png	AUV No. 3
dot_blue.png	AUV No. 4
dot_brown.png	AUV No. 5
dot_yellow.png	GPS point

- For replacing the mission points of the AUV replace the following files with the same name.

File	AUV No.
dot_m.png	1
dot_pink_m.png	2
dot_green_m.png	3
dot_blue_m.png	4
dot_brown_m.png	5

5.1 Acquainting with the Android program and Java program:-

The following are the images of the UML of the Java program and android program. One can use them to understand the code for the given program. Also there is a ".ucls" file present in each the folder of android app and java program. One can open them in eclipse and start exploring the code. The file names are "chaseboat_java_prog.ucls" and "android_app.ucls" respectively for java program and android app code

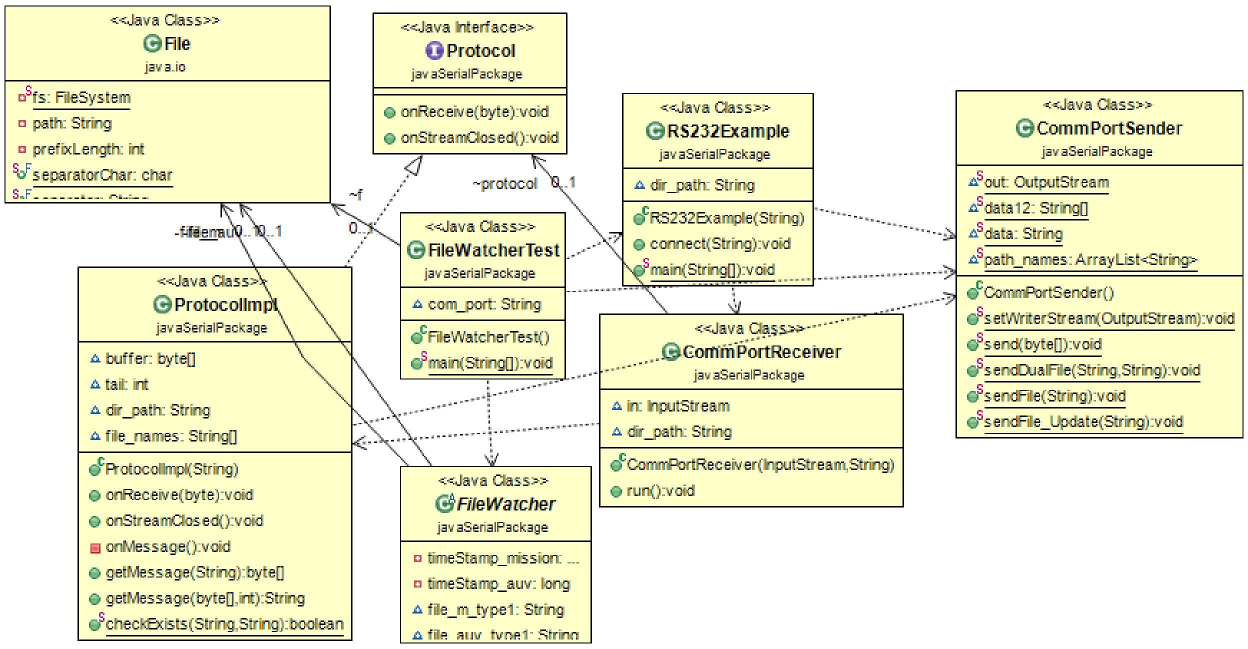

Fig.5 UML of the java program running continuously on agent side to detect any changes in the file

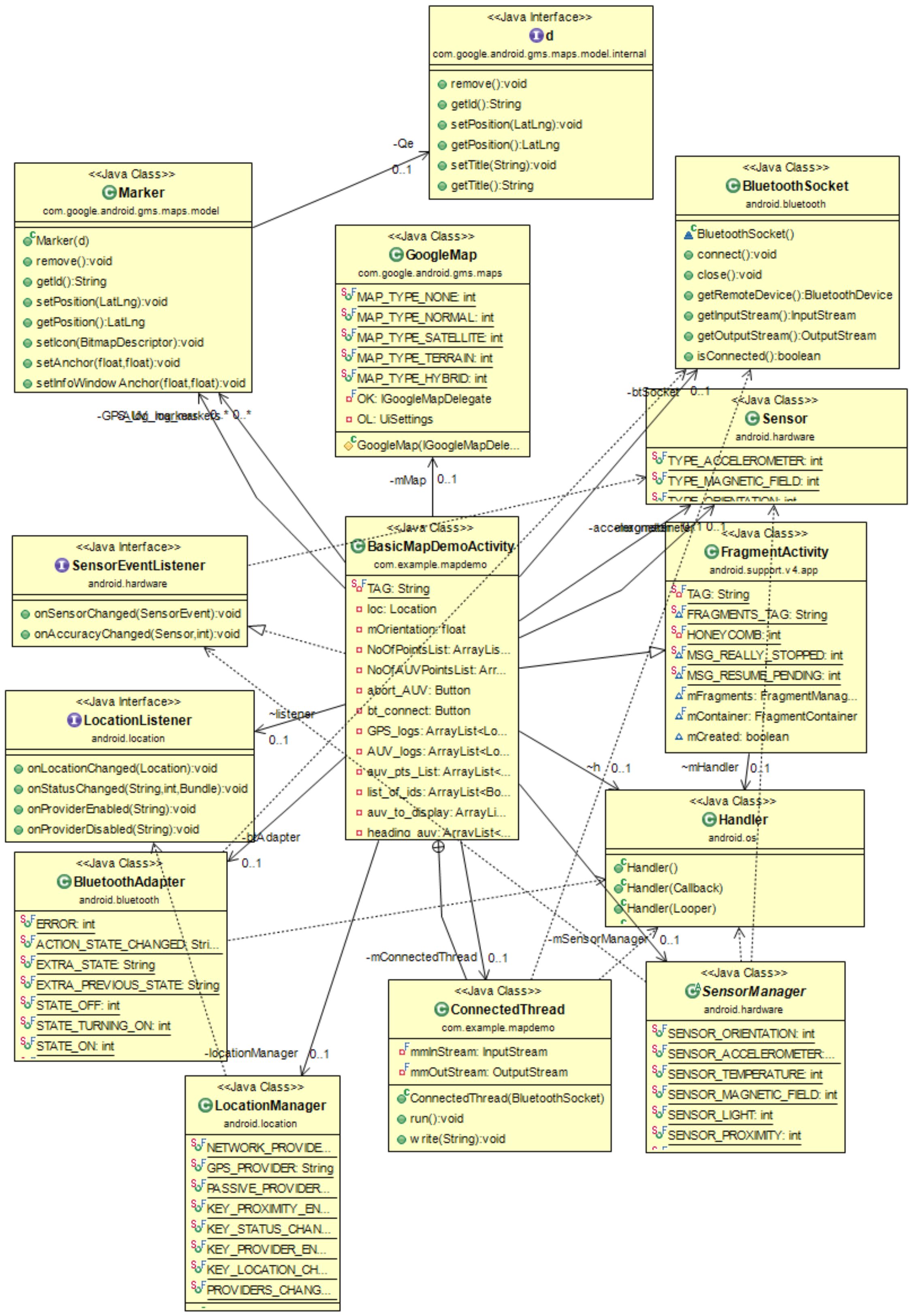

Fig.6 UML of the android app

6.0 Future Improvements:-

- Improve the format of data used in communication.
- Also implement the setup and develop the software for the two way communication. That is even transfer data from chase boat to agent. Now only one can transfer the Abort signal. Future implementation will be say we need to update the mission file sitting in the chase boat via the given hardware.
- Also improvise app to give more feature for the user.
- Improvise the hardware for robust communication. Also remove the redundancies .Instead of using Broadcast mode one can use addressed mode.
- One can add “Sequence number” in the format of the update string. Sequence number is basically the serial number related to the number of the AUV update points.
- Also by adding the sequence number for the mission points in the “lat-lon” stream would act as a delimiter to check for the transmission errors.
- Transmitter at the base station can transmit say last 10 update points whenever the update of the AUV comes on the base station. The advantage for doing this is that in case of loss of connectivity with the RF receiver we would get the last 10 updates whenever it gets connected. The disadvantage is that one has to remove the redundant data at the mobile side in order to get the current update.

7.0 References:-

- http://www.kuligowski.pl/java/rs232-in-java-for-windows,1
- http://www.rgagnon.com/javadetails/java-0490.html
- http://www.connectblue.com/products/wireless-ready-to-embed-modules/bluetooth-dual-mode-serial-port-module-obs421/
- https://developers.google.com/maps/documentation/android/start
- http://developer.android.com/index.html
- http://developer.android.com/guide/topics/location/strategies.html
- http://developer.android.com/guide/topics/ui/controls/spinner.html