



Semester Thesis Task Assignment for Girorgio Tresoldi (D-ITET)

A Flarm Receiver for the OpenSky Network

Main advisor:	Dr. Vincent Lenders (armasuisse)
Advisor ETH	Daniel Moser
Supervisor	Prof. Dr. Laurent Vanbever
Start Date:	March 6th, 2017
End Date:	July 7th, 2017

1 Introduction

Automatic broadcasting of position and velocity beacons has become a standard method in aviation for collision avoidance and to provide situational awareness to air traffic controls and pilots. The two most popular systems are ADS-B and FLARM. While ADS-B is mostly used by larger aircraft such as airliners and business jets, FLARM is optimized for the specific needs of light aircraft such as gliders, light airplanes and helicopters.

The OpenSky Network is a crowdsourcing initiative which aims at improving the security, reliability and efficiency of the air space usage by providing open access of real-world air traffic control data to the public. The OpenSky Network currently collects data from more than 400 ADS-B sensors around the world and provides an open API for researchers to access the data for their research. The network is constantly growing with volunteers adding new sensors and researchers performing studies and evaluations with the data.

As of this writing, the OpenSky Network lacks the possibility to collect and feed FLARM data.

2 Goals

The goal of this thesis is to develop a FLARM sensor and a server for feeding FLARM data to the OpenSky network. The sensor should rely on low-cost COTS hardware to allow volunteers to feed data to the network with little financial investment. The target platform for the sensor is therefore a Raspberry PI with a RTL-SDR dongle that can be tuned to 868 Mhz, the radio frequency used by FLARM. The server should be able to receive the data from many different sensors and to forward the data to the OpenSky backend for further processing and making the data available through the OpenSky API.

3 Thesis tasks

The detailed tasks of this thesis are:

1. Study the FLARM protocol [1] and the RX/TX modulation and coding schemes of the Nordic nRF chipset [2] used by FLARM devices.
2. Familiarize yourself with existing open source Flarm demodulators and decoders on Github. Study the code and experiment with these tools using a simple setup with a FLARM device provided to you by your advisor.
3. Evaluate the Openglidernet.org (OGN) sensor software stack.

4. Setup and install a FLARM sensor based on a Raspberry PI that feeds to OGN (e.g. at armasuisse in Thun or at the place of your choice)
5. Study the architecture and code of the OpenSky Mode S feeder and server
6. Based on the knowledge gained in Tasks 1-5, design a sensor software architecture for feeding FLARM data to the OpenSky Network.
7. Implement a sensor based on a Raspberry PI and a RTL-SDR dongle. The software on the sensor should include a FLARM packet demodulator and an OpenSky feeder. If appropriate, reuse code from existing open source FLARM decoders and the OpenSky Mode S feeder. The data feeder should export undecoded data with meta information such as the time of arrival and sensor identifier in the Avro format to the server.
8. Implement a simple server that receives Avro data from the sensors and forwards the data the Kafka queuing system of the OpenSky Network. If appropriate, reuse code from existing open source FLARM decoders and the OpenSky Mode S feeder.
9. Implement a FLARM decoding library in Java similar to the OpenSky Java ADS-B library [3] that allows you to decode and interpret the data received at the server.
10. Test you sensor with two FLARM devices provided to you by your advisor.
11. Deploy your sensor next to the OGN sensor from Task 4 and compare the performance between the two systems.
12. Optional: Extend your sensor with a 4G modem, and if a needed a solar panel, and deploy the sensor at a place of your choice in the Alps.

4 Deliverables

- At the end of the second week, a detailed time schedule of the thesis must be given and discussed with the advisors.
- At the middle of the thesis, a short discussion of 15 minutes with the supervisor and the advisors will take place. The student has to talk about the major aspects of the ongoing work using slides.
- At the end of the thesis, a presentation of 15 minutes must be given at ETH (in English). The presentations should give an overview as well as the most important details of the work. If possible, a demonstrator should be presented (offline after the talk).
- The final report should be written in English. It must contain a summary, the assignment and the time schedule. Its structure should include an introduction, an analysis of related work, and a complete documentation of all used hardware/software tools. Two written copies of the final report must be delivered to the main advisor along with a DVD that includes developments undergone during the thesis and the used data.

References

- [1] <http://pastebin.com/8ke6jnQZ>
- [2] http://infocenter.nordicsemi.com/pdf/nRF905_PS_v1.5.pdf
- [3] <https://github.com/openskynetwork/java-adsb>

armasuisse
Science and Technology
C4I Networks

Dr. Vincent Lenders
Thun, February 3, 2017