Air Traffic Monitoring System using GNU Radio

Final Project, CSCE-465/865, UNL School of Computing

Bala Subramanyam Duggirala (bduggirala2@huskers.unl.edu)
Samarth Vinod Mutnal (smutnal2@huskers.unl.edu)
Fahmida Afrin (fafrin2@huskers.unl.edu)

Abstract:

In this modern age, air-travel has become an inseparable part of our lives. This has led to an enormous increase in the air-traffic. At this very moment in which we are writing this sentence, there are somewhere between 7782 and 8755 commercial planes in the sky. Given the busy modern skies, it becomes crucially important to monitor and guide every airplane in the sky as part of keeping the air travel safe. The Air-Traffic-Control system which we have in place plays a major role in contributing to this safety, and wireless communication technology is its backbone. So, our project is about building one such system which is capable of monitoring the planes in the sky and getting useful information which would enable the ATC to help keep the skies safe.

Introduction:

Currently, technology that is widely implemented to support the ATC are the radar systems positioned nearby the ATC towers/facilities to get details about the aircraft that are flying in the airspace that they control. Popularity of the radar systems rose during the early 20th century as they played a key role in the outcomes of the wars during that period. The 1st generation radars had limited functionality compared to the modern technology. They were mostly implemented to serve as early-collision-warning systems with the sole purpose of alerting the operators about the proximity of nearby aircraft. Also, the resolution was very poor, and these radars operated by listening to echoes and emitting continuous radio signals simultaneously. They could only track the proximity, but could not get other critical information such as size of the aircraft, its altitude or its speed, type, and identity.

As the technology improved, modern antennas were built using improved materials and electronics, which led to an increase in efficiency and resolutions of these systems. But with these primary radars, there is still the disadvantage of limited information received by the ATC. This is where the modern technology can help in the form of an advanced secondary radar, which is capable of directly interrogating a specific aircraft and procure a wide variety of information. This is a reason why the present-day ATC authorities use a combination of primary and secondary radars. There are many types of such secondary radars such as Primary Surveillance Radars (PSR), Secondary Surveillance Radars (SSR), Mode-S and Surface Movement Radars to monitor traffic in the air and on the ground. In our project we focus on air traffic monitoring using one such system i.e. ADS-B/Mode-S receiver.

Problem definition and objective:

Our project is about building a secondary surveillance system which tracks the aircraft in the sky in real time. By tracking, we don't just mean the location of the aircraft, we aim to get a variety of information about the aircraft including it's Bearing, Range, Heading, Altitude, Climb (rate of ascent), etc. We intend to do so by using GNU Radio and Universal Software Radio Peripheral (USRP) – a software defined radio.

Objectives:

For this project our objectives are to implement this project our objectives are learn about and implement the following

- To learn about the GNU-Radio, and how to use it with a connected USRP-testbed.
- To get an idea of the environment and the dependencies that need to be installed/handled to successfully connect to and use the radio.
- To learn about the OOT modules in general, and about one OOT in specific, i.e., gr-air-modes. This is the main module which helps the project track flights.
- To become familiar with applications of gr-air-modes, like modes_rx and modes_gui.

Related Work:

According to our online exploration we discovered that some of the important work in the area of air traffic management id focused towards transition from old radar technology to a relatively modern Automatic Dependent Surveillance Broadcast(ADS-B) to track aircraft in the sky as well as on the ground. Since most of the aircraft that are currently in operation are not equipped with ADS-B/Mode-S receivers plans are being made to equip the same on these aircrafts [1].

The recent research shows that ADS-B could be a viable low cost replacement for the exist radar and helps the ATC to better track and guide the airplanes with improved precision and coverage of wider area. Also for NextGen(Next generation air transportation system) and SESAR (Single European Sky ATM (Air traffic Management) Research), which refers to the efforts of US federal aviation administration and FAA and its European counterpart, ADS-B is considered as the key technology that is to be used for transformation from currently used radar system to GPS surveillance.

Method:

The method we chose to implement for the surveillance system is by using ADS-B/Mode-S receiver on GNU-Radio. The USRP testbed that is connected to this GNU-Radio would receive and decode the ADS-B/Mode-S information beacons from real-world aircraft.

Automatic Dependent Surveillance (ADS-B) is used by an aircraft to broadcast its position that it gets from its built-in satellite navigation systems. One good advantage of ADS-B is that it is automatic because it does not need input from an external source like pilot or other application and just uses the flight's navigation system. So, the aircraft periodically broadcasts this signal and it can be used to track the airplane in the sky. Out-of-Tree modules (OOT) can be used to receive and process the beacons from the aircraft. OOT module is a component of the GNU Radio that is not present in the source tree of GNU Radio, and comes as an extension. The specific decoder we would be using is gr-air-modes. After installing the gr-air-modes OOT, we would use a graphic user interface application called modes_gui that has Google Maps integration. A list of visible aircraft is popped-up in a window with details like Bearing, Heading, Latitude, Longitude, Azimuth map, live data distributed across various tabs. The data can be outputted into a KML file as well which can be imported into Google Maps to display the location of the aircraft on Google Maps interface.

Our setup requirement for the project to run was to install multiple software including:

- Python 3
- PyZMQ
- NumPy
- SciPy
- SQLite 3.7 or later
- CMake >=2.6
- Gnuradio >=3.5.0
- Ettus UHD >=3.4.0 as a source
- gr-air-modes

Out of the above, the radios that were assigned to us had already come with some pre-installed software. So we additionally had to download and install SQLite, CMake, gr-air-modes. After installing these software, we cloned into the GitHub repository for gn-air-modes, https://github.com/bistromath/gr-air-modes.git. After successfully installing all the required software, and cloning the gr-air-modes repository, we will have access to two of the following gr-air-modes applications:

1. modes_rx
2. modes_gui.

<u>An understanding and background:</u>

<u>OOT (Out-of-Tree) Module:</u>

- It is a module that is by default not present in the GNU-Radio system.
- We can install it if we want to extend the functionality of the GNU-radio as per our requirement.
- This is not added to the source tree, but rather allows us to maintain the code with ourselves instead of requesting permission to modify the source code itself.

- Examples include gr-air-modes, GNU Radio Digital Audio Broadcasting module etc.,

gr-air-modes:

- This is an OOT for the GNU Radio system
- SDR receiver is implemented by gr-air-modes for Mode-S transponder signals. This also includes the transponder signals (ADS-B) given out by the aircraft.

Mode-S transponder:

- This is a transponder protocol used by modern aircraft
- It is used to respond to the radar interrogation signals sent out by either ground radar or other air-traffic using their Traffic Collision Avoidance Systems.
- This is an extended version of the Mode A/C protocol that is being used in transponders since a long time.
- ICAO number, short for International Civil Aviation Organization number, is an aircraft type designator which are used to identify the type of aircraft, and performance, which might affect the air traffic control.
- This ICAO number, along with the altitude of the aircraft is received in the Mode-S report.
- The Mode-S receiver listens to 1090 MHz downlink channel.

Automatic Dependent Surveillance-Broadcast (ADS-B):

- This is a communication protocol that uses the 'Extended Squitter' capability of the Mode-S transport layer.
- There are other implementations available for the ABS-B communication protocol but Mode-S remains its primary transport for the commercial use.
- The features of this protocol are as follows:
    - It is automatic and requires no pilot input
    - It is dependent on aircraft instrumentation
    - With surveillance, it provides the information about the aircraft transmitting
    - It broadcasts the signal to all receiving stations within range
    - The aircraft that are equipped with ADS-B periodically squitter (squitt means the periodic outburst of the aircraft data by the Mode-S transponder even without interrogation by any ground radar or other aircraft) their position, velocity and other details.
- Position reports are generated once every second and the flight identification details are generated every 5 seconds

- The signal is received using modes_rx and modes_gui (if we want a graphical user interface).
- We can use a receiver that is capable of 2Mbps sample rate.

Result:

The following is the default modes_rx output with a sample rate of 10e6:

```
$ modes_rx -r 10e6
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.git-202-g9e0861e1

-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Setting gain to 19
Gain is 19
Rate is 10000000
Using Volk machine: avx_64_mmx_orc
(-63 0.34132346) Type 0 (short A-A surveillance) from ab3639 at 8200ft (Vertical TCAS resolution only)
(-62 1.38402336) Type 0 (short A-A surveillance) from ac9fbe at 33025ft (Vertical TCAS resolution only)
(-62 1.44277776) Type 0 (short A-A surveillance) from ac9fbe at 33025ft (speed 300-600kt)
(-62 1.64927856) Type 11 (all call reply) from ab3639 in reply to interrogator 0 with capability level 6
(-63 1.68898826) Type 11 (all call reply) from ac9fbe in reply to interrogator 0 with capability level 6
(-65 2.36897966) Type 0 (short A-A surveillance) from ac9ff6 at 33025ft (Vertical TCAS resolution only)
(-69 2.91286006) Type 4 (short surveillance altitude reply) from 70b06 at 58300ft (AIRBORNE ALERT)
(-63 3.06877306) Type 17 BDS0,5 (position report) from ab3639 at (36.102610, -115.289183) at 8350ft
(-64 3.21826886) Type 0 (short A-A surveillance) from a46d29 at 11800ft (Vertical TCAS resolution only)
(-64 3.22831576) Type 0 (short A-A surveillance) from ab3639 at 8375ft (Vertical TCAS resolution only)
```

Modes Graphic User Interface:

After the GUI has started, you will need to set the following settings:

- Select UHD as Source
- Set Sample Rate
- Set Threshold
- Set Gain
- Optional

- Set Latitude
- Set Longitude
- Check KML and Set Output Filename

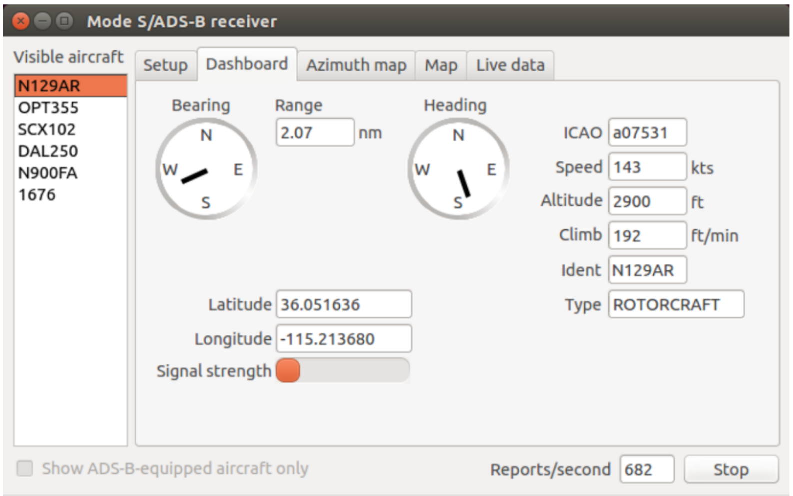
After adjusting the above settings, click "Start" to begin capturing ADS-B/Mode-S packets. You will see the Reports/second field fluctuate as packets are decoded; as aircraft are identified, their ID will populate the Visible Aircraft list.

After the GUI has started, you will need to set the following settings:

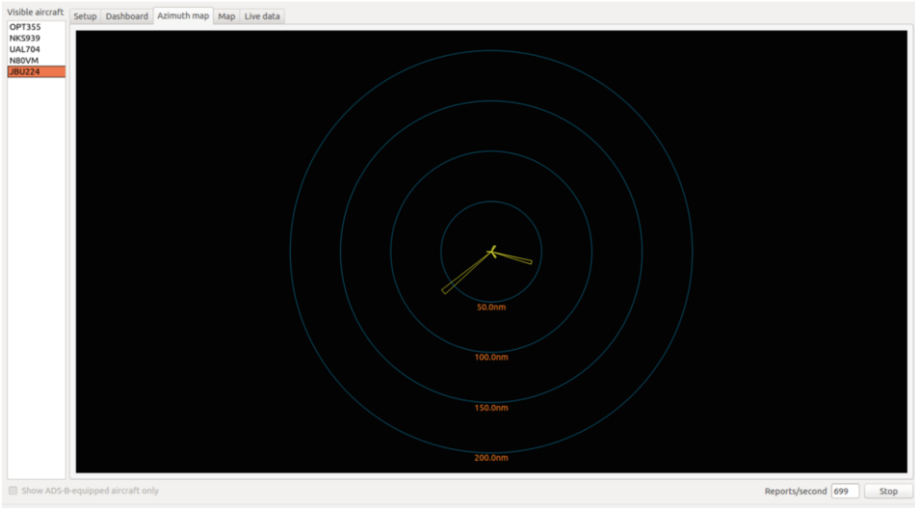
- Select UHD as Source
- Set Sample Rate
- Set Threshold
- Set Gain
- Optional:
  - Set Latitude
  - Set Longitude
  - Check KML and Set Output Filename

After adjusting the above settings, click "Start" to begin capturing ADS-B/Mode-S packets. You will see the Reports/second field fluctuate as packets are decoded; as aircraft are identified, their ID will populate the Visible Aircraft list.

Once aircraft have populated the Visible Aircraft list, if you highlight an ID, and click on the Dashboard tab, it will display the details of the aircraft including it's Bearing, Range, Heading, Speed, Altitude, Climb Rate, Latitude, Longitude and Signal Strength. Not all aircraft will broadcast every field of data.

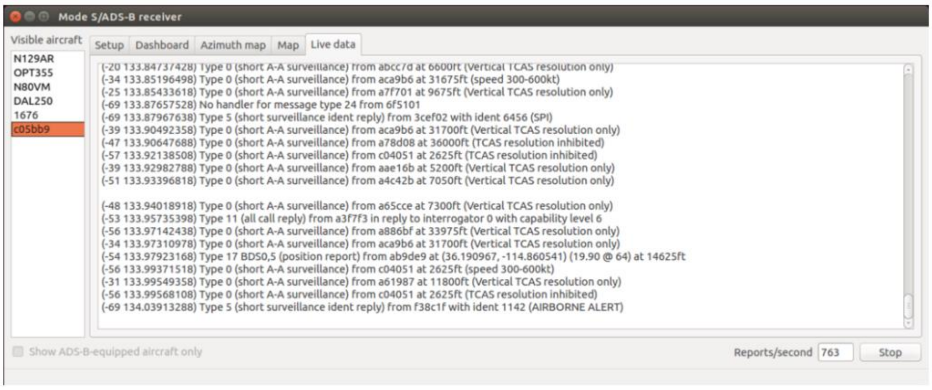


The Azimuth Map tab will display a plot of overall distance of all received packets that have included Latitude and Longitude data.



The Map tab will overlay the aircraft's broadcasted location in real time.

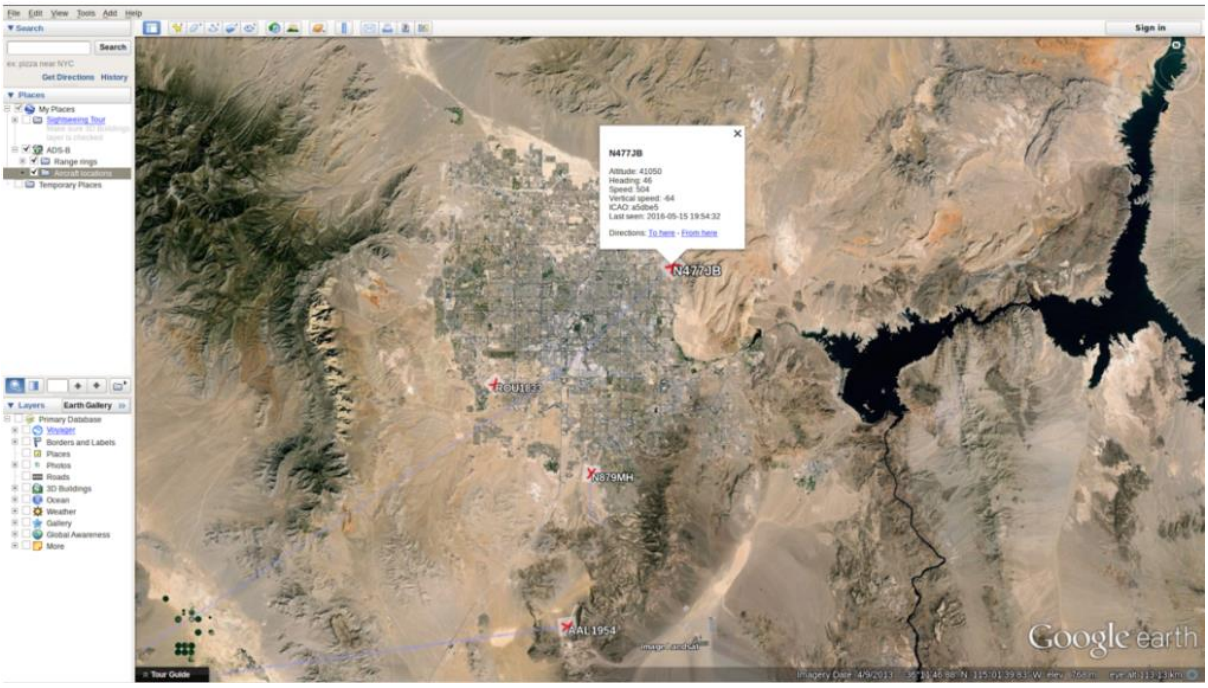
The Live Data tab will display a scrolling text field of all data packets decoded.



Plotting KLM Data:

- By enabling the KML setting within modes_gui, or with the --KML flag in modes_rx, it will output all aircraft data to a Google Earth compatible KML file. This KML file can be imported into Google Earth and viewed after the capture is complete.
- To import the KML data file and auto refresh follow the steps below:
    - Start Google Earth
    - Select "Add" -> "Network Link" from the menu bar
    - Enter a description into the "Name" field such as "ADS-B"
    - Click "Browse" and navigate to the location of the KML output file and select it
    - Select the "Refresh" tab
    - Under the "Time-Based Refresh" select "Periodically" and set a time frame you would like the data to refresh. (e.g. 4 seconds)

- Click "OK"



Conclusion and future work:

In summary, ADS-B is on track to transform the aviation industry's ATC system by providing a more dependable and accurate tracking system, at lower cost. This technology would be especially useful to convert the existing radar usage to a GPS system. This would overcome the shortcomings presented by the current radar systems and helps improve the safety of the air-travel.

Since the current research is still at phase of using a combination of primary and secondary radars, we think one good possible future work could be to unify these technologies and use a single tracking system that would serve as an all-in-one.

References:

- Main source: Implementation of an ADS-B/Mode-S Receiver in GNU Radio - Ettus Knowledge Base
- Research Paper by Boeing: AERO - New Air Traffic Surveillance Technology (boeing.com)
- Concepts: https://manpages.debian.org/testing/gr-air-modes/modes_rx.1.en.html