# Aircraft Collision Avoidance Systems:

# A Universal Receiver


Matthew Aleksis Snowdon

BEng Electrical and Electronic Engineering


2017


Aston University

ASTON UNIVERSITY

AIRCRAFT COLLISION AVOIDANCE SYSTEMS: A UNIVERSAL RECEIVER

MATTHEW ALEKSIS SNOWDON

BENG ELECTRICAL AND ELECTRONIC ENGINEERING

2017

# Summary

Across commercial and general aviation there are multiple systems in use that aim to reduce the risk of mid-air collisions. Each of these systems uses unique transmission methods and encoding protocols, which renders the information transmitted (such as aircraft position) inaccessible to users of other systems. Research was undertaken to understand the technical characteristics of each of these systems, where they are commonly used and the advantages and disadvantages between each of them for the end user. The ideal device would be able to both receive and transmit information across all systems.

This project has contributed to the technical knowledge required to develop such a device by focusing on two of these systems. These were Automatic Dependant Surveillance Broadcast (ADS-B), used mostly by large commercial aircraft, and the Open Glider Network Transport Protocol (OGNTP) which is used in general aviation. The GNU Radio software package was used to develop custom modules for receiving and decoding Open Glider Network (OGN) transmissions. In addition to this an existing project was utilised to receive ADS-B transmissions. The hardware used by both receivers are cheaply available Register Transfer Level Software Defined Radio (RTL-SDR) dongles, which serve as general purpose radio receivers.

Testing on the project has shown both receivers operate successfully, albeit with a limited range due to the antennas used.  The OGN receiver successfully decoded transmissions up to 100m away, whilst the ADS-B receiver successfully decoded transmissions up to 30km away. Conclusions show that it is possible to simultaneously receive transmissions from these two systems using open-source software and low cost hardware. This project serves as an invaluable reference for developers looking to add OGN reception functionality to their systems, which contributes to the overall aim of increasing cross-system compatibility. This will in turn help to reduce the risk of mid-air collisions.

*To Dad.*
*For beginning my interest in aviation.*

# Acknowledgments

I would like to primarily thank Dr Marc Eberhard (Aston University) for his advice, knowledge and guidance throughout the course of this project.

Secondly, I would like to thank the OGN community for their help, particularly Paweł Jałocha and Nick Bonniere.

Finally, thanks to Zahra Meghji for her unwavering support.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| ADS-B | Automatic Dependent Surveillance Broadcast |
| AOPA | Aircraft Owners and Pilots Association |
| API | Application Programming Interface |
| APRS | Automatic Packet Reporting System |
| ASK | Amplitude Shift Keying |
| ATC | Air Traffic Control |
| CAA | Civil Aviation Authority |
| DVB-T TV | Digital Video Broadcast - Terrestrial Television |
| EASA | European Aviation Safety Agency |
| FEC | Forward Error Correcting |
| FLARM | Portmanteau of “Flight” and “Alarm" |
| GA | General Aviation |
| GFSK | Gaussian Frequency Shift Keying |
| GNU | A recursive acronym meaning "GNU's not Unix" |
| GPS | Global Positioning System |
| GRC | GNU Radio Companion |
| GUI | Graphical User Interface |
| ICAO | International Civil Aviation Organisation |
| IFF | Identification, Friend or Foe |
| IMU | Internal Measurement Unit |
| JSON | Java Script Object Notation |
| LCD | Liquid Crystal Display |
| LDPC | Low Density Parity Check |
| LED | Light Emitting Diode |
| LPAT | Low Power ADS-B Transceiver |
| LSB | Least Significant Bit |
| NATS | National Air Traffic Services |
| NextGen | Next Generation Air Transportation System |
| NM | Nautical Mile |
| NMEA | National Marine Electronics Association |

| | |
|---|---|
| OGN | Open Glider Network |
| OGNTP | Open Glider Network Transport Protocol |
| PAW | PilotAware |
| P3I | PilotAware Protocol |
| PPL | Private Pilot's License |
| PPM | Pulse Position Modulation |
| RF | Radio Frequency |
| RPi | Raspberry Pi |
| RTL-SDR | Register Transfer Level - Software Defined Radio |
| SDR | Software Defined Radio |
| SRD | Short Range Device |
| SSR | Secondary Surveillance Radar |
| TCAS | Traffic Collision and Avoidance System |
| TEA | Tiny Encryption Algorithm |
| TTL | Transistor to Transistor Logic |
| UART | Universal Asynchronous Receiver / Transmitter |
| UAV | Unmanned Aerial Vehicle |

# Introduction

As airspaces around the world become busier and more complex, maintaining a safe separation distance between aircraft becomes a considerable challenge [1]. Airspace is usually divided into three-dimensional segments, which are allocated to a class depending on what kind of traffic passes through the area [2]. These classes can either be controlled or uncontrolled. In controlled airspace, air traffic controllers will direct every aircraft. They will consider parameters such as weather conditions and proximity to other traffic to decide the safest and most efficient route for each aircraft. Because they can view the location of all aircraft within their sector, they can effectively ensure that minimum vertical and horizontal separation distances are maintained. Areas around major airports and areas that see a large amount of traffic are common examples of controlled airspace [3].

However, uncontrolled airspace can also be found in many areas. In fact, it makes up around half of UK airspace. Uncontrolled airspace has no restrictions on which aircraft can enter it, what equipment they must carry or which routes they can take [4]. Whilst there are sometimes air traffic control (ATC) services available, it is up to the pilot's discretion whether to utilise them. When the topic of aircraft collisions arises, accidents involving large commercial aircraft are likely to spring to mind. However, most collisions happen between smaller aircraft that fall under the general aviation (GA) category. Since no one has a complete picture of all the aircraft in the area, accidents involving a range of small aircraft and gliders unfortunately continue to occur.

To improve pilot awareness of other traffic, there have been several systems developed that are in widespread use today. This project will analyse in detail the technical and commercial aspects behind each of these systems, and will also address a major issue that affects all pilots. That is, these systems are largely incompatible with each other. This means that despite investment to reduce the risk of mid-air collisions, there is still a chance that two aircraft could be transmitting their position but still be invisible to one another. By demonstrating that it is possible to create a system that can simultaneously receive transmissions from two of these systems, this project will show that this issue can be effectively mitigated without requiring large hardware investment or a complicated software setup.

This project was proposed after consulting with Dr Marc Eberhard of Aston University. Dr Eberhard holds a Private Pilot's License (PPL) and has been able to offer considerable insight into this subject area. The work was undertaken by final year BEng undergraduate student Matthew

Snowdon. After the initial proposal in September 2016, research was undertaken and assembly of the project started in January 2017. After verifying component functionality, software development was undertaken and completed by April 2017. The remaining month was used for testing and refinement purposes.

This report will explain how information into the different subject areas was collated and applied to produce the required solution. It will also give context as to how the technical aspects of this project have relevance to real world scenarios. This report covers the background and current day methods used in collision avoidance systems, to ensure the reader understands how the technology has developed into the systems used today. The report will then go on to describe the procedure undertaken to produce different aspects of this project, as well as presenting quantitative results. Finally, the report will conclude with an analysis of the technical achievements of this project and explain how they are relevant to the future of air safety. There will also be a discussion on potential avenues for future development within this project.

The primary objective of this project is to show that it is possible to receive and decode information from two different collision avoidance systems on a single device. The systems chosen are Automatic Dependent Surveillance Broadcast (ADS-B) and the Open Glider Network Transport Protocol (OGNTP). Due to the nature of an aircraft cockpit, the device will be of a small form factor, portable and independent of any aircraft systems. To facilitate a consistent development environment, it will be necessary to assemble a transmitter (identical to the ones used in real light aircraft and gliders) to verify the receiver's performance. The project requirements as discussed with Dr Eberhard are as follows:

1. The system needs to run on a portable computer.
2. The system needs to be powered by battery for at least 3 hours.
3. The system needs a way to allow the user to interface with the computer.
4. I will need a signal source for the OGNTP protocol.
5. The system needs to be able to receive and decode ADS-B.
6. The system needs to be able to receive and decode OGN.
7. The system needs to display aircraft and traffic information to the user.

Some modifications to the originally envisaged solution became necessary as the project progressed, details of which are discussed in the Results section of this report. The main priority of the project is to create a program that can simultaneously receive and decode transmissions from the two different collision avoidance systems. The priority is not creating a full-featured user interface to display the decoded information. Once decoded information on nearby aircraft is in a format that can be processed by other applications, it would be an exercise in software engineering to create an interface that meets the requirements of different users (such as pilots, air traffic controllers or data loggers). As such, a demonstration that the system can decode the received packets successfully and a simple graphical plot of the results will be sufficient. Once the main aims of the project are realised, expansion objectives may be undertaken. Figure 1 is a block diagram showing the different components of the proposed receiver and how they interface with each other. Figure 2 is a block diagram showing the OGN Tracker components.

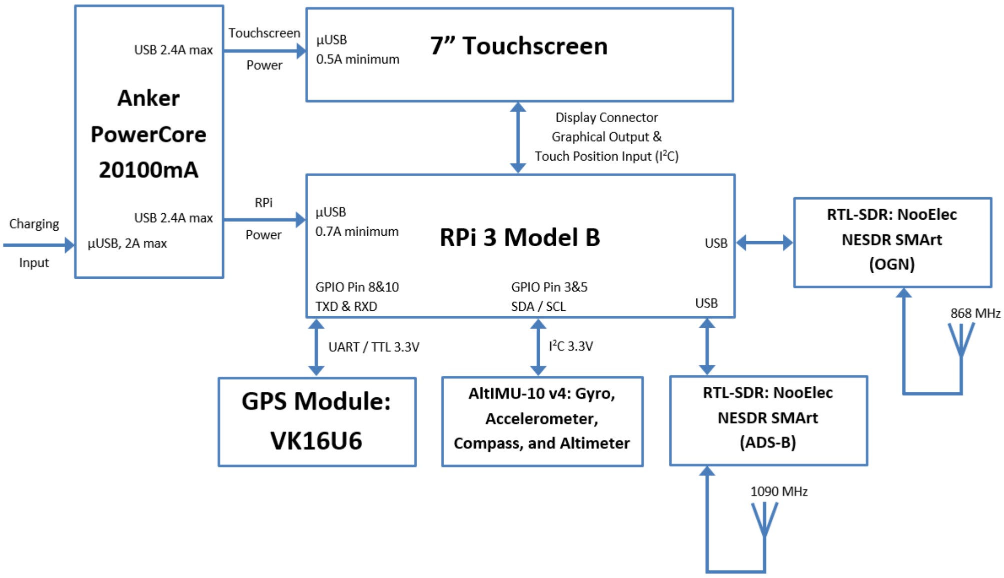



**Figure 1 - Block diagram of the proposed receiver**

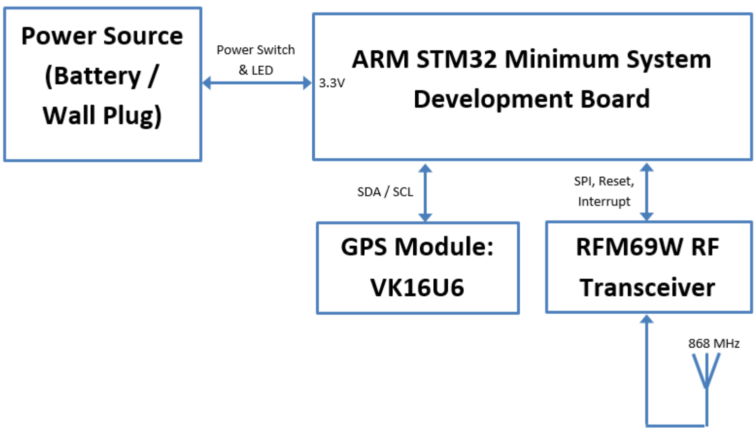



**Figure 2 - Block diagram of an OGN tracker**

Testing of the receivers will be realised by comparing their results against reference sources. For the OGN receiver, this will be the official receiver software produced by the OGN community and the known OGN Tracker position. For ADS-B, this will be public websites that utilise a network of ADS-B receivers to produce a database of aircraft movements (e.g. Flightradar24.com).

# 1 Background

In today's world, the requirement for advanced air traffic management has become essential due to the high density of aircraft operating in certain areas. For example, London Heathrow sees on average 1,300 departures and arrivals daily. This equates to an aircraft departing or arriving every 45 seconds. An array of different technologies, as well as careful planning, ensure the safety of the 75 million passengers that pass through each year [5].

Air traffic controllers are required to keep aircraft separated by minimum vertical and horizontal distances, which are situationally dependent. In most cases, they are defined by the International Civil Aviation Authority (ICAO) as 1000 feet vertically and 15NM (nautical miles) horizontally [4]. This section of the report will look in detail at the history of air traffic management technology, and explain how the different systems in use today ensure aircraft meet these requirements.

## 1.1 What is a Traffic Collision Avoidance System?

An important distinction needs to be made between the two different approaches used for managing air traffic. The most widely recognised concept is probably manned ATC. A system is used to detect the position of all aircraft within a certain area. This information is then displayed to an air traffic controller on a screen, as can be seen in Figure 3. This allows the controller to have situational awareness over all traffic in the area, which in turn enables them to issue routing instructions to each aircraft via radio. This approach is effective in areas that see a high amount of regular traffic, such as airports or international airways. Radar systems are often used, which means aircraft will be detected regardless of how they are equipped or what their motivations are. However, ground based ATC facilities require a significant amount of investment in equipment and personnel, which makes them unsuitable for areas with low amounts of traffic.


**Figure 3 - Example of an ATC screen showing information on aircraft in the sector [60]**

For low traffic areas, a traffic collision avoidance system (TCAS) can be utilised to reduce the incidence of mid-air collisions between aircraft. This system will monitor the airspace surrounding the aircraft for any other aircraft transmitting their position in a compatible format [6]. It is independent, works automatically and can directly warn the pilot of any collision risk. An example of what the pilot might see can be seen in Figure 4. Advantages of this approach include lower equipment cost, as little ground infrastructure is required, and consistent performance as there is no human factor involved. The disadvantage is that for the system to work effectively, all aircraft need to be using equipment that operates to the same standard. If this is not the case, then nearby aircraft could potentially be invisible to each other.

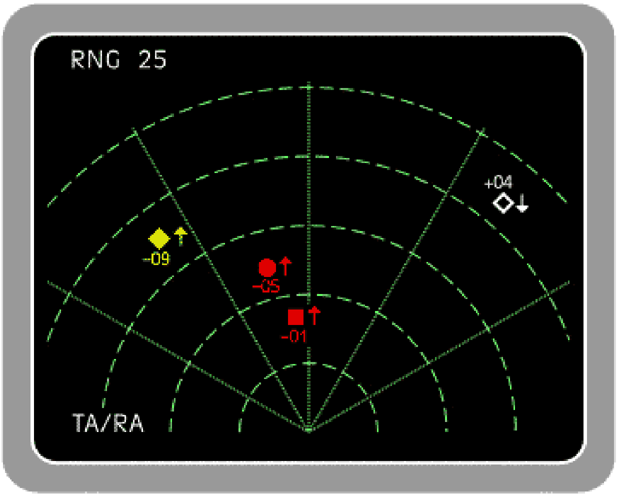


**Figure 4 - Example of a TCAS display. Symbols denoting nearby aircraft are displayed, with their appearance conveying the risk of collision [61]**

In practice, both approaches are used to manage large aircraft, as the ICAO mandates that all aircraft over a certain weight must be TCAS equipped [4]. However, there are no similar requirements for general aviation. As the price and availability of suitable electronic components has improved, the TCAS approach has become viable for the GA community. Previously the only preventative measure in areas that lack ATC coverage was manual observation by the pilot. The issue with this method is described by the following journal excerpt:

*"Even if a converging aircraft is unobstructed, it appears small, motionless, camouflaged, and inconspicuous until imminent impact. A statistical model reveals that the probabilities of seeing and avoiding a converging 40-ft aircraft, for an optimal observer or theoretical pilot scanning 2/3 or 1/3 of the time, respectively, are less than 0.91, 0.60, 0.30 at 200 kn; 0.49, 0.32, 0.16 at 300 kn; 0.28, 0.18, 0.09 at 400 kn; and 0.15, 0.10, 0.05 at 500 kn. The see-and-avoid concept has striking physical and behavioral limitations."* [7]

Clearly a technological based solution could offer much more consistent performance. After each serious accident, there are renewed calls for the GA community to adopt a universal solution. But as with any community, encouraging all members to make a significant investment is a challenge. To understand how the technology in widespread use today has developed, it is essential to understand its predecessors. The next section of this report will look at this in detail.

## 1.2 History

### 1.2.1 Primary Radar

**Figure 5 - A primary radar antenna [62]**

The first form of ATC began with the advent of the practical radar system in 1935 by the British physicist, Sir Robert Watson-Watt [8]. By obtaining precise information of the position of aircraft in real time, the minimum horizontal and vertical separation could be safely reduced. This allowed more efficient aircraft routing to become possible. We now know this system as Primary Radar. Based on the principle of echolocation, Primary Radar can detect anything that reflects the radio signals transmitted from its high power, rotating antenna (an example of which can be seen in Figure 5). This includes aircraft, birds, weather and land features. This enables a controller to detect any suitably reflective object within range, whether they wish to cooperate with ATC or not. However, a significant drawback is the inability to identify these objects. This was originally mitigated by asking the pilot to do a specific turn, which allowed the controller to differentiate their target from the other targets on their scope. Nowadays Primary Radar is mainly used to complement more advanced systems. Despite lacking features that other systems can offer, Primary Radar is still the most resilient method for detecting aircraft. In the current security climate, it is unlikely that Primary Radar will be phased out in favour of systems that could be susceptible to jamming or spoofing.

### 1.2.2 Secondary Surveillance Radar

During the Second World War, rising cases of friendly fire led to another ATC development; the Identification, Friend or Foe (IFF) system [9]. Based on an interrogation and response methodology, it enabled the identification of both military and civilian aircraft. An aircraft equipped with IFF could detect if another suitably equipped aircraft was allied, as well as determine its bearing and range. IFF relied on a radio device fitted to the aircraft called a transponder. This consisted of a receiver operating at a frequency of 1030MHz, and a transmitter operating at 1090MHz. Ground stations, usually located next to a primary radar site, sent out interrogation “pings” which caused the transponder to reply with a coded message. The system was adopted for civil use by combining its identification capability with primary radar, upon which the system became known as Secondary Surveillance Radar (SSR). SSR supports several modes of interrogation. Table 1 describes the modes allocated for civilian use.

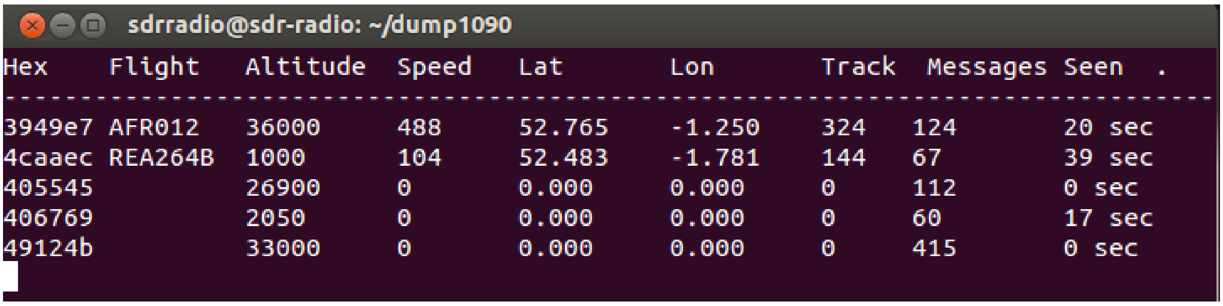
**Table 1 - Summary of civilian SSR interrogation modes [10]**

| SSR Mode | Description |
|---|---|
| A | Provides a 4-digit octal identification code for the aircraft, assigned by air traffic control but entered by the pilot into the transponder module. |
| C | Provides the aircraft's pressure altitude. Often combined with Mode A to provide an identification code and altitude (known as Mode A and C). |
| S | Provides multiple aircraft parameters. Each aircraft is assigned a fixed 24-bit address. Provides data for the ADS-B system. |

### 1.2.3 Automatic Dependant Surveillance - Broadcast

Building upon the additional capability added to transponders through SSR (i.e. the ability to identify aircraft), the next generation of transponder includes additional functionality called Mode S (Select). Mode S transponders support a system called ADS-B. ADS-B utilises a Global Positioning System (GPS) receiver to incorporate highly accurate positional information in each transmission. This enables controllers to reduce the minimum separation distance between aircraft, since the positional information is more accurate than conventional systems could provide [11]. Mode S transponders still maintain compatibility with both SSR and TCAS.

Figure 6 shows some of the information contained within an ADS-B transmission. ADS-B is now a vital technology in busy airspace, but it can also provide a significant advantage in areas that lack ATC services. ADS-B-capable transponders can interrogate other aircraft and display traffic information directly to the pilots. This allows them to have similar information that an air traffic controller would have access to, which increases situational awareness. TCAS technology uses both SSR and ADS-B as data sources.


sdrradio@sdr-radio: ~/dump1090

| Hex | Flight | Altitude | Speed | Lat | Lon | Track | Messages | Seen . |
|---|---|---|---|---|---|---|---|---|
| 3949e7 | AFR012 | 36000 | 488 | 52.765 | -1.250 | 324 | 124 | 20 sec |
| 4caaec | REA264B | 1000 | 104 | 52.483 | -1.781 | 144 | 67 | 39 sec |
| 405545 | | 26900 | 0 | 0.000 | 0.000 | 0 | 112 | 0 sec |
| 406769 | | 2050 | 0 | 0.000 | 0.000 | 0 | 60 | 17 sec |
| 49124b | | 33000 | 0 | 0.000 | 0.000 | 0 | 415 | 0 sec |

**Figure 6 - Decoded ADS-B packets, as seen in the decoding program dump1090 [63]**

ADS–B is an element of the US Next Generation Air Transportation System (NextGen). This intends to transform North America's ATC system from a radar-based system (with radio communication) to a satellite-based one [12]. In both the US and Europe there are upcoming mandates which will require most commercial aircraft to be fitted with ADS-B capable transponders. In the Eurocontrol region, which covers most European airspace, this comes into force in December 2017 for aircraft with a maximum takeoff weight greater than 5,700 kilograms, or a maximum cruising airspeed greater than 250 knots [13]. This will affect almost all commercial aircraft fleets.

Whilst ADS-B transponders are commonplace on commercial aircraft, there is a major factor preventing their widespread adoption in GA: cost. Since the required equipment can cost a significant fraction of a GA aircrafts entire value, many owners are understandably hesitant to make the investment. This has led to a unique situation where several organisations are trying to fill this market gap for low-cost collision avoidance technology. The next section of this report will look at each of these in turn.

## 1.3 Comparison of Low-Cost Collision Avoidance Technology

#### 1.3.1 FLARM

FLARM (a portmanteau of "flight" and "alarm") is a European Aviation Safety Agency (EASA) approved system that alerts pilots to potential collisions between aircraft. Designed specifically for the needs of general aviation, rather than long-range communication or ATC interaction, it's the second most popular collision avoidance system (after ADS-B) with approximately 30,000 units in operation worldwide. A FLARM unit (Figure 7) will transmit position information and a predicted future path [14]. Additionally, each unit will process received data from other nearby units and alert the pilot to any hazardous conditions via visual or audio cues. Some unique features of FLARM include motion prediction algorithms that can predict potential conflicts for up to 50 other aircraft, as well as an integrated obstacle collision warning system which references an obstacle database [14]. Newer FLARM devices can also receive ADS-B and SSR Mode A/C data using integrated components, which is displayed to the user and incorporated into their collision prediction algorithm.


**Figure 7 - PowerFLARM Core**

Each FLARM system utilises a GPS receiver to determine its position and altitude. Based on parameters such as speed, acceleration, track, turn radius, wind, altitude, vertical speed and aircraft type it can calculate a projected flight path. This flight path, along with additional information such as a unique identification number, is encrypted and transmitted over the Non-

Specific Short Range Device (SRD) frequency band using proprietary hardware. In Europe, the 863 to 870 MHz band has been allocated for license-free operation subject to some restraints (e.g. transmitter power level, duty cycle) [15]. Because of the ease of operating and licensing products that utilise the SRD frequency band, it has become a common factor across every low-cost collision avoidance platform.

Figure 8 shows some of the available methods for displaying information from the FLARM unit to the pilot. The display can be as straightforward as simple light emitting diodes (LEDs) that indicate the direction and distance to a nearby aircraft. Alternatively, a pilot might opt for a colour liquid crystal display (LCD) that displays a map showing aircraft in the vicinity. A popular option is integrating the data from the FLARM system into an existing display, such as a GPS unit or a tablet-based flight planning app. Although these displays can often become cluttered due to the amount of information shown, it is still a popular approach as it negates the need for additional equipment and space on the instrument panel.

**Figure 8 - A selection of available FLARM units and displays [14]**

However, a major downside with FLARM is that the company behind it, FLARM Technology, has opted to encrypt their communications protocol. This effectively makes FLARM units invisible to any other collision avoidance system, and vice-versa. FLARM Technology argues that they have a right to protect their commercial property, and despite several discussions taking place in the last few years it seems unlikely that they will change their stance. Some members of the aviation community argue that by using a closed-source and encrypted communications protocol, FLARM Technology is placing commercial interests above aviation safety. Although FLARM is currently the market leader in collision avoidance systems, there are a comparable number of users on competing platforms. Some have argued that FLARM Technology is using encryption to protect their monopoly on the market. Regardless of whether this is true, the fact remains that a fragmented user base across non-interoperable systems provides no benefit to GA safety.

### 1.3.2 PilotAware

PilotAware is an alternative collision avoidance system, designed by the UK company PilotAware Limited. Based around a Raspberry Pi (RPi) 2B [16], it's designed as a small, stand-alone unit that can be easily added to any GA aircraft. An example of a PilotAware Classic unit is shown in Figure 9. Unlike FLARM, it does not provide a display but instead relies on integration with a phone or tablet app. Widely used navigation apps, such as Sky Demon and Easy VFR, support the display of traffic information as an overlay on a moving map [17]. This has the advantage that the pilot can refer to one display for navigation, weather, traffic and other information. However, it also has the disadvantage that the pilot must rely on a battery powered device, rather than an instrument panel mounted display. Unlike FLARM, PilotAware requires an annual subscription for use of its products.


**Figure 9 - PilotAware Classic [64]**

The technical approach behind PilotAware is similar to that of FLARM. It utilises a GPS receiver to calculate the aircrafts position and track, before transmitting this information using their own protocol (P3I). They use a proprietary radio frequency (RF) transceiver that other PilotAware equipped aircraft can receive [18]. Again, this protocol utilises the SRD frequency band. Unlike FLARM, PilotAware does not include any collision avoidance algorithms to provide warnings to the pilot. It simply conveys information on nearby aircraft and leaves the pilot to decide what action to take. Information is conveyed to the pilot via a connected app or audible cues. Like FLARM, PilotAware can also incorporate ADS-B and SSR Mode A/C information. The hardware used for this is an inexpensive register transfer level software defined radio (RTL-SDR) dongle.

Where PilotAware begins to offer an advantage over FLARM is the number of protocols it supports, which are summarised in Figure 10. As well as receiving P3I, ADS-B and SSR Mode A/C it also has the capability of receiving FLARM transmissions [17]. However, adding this optional functionality comes at a significant cost. Since the FLARM protocol is closed source and encrypted, to receive FLARM transmissions it is necessary to purchase a FLARM Mouse. This device has the

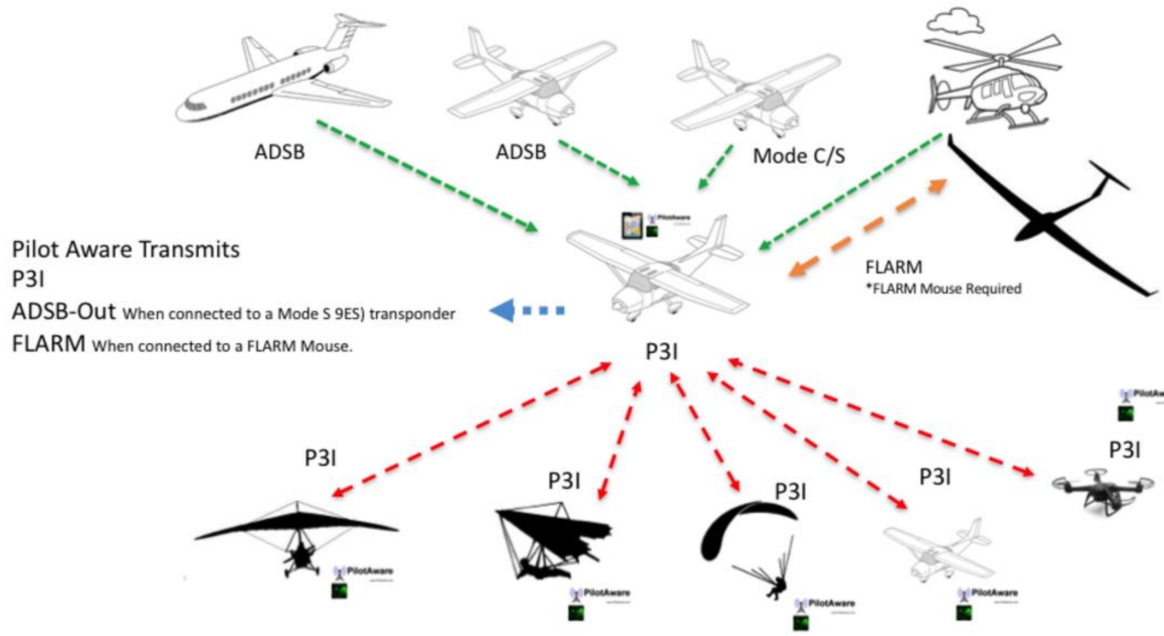



**Figure 10 - Summary of protocols supported by PilotAware [17]**

same functionality and form factor as a normal FLARM unit, but it offers no user display and can only be controlled via serial port. The FLARM Mouse is effectively a bridge that allows other systems to interface with the FLARM network, without accessing any of its proprietary technology. The significant obstacle to adding this functionality to PilotAware is cost. A FLARM Mouse retails in the UK at £732.00 [14], which is several times the cost of the PilotAware unit itself. At this price range, it seems unlikely that there is much of a market for adding FLARM capability to the PilotAware system.

PilotAware also has improved transmission capabilities over FLARM. As well as supporting P3I natively and FLARM with the addition of a FLARM Mouse, it can also transmit ADS-B when combined with a suitable transponder. The transponder must support Mode S and have a compatible input port to interface with the PilotAware unit. This is a significant advantage over FLARM, which has no capability to transmit via ADS-B.

Although the software behind PilotAware is closed-source, it runs on an open hardware design and its communications protocol P3I is open-source. This should theoretically allow anyone to utilise the P3I protocol on their system. However, the documentation required to do this is not easily accessible, although this has improved slightly over the course of this project. PilotAware also sell a product called the PilotAware Bridge, retailing for £84.99, which is their equivalent to the FLARM Mouse. A computer could be interfaced with this module to provide reception and transmission of the P3I protocol, without knowing the intricacies of the modulation method or packet format. PilotAware is a commercial enterprise, and it is possible that the obligations that come with that may have an impact on the purported open-source nature of their platform.

### 1.3.3 Open Glider Network

The objective of the OGN is to create and maintain a unified tracking platform for gliders and other GA aircraft [19]. Figure 11 is an overview of the network's system architecture. Currently OGN focuses on tracking aircraft equipped with FLARM, or their own OGN Tracker. OGN is an open source community project that consists of [19]:

- A pair of Automatic Packet Reporting System (APRS) Linux-based servers that receive and forward data, which includes device location information, status of receivers and the status of the OGN APRS network.
- A database of aircraft.
- OGN Trackers fitted to gliders and light aircraft. Based on cheap, readily available hardware they transmit regular position reports via the open-source OGNTP.
- OGN ground receivers, based at airfields, gliding clubs, summits of mountains or private houses of its members. These listen for and decode the location data of aircraft in their vicinity, before sending position reports via the internet to the APRS servers. As of March 2017, there are around 500 active receiver stations across Europe, which have observed over 1000 unique aircraft [20].
- Websites and applications that can use and display the data.

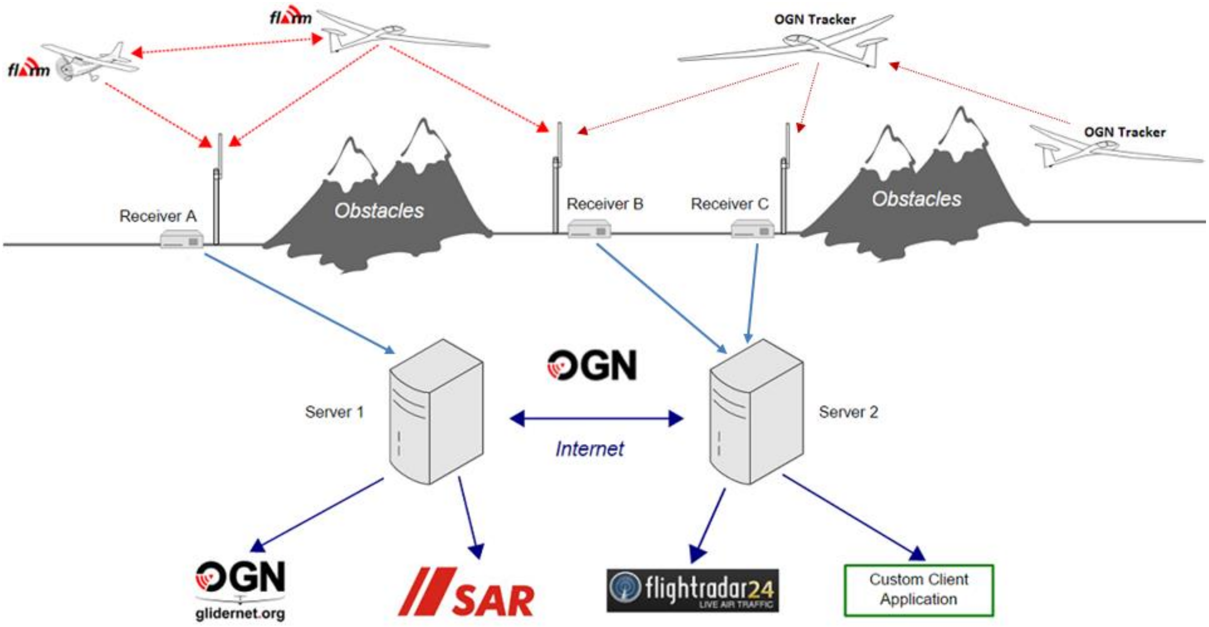



**Figure 11 - OGN system architecture [65]**

OGN differs significantly from both FLARM and PilotAware because their focus is tracking and collating glider movements over large areas, rather than localised collision avoidance. However, with the introduction of their "Do-It-Yourself" OGN Tracker [21] (pictured in Figure 12), they have provided the cheapest available tracking solution to anyone that is comfortable doing basic soldering. Like FLARM and PilotAware, the OGN Tracker uses a GPS receiver to collect information on the aircrafts location and track, to make regular position reports via an RF transceiver. Based around an ARM Cortex-M3 microcontroller [21], the OGN Tracker has a smaller form factor and less power requirements than both FLARM and PilotAware. However, the main advantage is that the total component cost for each tracker is around £35 [21]. This makes it much more viable for the GA community, and especially glider enthusiasts who don't have the capability to support high power or large devices in their aircraft. Since the tracker's primary objective is to provide data for ground based receivers, it lacks any display functionality. A small buzzer is the primary method for warning pilots of hazards, although some users have incorporated an LCD display to show more information (pictured in Figure 13).

Since the OGN Tracker is designed primarily as a tracking solution, it would require some modification to act as a collision avoidance system. However, because anyone can utilise the OGNTP protocol in their own designs, there are no constraints on who can benefit from the traffic information. OGN was used for live tracking of the gliding event at the 2015 World Air Games [22], hosted in Dubai, which shows how prevalent this technology is across the gliding community. However, there is no reason the benefits of the OGN approach couldn't be shared

**Figure 12 - OGN tracker with potentiometer [21]**


**Figure 13 - OGN tracker with LCD display [21]**

across the larger GA community. Since the hardware is straightforward to assemble and configure, it could be adapted for use at small airfields in the form of a cheap air traffic control console or as a navigational aid.

The OGN ground-based receivers run from a Linux-based computer, the most popular option being a Raspberry Pi. By utilising the same RTL-SDR dongle that PilotAware uses, each ground station can receive and decode both OGN and FLARM transmissions. Received information is then uploaded to central servers, where it is collated and broadcast to be used by a variety of tools. An important distinction to make is that although the OGN Tracker only transmits using the OGNTP protocol, the ground-based receivers can decode both OGNTP and FLARM. This is despite FLARM using an encrypted, proprietary protocol.

The argument from the OGN perspective is that FLARM is simply a source of positioning information. OGN do not use their collision detection aspects or algorithms, and FLARM broadcasts data on publicly available frequencies. Despite this, decoding FLARM without the permission of the developers is most likely illegal under Article 6 of EU Directive 2013/40/EU [23]. However, despite many of the 7000 devices [20] registered on the publicly available OGN aircraft database being FLARM units, there have not yet been any legal objections to the activities of the OGN community. This could be helped in part by the fact that the OGN has kept their FLARM decoding software closed-source.

#### 1.3.4 Trialling ADS-B in GA and LPAT

Ongoing work by the UK National Air Traffic Services (NATS) is focused on enabling ADS-B transponder functionality across GA [24]. NATS began a trial in 2014 that allowed GA pilots to use the full functionality of a Mode S transponder for the first time. This was made possible by relaxing safety regulations to allow the connection of a non-certified GPS receiver to a Mode S transponder. Under current regulations, GPS receivers must have their accuracy certified to Civil Aviation Authority (CAA) standards. The purpose of the trial is to assess the following [25]:

1. The typical accuracy of ADS-B reported positions from non-certified GPS sources.
2. The impact of low integrity GPS data on ATC surveillance systems.
3. The interest of GA users in ‘situational awareness’ applications that assist a pilot to visually acquire nearby traffic.

A significant advantage with this approach is that by widening the availability of the current leading collision avoidance platform, coverage can be increased whilst utilising existing techniques and infrastructure. Since ADS-B is a well-defined and globally recognised standard, this is the most feasible route to unite both commercial and private aircraft on a single platform.

Recognising that cost, weight and power restrictions are issues for many GA pilots, NATS are also trialling a new prototype device called the Low Power ADS-B Transceiver (LPAT) (as seen in Figure 14), in conjunction with Funke Avionics [26]. LPAT is positioned as a portable, battery powered, affordable device that can provide the minimum functionality required to make a GA pilot visible to other airspace users. NATS GA Lead Jonathan Smith said [27]:

*"For those that either can't afford to invest in a transponder, or don't have the power to run one or are weight restricted – like some very light aircraft, gliders or perhaps a hot air balloon – LPAT could be the answer. Transponders are the ideal solution for some GA aircraft, but any additional information pilots and NATS can have about aircraft in the sky, the better and safer for everyone."*

LPAT is based on two existing Funke Avionics products, the TM250 Traffic Monitor and the TRT800H Mode S transponder [26]. It receives traffic information via ADS-B and FLARM, and transmits reduced power ADS-B messages. It's rumoured that the eventual cost per unit will be under £1000, but since this system is not yet commercially available it has been excluded from further comparison with existing platforms.

However, since this product has been instigated by NATS themselves, when it does become available beyond the scope of the trial it has the potential to become quickly widespread. Combined with allowing non-certified GPS sources to be connected to Mode S transponders, the conclusions of the trial will have big implications for collision avoidance in GA and especially for the existing commercial platforms.


**Figure 14 - Prototype LPAT unit [26]**

### 1.3.5 Comparison Table

**Table 2 - Comparison of collision avoidance systems**

| | SSR Mode A/C | SSR Mode S | FLARM | PilotAware | OGN |
|---|---|---|---|---|---|
| Main parameters transmitted | 4-digit octal identification code. Aircraft's pressure altitude [9]. | Altitude, latitude, longitude, horizontal velocity, vertical velocity, heading, and callsign [9]. | Position, altitude, speed, acceleration, track, turn radius, wind, vertical speed, aircraft type and unique identification number [14]. | ICAO identifier, longitude, latitude, altitude, track, speed, aircraft type [17]. | Aircraft ID, emergency flag, relay count, stealth flag, time, latitude, longitude, altitude, climb rate, speed, heading and turn rate [19]. |
| Manufacturer | Multiple avionics manufacturers. | Multiple avionics manufacturers. | FLARM Technology Ltd. | PilotAware Ltd. | Open source design. User assembled. |
| Hardware required | Mode A/C compatible aircraft transponder. | Mode S compatible aircraft transponder. | PowerFlarm Core or PowerFlarm Portable unit. | PilotAware Classic unit. | User assembled OGN tracker. Display requires modification. |
| Price (ex VAT) | From £595.00 [28] | From £1650.00 [29] | From £1225.00 [14] | £159.99 [17] | ~£35.00 [21] |

| | SSR Mode A/C | SSR Mode S | FLARM | PilotAware | OGN |
|---|---|---|---|---|---|
| Subscription cost | - | - | - | £15.00 annually [17] | - |
| Receiving capabilities | SSR Modes A and C. | SSR Modes A, C and S. | SSR Modes A, C and S when combined with an appropriate transponder. FLARM [14]. | SSR Modes A, C and S. FLARM when combined with a FLARM mouse. P3I [17]. | OGNTP.<br><br>Ground-based receivers support OGNTP and FLARM [19]. |
| Transmitting capabilities | SSR Modes A and C. | SSR Modes A, C and S. | FLARM. | FLARM when combined with a FLARM mouse. P3I [16]. | OGNTP. |
| Estimated operating range | 170km [30] | 300km [31] | 25km [32] | 25km [33] | 25km [34] |
| Estimated userbase | > 50,000 | > 50,000 | > 30,000 | > 1,000 | > 7,000 |
| Primary market | Commercial aviation. | Commercial aviation. | General aviation and gliders. | General aviation and gliders. | General aviation and gliders. |
| Form factor | Panel integrated instrument. | Panel integrated instrument. | Panel mounted instrument. | Aircraft mounted. | Aircraft mounted. |

| | SSR Mode A/C | SSR Mode S | FLARM | PilotAware | OGN |
|---|---|---|---|---|---|
| Displays available | Multipurpose or specialised integrated displays. | Multipurpose or specialised integrated displays. | Panel mounted display, phone or tablet app. | Phone or tablet app. | None, auditory only. Possible for user to add their own display. |
| Collision avoidance capability | Limited as no positional information conveyed. | Proximity warnings. Automated instructions can be issued through TCAS. | Proximity warnings. | No warnings given, only information on the position of nearby aircraft. | None. Would require user modification. |
| Unique features | Certified to aviation regulatory standards. | Certified to aviation regulatory standards. | Custom collision avoidance protocols. Obstacle database. Certified flight path recording. | Can receive and transmit over three protocols. Connection to phone app. | Very inexpensive. Possibility to customise hardware design with relevant electronics experience. |
| Source code and protocol availability | Published, publicly available standard. | Published, publicly available standard. | Closed source. Protocol encrypted. | Open hardware design. Some protocol information published. | Open source. FLARM decoder source code is closed source. |

| | SSR Mode A/C | SSR Mode S | FLARM | PilotAware | OGN |
|---|---|---|---|---|---|
| Processor | Various | Various | Unpublished | RPi P2B [16] | ARM Cortex-M3 [21] |

## 1.4 Controversies

Despite advances in technology and emphasis on proper piloting technique, around 150 Airprox events still occur annually in UK airspace [35]. An Airprox is a situation in which, in the opinion of a pilot or air traffic services personnel, the distance between aircraft as well as their relative positions and speed have been such that the safety of the aircraft involved may have been compromised [35]. These occur most often in airspace near aerodromes, where aircraft fly the same departure, approach and circular patterns. Figure 15 shows the Airprox 10-year trend grouped into categories, with Category A indicating the most serious events. More detailed category descriptors are available in the appendix.

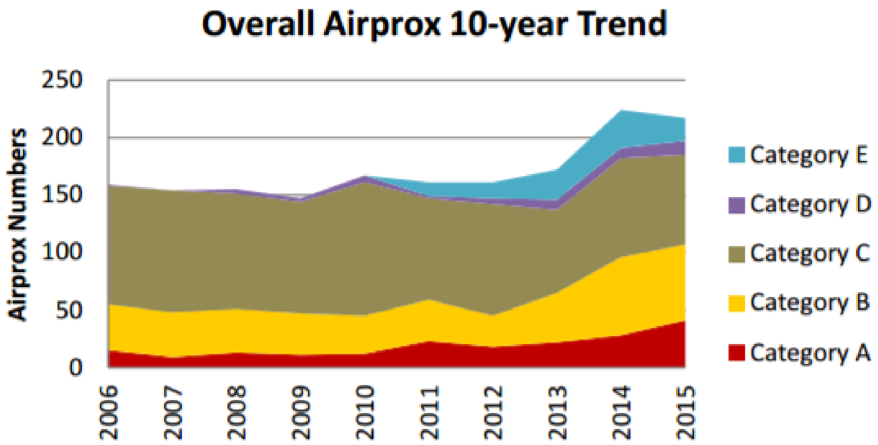



**Figure 15 - Overall Airprox 10-year Trend [35]**

As we can see in Figure 15 the number of annual Airprox events is quite consistent, and has even shown a slight upward trend in the last 5 years. Whilst the clear majority of these fall into Category C or below (no risk of collision), it is apparent that if the overall number of Airprox events increases then the proportion of serious or fatal events is also likely to increase. As such, careful thought must be given into how technology can be used to reverse this increasing trend.

To understand how mid-air collisions fit into the overall picture of GA accidents, a useful source to consider is the Joseph T. Nall Report [36]. This is an annual report by the Aircraft Owners and Pilots Association (AOPA) which reviews GA accidents in the US. The latest report available states that there were 6 mid-air collisions in 2013, of which 3 were fatal [36]. Whilst this type of accident only makes up a small percentage of the 709 total accidents of that year, it is likely that collision avoidance technology could have prevented these accidents. In addition, it is likely that the number of incidents that came close to being a mid-air collision is much higher. Another worrying trend in both general and commercial aviation is the increasing availability of Unmanned Aerial Vehicles (UAVs) [35]. CAA regulations state that recreational UAVs must be flown below 400ft, at least 50m from any buildings or people and kept within line of sight [37]. Despite this there were 56 Airprox reports involving drones in 2016, nearly double the previous year [35]. Since drones are relatively low cost, easily available and remotely operated they are set to pose an increasingly serious issue in airspace across the world. In the last year, there have been multiple reports of near misses between drones and commercial aircraft including:

- A narrow miss between an A320 and a white quadcopter at 6000ft over Hertfordshire in August 2016 [38].
- A narrow miss between a Eurocopter 145 and a drone at 1900ft over London in August 2016 [39].
- A drone sighting around 500m from a Bombardier Dash 8, which was at a height of 500ft on approach to Birmingham International Airport in September 2016 [40].
- A drone sighting around 30m from an A320, which was at a height of 10,000ft on approach to Heathrow Airport in November 2016 [41].

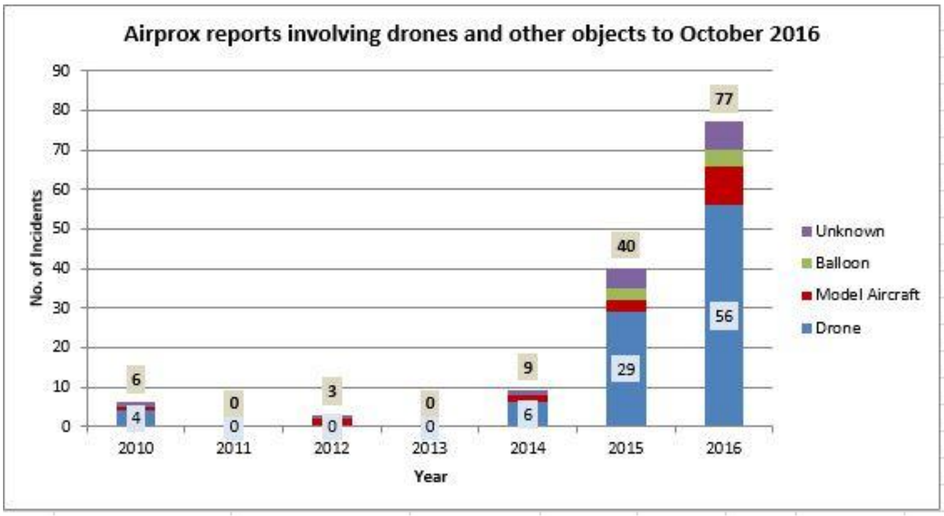



**Figure 16 - Airprox reports involving drones from 2010-2016 [35]**

Clearly drones are being operated with disregard to the CAA regulations, and now pose a serious threat to commercial aircraft. Figure 16 shows this increasing trend. Soon it will become essential for drones to be fitted with collision avoidance technology to make them visible to pilots. A low-cost platform like the OGN tracker would be a potential solution. Researchers at the Nanyang Technological University and the CAA of Singapore are working on a system of air lanes to keep drones on a safe path [42]. This is in preparation for a future where there are likely to be large numbers of drones operating in small areas, such as city centres. By establishing a network of set routes, the risk of collision can be minimized. In addition, the system can geofence certain areas (such as government buildings or crowds) to prevent drones flying in that area.

# 2 Procedure

An overall theme throughout both the hardware and software development was verifying the individual functionality of each modular section. During the initial hardware assembly, this was achieved by confirming the functionality of each component at its most basic level, before going on to integrate it with the rest of the system. During software development, I took advantage of a useful feature of the chosen environment (GNU Radio): the ability to instantly see graphical results when adding or modifying sections of code. Combining this with the ability to test the system with pre-recorded, consistent data meant I could apply an easily verifiable testing methodology throughout the different aspects of program design.

## 2.1 Hardware Environment

#### 2.1.1 Raspberry Pi

1. Raspberry Pi 3 Model B

I chose the RPi 3 because it is easy to source, has a relatively low cost and is quite user friendly since its primary market is hobbyist and educational use. Additionally, it also has more than enough processing power for my application. There are a range of purpose-built accessories (such as the touchscreen) which can be quickly added without having to design or build anything myself. This component fulfils the portability requirement.

2. 7" Touchscreen Display

As the system operates as a stand-alone unit, there needs to be some way for the user to control the RPi and view the output from the receivers. The official 7” RPi touchscreen is perfect for this and it comes pre-assembled for fast implementation.

3. Anker PowerCore 20100

The system is intended to become a part of the “flight kit” that every pilot has. As such it needs to be portable to move between aircraft. Additionally, by using its own dedicated power source the system is isolated from the rest of the aircraft instruments, which will prevent any interference with critical systems and ensure that in the event of an aircraft electrical failure the system is still operable. The Anker PowerCore 20100 more than meets the requirements. As it has two USB ports that can both output up to 2.4A, there is no risk of reaching the current limit. Full power requirement calculations are available in the appendix.

4. NooElec NESDR SMArt

To receive the ADS-B and OGN transmissions, I used an RTL-SDR dongle. RTL-SDR is an ultra-cheap software defined radio based on DVB-T TV tuners with RTL2832U chips. The RTL-SDR can be used as a wide band radio scanner. It's popular with ham radio enthusiasts, hardware hackers, tinkerers and anyone interested in RF. The discovery that these DVB-T TV tuners could be utilised as a wide band radio scanner was a revelation in the field of hobbyist electronics. Never has such capable RF equipment been so widely available, and its advent has encouraged a new generation of radio enthusiasts to get involved with SDR.

**Figure 17 - RTL-SDR dongles**

5. VK16U6 GPS Module

To provide GPS position, I connected a GPS receiver (the same model as the OGN Tracker) to the RPi. This provides standard National Marine Electronics Association (NMEA) sentences via the RPi's universal asynchronous receiver/transmitter (UART) interface.

6. AltIMU-10 v4

This sensor board contains a compass, accelerometers, gyroscopes and an altimeter. By utilising this data, I can provide additional parameters to the user through a basic graphical user interface (GUI).

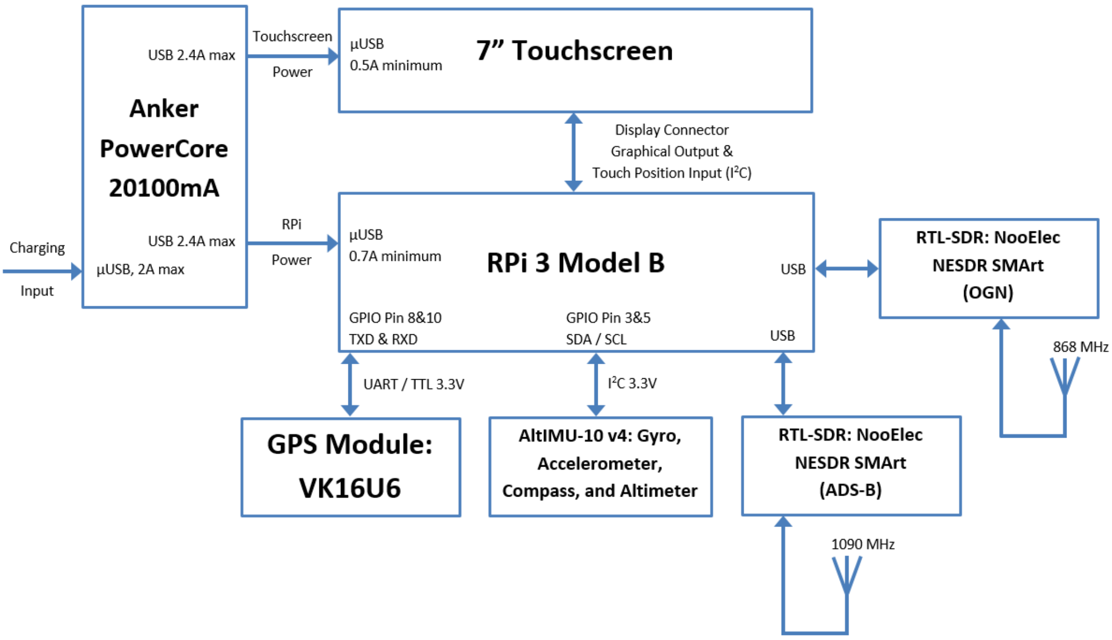



**Figure 18 - Block diagram of the proposed receiver**


**Figure 19 - Assembled RPi with GPS and AltIMU connected**


**Figure 20 - Assembled RPi with GPS and AltIMU connected**

### 2.1.2 OGN Tracker

The assembly of the OGN Tracker followed the DIY OGN Tracker design [21]. The components used were:

1. STM32F103C8T6 Cortex-M3 System Development Board

   The main CPU. This is an advanced microcontroller that offers features such as high-speed embedded memory, an extensive range of I/O ports and peripherals, 12-bit analogue to digital converters (ADC's), multiple general purpose timers and a variety of industry standard interfaces [43].

2. HopeRF RFM69HW 868MHz Transceiver

   The RFM69W is a transceiver module capable of operation over the 868MHz license-free frequency bands. All major RF communication parameters are programmable and most of them can be dynamically set. The RFM69W offers the unique advantage of programmable narrow-band and wide-band communication modes. The RFM69W is optimized for low power consumption while offering high RF output power and channelized operation [44].

3. Male SMA to Female SMA, 167mm Coaxial Cable

4. 868MHz Antenna

5. VK16U6 GPS Module

   To provide GPS position. Outputs standard NMEA sentences via UART.

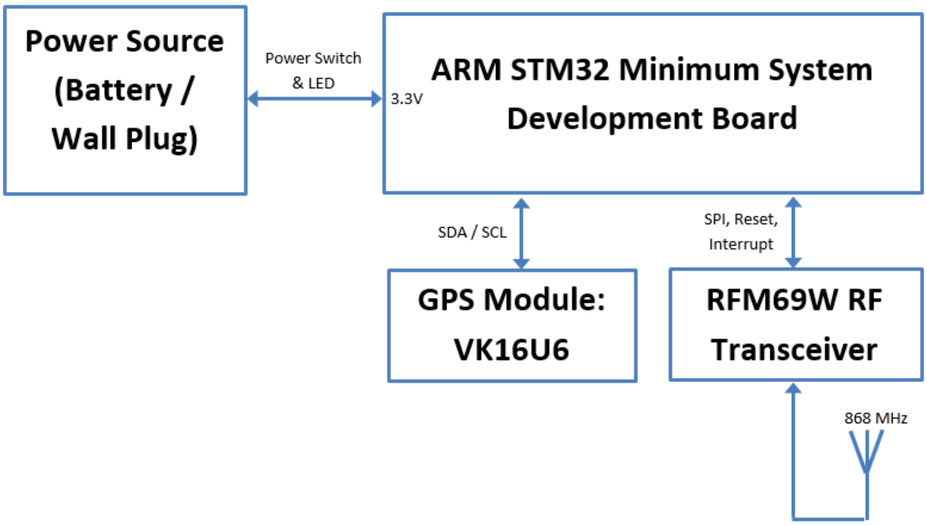



**Figure 21 - Block diagram of an OGN Tracker**

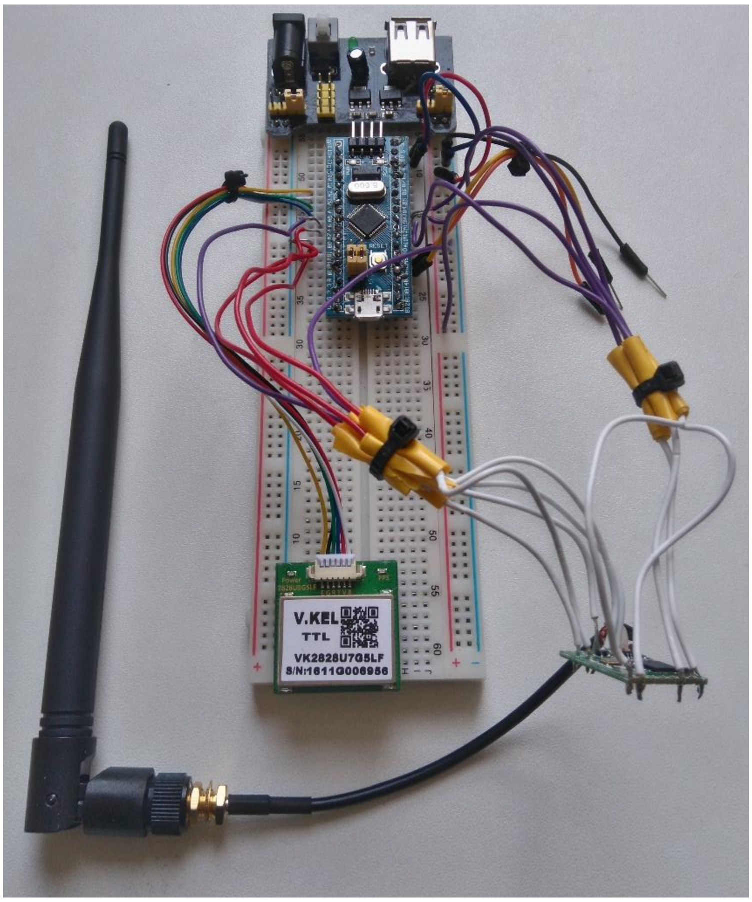



**Figure 22 - OGN Tracker implemented on a breadboard**

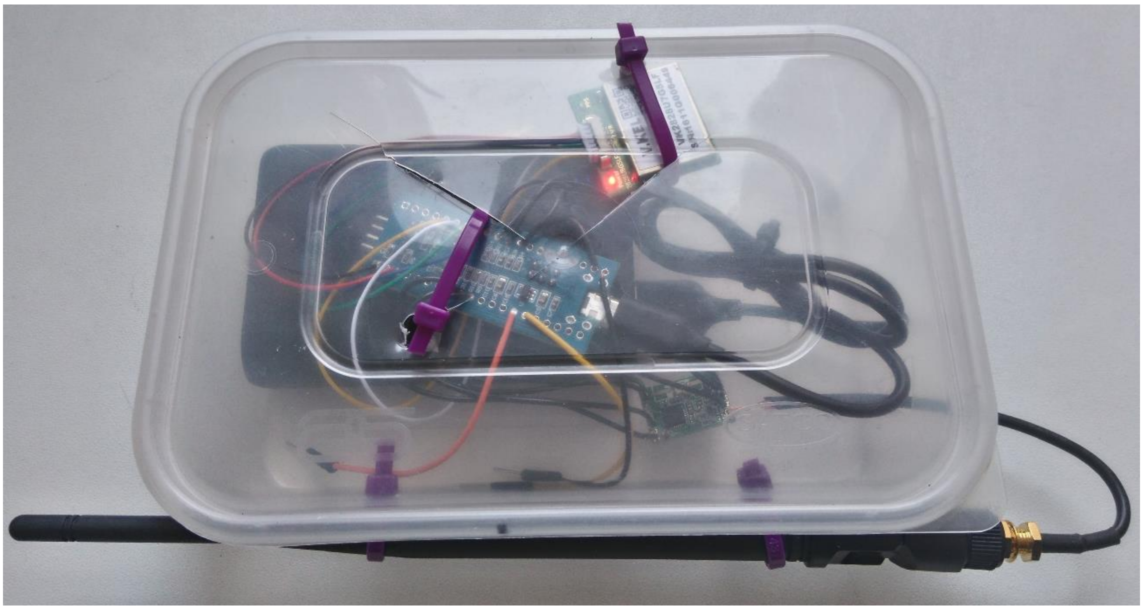


**Figure 23 - OGN Tracker soldered and packaged inside a box**

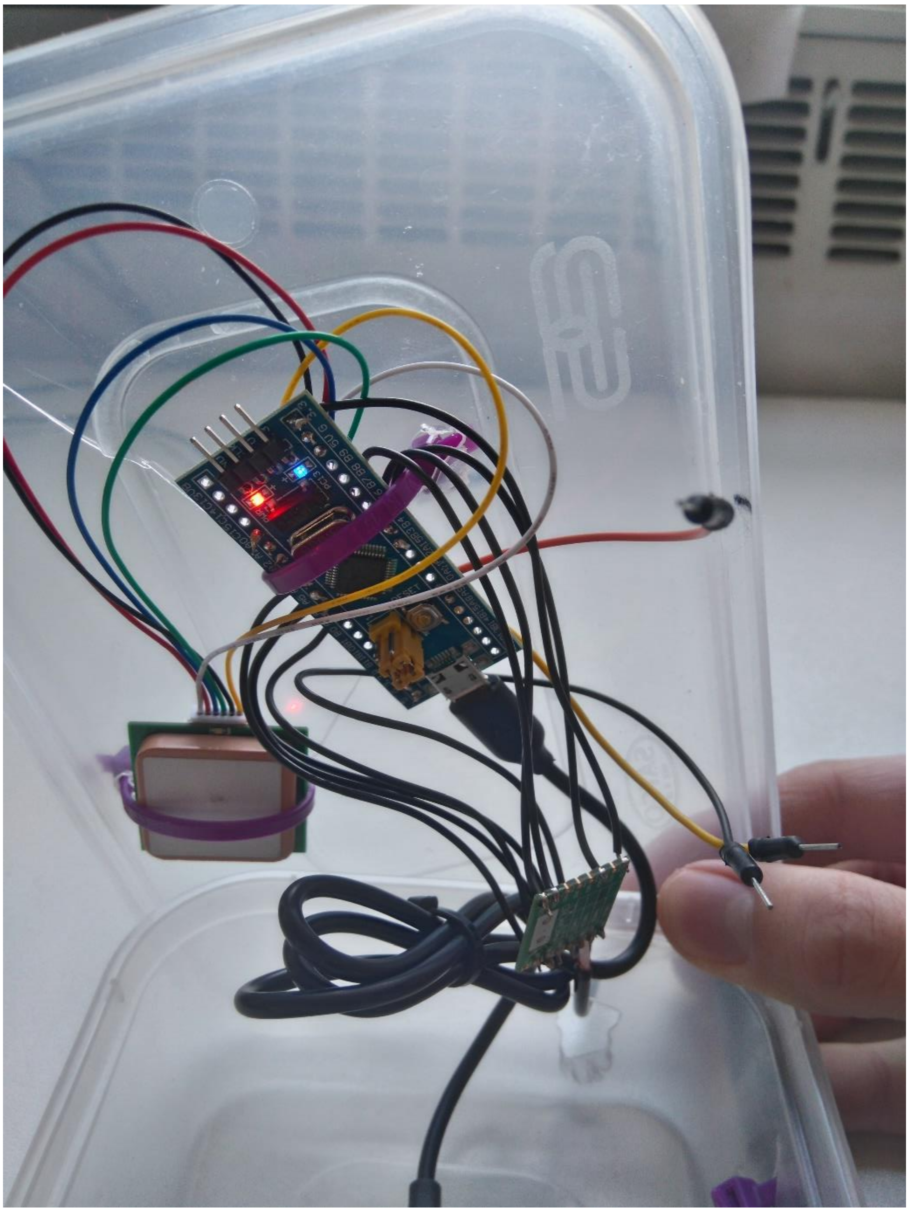


**Figure 24 - The microcontroller, GPS receiver and RF transceiver that make up the OGN Tracker**

## 2.2 Software Environment

### 2.2.1 GNU Radio

GNU Radio is a free software development toolkit that provides signal processing blocks that allow users to create their own software defined radios and custom signal-processing systems. It can be used in both a simulation environment as well as with external RF hardware. It has seen widespread use across hobbyist, academic and commercial environments in a variety of applications [45].

GNU Radio provides the framework and tools to build and run software radio or general signal-processing applications. The applications themselves are known as flowgraphs, and are made up of discrete blocks that each perform some functionality. Data flows between these blocks and different data types are represented by different colours. There are also many graphical options available for providing display and control to the user. GNU Radio provides a powerful platform that allows users to quickly develop applications with a variety of different hardware sources.

GNU Radio was chosen for this project because it is compatible with the chosen hardware and there are examples and tutorials available on how to interface with it. In addition, the graphical nature of this software package means I could save time in the learning process and quickly build functionality. Finally, the ability to add custom blocks (in C++ or Python) meant I could add new functionality when required by my application.

### 2.2.2 Raspberry Pi

1) Raspberry Pi 3 Model B

The latest Raspbian operating system (based on Debian Jessie) was installed.

2) NooElec NESDR SMArt

The librtlsdr driver was used to turn the RTL2832U (RTL-SDR dongle) into an SDR [46].

3) GNU Radio

GNU Radio 3.7.5 was installed to run the SDR application.

4) VK16U6 GPS Module

This was connected to the RPi's UART port. I then used gpsmon, a real-time GPS packet monitor and control utility, for initial testing before using pynmea2 to develop my own Python script [47] [48].

5) AltIMU-10 v4

I used the RTIMULib2 Python IMU library to confirm functionality [49].

6) ogn-rf and ogn-decode

To verify the OGN Trackers functionality, I used the pre-packaged binary offered by the OGN for their receiver stations. These closed-source programs can receive and decode OGN packets [50].

7) dump1090

To verify ADS-B reception a well-established ADS-B decoding program, dump1090, was used [51].

### 2.2.3 OGN Tracker

The STM32 Flash Loader Demonstrator software utility was used on a Windows 10 machine to flash the source binary file. This was provided by the OGN and flashed to the STM32F103C8T6 via a USB to TTL cable [52] [21].

### 2.2.4 Laptop Computer

For the GNU Radio development process, most of the work was done on a laptop computer. By using this method, I could take advantage of the greater processing power and full featured desktop environment available from Ubuntu before deploying to the RPi.

The laptop was installed with Ubuntu 14.04 LTS and Windows 10. This allowed programming of the OGN Tracker and programming in GNU Radio (release 3.7.10) on the same device.

## 2.3 ADS-B Packet Protocol

Unlike transponders, which usually need to receive an interrogation pulse before transmitting, ADS-B systems continuously transmit aircraft parameters (known as a squitter). These are then picked up by dedicated ground stations that relay the information to ATC for precise tracking of aircraft, without the need for additional interrogation from the ground radar. Figure 25 shows the reasoning behind the ADS-B acronym.

ADS-B: [68]

Automatic - Requires no pilot or external input

Dependant - Depends on accurate information from other aircraft systems (e.g. GPS receiver)

Surveillance - Provides aircraft position and other data for surveillance purposes

Broadcast - Information is continually broadcast for other aircraft or ground stations to receive

**Figure 25 - ADS-B Description**

ADS-B data is broadcast every half-second on a 1090MHz digital data link. Data is encoded using pulse position modulation (PPM) at a rate of 1Mbit per second. To understand PPM, we must first understand amplitude-shift-keying (ASK). In ASK, binary data is sent by turning the carrier on or off. PPM uses the Manchester encoding variant of ASK. The idea behind Manchester encoding is that within each bit transmission, there is always a transition that occurs in the middle of the

```
BIN   | 10001   101 | 010010000100 | [00100]0000010110011000011011 | 010101110110
      |             | 000011010110 | 10001110000110010110011100000 | 000010011000
------+-------------+--------------+-------------------------------+--------------
DEC   |  17     5   |              | [4] ......................     |
------+-------------+--------------+-------------------------------+--------------
         DF     CA      ICAO          [TC] ------ DATA ----------     PI
```

**Figure 26 - A partially decoded ADS-B packet**

interval. This can reduce error rates, and has the advantage that the signal becomes self-clocking. This allows the receiver to recover the clock from the data stream.

An example of a PPM signal can be seen in Figure 27. Within each frame, the signal will be present at either the start or end depending on the value encoded. This is shown in Figure 28. This results in the middle of each frame denoting a 1 with a falling edge, or a 0 with a rising edge. As shown in Figure 28, this adheres to the G.E Thomas standard for Manchester encoding.

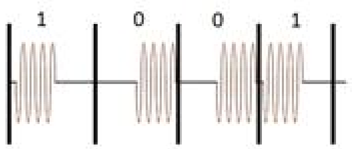



**Figure 27 - Modulated Manchester encoding [66]**

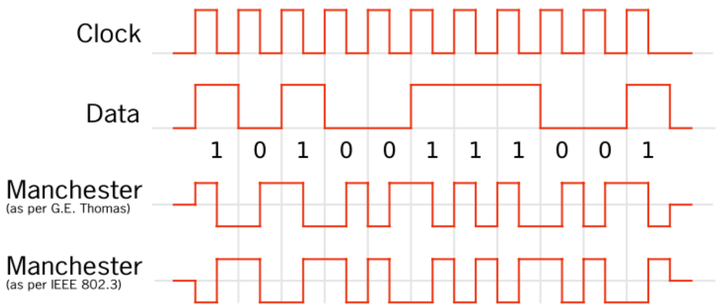



**Figure 28 - An example of Manchester encoding showing both possible conventions [67]**

An ADS-B downlink broadcast is made up of several fields, as can be seen in Table 3. To understand more about what each of these fields mean, let's study an example transmission (Figure 26):

- Downlink Format: For an ADS-B transmission this will always be 17 (10001 in binary), encoded in the first 5 bits.
- Capability: Sub type of the ADS-B message. This has different meanings within different types of ADS-B message, encoded in the next 3 bits.
- ICAO Aircraft Address: This is a unique identifying address for each aircraft, 24 bits wide.
- Type Code: Bits 33-37 (the first 5 bits of the Data segment) indicate the message type, which are described in Table 4.
- Data: The encoded aircraft information, specified by the type code.
- Parity Information: Error detection code.

**Table 3 - Sections of an ADS-B broadcast [69]**

| nBits | Bits | Abbr. | Name |
|---|---|---|---|
| 5 | 1 - 5 | DF | Downlink Format (17) |
| 3 | 6 - 8 | CA | Capability (additional identifier) |
| 24 | 9- 32 | ICAO | ICAO aircraft address |
| 56 | 33 - 88 | DATA | Data |
| | [33 - 37] | [TC] | Type code |
| 24 | 89 - 112 | PI | Parity/Interrogator ID |

**Table 4 - ADS-B type codes [69]**

| TC | Content |
|---|---|
| 1 - 4 | Aircraft identification |
| 5 - 8 | Surface position |
| 9 - 18 | Airborne position (w/ Baro Altitude) |
| 19 | Airborne velocities |
| 20 - 22 | Airborne position (w/ GNSS Height) |
| 23 - 31 | Reserved for other uses |

An ADS-B packet has the capability to contain a range of different information, depending on the specific mode. The type code is important and should be carefully considered to correctly interpret the data field. Finally, each ADS-B packet starts with a preamble made up of four specifically spaced pulses over a period of 8us. This is a straightforward way to mark the start of each packet.

## 2.4 OGN Packet Protocol

The OGN Transport Protocol (OGNTP) uses a Gaussian Frequency Shift Keying (GFSK) modulated carrier. FSK is a modulation method in which the frequency of the carrier signal is modified to embed data. For example, with a 2-level FSK system there are only two different frequencies the carrier signal can take. These correspond to 0 and 1. Once the carrier has been modulated with a binary stream, it will hop between the two set frequencies as it runs through the binary data, as can be seen in Figure 29.

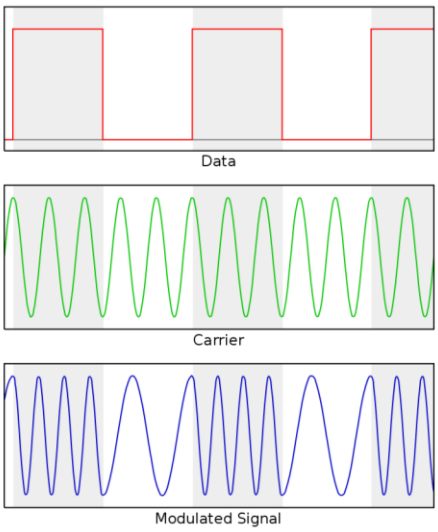



**Figure 29 - Binary FSK [70]**

GFSK refers to the practice of applying a Gaussian filter to the input data stream, before modulating the carrier signal. This means that instead of a sudden jump e.g. -1 to 1, there will be intermediate levels e.g. -1, -0.98, -0.96 etc. This is beneficial because it makes the transitions between the different frequency levels smoother. This in turn reduces sideband power and reduces interference with neighbouring channels.

The OGN Tracker transmits identical packets on 868.2MHz and 868.4MHz. It uses GFSK with a modulation index equal to 0.5. This means that the phase shift is equal to +/- Ω/2 every bit period [53]. The bitrate is equal to 100kbps, but since Manchester Encoding is used (IEEE standard) the effective bitrate is 50kbps. It takes 20us to transmit a single bit. The FSK deviation is +/- 50KHz. Manchester encoding is used to prevent long streams of 0s or 1s, which in turn reduces the likelihood of losing alignment at the receiver. In addition, an implementation of the Tiny Encryption Algorithm (TEA) is used on several fields for the same purpose, somewhat redundantly [54].

Each OGN packet begins with an 8-bit preamble, which differs between RF transceivers so should not be relied upon for identifying packets. Following this, there is a 4-byte SYNC pattern which is set in the source code to be equal to the hexadecimal value 0x0AF3656C. Since this is more consistent it is recommended to look for this when identifying packets. Following this there is a 4-byte header and a 16-byte data field. Finally, there are 6-bytes assigned for Forward Error Correcting (FEC) using Low Density Parity Check (LDPC) code. It should be noted that several fields have variable resolution coding to accommodate a greater range of values with a limited bit width. Fields are also stored using little endian, which means the least significant bit (LSB) is sent first. An example of this is shown in Figure 30. Table 5 lists the OGNTP packet fields in more detail. Figure 31 and Figure 32 are detailed examples of how to decode some fields of an OGNTP packet.

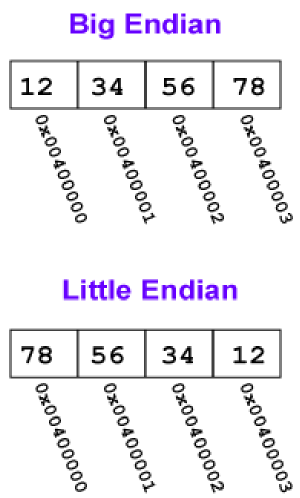



**Figure 30 - Big and little endian notations [81]**

**Table 5 - OGN packet parameters [54] [55]**

| Parameter | Width (bits) | Range | Resolution | Unit | Description |
|---|---|---|---|---|---|
| Header Field | | | | | |
| Emergency Flag | 1 | | | | Can be set manually or automatically based on sensors |
| Encrypt Flag | 1 | | | | Can be set if custom protocol is used |
| Relay Count | 2 | 0-3 | | | Number of relays between trackers |
| Parity | 1 | | | | |
| Meteo Flag | 1 | | | | Meteorological report on conditions |
| Address Type | 2 | 0-3 | | | 1 = ICAO, 3 = OGN |
| Address | 24 | 0x000000-0xFFFFFF | | | ICAO or OGN internal database |
| Data Field | | | | | |
| GPS Fix Quality | 2 | | | | 0 = no fix, 1 = GPS, 2 = D-GPS, 3 = other |
| Time | 6 | 0-59 | 1 | Second | UTC second, 0x3F=out of range |
| Latitude | 20 | +/- 90 | 0.0008/60 | Degree | 1.5m accuracy |
| GPS Fix Mode | 1 | | | | 0 = 2D, 1 = 3D |
| Barometer Flag | 1 | | | | For trackers with a barometer and an algorithm to correlate GPS altitude with pressure altitude |
| GPS DOP | 6 | | | | GPS dilution of precision |
| Longitude | 20 | +/- 180 | 0.0016/60 | Degree | 3 Meters on the equator |

| Parameter | Width (bits) | Range | Resolution | Unit | Description |
|---|---|---|---|---|---|
| Turn Rate | 8 | +/- 47.2 | 0.1-0.8 | Degrees / second | |
| Speed | 10 | 0-766 | 0.2-1.6 | Knots | Ground speed |
| Altitude | 14 | 0-61432 | 1-8 | Meters | Above Geoid or above sea level |
| Temperature | 8 | +/- 236 | 1 | °C | |
| Aircraft Type | 4 | 0-15 | | | Glider, tow plane, helicopter etc. |
| Private Flag | 1 | | | | Can be set if position should not be published online (e.g. in competitions) |
| Climb Rate | 9 | +/- 95.2 | 0.1-0.8 | Meters / second | Sourced from GPS or barometer |
| Heading | 10 | 0-360 | 0.1 | Degree | Ground heading |

Raw data: 0001 0101 1001 0001 0001 1011 0000 0011

-Split this into 4 bytes and reorder, beginning with the last byte

-First 32-bit word: 0x031B9115 (stored in little endian)

-First byte: 0x03 - 0000 0011

Emergency flag: 0

Encrypt flag: 0

Relay count: 00

Parity: 0

Meteo flag: 0

Address type: 11 (3 = OGN)

-Next 3 bytes: 0x1B9115 is aircraft ID 1B9115

**Figure 31 - Decoding the header field [55]**

Raw data: 1011 0100 1111 1110 1111 0111 1001 1100

-Split this into 4 bytes and reorder, beginning with the last byte

-Second 32-bit word: - 0x9CF7FEB4 (stored in little endian)

-Need to apply TEA_Decrypt, which with a key of 0 yields 0x423C1151

-First byte: 0x42 - 01000010

GPS fix quality: 01 (1 = GPS)

Time: 000010 (2 seconds)

-Latitude: 0x003C1151

-To recover the sign, we can shift left to move the sign bit into the top bit: 0x3C115100

-Undoing a possible rounding can be done by adding 0x00000080: 0x3C115180

-Scale convert to a decimal floating point value and scale to get a degree value:

0x3C115180 -> 1007767936 / 600000 / 32 = 52.48791 degrees

**Figure 32 - Decoding the first three parameters of the data field [55]**

## 2.5 GNU Radio Companion Flowchart

### 2.5.1 ADS-B

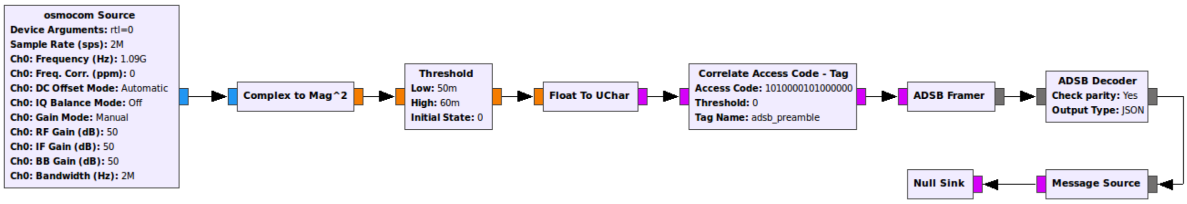



**Figure 33 - GNU Radio Companion (GRC) flowchart for the ADS-B receiver**

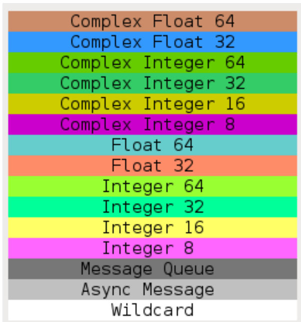



**Figure 34 - GRC variable colours**

As mentioned in the Procedure section of this report, each GNU Radio Companion program is represented by a flowchart which is made up of discrete blocks. Each of these blocks performs a certain function on the data flowing through it. As can be seen in Figure 33 and Figure 34, the colours of the input and output ports on the blocks change as data is converted between different variable types. In this section, the flow chart will be broken down and the purpose of each of these blocks will be explained. The ADS-B receiver is a modified version of work published by Wolfgang Nagele under his gr-adsb project [56].

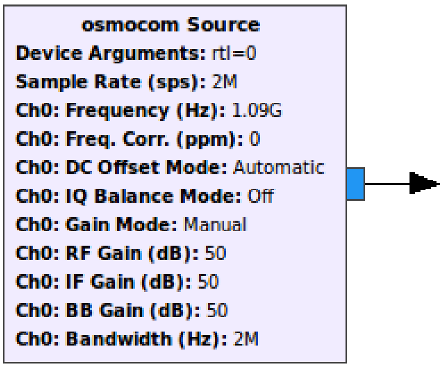



**Figure 35 - osmocom Source Block**

This block originates from the gr-osmosdr package [71]. It provides a hardware reference for a range of RTL-SDR devices. This block allows the user to set the sampling rate, centre frequency and a range of other parameters for the hardware.

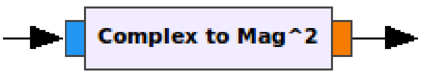



**Figure 36 - Complex to Mag^2 Block**

Since ADS-B uses PPM, the magnitude of the signal at each point in time is required. By using this block, the data stream is converted from complex to float, thus potentially losing some information (e.g. phase). This is the most sensible approach for this application.

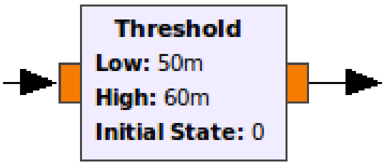



Figure 37 - Threshold Block

The Threshold block will take an input stream and output a 1 if the input value is above the High Limit, and a 0 if it is below the Low Limit. In this application, it serves to clean up the incoming binary signal into two discrete values, rather than a range.

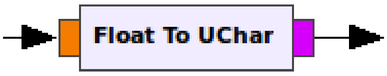



Figure 38 - Float to Unsigned Character Conversion Block

Since the data stream will now only be 1 or 0, it makes sense to convert to a more sensible format. UChar has a range of 0-255.

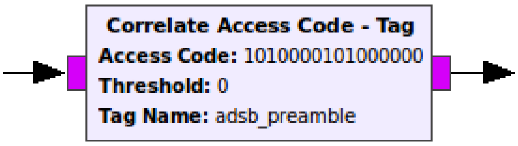



Figure 39 - Correlate Access Code - Tag Block

The Correlate Access Code - Tag block is used to detect the ADS-B preamble. When the specified sequence is found, a tag will be inserted into the data stream. This has no effect on the data itself, but other blocks can identify the tag and recognise the start of the data packet. The ADS-B preamble consists of four high pulses, which can be seen in the Access Code.

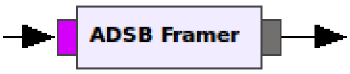



Figure 40 - ADS-B Framer Block

This block has been created by the author of the ADS-B decoding project. The Framer block can detect the "adsb_preamble" tag inserted by the previous block. Upon detection, it will convert the incoming data stream from the modulated format into a usable format. To do this it will look for the rising or falling edge in the middle of each bit period, as mentioned in the Method section, and determine whether that bit should be a 0 or a 1. Once the block has processed the number of bits required for a packet, it will output the data to a message queue. This queue can store the data packets until they are ready to be processed by the next block.


**Figure 41 - ADS-B Decoder Block**

The ADS-B Decoder block processes messages added to the queue by the previous Framer block. Packets are checked for errors that occurred in transmission and each field is decoded according to its specific format. Once a complete decoded message has been constructed, it is output to the console and to the log files.

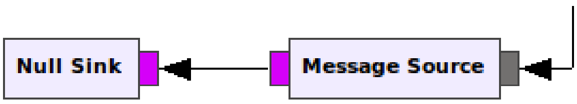



**Figure 42 - Null Sink**

The original version of this decoder had further processing in the GRC flowchart. Since the message has already been written to a log, the Null Sink simply discards the information.

### 2.5.2 OGN

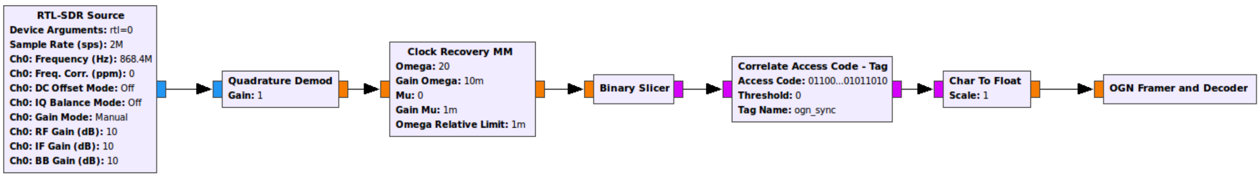



**Figure 43 - GRC flowchart for the OGN receiver**

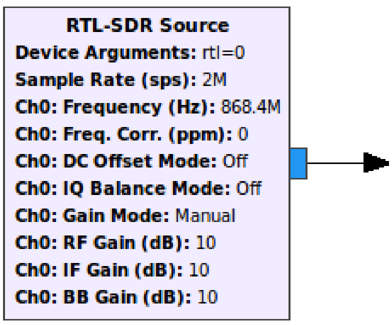



**Figure 44 - RTL-SDR Source Block**

This block originates from the gr-osmosdr package [71]. It provides a hardware reference for a range of RTL-SDR devices. This block allows the user to set the sampling rate, centre frequency and a range of other parameters for the hardware. It can be used interchangeably with the osmocom Source block.

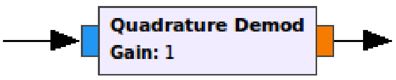



**Figure 45 - Quadrature Demodulator Block**

The quadrature demodulator passes the modulated carrier signal through an LC tank circuit that shifts the signal by 90° at the centre frequency (Figure 46) [72]. The phase shift is then either greater or less than 90°, depending on the direction of deviation. The demodulated baseband signal is compared to the phase-shifted signal using a phase detector. The different phase offsets correspond to different levels in the GFSK modulation scheme.

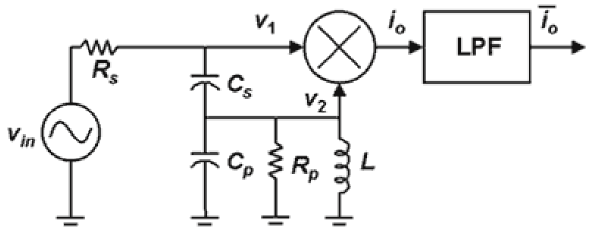



**Figure 46 - Quadrature demodulator block diagram [72]**

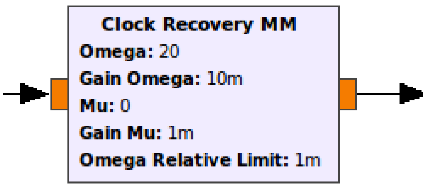



**Figure 47 - Clock Recovery MM Block**

The Mueller and Müller (M&M) clock recovery block can recover samples from a signal with the same frequency and phase as used by the transmitter. It allows the receiver to synchronize with the centres of 1s and 0s present in the signal, which is demonstrated in Figure 48 [73].

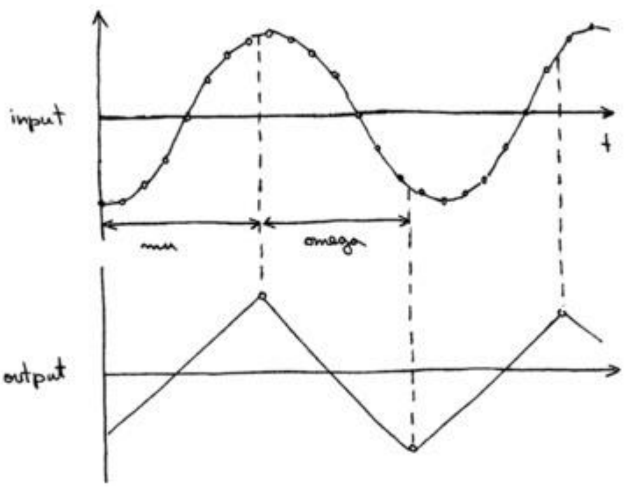



**Figure 48 - Clock recovery theory [73]**

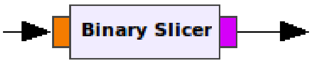



**Figure 49 - Binary Slicer Block**

The Binary Slicer outputs a 0 if the input value is below 0, and outputs a 1 if the input value is above 0. Performs the equivalent role to the Threshold block in the ADS-B decoder.

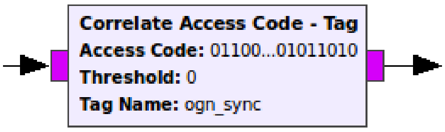



**Figure 50 - Correlate Access Code - Tag Block**

The Correlate Access Code - Tag block is used to detect the OGN SYNC pattern, 0x0AF3656C, which is a 28-bit sequence. When the specified sequence is found, a tag will be inserted into the data stream. This has no effect on the data itself, but other blocks can identify the tag and recognise the start of the data packet.

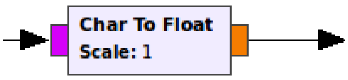



**Figure 51 - Character to Float Conversion Block**

For ease of working with the data stream within the following custom block, the stream is first converted to floating point.


**Figure 52 - OGN Framing and Decoding Block**

This block performs the same high level functions as the ADS-B Framer and Decoder. It was created during this project. The Framing section will wait for the “ogn_sync” tag before buffering 320 bits of data. This could take several iterations of the program due to the way GNU Radio handles streaming data. Once the buffer is full, the Framer will attempt to apply the Manchester Decoder to the information. If this completes without error, the 160 user bits are passed on to the Decoder.

The Decoder separates the data out to the fields listed in Table 5 and performs the decryption required to convert the data into its final format. This is then output to the console and written to the log files. Full source code is available in the Annexe.

## 2.6 Recording and Displaying Data

Data is recorded and displayed to the user in the following ways:

- Each time a valid packet is detected by either receiver, it is output directly to the console.
- Each time a valid packet is detected by either receiver, it is recorded into a log file which is in plain text format. An example of this is available in the Annexe.
- Each time a valid packet is detected by either receiver, it is recorded into a JavaScript Object Notation (JSON) log. An example of this is available in the Annexe.
- By utilising the built-in web server for Python, SimpleHTTPServer, as well as a Javascript webpage that utilities the Google Maps application programming interface (API), the user can open a webpage that will read the JSON log files and plot each received aircraft transmission in a Google Maps window. Full source code for the webpage is available in the Annexe. Figure 53 and Figure 54 show an example of this.

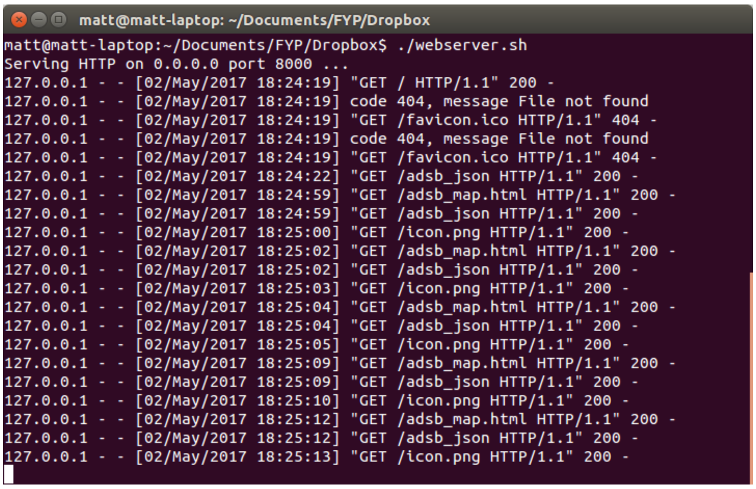



**Figure 53 - SimpleHTTPServer**

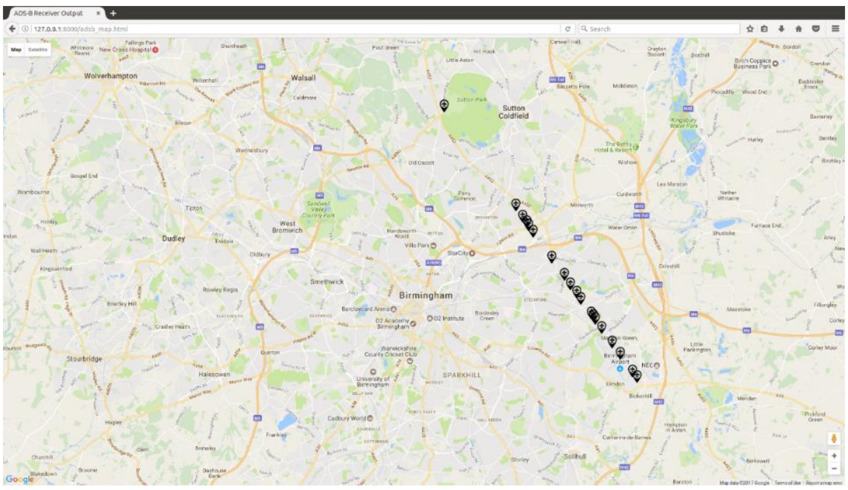



**Figure 54 - Google Maps view of received aircraft positions**

# 3 Results

## 3.1 System Alterations

During the software development phase of the project, several issues relating to the RPi became apparent. The RPi has less processing power than a typical desktop computer, which means it can take longer to accomplish some tasks. It's often necessary during software development to have several programs running at once, and whilst doing this the reduced processing power become apparent. In addition to this there are more complexities with installing software on the RPi's operating system (Jessie) than a full-fledged Linux environment. Even when programs are supported, they may not be the most up to date.

These issues were manageable, however an issue encountered with GNU Radio was not. Since GNU Radio is quite a demanding program, the reduced processing power became very apparent. GNU Radio was often unstable and exhibited unexpected behaviour, especially with graphically intense programs. None of these issues occurred when using my laptop, which I ended up using for most of the software development.

Once the OGN receiver was successfully working on my laptop, I attempted to run the program in the same simulated environment on the RPi. It failed testing. Upon further investigation, I determined that the issue was independent from my project. The simple flowgraph in Figure 55 should give identical results across different computers, which it failed to do (as can be seen in Figure 56 and Figure 57).

After attempting to troubleshoot the issue for some time, the decision was made to continue development on the laptop due to time constraints. It is likely the issue could be resolved by reinstalling or updating GNU Radio, but reconfiguring the software environment was not possible with the time that remained. However, despite the system not utilising the originally intended form factor, it is still possible to demonstrate the work that has been successfully completed.

During the project, I also decided to prioritise the SDR objectives over the AltIMU sensor board GUI. After successfully testing the functionality of the sensor board upon arrival, it was discovered that it failed to communicate a few weeks later. After unsuccessful troubleshooting, a replacement was ordered. By the time this arrived, other objectives had already taken longer than anticipated. Therefore, it was decided that time should be focused on the key objectives that form the real technical work on this project, SDR, rather than elements that are more in the realm of software engineering (like creating a GUI). Combined with the fact that the form factor of the system had been changed to a laptop, creating a GUI for this sensor board would not have been as useful as fully functioning ADS-B and OGN receivers.

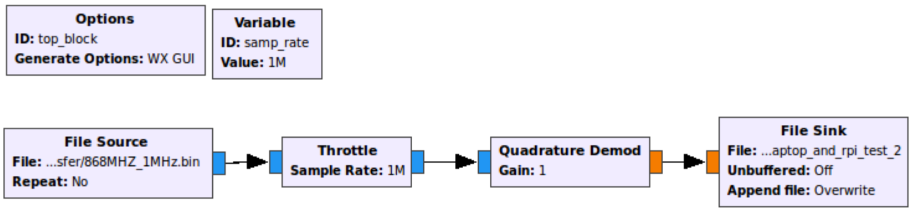



**Figure 55 - GRC flowchart for testing between the RPi and laptop computer**

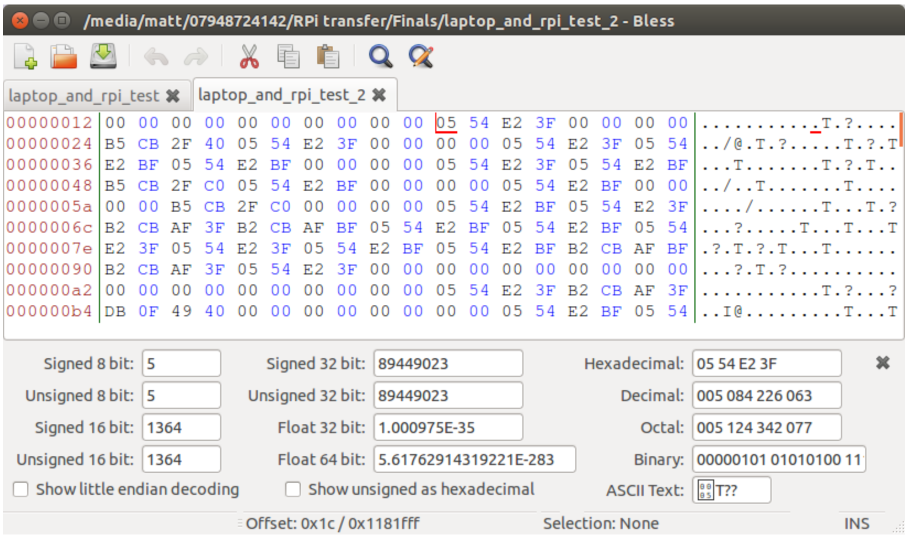



**Figure 56 - Flowgraph output on the RPi, as viewed in a hex editor**

Immediate differences can be seen between the results on the RPi and the laptop.

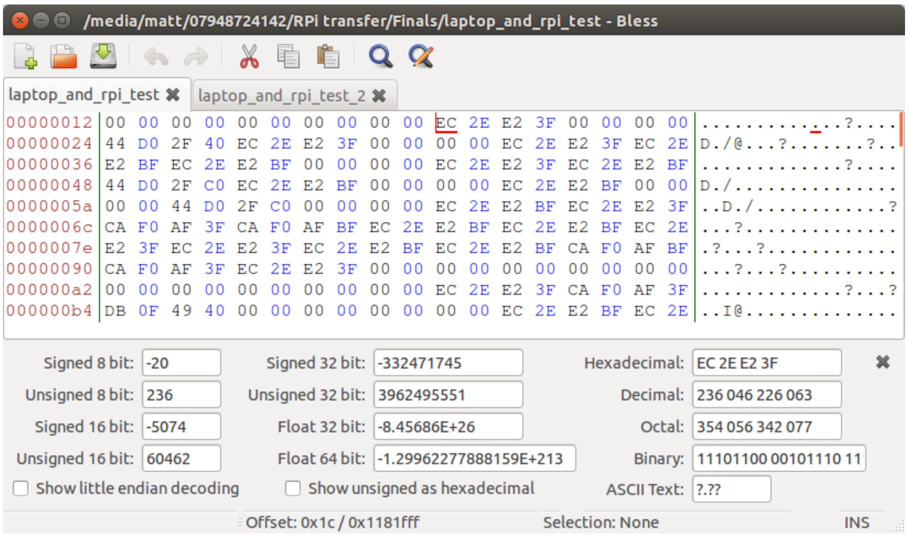



**Figure 57 - Differing flowgraph output on the laptop computer, as viewed in a hex editor**

## 3.2 ADS-B Receiver

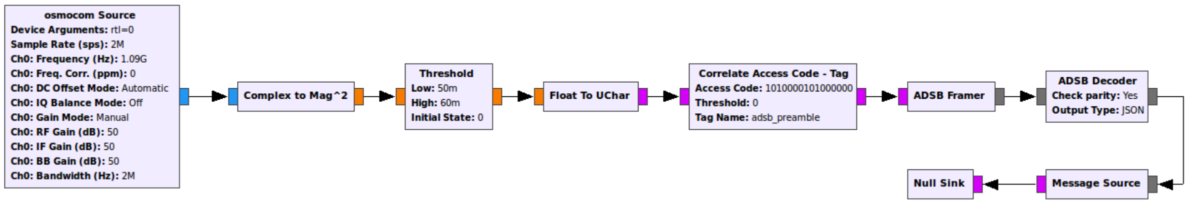



**Figure 58 - GRC flowchart for the ADS-B receiver**

### 3.2.1 Recorded Data

To have a consistent signal source to test the receiver against, I recorded a sample of IQ data that was known to contain valid ADS-B transmissions. The following section will show how the receiver goes from this raw data to a decoded transmission.

Figure 59 shows the raw IQ data in the recording. The sample was recorded using the same parameters as the hardware would normally use, i.e. a centre frequency of 1090MHz and a sample rate of 2M samples per second. As can be observed, there is quite a lot of activity on this frequency as well as a constant source of noise.

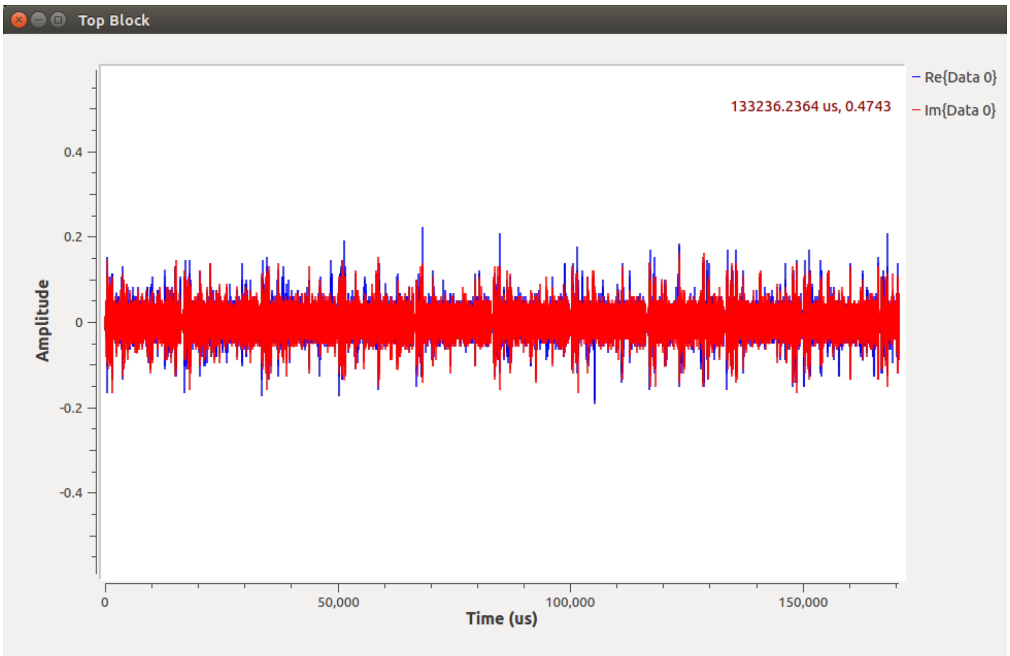



**Figure 59 - Raw input signal**

The first block in the ADS-B decoding chain is a Complex to Mag ^2 block (Figure 36). We can observe the output from this block in Figure 60. Firstly, we can see that since we are now observing the signal magnitude, we have lost the two IQ channels visible in Figure 59. We can still observe that this is quite a busy frequency, and there are several transmissions (or fragments of transmissions) visible. However, this block has managed to clear out a lot of the ambient noise.

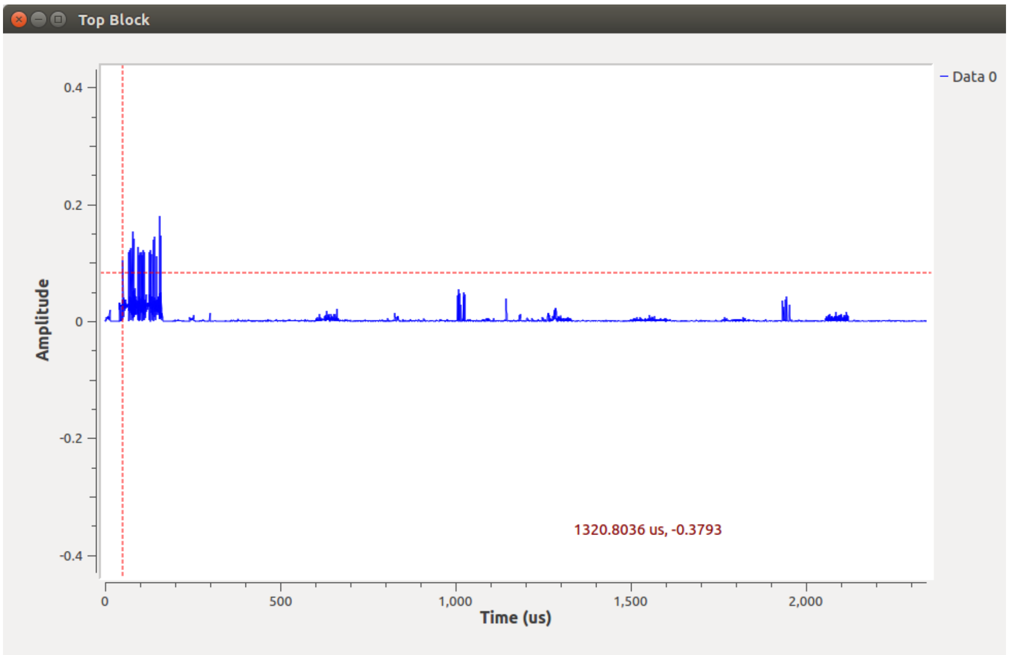



**Figure 60 - Output from Complex to Mag^2 Block**

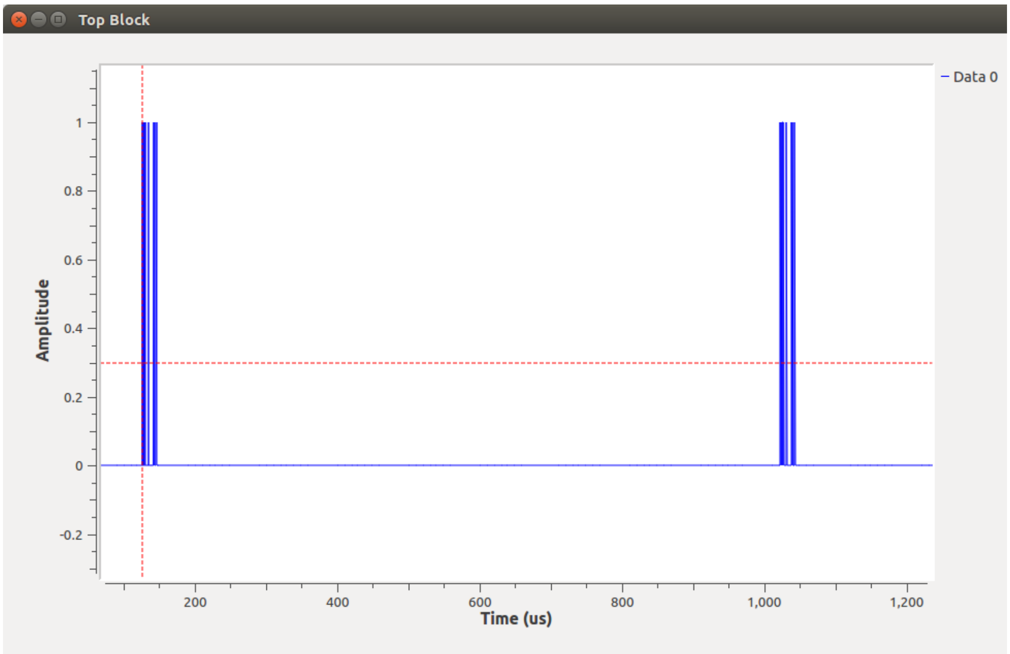



**Figure 61 - Output from the Threshold Block**

Figure 61 shows the output from the Threshold block. As previously described, this block converts the stream into a digital format. We can see that the bursts of activity previously visible now range between 0 and 1. Some care needs to be taken when setting the Threshold limits with each new installation to an appropriate level, depending upon the strength of the received signals. Figure 62 shows the output from the Float to Uchar block. We can now start to distinguish between the different width pulses.

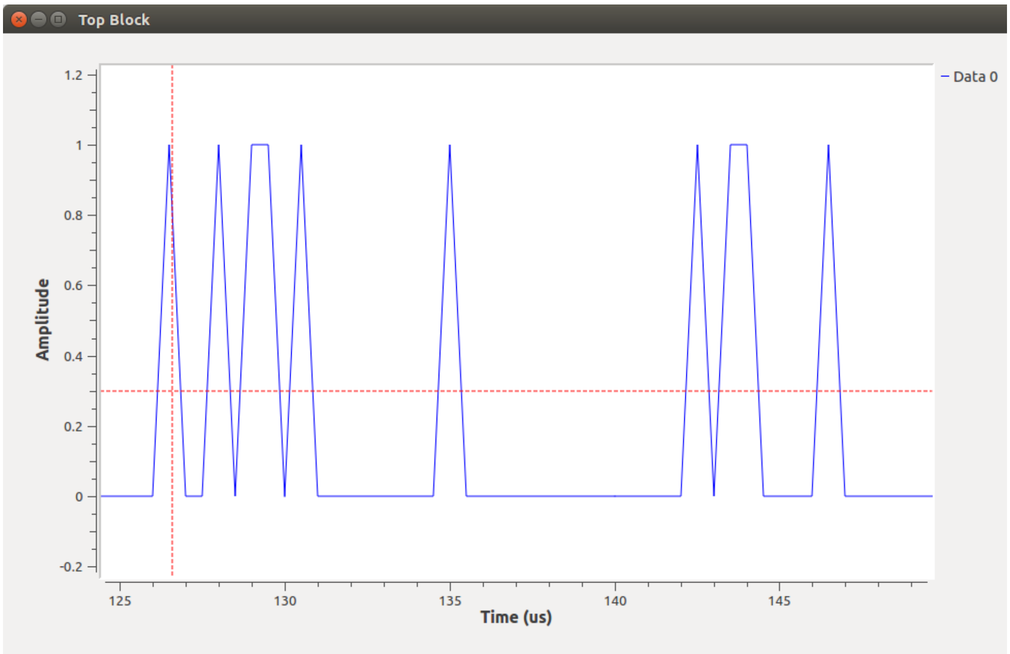



**Figure 62 - Output from the Float to Uchar Block**

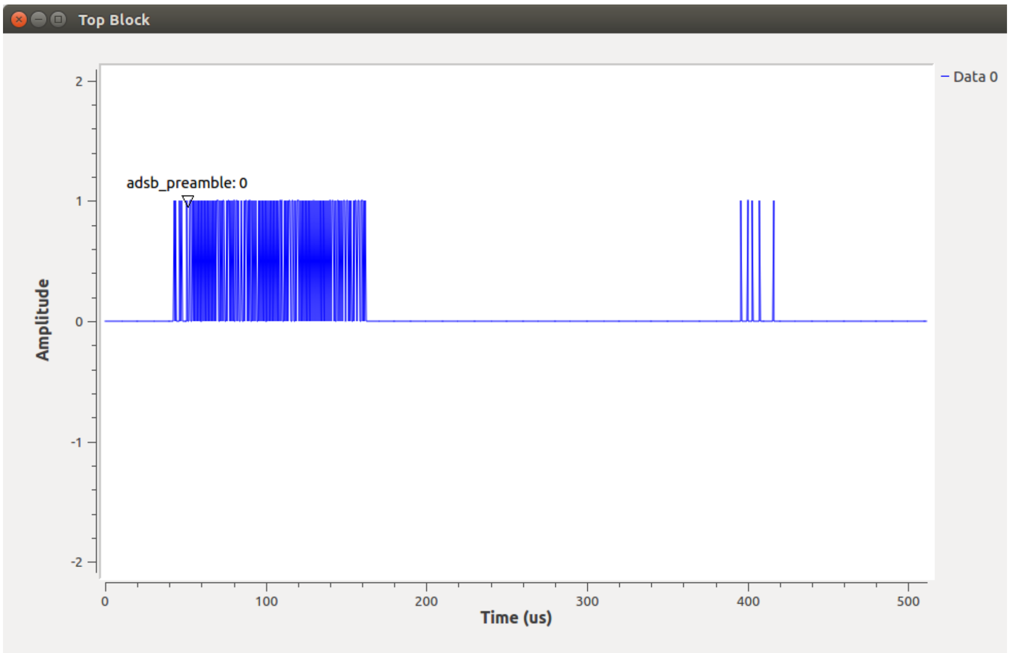



**Figure 63 - Output from the Correlate Access Code - Tag Block**

Figure 63 shows the output from the access code detecting block. It is possible to trigger the graph from the tag itself for easy viewing. We can see that one of the bursts of activity that was visible further back in the chain has been matched with the ADS-B preamble. Figure 64 shows this in closer detail, and we can clearly see the four distinct pulses that make up the preamble. The binary data following this is the data packet, of which we can see the entirety of in Figure 65. The packet is now ready to be sent into the Framing and Decoding blocks, which can interpret and output the individual packet fields.

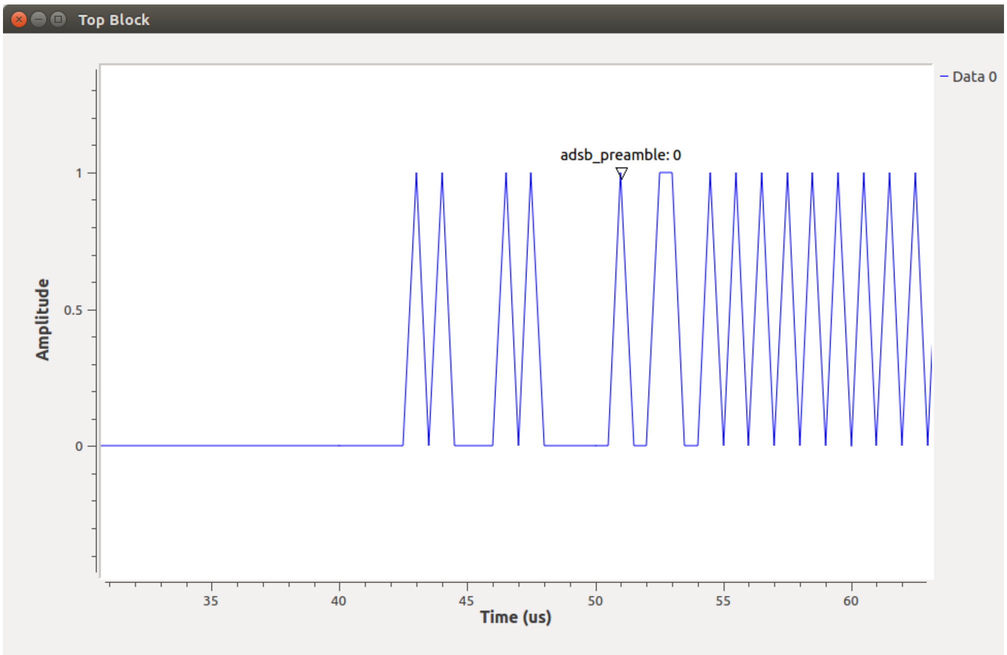



**Figure 64- ADS-B Preamble**

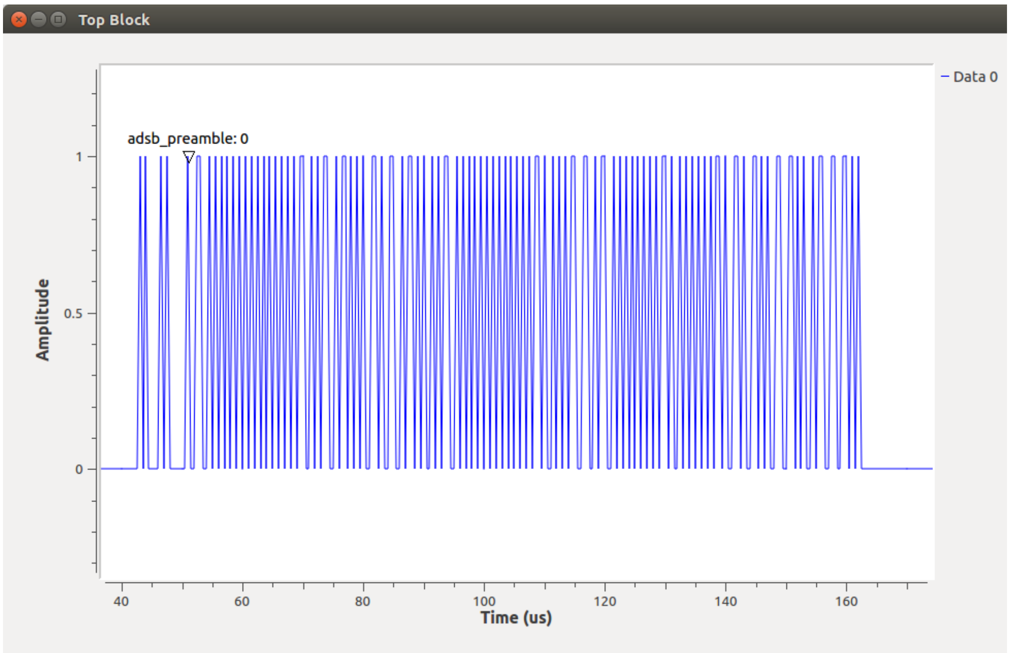



**Figure 65 - An entire ADS-B packet**

The console output for this detected ADS-B packet is:

```
2017-04-29 13:42:19: ID: A596B1 Lat: 52.61 Lon: -1.87 Src: ADS-B
```

This corresponds to a United Airlines Boeing 787-900 that was flying in the vicinity when the sample data was recorded. By viewing the Google Maps output (Figure 66), we can see that it was flying to the north of the city centre.

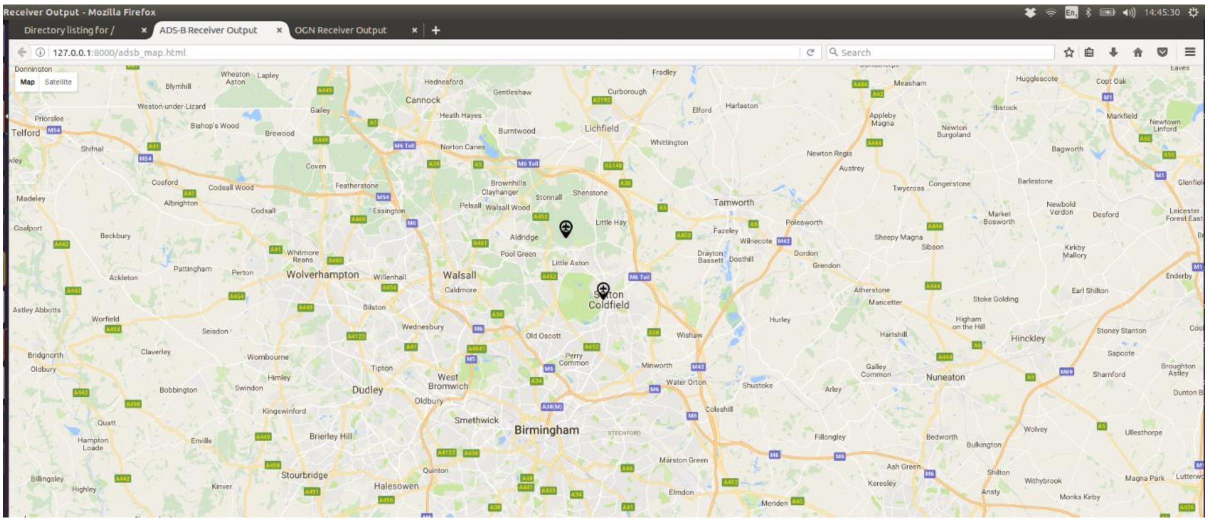



**Figure 66 - Google Maps output for the recorded ADS-B data**

### 3.2.2 Live Data

The following tests received live ADS-B transmissions from real aircraft. For all tests a 12cm antenna was used, which was mounted magnetically to a window on either the 4th floor of Aston University or a 2nd floor residence nearby. The range of the receiver was found to be around 30km (Figure 67), although this was heavily dependent on line of sight. Figure 67 shows all received transmissions from a 15-minute test. This test was performed by a north-east facing window, which explains why most received transmissions lie in that direction.

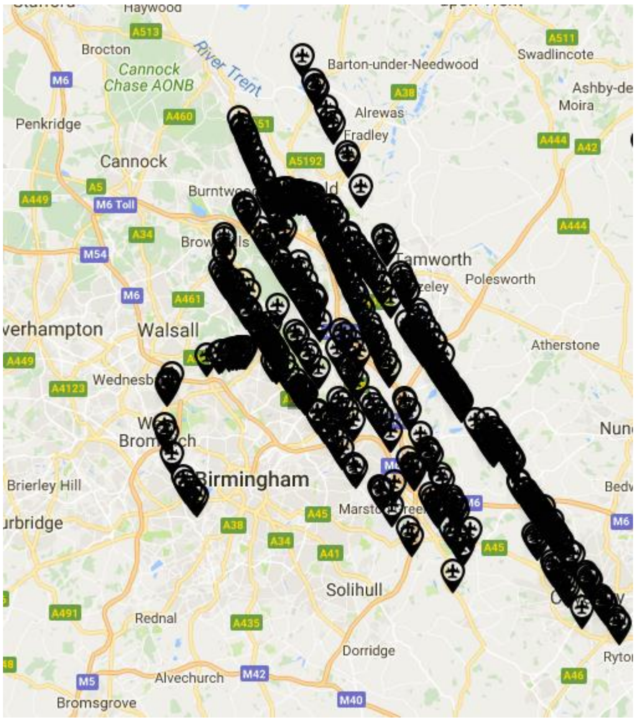



**Figure 67- All received transmissions from a 15-minute test**

I will now pick out several specific examples of received transmissions, and cross reference the data with Flightradar24.com to test the receiver's functionality.

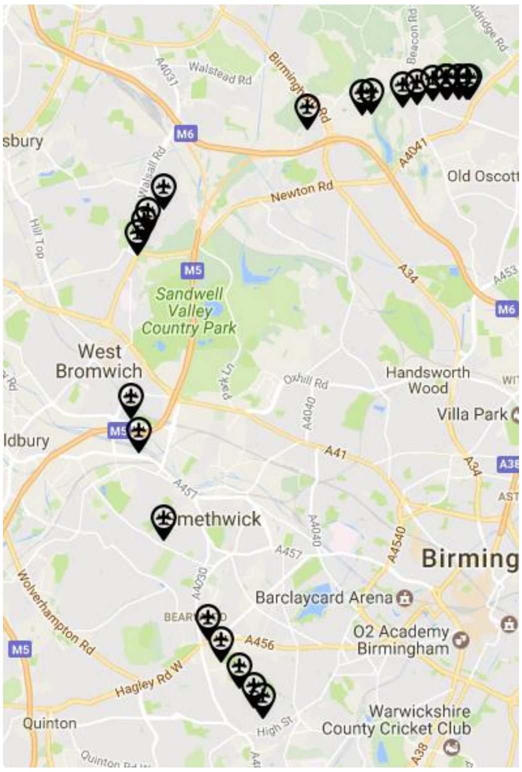



Figure 68 - Trace of flight LH2508


Figure 69 - Details of flight LH2508 [75]

ADS-B Receiver Log:

2017-05-02 08:23:21: ID: 3C66B6 Lat: 52.460358 Lon: -1.952057 Src: ADS-B

Flightradar24.com Log:

2017-05-02 LH2508 Munich (MUC) 08:58 CEST -> Birmingham (BHX) 09:30 BST

Figure 68 shows a Lufthansa Airbus A320 flying over Birmingham city centre, whilst on approach to Birmingham International airport.

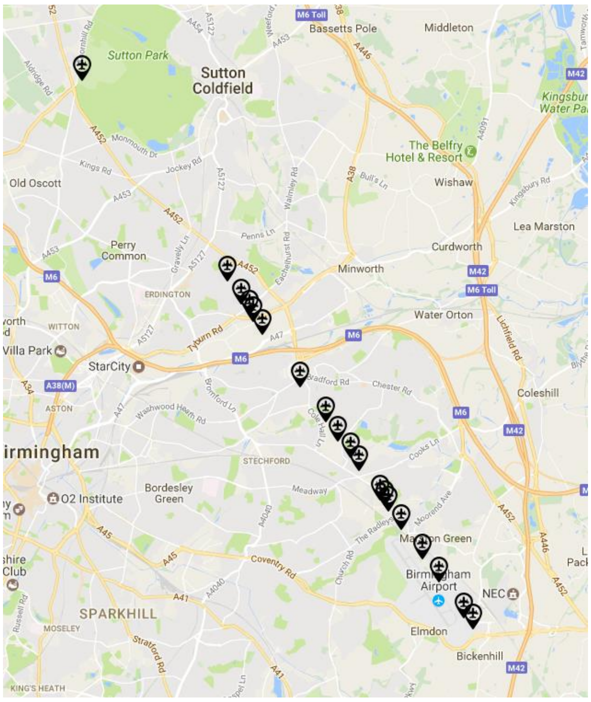



Figure 70 - Trace of flight AA51

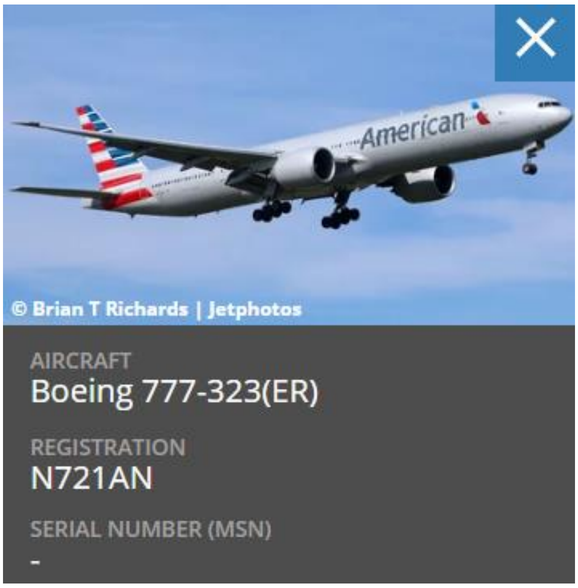



Figure 71 - Details of flight AA51 [76]

<u>ADS-B Receiver Log:</u>

2017-05-02 09:15:31: ID: 4006A9 Lat: 52.570312 Lon: -1.876526 Src: ADS-B

<u>Flightradar24.com Log:</u>

2017-05-02 AA51 London (LHR) 09:31 BST -> Dallas (DFW) 13:25 CDT

Figure 70 shows an American Airlines Boeing 777-323 east of the city, en route to Dallas.


Figure 72 - Trace of flight MT957

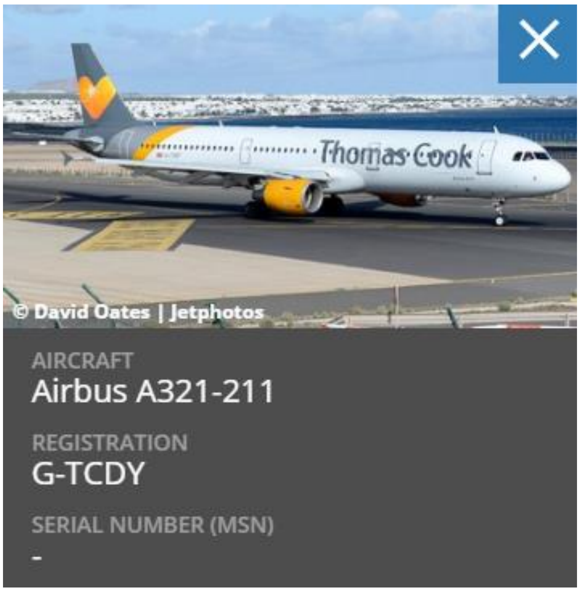



Figure 73 - Details of flight MT957 [77]

ADS-B Receiver Log:

2017-04-06 10:46:37: ID: 406CC9 Lat: 52.49 Lon: -1.79 Src: ADS-B

Flightradar24.com Log:

2017-05-02 MT957 Manchester (MAN) 09:14 -> Birmingham (BHX) 09:35

Figure 72 shows a Thomas Cook Airbus A321 over Sutton Park, on approach to Birmingham International Airport. This short flight carried no passengers, so it's likely the aircraft was being transported for scheduling or maintenance reasons.

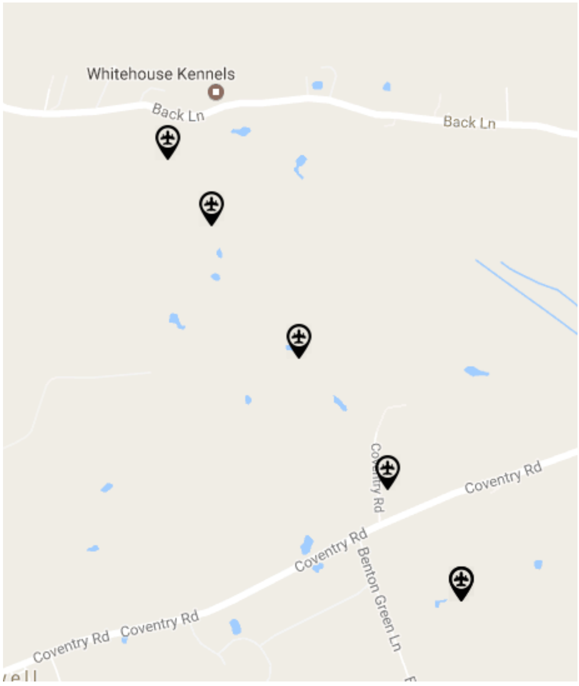



**Figure 74 - Trace of flight VS8**

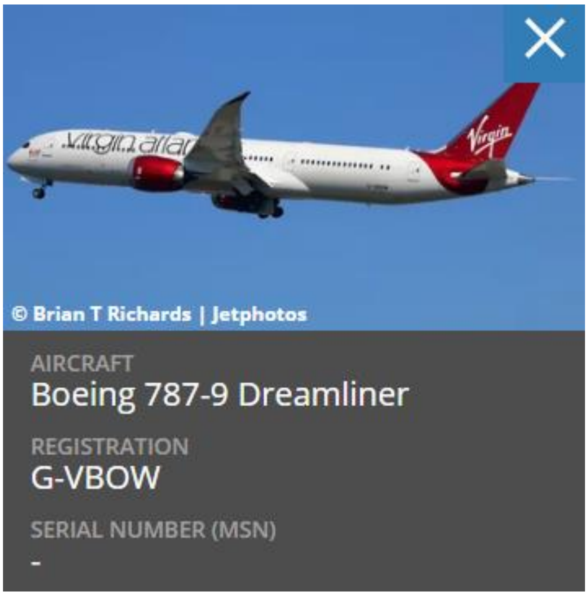



**Figure 75 - Details of flight VS8 [78]**

ADS-B Receiver Log:

2017-05-02 08:40:12: ID: 407131 Lat: 52.420532 Lon: -1.633148 Src: ADS-B

Flightradar24.com Log:

2017-05-01 VS8 Los Angeles (LAX) 16:07 PDT -> London (LHR) 10:29 BST

Figure 74 shows a Virgin Atlantic Boeing 787-9 recorded between Birmingham and Coventry, en route to London Heathrow.

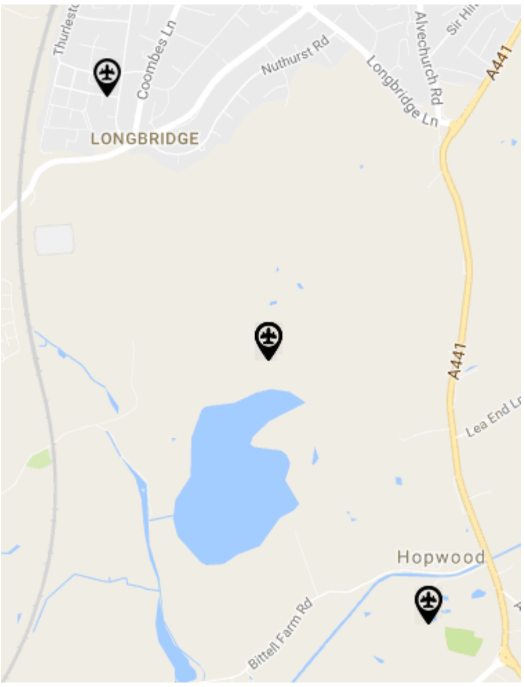



Figure 76 - Trace of aircraft N71UK


Figure 77 - N71UK [79]

ADS-B Receiver Log:

2017-04-30 14:13:53: ID: A97B80 Lat: 52.52 Lon: -1.88 Src: ADS-B

Flightradar24.com Log:

2017-04-30 N71UK / Cirrus SR22T Stornoway (SYY) 12:47 -> Chichester (QUG) 16:00

Figure 76 shows a general aviation aircraft that's equipped with an ADS-B compatible transponder. N71UK was recorded flying past Birmingham en route to Chichester on 30/04.

To establish the accuracy of the positions decoded by the ADS-B receiver, I compared the recorded trace of a flight against the exact track recorded by Flightradar24. It should be noted that Flightradar24 utilises a network of ground based ADS-B receivers, which each feed into a central network via the internet. Therefore, the positions received should be identical.

This test focused on an Avro RJ100, operated by Brussels Airlines, that landed at Birmingham International Airport on the morning of 30th April. Figure 78 and Figure 79 compare the recorded flight trace between Flightradar24 and the ADS-B receiver. As can be seen, they are identical. Flightradar24 can provide greater coverage from their network of optimised ADS-B receivers, but when the ADS-B receiver used in this project is in range of the aircraft, the decoded positions are identical. Reception is dependent upon ground objects obstructing the line of sight, but with a clear view the receiver can pick up transmissions from the aircraft until just before landing.

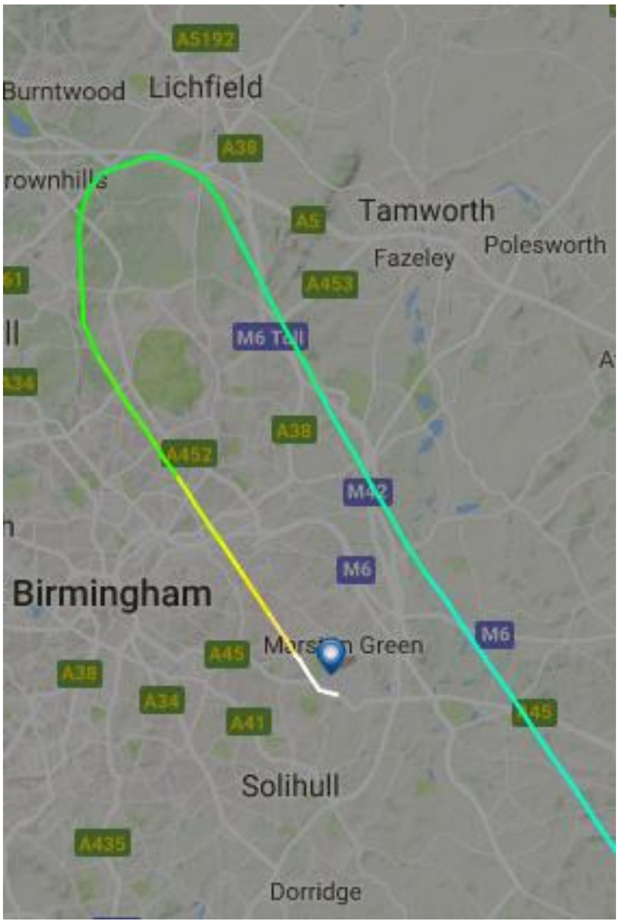



**Figure 78 - Trace of flight SN2037 from Flightradar24 [80]**

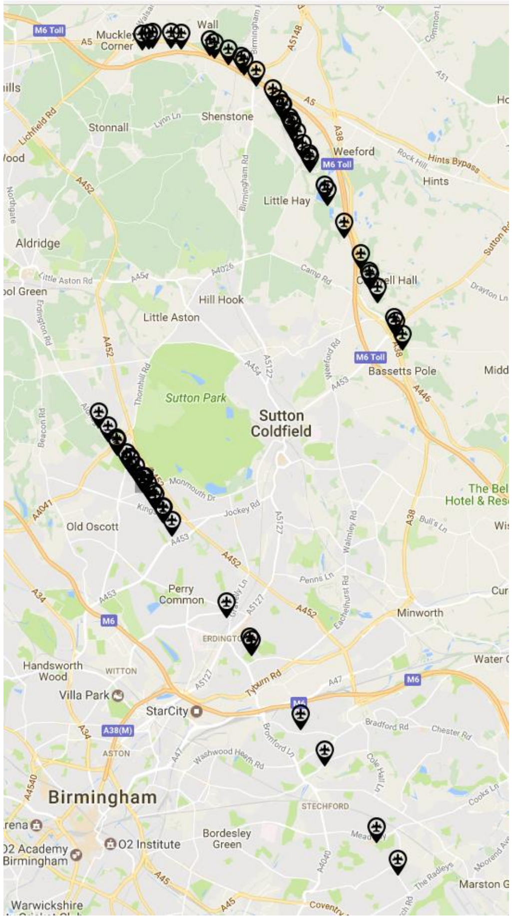



**Figure 79 - Trace of flight SN2037 as recorded by the ADS-B receiver**

<u>ADS-B Receiver Log:</u>

2017-04-30 16:06:21: ID: 4492E4 Lat: 52.644855 Lon: -1.795715 Src: ADS-B

<u>Flightradar24.com Log:</u>

2017-04-30 SN2037 Brussels (BRU) 08:27 UTC -> Birmingham (BHX) 09:20 UTC

Figure 78 and Figure 79 compare the received transmissions from flight SN2037.

## 3.3 OGN Receiver

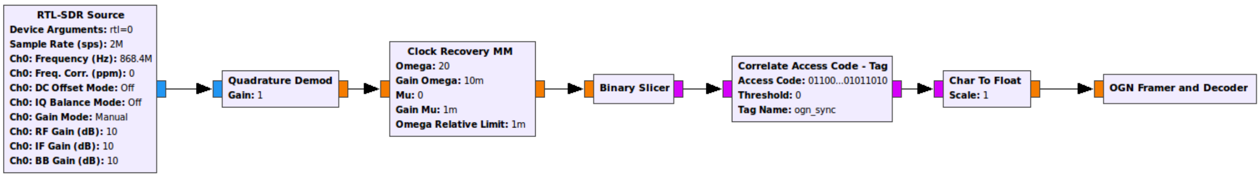



**Figure 80 - GRC flowchart for the OGN receiver**

### 3.3.1 Recorded Data

To test the OGN receiver, a recorded sample of data was also used. Since the transmitter was within the same room as the receiver, we can see that the received signal has a much higher amplitude than the background noise (Figure 81). Figure 82 shows a closer look at the raw IQ data from the recorded sample.

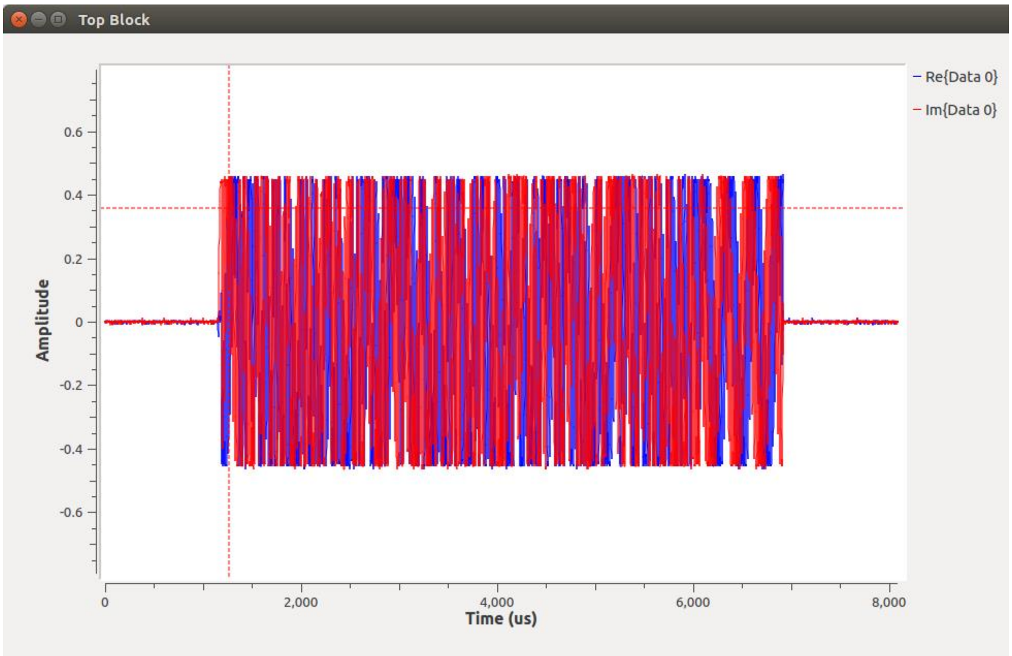



**Figure 81 - Raw signal from an OGN transmission (IQ data)**

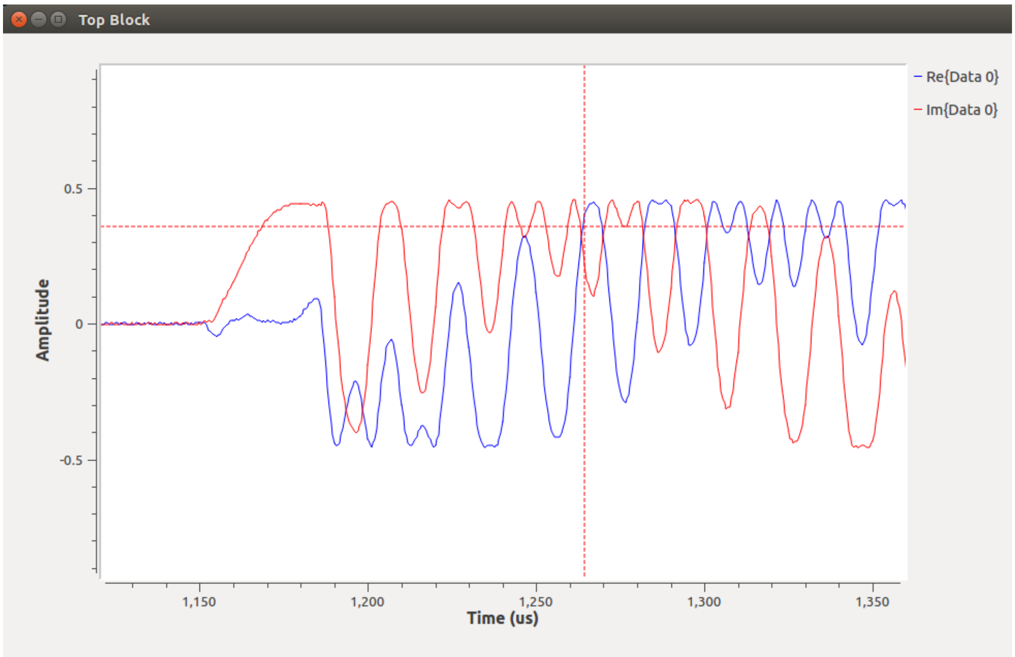



**Figure 82 - A closer look at the raw data**

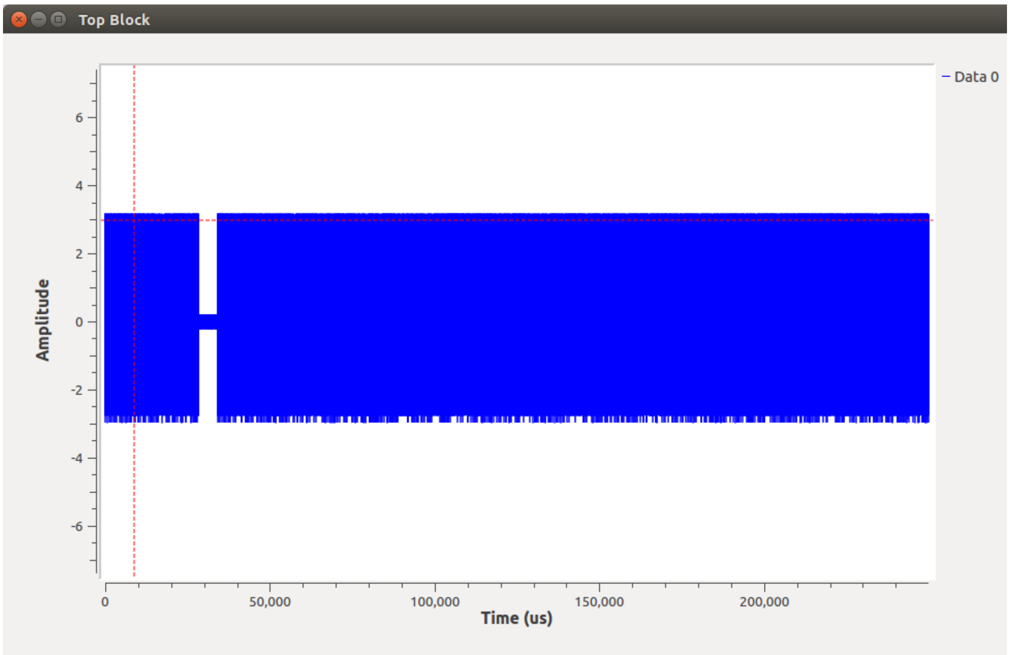



**Figure 83 - Output from the Quadrature Demodulator Block**

The next stage of the decoding chain is the quadrature demodulator, which outputs the phase difference at each point in time. A consequence of this is the magnitude of the packet ends up being significantly less than the background noise. The packet data is visible in increasingly more detail in Figure 83 to Figure 86. In Figure 86 we can begin to see the digital waveform, although in quite a noisy format.

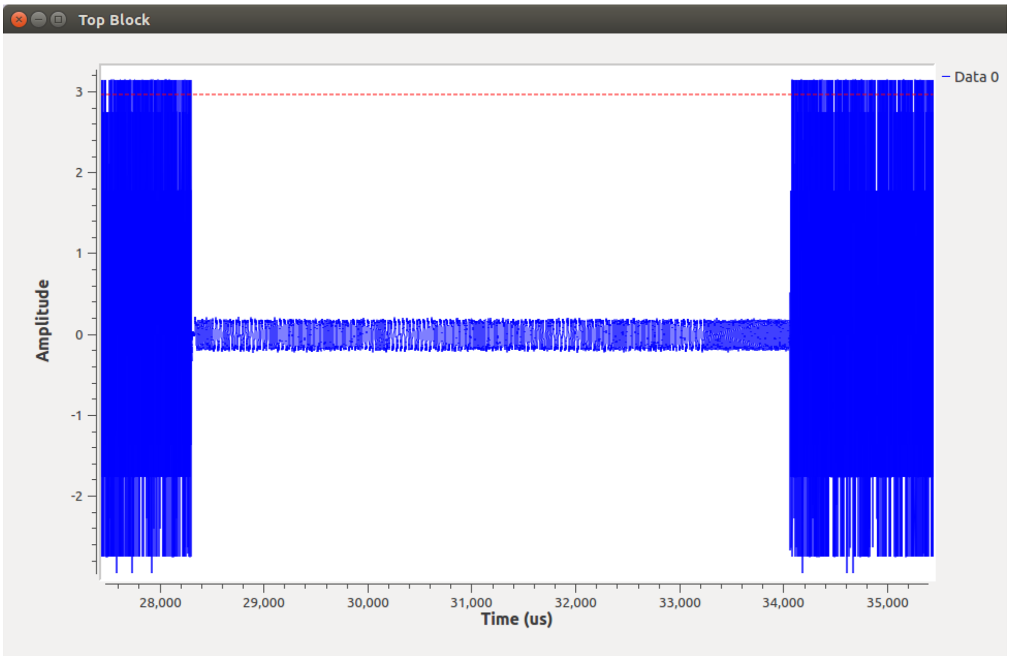



**Figure 84 - Output from the Quadrature Demodulator Block**

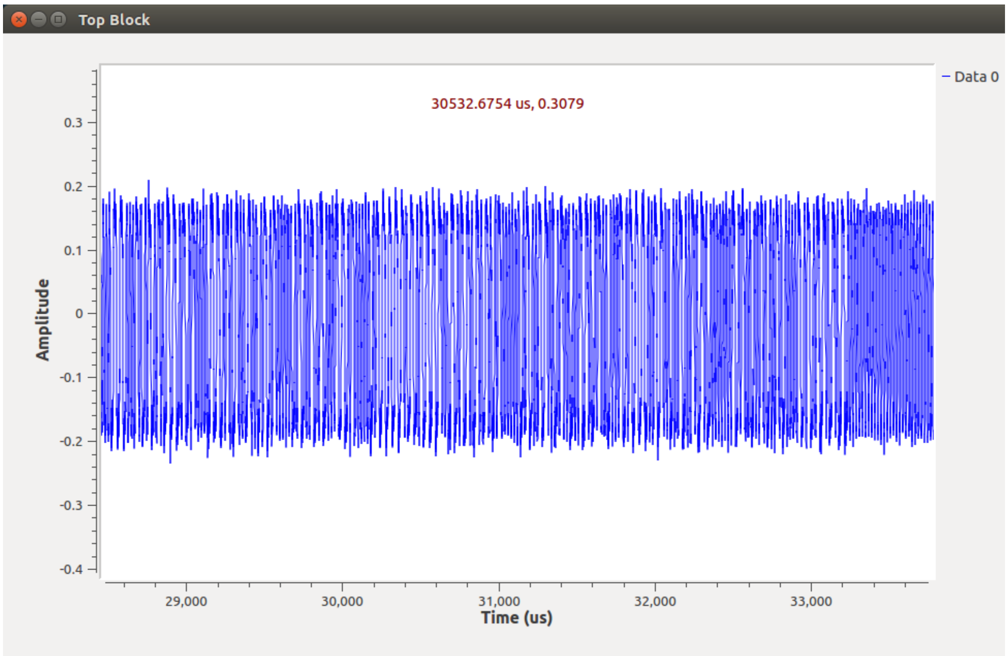



**Figure 85 - Output from the Quadrature Demodulator Block**

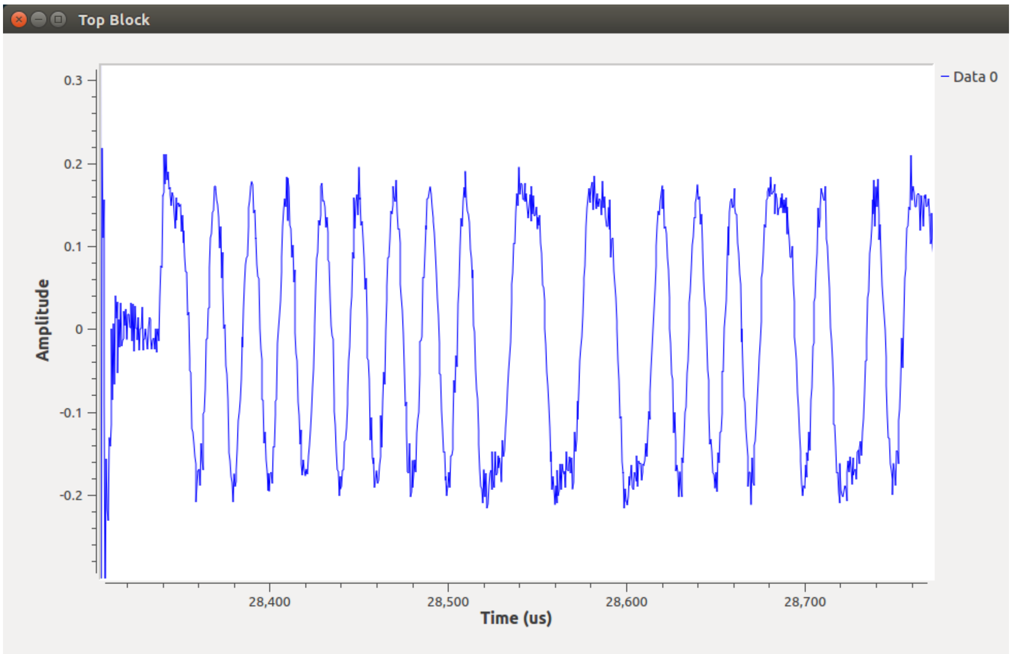



**Figure 86 - Magnified output from the Quadrature Demodulator Block**

At this stage, the signal is already clear enough to identify the preamble. This is usually a series of around 8 short pulses in a row. Figure 86 shows an example of this.

The next block in the decoding chain is the Clock Recovery block. As mentioned before, this block should ideally produce a single sample for each bit period, sampling in the centre of the period. The resulting binary output is then nearly at a stage where the information can be directly extracted. The output of the block suffers from the same effect as the quadrature demodulator block, in which the background noise has produced an output that is significantly higher in amplitude than the actual signal. Figure 87 to Figure 89 shows the output of the Clock Recovery block.

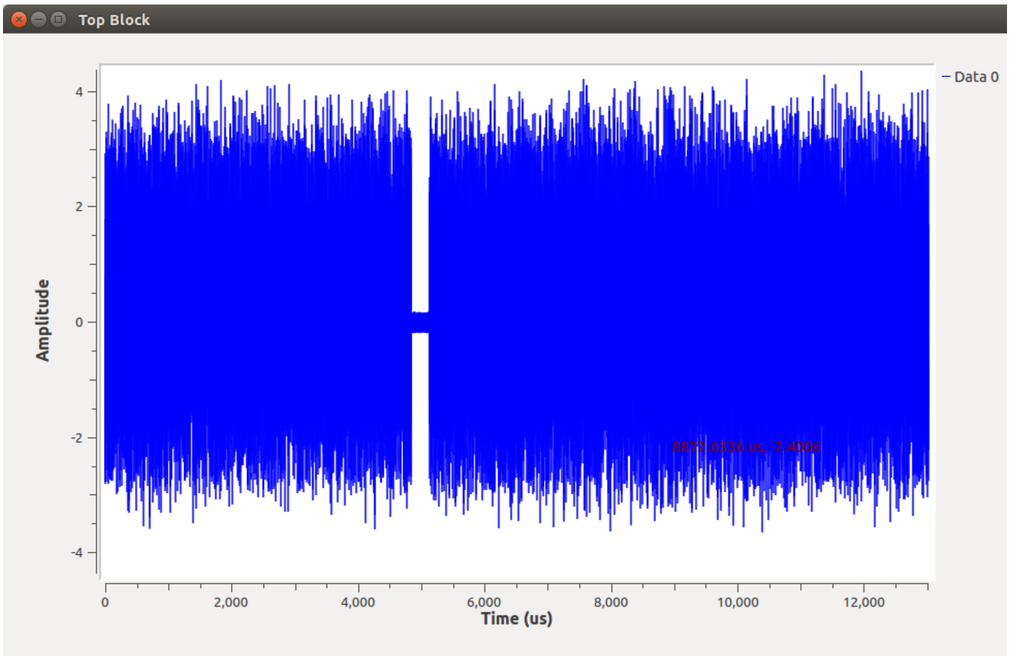



**Figure 87 - Output from the Clock Recovery Block**

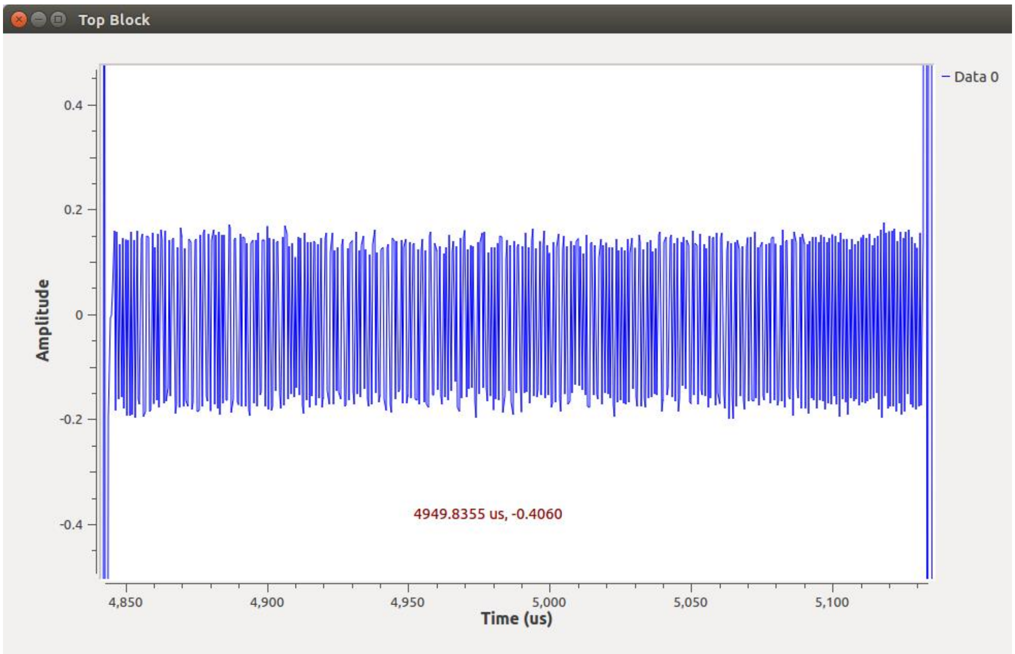



**Figure 88 - Output from the Clock Recovery Block**

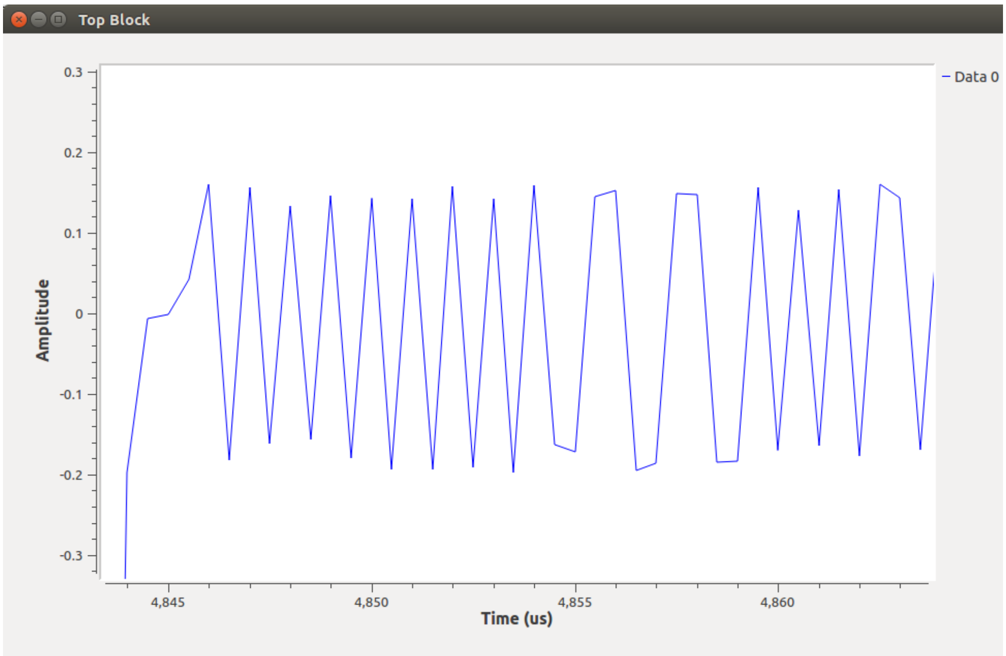



**Figure 89 - Output from the Clock Recovery Block**

Figure 89 shows the OGN preamble and the beginning of the SYNC field. At this stage, we can see that there is one sample per user bit. This is evident in the single points visible on the graph in the OGN preamble. We can also clearly identify the difference between the short and wide pulses.

Figure 90 shows the output from the Binary Slicer block. This converts the data stream into binary format. We can see that the signal now ranges between 0 and 1.

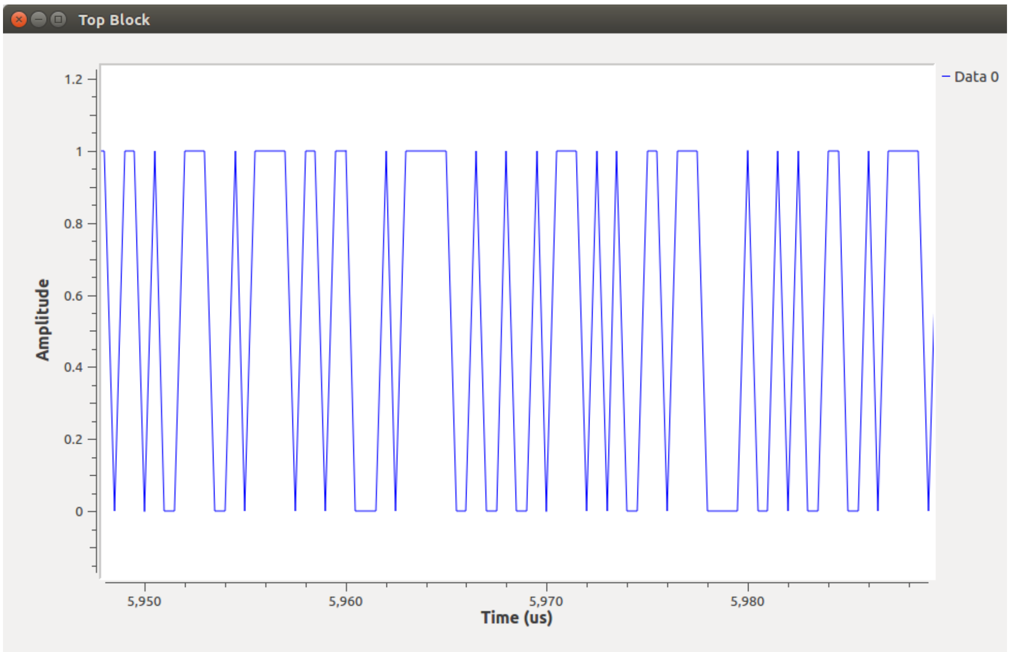



**Figure 90 - Output from the Binary Slicer Block**

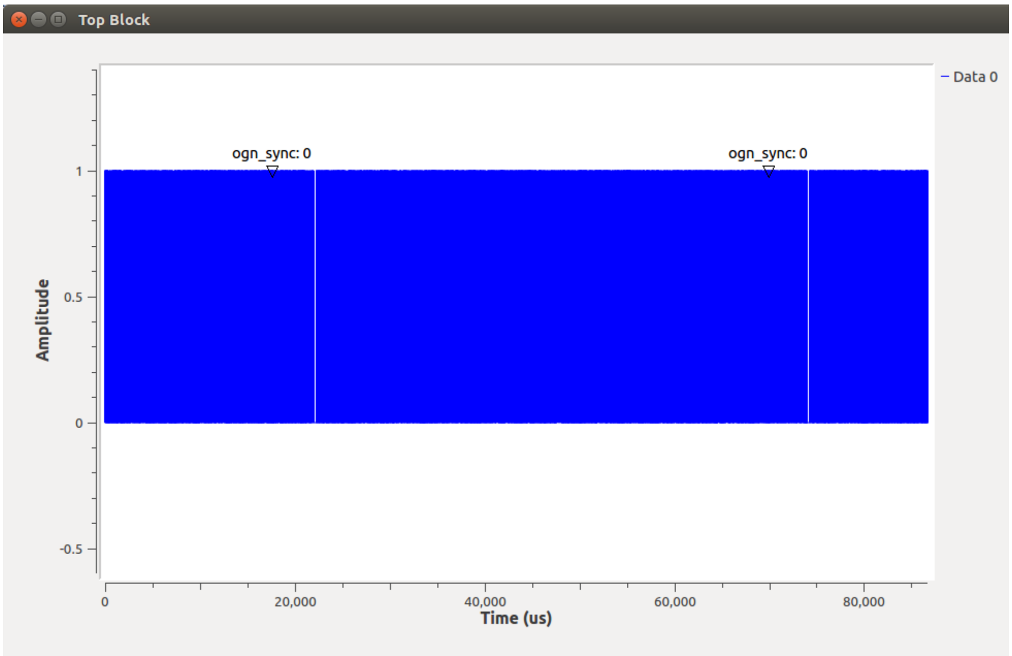



**Figure 91 - Tags inserted as the OGN SYNC pattern is detected**

The next block in the decoding chain is the Correlate Access Code - Tag Block. Now that we have the data in a suitable binary format, we can look for the specific pattern that occurs in the OGN SYNC field. Every time this block detects the 28-bit pattern, it inserts an "ogn_sync" tag. Figure 91 shows two detected transmissions. Figure 92 shows one of these packets in closer detail. A distinctive feature that allows for easy visual identification is the FEC field, contained at the end of the packet. This defines a clear boundary between the packet and the background noise.

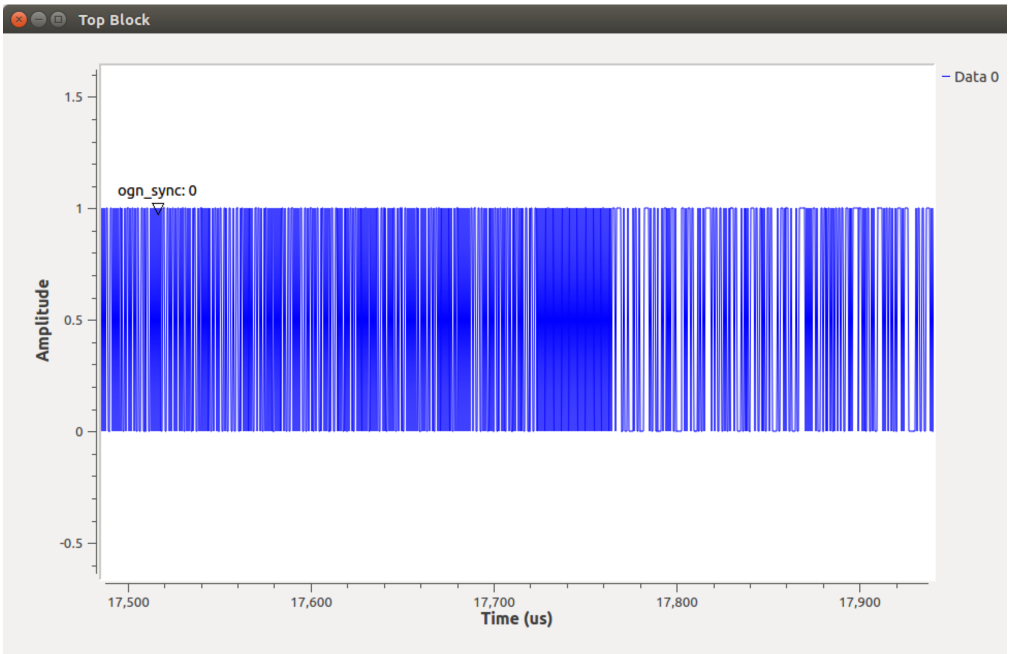



**Figure 92 - A closer look at an OGN packet**

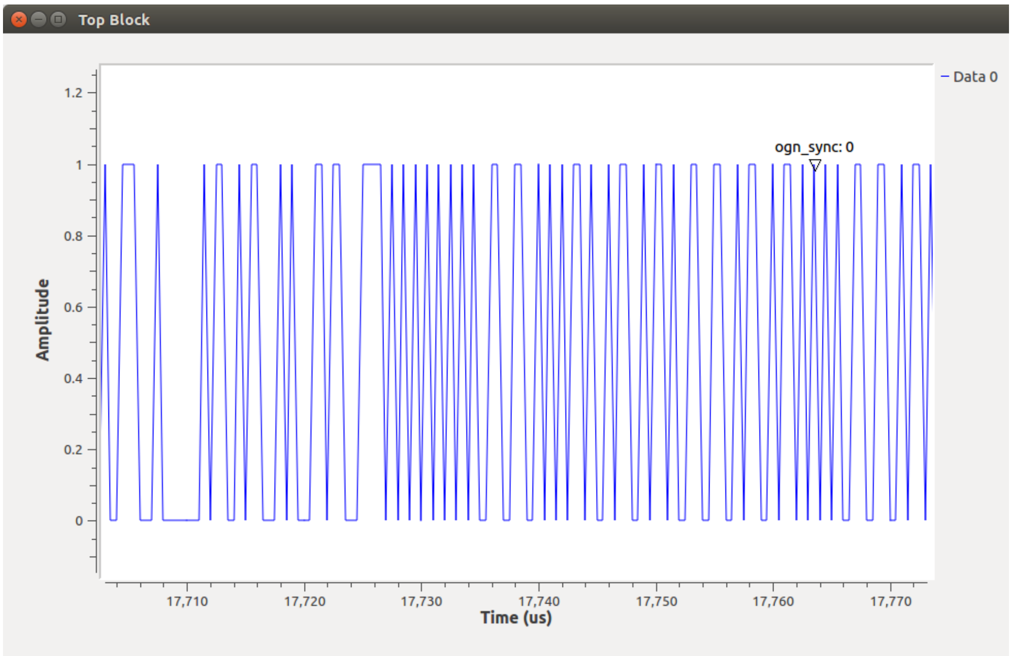



**Figure 93 - The OGN preamble and SYNC field**

Figure 93 shows the OGN preamble and SYNC header that are used to identify the start of a packet. We can clearly identify the 8 short pulses that make up the preamble, and the data between this point and the tag is the SYNC pattern.

Figure 94 shows the header and data portion of the OGN packet. This is where the actual aircraft parameters are encoded. Again, we can see that this is clearly bordered by the FEC at the end of the packet.

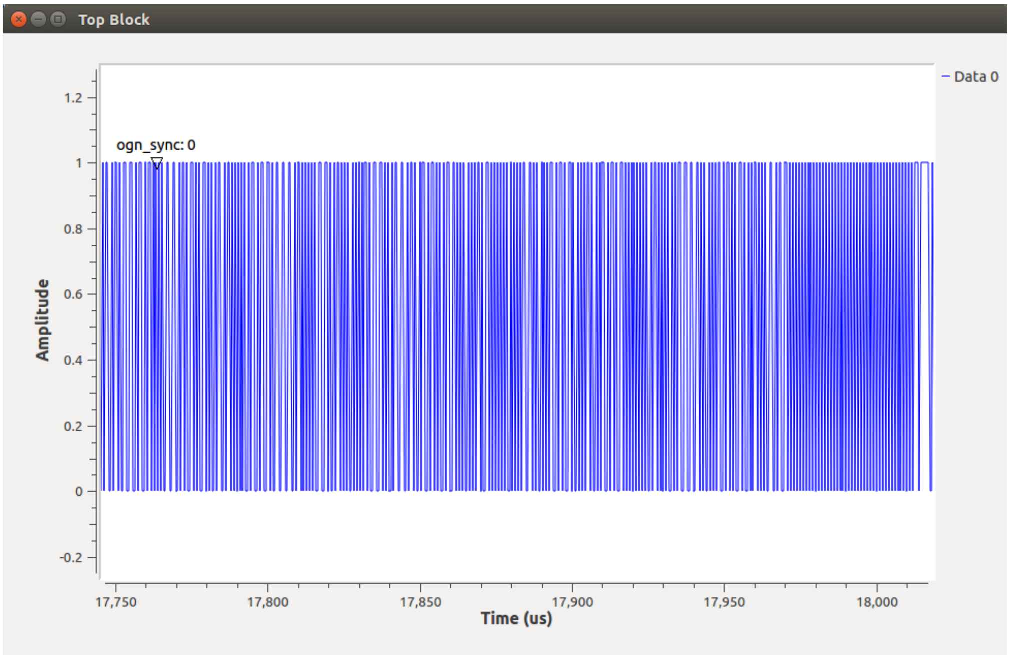



**Figure 94 - Header, data and FEC fields of an OGN packet**

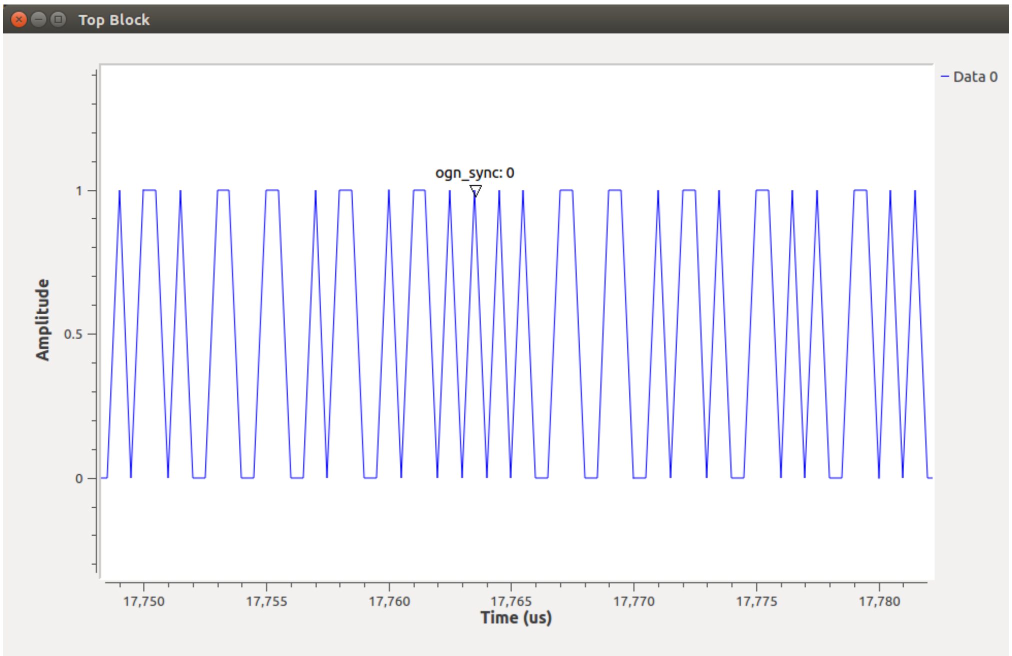



**Figure 95 - Transition between OGN SYNC and header fields**

The console output for this detected OGN packet is:

| |
|---|
| 2017-04-29 13:42:19: ID: 1b9115 Lat: 52.49 Lon: -1.89 Src: OGN |

The transmissions are consistently detected around once per second, as the transmitter is in such close proximity to the receiver. Since the transmitter did not move when this sample data was recorded, the Google Maps plot shows many points overlaid at the same location.

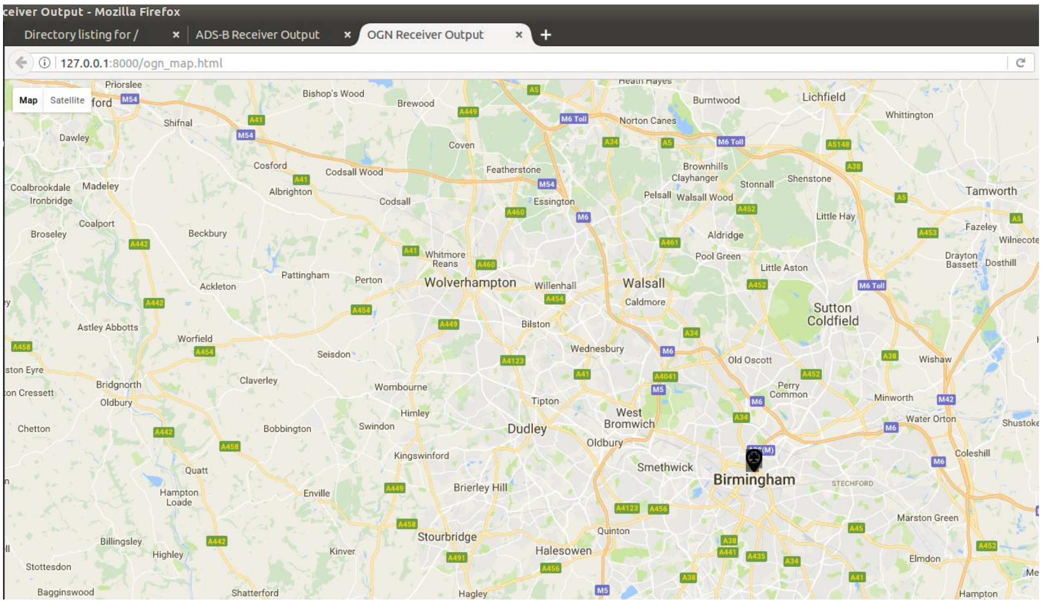



**Figure 96 - Google Maps output for the recorded OGN data**

### 3.3.2 Live Data

Test 1

Once receiver functionality was established within the same room, I began testing at greater distances. The first of these tests was across the open area in front of my house. After some experimentation, it was determined that all the antennas available gave similar performance; with the 12cm antenna giving slightly more consistent results. For the first test the antenna was placed on a chair outside my front door and attached to a metal fixture. The gain settings in the GRC RTL-SDR source block were increased to 50. I then carried the OGN tracker to the furthest point within line of sight, which was approximately 75m from the receiver. Figure 97 shows the received transmissions, with the red line indicating the actual path the tracker took.

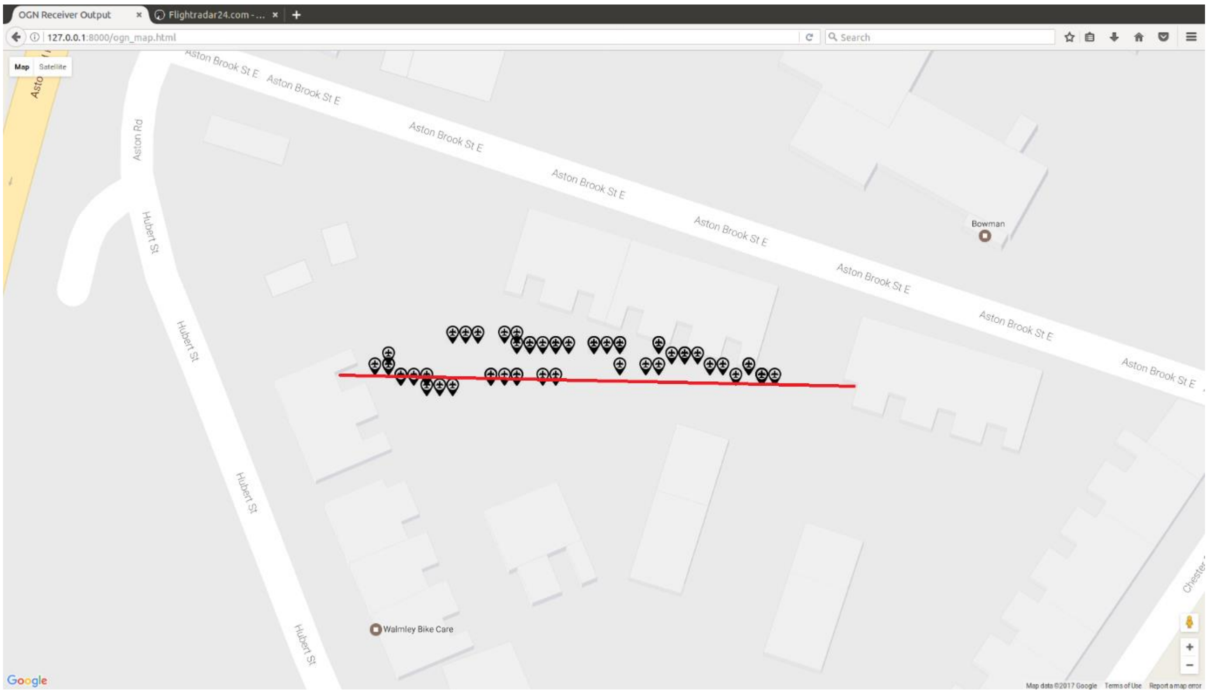



**Figure 97 - Results from Test 1**

Test 2

For the second test the antenna was placed outside my upstairs bedroom window, with the magnetic base attached to the frame. I then carried the OGN tracker on a lap around the block. Figure 98 shows the results from the receiver. The reception is lost after around 50m, although some transmissions were received despite there being no line of sight between the transmitter and the receiver.

Test 3

For the third test the antenna was kept in the same place, but the magnetic base was moved away from any metal so the antenna was hanging free. I then walked to the end of the street outside my house holding the OGN tracker. As can be seen in Figure 99 the reception appears to perform better than in the second test, with the furthest packet received from around 125m away. This indicates that the antenna is probably not optimized for this application, as such a small change had a large impact on the system performance.

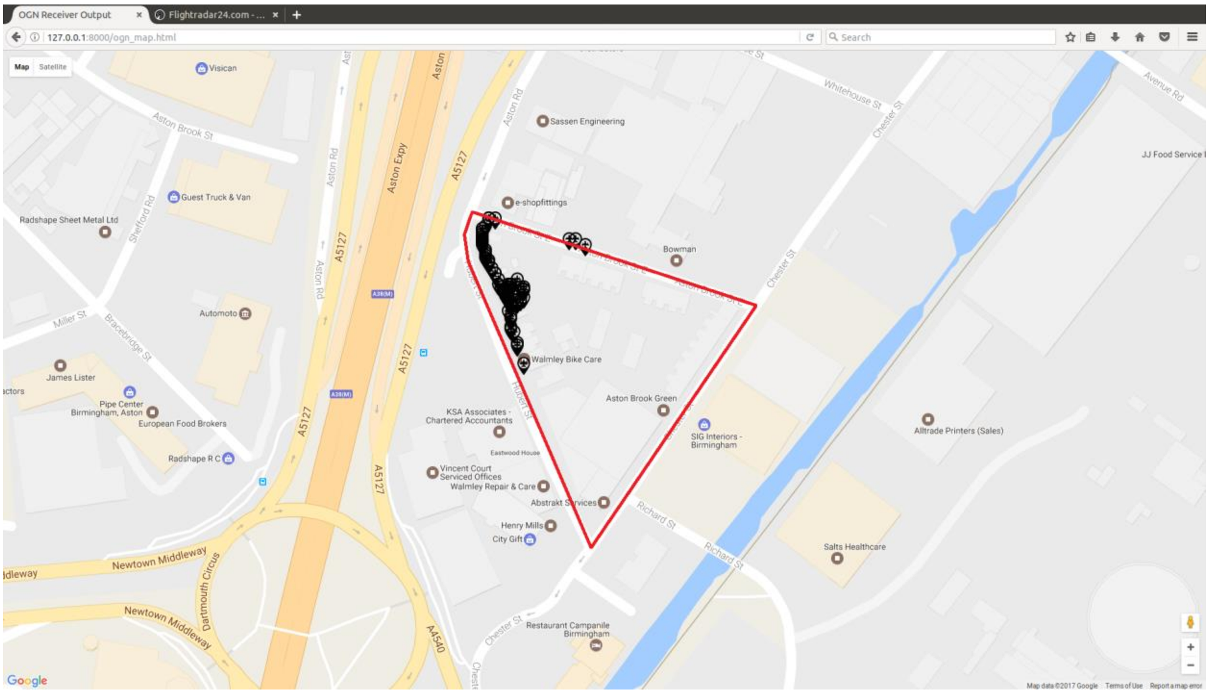



**Figure 98 - Results from Test 2**

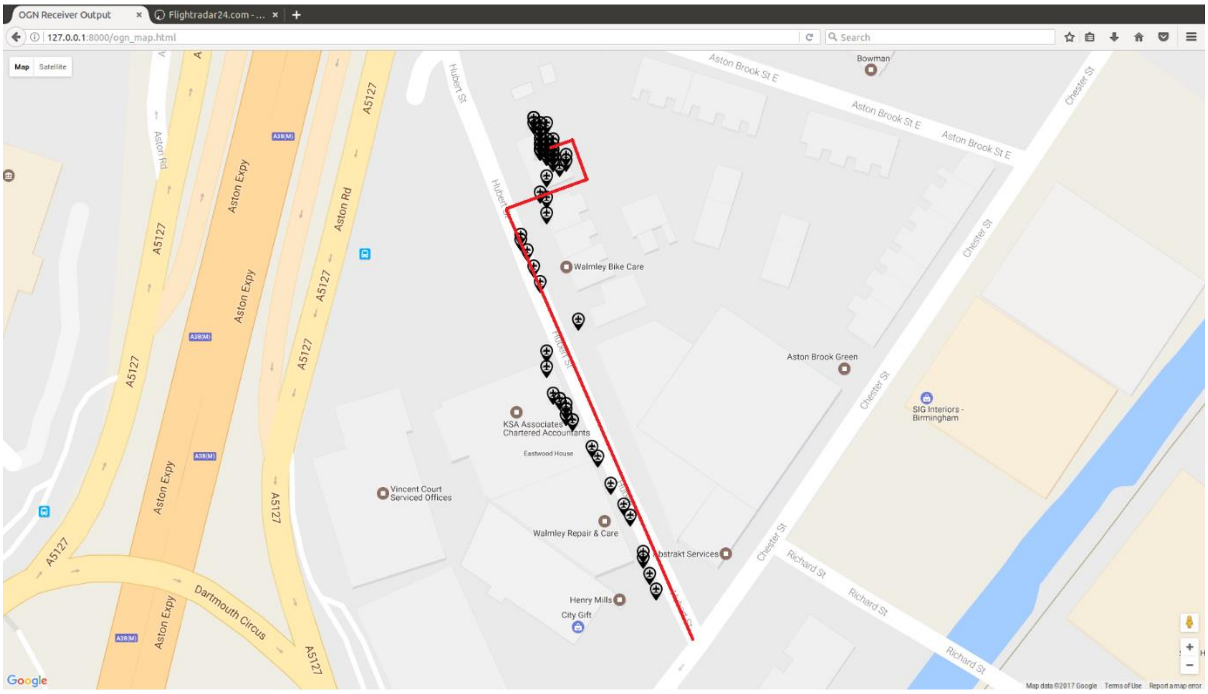



**Figure 99 - Results from Test 3**

<u>Test 4</u>

For the final test the OGN tracker was placed within a car, which was driven to the end of a straight road approximately 600m away. Line of sight was maintained throughout the test. I held the antenna in my hand, thus the magnetic base was not used. Figure 100 shows the results from this test, which again indicates that the maximum range of the receiver is around 100m.

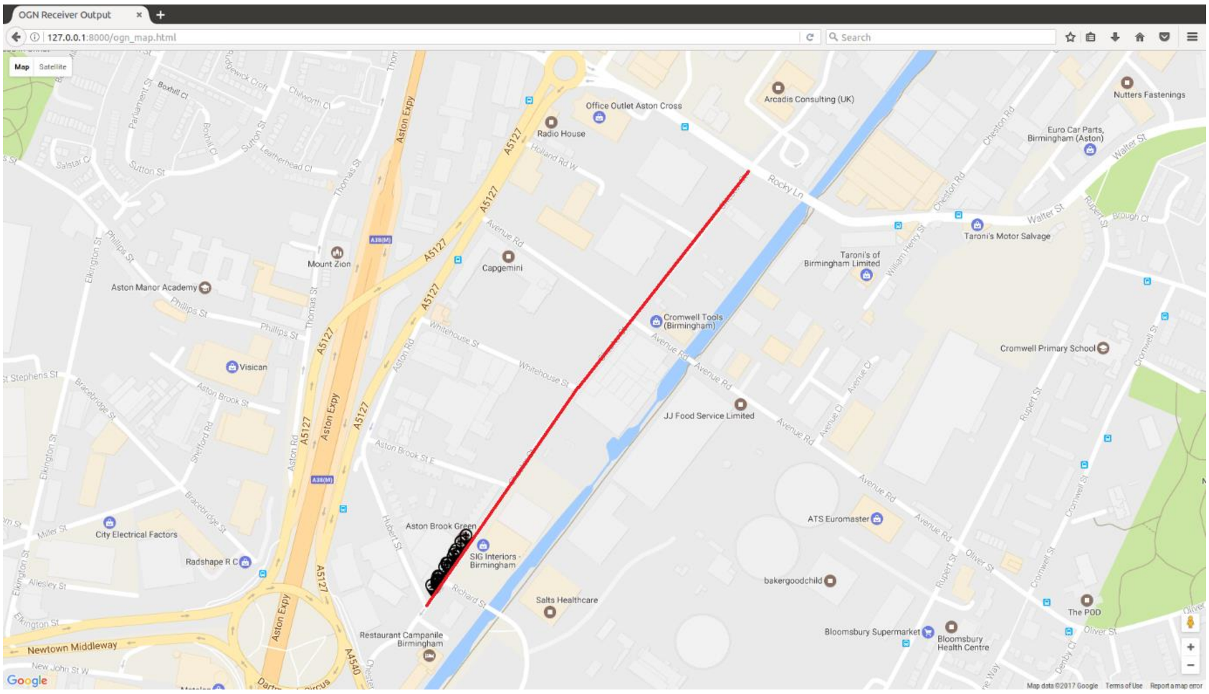



**Figure 100 - Results from Test 4**

The performance of the receiver is significantly less than that of the official OGN base stations, which reportedly have an operating range of at least several kilometres. The probable reason for this is that the receiving antenna is not optimised for this application. The transmissions are low power, therefore having a good antenna can make a big difference. Compared to ADS-B transmissions, OGN packets are 1000-10000 (30-40dB) weaker in power. There are designs available on the OGN community for constructing a high-performance collinear antenna [57]. Additionally, the receiver has no error correcting capability. There is FEC information contained in every OGN transmission that if utilised would increase reception capabilities. However, advanced fine tuning of the receiver is out of the scope of this project. The aim is not to produce a perfect receiver, but to demonstrate the key detection and decoding aspects required. With additional time, it would probably be possible to improve the performance of the receiver. For the purposes of this project these tests clearly show that both the tracker and receiver function correctly.

# 4 Analysis

## 4.1 Comparison Against Initial Design

Due to the issues faced when porting the GRC flowchart from the laptop to the RPi, the final system had some significant differences to the original design. Each decision during the project considered which objectives had the highest priority, and the feasibility of undertaking work with the amount of time remaining. We can compare the outcome against the original requirements to understand the differences:

1. The system needs to run on a portable computer.
This requirement has been met, but not in the originally intended way. The reason for this is that during the software development process, there was an issue discovered with GNU Radio on the RPi. After further investigation, it was determined that this issue was independent from my code and therefore likely to be caused by either the RPi's processing power limitations or the GNU Radio installation. Since the system is now laptop based, it can still be described as portable however it is not a form factor that can be easily used inside a small aircraft.

2. The system needs to be powered by battery for at least 3 hours.
This requirement has been met. The RPi can easily be powered for 3 hours by the Anker PowerCore. The laptop can run from its internal battery for at least 3 hours.

3. The system needs a way to allow the user to interface with the computer.
This requirement has been met. Users can interact with the RPi via the touchscreen. The laptop based system can be interacted with using the integrated keyboard and mouse.

4. I will need a signal source for the OGNTP protocol.
This requirement has been met. During the project I assembled two OGN Trackers. One of these was based on a breadboard, and the other soldered directly. The second tracker has been fitted inside a plastic container so it can be moved around outside safely.

5. The system needs to be able to receive and decode ADS-B.
This requirement has been met. The GNU Radio program outputs received ADS-B packets to the console, a log file and a JSON log. These can then be viewed using the Google Maps viewer. This section of the project was adapted from an existing project.

6. The system needs to be able to receive and decode OGN.
This requirement has been met. The GNU Radio program outputs received OGN packets to the console, a log file and a JSON log. These can be viewed using the Google Maps viewer. This section of the program is entirely original work. There are no examples of decoding OGNTP packets in GNU Radio available.

7. The system needs to display aircraft and traffic information to the user.
This requirement has been partially met. The system does display traffic information to the user. However, due to issues with the AltIMU board it was not possible to create an interface to display aircraft information to the user in the time available. The areas of the project that involved developing a SDR were deemed to have higher priority.

With more time, it is likely that the issue with the RPi could be resolved. The behaviour exhibited is not expected. GNU Radio by its very nature is an open source, community driven project. With these kinds of projects there is a vastly different level of support available compared to a commercial product. There is generally no guarantee on availability of functionality or program performance, and there are no official support channels. Work on the development of GNU Radio itself, as well as any documentation or learning resources, is largely done by volunteers. Combined with the challenges faced when working in a Linux-based environment, it can sometimes be more difficult than anticipated to develop applications. Since I could prove that the issue faced was independent of the code I had written, and confirmed through the online community that the behaviour was not expected, spending a significant amount of time troubleshooting software installation was deemed a low priority for this project.

After the delay caused by the faulty AltIMU sensor board, it was decided not to continue development with a GUI. With more time and guaranteed sensor functionality, there is no reason that a suitable interface could not be developed. However, since the core of this project was the SDR development, creating a GUI (which is more in the realm of software engineering) was deemed to be a lower priority for this project.

## 4.2 Summary of Testing

#### 4.2.1 ADS-B Receiver

As demonstrated in the Results section, the receiver can successfully detect and decode ADS-B packets. The reception frequency is dependent upon nearby traffic, but can range from one packet decoded every 10 seconds to several packets decoded per second. By comparing the data received against data published by Flightradar24, I have proved that the receiver functions correctly. The reception range was shown to be around 30km. With an optimised antenna and more robust error correction functionality, it is likely that this could be significantly increased. However, the ADS-B receiver meets the requirements set out in this project.

### 4.2.2 OGN Receiver

As demonstrated in the Results section, the receiver can successfully detect and decode OGNTP packets. With the OGN tracker in the same room it can be observed that the receiver detects around one transmission a second, which corresponds to a 100% success rate. By observing the console output, the data can be seen to be corresponding to the trackers position in real time. This indicates the receiver functions correctly. As the distance between the OGN Tracker and the receiver is increased, the number of packets detected decreased to a point where no packets were detected. This was experimentally found to be around 100m. This is significantly less than the performance of the official OGN base stations. The reasons for this include a non-optimised receiver antenna, and no error detection or correction capabilities. However, the aim for this project was to develop a working detection and decoding system, which has been thoroughly demonstrated, and not to build a highly-optimised receiver.

## 4.3 Implications

As we saw in the Introduction, there are many different systems currently available for aircraft collision avoidance. Some of these have the capability to receive or transmit across multiple protocols. However, most are incompatible with each other. Figure 101 summarises the cross-compatibility between the systems.

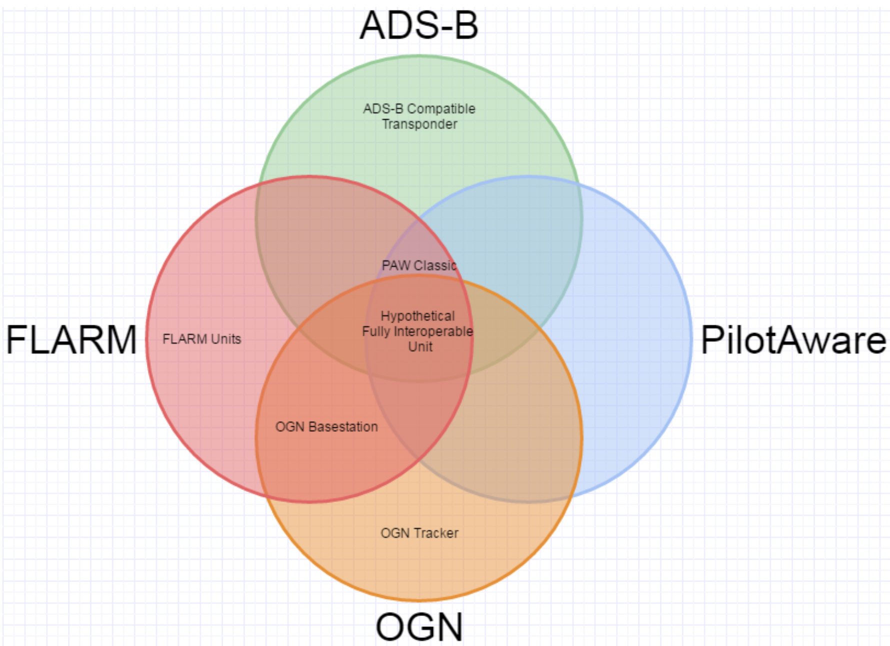



**Figure 101 - Cross compatibility between systems**

One of the requirements for this project was to develop a receiver for OGN transmissions, using GNU Radio. Now that this has been achieved, I will upload the source code for the OGN community and wider audience to utilise. This can have a real impact, because so far there are no open-source examples showing how to decode OGN transmissions. By making my OGN receiver project freely available, it can be used as a template for other developers to include OGN reception in their applications. It also serves as a useful reference for the format and structure of OGNTP transmissions, of which documentation was found to be sparse throughout the course of this project. This in turn increases the cross-compatibility of collision avoidance systems and goes some way to reducing collision risk. The full repository is publicly available on GitHub [58].

## 4.4 Current Systems and Future Developments

#### 4.4.1 FLARM

As the current market leader in collision avoidance technology for gliders, FLARM is in a stronger position than PilotAware or OGN. They have several products that require no user assembly, just installation into the aircraft. FLARM trackers have even been installed on the RAF's Battle of Britain Memorial Flight (BBMF) aircraft, as well as the Air Cadets glider fleet [59]. Their reputation for quality and reliability appears to justify the high unit cost to many users. Despite some objection from the community to their closed source nature, it's unlikely FLARM will be eclipsed by another platform in the near future.

#### 4.4.2 PilotAware

Since PilotAware has not released any usage statistics, it is difficult to gauge how widespread their devices are. Judging by online activity it is likely that there are PilotAware devices at most UK airfields. The drastically lower unit cost than FLARM is a big factor to PilotAware's success. However, the fact that the PilotAware tracker is based on a RPi, rather than bespoke hardware (like FLARM) reduces its professionalism, which might have some impact when big organisations (such as the RAF) are reviewing the options available for collision avoidance. If PilotAware could achieve a similar size user base to FLARM then it would become a real competitor. However, it will probably take a significant amount of time for this to happen.

#### 4.4.3 OGN

As a community driven project, OGN has a vastly different position to both FLARM and PilotAware. Since their software and hardware designs are developed entirely by volunteers, OGN sits most definitely on the hobbyist side of the spectrum. The fact that they openly decode FLARM transmissions and haven't faced any legal backlash is an interesting situation. There is probably an advantage for FLARM to ignore this activity, as there now exists the infrastructure (e.g. ground based receivers, servers) for FLARM equipped aircraft to be tracked across much of Europe, at no cost to FLARM themselves.

The very low-cost OGN Tracker is a versatile design that could be utilised in several applications, such as drone collision avoidance technology. The fact which applies to PilotAware is also true here, in that if OGN gained widespread usage then it could become a real competitor to FLARM. However, whilst the low unit cost creates a wider audience, the electronics experience required for tracker assembly will certainly exclude a proportion of the market.

#### 4.4.4 ADS-B in GA

If, following the NATS trial, the CAA decided to allow non-certified GPS sources to be connected to ADS-B compatible transponders, there would be significant implications for GA and the collision avoidance market. In a lot of ways, it makes sense for ADS-B to be used in GA. It is already the most used collision avoidance system in aviation worldwide, so it would follow that the best chance of having every aircraft on the same system would be to increase coverage of the leading platform. It's a well-defined standard that has regulatory approval and support from many manufacturers.

Although the current unit cost for pilots is relatively high, in the next decade this is likely to decrease as technology advances. Planning in aviation regulation must take a long-term view, since it takes a significant amount of time and resources for regulations to propagate globally and for pilots and airlines to invest in equipment. In the long-term ADS-B is likely to be the best option for uniting commercial and general aviation under the same collision avoidance technology.

## 4.5 Expansions

#### 4.5.1 Improving Reception

The antennas that were supplied with the RTL-SDR dongles are not optimised for receiving ADS-B or OGN transmissions. Detailed designs are available online for optimal antennas for both applications. Utilising these designs would certainly improve the reception range and the decoding rate.

In addition to this, there is limited error detection and correction capability present in the receivers. By making use of the error correction methods that are used by both the ADS-B and OGN protocols, the decoding rate could be increased. By including some simple checks, such as if it's feasible for an aircraft to have moved the distance decoded in the time available, obviously spurious results could be excluded.

### 4.5.2 RPi Implementation

With some additional time, it should be possible to implement the receivers on the RPi. Combining these with the GPS and AltIMU sensor boards would create a capable information system that could provide a real benefit to pilots. This project has managed to achieve the key technical objective of creating a working OGN receiver. To get this functioning on a RPi would simply be an exercise in configuring software.

### 4.5.3 Improved GUI

Now that the decoded data from both the ADS-B and OGN receivers are accessible in a plain-text format, many opportunities for utilising this data are possible. A GUI that takes the form of a radar display to show the user nearby traffic would be very useful, and it would also be possible to write collision detection algorithms to warn the user of any impending proximity risks. In addition to this, creating a GUI for a ground based station that could be installed at airfields that lack official ATC facilities would also be possible.

### 4.5.4 Increased Cross-Compatibility

This project has shown that it is possible to receive transmissions from two of the available collision avoidance systems. It should also be possible to add functionality for receiving PilotAware transmissions with the addition of another RTL-SDR device. Alternatively, a single device with sufficient bandwidth could be used to monitor multiple channels. Reception of FLARM transmissions would be possible with the addition of a FLARM mouse, or could likely be reverse-engineered (although again this bring up legality issues). Finally, utilising cheaply available RF transceiver boards (such as the HopeRF board) would add transmission capabilities.

# 5 Conclusions

Across commercial and general aviation there are multiple systems in use that aim to reduce the risk of mid-air collisions. Each of these systems uses unique transmission methods and encoding protocols, which renders the information transmitted (such as aircraft position) inaccessible to users of other systems.

This project has examined in detail the commercial and technical aspects behind each system, and has shown how they each fit into this complicated market. In addition, the project has also studied aircraft accident rates to understand how non-compatible collision avoidance systems contribute to accidents. Future trends such as drone collision avoidance technology have also been considered.

The technical aspect of this project focused on developing a SDR receiver for one of the collision avoidance technologies, the OGN Tracker. This was achieved using GNU Radio and readily-available RTL-SDR dongles. By applying this novel approach, the project has shown that adding OGN reception to an application can be inexpensive and straightforward.

By contributing the achievements of this project to the wider GA community, this project can serve as a useful example on how to decode OGN transmissions. At the time of writing there are no detailed examples or documentation on how to do this available. This contributes to the overall aim of increasing cross-compatibly between collision avoidance systems, which in turn can reduce the risk of mid-air collisions between aircraft.

# 6 References

[1] G. Topham, “Number of planes to double in next two decades, Boeing forecasts,” 11 06 2013. [Online]. Available: https://www.theguardian.com/business/2013/jun/11/boeing-commercial-planes-double-asia-pacific. [Accessed 21 04 2017].

[2] Civil Aviation Authority, “How is UK airspace structured?,” [Online]. Available: https://www.caa.co.uk/Consumers/Guide-to-aviation/Airspace/How-is-UK-airspace-structured-/. [Accessed 20 04 2017].

[3] National Air Traffic Services, “Introduction to Airspace,” [Online]. Available: http://www.nats.aero/ae-home/introduction-to-airspace/. [Accessed 20 04 2017].

[4] International Civil Aviation Organisation, Rules of the Air, 2005.

[5] LHR Airports Limited, “Facts and Figures | Heathrow,” [Online]. Available: http://www.heathrow.com/company/company-news-and-information/company-information/facts-and-figures. [Accessed 20 04 2017].

[6] Federal Aviation Authority, Introduction to TCAS II 7.1, 2011.

[7] C. C. Morris, “Midair collisions: Limitations of the see-and-avoid concept in civil aviation,” *Aviation Space and Environmental Medicine,* pp. 76(4):357-65, 2005.

[8] P. Vabre, “Primary and Secondary Radar,” [Online]. Available: http://www.airwaysmuseum.com/Surveillance.htm. [Accessed 20 04 2017].

[9] C. Allen, “IFF AND MODE 5: PAST PRESENT AND FUTURE,” Tel Instrument, [Online]. Available: https://www.telinstrument.com/avionics-news/industry-articles/101-iff-and-mode-5-past-present-and-future.html. [Accessed 20 04 2017].

[10] Y. Mansolas, “From SSR to Mode-S,” [Online]. Available: http://imansolas.freeservers.com/ATC/Mode_S.html. [Accessed 20 04 2017].

[11] C. Wolff, “Radar Basics,” [Online]. Available: http://www.radartutorial.eu/13.ssr/sr20.en.html. [Accessed 20 04 2017].

[12] “Next Generation Air Transportation System (NextGen),” FAA, [Online]. Available: https://www.faa.gov/nextgen/. [Accessed 20 04 2017].

[13] J. Law, “European ADS-B Regulation,” [Online]. Available: http://www.icao.int/APAC/Meetings/2010/adsb/sp22.pdf. [Accessed 20 04 2017].

[14] FLARM Technology Ltd, “The Affordable Collision Avoidance Technology for General Aviation and UAV,” 16 02 2016. [Online]. Available: https://flarm.com/wp-content/uploads/man/FLARM-General-EN.pdf. [Accessed 21 04 2017].

[15] Ofcom, “Short Range Devices operating in the 863 - 870 MHz frequency band,” 2010.

[16] PILOTAWARE LIMITED, “Specifications,” [Online]. Available: http://www.pilotaware.com/specifications/. [Accessed 21 04 2017].

[17] PILOTAWARE LIMITED, “PilotAwareIntroduction,” [Online]. Available: http://www.pilotaware.com/pilotawareintroduction/. [Accessed 21 04 2017].

[18] PILOTAWARE LIMITED, “P3I Protocol,” [Online]. Available: http://www.pilotaware.com/wp-content/uploads/2017/03/Protocol.pdf. [Accessed 21 04 2017].

[19] OGN, “About OGN,” 06 04 2016. [Online]. Available: http://wiki.glidernet.org/about. [Accessed 21 04 2017].

[20] OGN, “Stats - Open Glider Network,” 16 02 2017. [Online]. Available: http://wiki.glidernet.org/stats. [Accessed 21 04 2017].

[21] OGN, “Cheap Do-It-Yourself OGN tracker,” 22 06 2016. [Online]. Available: http://wiki.glidernet.org/ogn-tracker-diy. [Accessed 21 04 2017].

[22] OGN, 06 11 2016. [Online]. Available: https://www.facebook.com/OpenGliderNetwork/posts/1071765049589482. [Accessed 21 04 2017].

[23] Official Journal of the European Union, “DIRECTIVE 2013/40/EU OF THE EUROPEAN PARLIAMENT,” p. L 218/8, 2013.

[24] J. Smith, “Helping General Aviation stand out from the crowd,” 28 01 2015. [Online]. Available: http://nats.aero/blog/2015/01/helping-general-aviation-stand-crowd/. [Accessed 21 04 2017].

[25] A. Price, “Re: General Aviation ADS-B Trial in Southern England,” 27 01 2015. [Online]. Available: http://www.nats.aero/wp-content/uploads/2015/01/LPAT-Trial-v.2-Generic.pdf. [Accessed 21 04 2017].

[26] f.u.n.k.e. AVIONICS GmbH, “Avionics Engineering,” [Online]. Available: http://www.funkeavionics.de/45.html?&L=1. [Accessed 21 04 2017].

[27] NATS, “NATS to enable ADS-B transponder functionality for GA,” 28 01 2015. [Online]. Available: http://www.nats.aero/news/nats-enable-ads-b-transponder-functionality-ga-community/. [Accessed 21 04 2017].

[28] Mendelssohn Pilot Supplies, “Bendix-King KT76C Transponder,” [Online]. Available: http://www.gps.co.uk/bendix-king-kt76c-transponder.-recon/p-0-1760/?gclid=google&utm_medium. [Accessed 21 04 2017].

[29] Mendelssohn Pilot Supplies, “Trig TT31 Mode S Transponder,” [Online]. Available: http://www.gps.co.uk/trig-tt31-mode-s-transponder-unit-only/p-0-283/?gclid=google&utm_medium. [Accessed 21 04 2017].

[30] ICAO, “Planning Criteria for the assignment of SSR Mode S Interrogator Identifier (II) Codes in the AFI Region”.

[31] ADS-B Technologies, LLC, 2016. [Online]. Available: http://ads-b.com/faq-12.htm. [Accessed 21 04 2017].

[32] Navboys, “Improving Flarm coverage,” 2014. [Online]. Available: http://www.navboys.com/services/support/improving-flarm-coverage/. [Accessed 21 04 2017].

[33] L. Moore, “Transmit power and range,” 30 09 2015. [Online]. Available: http://forum.pilotaware.com/index.php?topic=105.0. [Accessed 21 04 2017].

[34] M. Jenkins, “Open Glider Network Range,” [Online]. Available: http://ognrange.onglide.com/#,max,all,,,#00990000:#009900ff,circles;. [Accessed 21 04 2017].

[35] UK AIRPROX BOARD, “Analysis of Airprox in UK Airspace,” 2015.

[36] AOPA Air Safety Institute, “25TH JOSEPH T NALL REPORT GENERAL AVIATION ACCIDENTS IN 2013,” 2013.

[37] NATS / CAA, “Drone code,” [Online]. Available: http://dronesafe.uk/drone-code/. [Accessed 21 04 2017].

[38] BBC, “Drone 'narrowly avoids' plane collision over Hertfordshire,” 12 02 2017. [Online]. Available: http://www.bbc.co.uk/news/uk-england-beds-bucks-herts-38906501. [Accessed 21 04 2017].

[39] BBC, “Prince William's air ambulance in near-miss with drone,” 26 03 2017. [Online]. Available: http://www.bbc.co.uk/news/uk-england-cambridgeshire-39398409. [Accessed 21 04 2017].

[40] BBC, “Drone in 'near miss' over Birmingham Airport,” 27 01 2017. [Online]. Available: http://www.bbc.co.uk/news/uk-england-birmingham-38774718. [Accessed 21 04 2017].

[41] BBC, “Passenger jet approaching Heathrow in drone 'near-miss',” 31 03 2017. [Online]. Available: http://www.bbc.co.uk/news/uk-england-london-39457371. [Accessed 21 04 2017].

[42] BBC, “An air traffic control system... for drones,” 10 04 2017. [Online]. Available: http://www.bbc.co.uk/news/business-39530054. [Accessed 21 04 2017].

[43] ST, “STM32F103C8,” [Online]. Available: http://www.st.com/en/microcontrollers/stm32f103c8.html. [Accessed 21 04 2017].

[44] HopeRF, “RFM69W ISM TRANSCEIVER MODULE,” 2006. [Online]. Available: http://www.hoperf.com/upload/rf/RFM69W-V1.3.pdf. [Accessed 21 04 2017].

[45] GNU Radio, [Online]. Available: https://www.gnuradio.org/about/. [Accessed 28 04 2017].

[46] S. Markgraf, “steve-m/librtlsdr,” [Online]. Available: https://github.com/steve-m/librtlsdr. [Accessed 25 04 2017].

[47] E. S. Raymond, “gpsmon,” [Online]. Available: http://catb.org/gpsd/gpsmon.html. [Accessed 25 04 2017].

[48] T. Flanagan, “pynmea2,” [Online]. Available: https://github.com/Knio/pynmea2. [Accessed 25 04 2017].

[49] “RTIMULib2,” [Online]. Available: https://github.com/RTIMULib/RTIMULib2. [Accessed 25 04 2017].

[50] OGN, “Raspberry Pi Installation,” 03 02 2017. [Online]. Available: http://wiki.glidernet.org/wiki:raspberry-pi-installation. [Accessed 25 04 2017].

[51] S. Sanfilippo, “dump1090,” [Online]. Available: https://github.com/antirez/dump1090. [Accessed 25 04 2017].

[52] STMicroelectronics , “FLASHER-STM32,” [Online]. Available: http://www.st.com/en/development-tools/flasher-stm32.html. [Accessed 25 04 2017].

[53] “What is the different GMSK and GFSK?,” 2003. [Online]. Available: http://www.edaboard.com/thread5788.html. [Accessed 26 04 2017].

[54] OGN, “OGN Tracking Protocol (OGNTP),” 08 09 2015. [Online]. Available: http://wiki.glidernet.org/ogn-tracking-protocol. [Accessed 28 04 2017].

[55] N. Bonniere, “OGNTP Packet Structure,” 29 03 2017. [Online]. Available: https://groups.google.com/d/msgid/openglidernetwork/0BAE103C2A5F4729846CD8D1DD41D9FD%40SR2180nx?utm_medium=email&utm_source=footer. [Accessed 28 04 2017].

[56] W. Nagele, "GNU Radio ADSB decoder and framer," [Online]. Available: https://github.com/wnagele/gr-adsb. [Accessed 28 04 2017].

[57] P. Jałocha. [Online]. Available: http://openglidernetwork.wdfiles.com/local--files/links/collinear_antenna.pdf. [Accessed 01 05 2017].

[58] M. Snowdon, "gr-ogn," 07 05 2017. [Online]. Available: https://github.com/mattsnow/gr-ogn. [Accessed 07 05 2017].

[59] RAF, "Track our Aircraft Live!," 12 09 2014. [Online]. Available: http://www.raf.mod.uk/bbmf/news/index.cfm?storyid=122DB3D3-5056-A318-A8946DB778012B50. [Accessed 01 05 2017].

[60] BBC, [Online]. Available: http://news.bbcimg.co.uk/media/images/63563000/jpg/_63563648_atc.jpg. [Accessed 20 04 2017].

[61] NASA, [Online]. Available: https://asrs.arc.nasa.gov/publications/directline/dl4_tcas_unit.gif. [Accessed 20 04 2017].

[62] [Online]. Available: http://www.artofthestate.co.uk/photos/Heathrow_radar_tower.jpg. [Accessed 20 04 2017].

[63] [Online]. Available: http://landoflinux.com/images/linux_dump_1090_interactive_01.png. [Accessed 21 04 2017].

[64] PILOTAWARE LIMITED, [Online]. Available: http://pilotawarehardware.com/wp-content/uploads/2016/08/PAW6006001.jpg. [Accessed 21 04 2017].

[65] OGN, [Online]. Available: http://openglidernetwork.wdfiles.com/local--files/start/OGN_Arch.png. [Accessed 21 04 2017].

[66] A. S. A. A. A. M. T. K. H. B. S. Abdulrazaq Abdulaziz, "Optimum Receiver for Decoding Automatic Dependent Surveillance Broadcast (ADS-B) Signals," *American Journal of Signal Processing,* 2015.

[67] S. Schmidt, "Manchester encoding both conventions," 27 09 2006. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/9/90/Manchester_encoding_both_conventions.svg. [Accessed 26 04 2017].

[68] Airservices Australia, "How ADS-B works," 18 09 2015. [Online]. Available: http://www.airservicesaustralia.com/projects/ads-b/how-ads-b-works/. [Accessed 26 04 2017].

[69] J. Sun, “ADS-B Decoding Guide,” TuDelft , 2017.

[70] Ktims, “File:Fsk.svg,” 15 03 2006. [Online]. Available: https://commons.wikimedia.org/wiki/File:Fsk.svg. [Accessed 26 04 2017].

[71] osmocom, [Online]. Available: https://github.com/osmocom/gr-osmosdr. [Accessed 28 04 2017].

[72] E. M. Danielle Coffing, “A Quadrature Demodulator Tutorial,” EE Times, 06 04 2001. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1275839. [Accessed 28 04 2017].

[73] T. Šolc, “NOTES ON M&M CLOCK RECOVERY,” 12 03 2015. [Online]. Available: https://www.tablix.org/~avian/blog/archives/2015/03/notes_on_m_m_clock_recovery/. [Accessed 28 04 2017].

[74] G. Maxson, “ADS-B for General Aviation,” [Online]. Available: http://adsbforgeneralaviation.com/new-home-page/. [Accessed 01 05 2017].

[75] flightradar24, “Playback of Lufthansa flight LH2508,” [Online]. Available: https://www.flightradar24.com/data/flights/lh2508/#d40a201. [Accessed 02 05 2017].

[76] flightradar24, “Playback of flight AA51,” [Online]. Available: https://www.flightradar24.com/data/aircraft/n721an/#d40c3ba. [Accessed 02 05 2017].

[77] flightradar24, “Playback of Thomas Cook Airlines flight MT957,” [Online]. Available: https://www.flightradar24.com/data/flights/mt957/#d40b98e. [Accessed 02 05 2017].

[78] flightradar24, “Playback of Virgin Atlantic Airways flight VS8,” [Online]. Available: https://www.flightradar24.com/data/flights/vs8/#d3fdab4. [Accessed 02 05 2017].

[79] flightradar24, “Playback of flight / N71UK,” [Online]. Available: https://www.flightradar24.com/data/aircraft/n71uk/#d3bda85. [Accessed 02 05 2017].

[80] flightradar24, “Playback of Brussels Airlines flight SN2037,” [Online]. Available: https://www.flightradar24.com/data/flights/sn2037/#d40c371. [Accessed 02 05 2017].

[81] Central Connecticut State University, [Online]. Available: http://chortle.ccsu.edu/AssemblyTutorial/Chapter-15/bigLittleEndian.gif. [Accessed 04 05 2017].

# 7 Appendices

## 7.1 Appendix A

**Table 6 - Airprox Category Descriptions**

| Risk Category | UK Airprox Board Risk Descriptor |
|---|---|
| A | **Providence - serious risk of collision.** Situations where separation was reduced to the bare minimum and/or which only stopped short of an actual collision because chance played a major part in events: the pilots were either unaware of the other aircraft or did not/could not make any inputs in time to materially improve matters. |
| B | **Safety much reduced / not assured.** Situations where aircraft proximity resulted in safety margins being much reduced below the norm either due to serendipity, misjudgement, inaction, or where emergency avoiding action was taken at the last minute that materially increased separation and averted a likely collision. |
| C | **Safety degraded - no risk of collision.** Situations where safety was degraded but either fortuitous circumstances or early enough sighting, information or action allowed one or both of the pilots to either simply monitor the situation or take timely and effective avoiding action to prevent the aircraft from coming into close proximity. |
| D | **Non-assessable - insufficient, inconclusion or irresolvable information.** Situations where insufficient information was available to determine the risk involved, or |

| | |
|---|---|
| | inconclusive/conflicting evidence precluded such determination. |
| E | **Non-proximate - benign.**<br>Situations that met the criteria for reporting but where the occurrence was in fact benign and normal procedures, safety standards and parameters were considered to have pertained. |

## 7.2 Appendix B

**Table 7 - RPi specifications**

| Product: | Recommended PSU current capacity: | Maximum total USB peripheral current draw: | Typical bare-board active current consumption: |
|---|---|---|---|
| Raspberry Pi 3 Model B | 2.5A | 1.2A | ~400mA |

**Table 8 - Estimated system current consumption**

| Product: | Current Consumption (mA): | Conditions: |
|---|---|---|
| Raspberry Pi 3 Model B | 400 | Typical bare-board active current consumption |
| 7" Touchscreen Display | 640 | Peak current consumption (screen driven black) |
| RTL SDR USB | 290 | R820T dongle typical operation |
| VK16U6 GPS Module | 55 | Typical operation |
| AltIMU-10 v4 | 6 | Typical operation |
| Total | 1391 | Estimated maximum current draw |

**Table 9 - Anker PowerCore specifications**

| Product: | Capacity: | Output: | Input: |
|---|---|---|---|
| Anker PowerCore 20100 | 20100mAh | 5V / 4.8A (max 2.4A per USB port) | 5V / 2A (supports standard USB ports) |

$$\frac{Battery\ Capacity\ (mAh)}{System\ Current\ Consumption\ (mA)} = \frac{20100}{1391} = \sim 14\ hours$$