Mode-S Receiver and ADS-B Decoder

Date 12/4/2014
Sponsor: Boeing

Group 24 - ***Sand5***
Michael Vose
Sean Koceski
Long Lam

Table of Contents	Page

1. Executive Summary

National airspace systems (NAS) around the world are undergoing infrastructure upgrades to enhance or replace aging radar systems. The new technology is Automatic Dependent Surveillance – Broadcast (ADS-B). ADS-B is based on the Global Positioning System (GPS) and is a key element of the U.S. Federal Aviation Administration's (FAA) Next Generation Air Transportation System. By using ADS-B, an aircraft determines its position by GPS coordinates and periodically broadcasts it, enabling it to be tracked by air traffic control ground stations. However, ADS-B data is not encrypted and represents an exposure of highly precise and potentially exploitable commercial aircraft location information [McCallie].

Military aircraft transponders and ground stations already implement an encrypted digital protocol to address this concern referred to as Mode-5. The application of encryption (where law and treaty permit) to Mode-S will protect the location data of such aircraft from being misused. To that end, this project will simulate the broadcast of an encrypted ADS-B transponder in software, as an actual ADS-B broadcast would be forbidden. Bypassing our Mode-S receiver, our enhanced ADS-B decoder would then decrypt the signal and demonstrate the receipt of the simulated communication. Meanwhile a standard ADS-B decoder would not be able to determine the aircraft's location.

The functional stages of the programmable Mode-S receiver and ADS-B decoder for this project are illustrated in figure-1. In general, the ADS-B data stream is captured on two frequencies and digitally merged. This result is transferred to an Android smartphone via USB or wireless protocol. The decrypted result is shown on the smartphone.

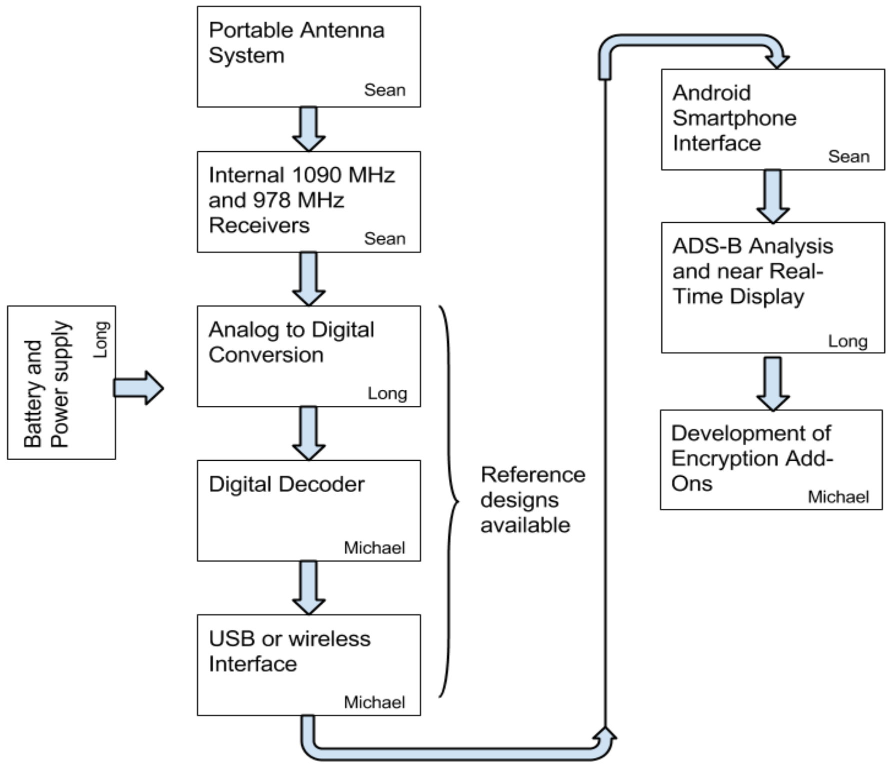


Figure 1 – Functional Stages

2. Project Description

2.1. A Global Motivation

Automatic Dependent Surveillance – Broadcast (ADS–B) is a key element of the U.S. Next Generation Air Transportation System. By using ADS-B, an aircraft determines its position by GPS coordinates and broadcasts it every second, enabling it to be tracked by air traffic control ground stations that are listening to ADS-B. This digital signal can also be received by other ADS-B equipped aircraft to provide situational awareness and help to avoid collisions in the air and on the runway. In general, ADS-B is a significant improvement over the traditional primary and secondary radar systems in use until now. It costs much less to implement and operate and it does the job faster, with greater precision, and with little restriction to the placement of ground stations. ADS-B promises even more benefits in the future. Once everyone is using ADS-B, the current 80-miles-apart safety margin for aircraft in flight can be reduced. This will result in significant fuel savings, fewer airport delays and fewer diverted flights, [McCallie]. The conversion of aircraft transponders and ground stations to the ADS-B standard is well ahead of the FAA scheduled 2020 deadline in the United States, (and several other nations are adopting aggressively also,) because the advantages of this new technology are compelling and profitable, [Nickels].

However, the move from radar, a costly and somewhat exclusive technology, to ADS-B, a relatively inexpensive digital technology, has had unintended consequences. Not everyone could successfully build a two-story radar complex, but a few engineers could design an inexpensive digital radio receiver to monitor aircraft – which millions of hobbyists could potentially build and even more interested persons could then use to monitor their local airspace. Vast networks of such monitoring stations operating over the internet now exist and the airspace information they provide approaches the precision and timeliness of that used by the Federal Aviation Administration in the United States. Flightaware.com, and flightradar24.com are two such examples, [Flightaware], [Flightradar24].

To a large degree, this represents a loss of control of local airspace information. Granted, ADS-B broadcast is strictly regulated, but this would not deter a dedicated attacker. And, unrestricted reception of ADS-B empowers everyone with a dark motive to contemplate how easy it *might* be to engage the system in subversive ways. Too much is at stake to dismiss the potential for abuse. It has been demonstrated in other research that unauthorized ADS-B broadcast could be used to spoof the presence of a phantom aircraft or to flood the local airspace with many such phantoms. Could the resulting confusion create an exploitable vulnerability in the air traffic control system? Could the greater speed and precision that ADS-B provides be used as an inexpensive substitute for a tracking system used to target an anti-aircraft weapon? These questions are not asked to generate fear, but rather to highlight the need to complement the worldwide adoption of ADS-B with security protocols to mitigate specific risks.

Global leaders have recognized the need to act, and experts in aviation and related fields have been carefully considering this issue. In his 2013 State of the Union Address, President Barack Obama said, “America must also face the rapidly growing threat from cyber-attacks… our enemies are also seeking the ability to sabotage our power grid, our financial institutions, our air traffic control systems. We cannot look back years from now and wonder why we did nothing in the face of real threats to our security and our economy.” [Obama]. The National Institute of Standards and Technology (NIST), and the American Institute of Aeronautics and Astronautics (AIAA) have recommended specific approaches to harden our aviation infrastructure against cyber-attack, [Automatic], [Framework]. Meanwhile, the International Civil Aviation Organization (ICAO), and the Air Force Institute of Technology have published specific recommendations to harden, (e.g. improve the security of,) ADS-B [Security], [McCallie].

The recommendations of the Air Force Institute of Technology are of particular interest, because aircraft vulnerability to cyber-attack is a long-studied subject of military interest. These are discussed in a later section. To understand them, we first need to distinguish between military and civilian protocols for aircraft communication. Military aircraft transponders and ground stations already implement an encrypted protocol referred to as Mode-5 for secure communication during missions that require it. In contrast, commercial aircraft transponders use non-encrypted protocols known as Mode-A (for identity) and Mode-C (for altitude). A more general form of Mode-A and Mode-C is Mode-S (for mode select). Mode-S is a broader standard and so also allows civilian aircraft, ground support, and general aviation to participate in the use of the protocol. Mode-S is used to transmit and receive the ADS-B transponder information from all civilian (aka commercial) aircraft in our local airspace on 1090-MHz. In the United States a distinction is made for low-altitude general aviation. These aircraft are encouraged to use the lower frequency; 978-MHz. Figure-2 summarizes the relevant aviation modes.

Transponder Modes by Purpose and Aviation Type			
Message Purpose	Military Transponder	Civilian Transponder	General Aviation (US only)
Aircraft Identification	Mode-3	Mode-A	Universal Access Transceiver
Aircraft Altitude		Mode-C	
Secure Communication	Mode-5	--	--
Purpose by (S)election	Mode-S		

Figure 2 - Transponder Modes

Comparing Mode-5 and Mode-S in detail is too cumbersome. Obviously, military aviation requirements and civilian aviation requirements will be different. But, comparing the two protocols in general suggests that the key difference is the encryption Mode-5 uses to protect the information being communicated. The recommendations of the ICAO and the Air Force Institute of Technology also point to encryption as the key step in the solution. However, not everyone agrees with this idea.

2.2. Controversy and Dissent

In 2009, the Federal Aviation Administration (FAA) released its own analysis on the security of ADS-B. Using the Security Certification and Accreditation Procedures (SCAP) published by the National Institute of Standards and Technology (NIST), they used specific language to omit all pertinent details in the name of National Security stating, “This assessment contains Sensitive Security Information that is controlled under 49 CFR parts 1 and 1520, and its content is otherwise protected from public disclosure. While the agency cannot comment on the data in this study, it can confirm, for the purpose of responding to the comments in this rulemaking proceeding, that using ADS-B data does not subject an aircraft to any increased risk compared to the risk that is experienced today." [FAA 75].

Naturally, this statement has generated considerable speculation regarding the FAA's motives. In any case, the reasons behind their conclusion will not be subject to public review. So, the question remains why would some believe that securing information about our airspace is the wrong choice? As one interviewed aviator put it, the point of the transponder is so no one ever loses track of you. Indeed, if the idea is truly to never be hidden to anyone then the idea of secure communication is pointless. At first glance, recent events like the loss of Malaysia Airlines Flight 370, lend credence to this idea. The fact that both transponders were disabled complicated the investigation of those events nearly to the point of hopelessness. On the other hand, the latest information from the ongoing investigation suggests the attack vector was from within the Electronic and Equipment (E/E) bay of the aircraft – which puts the focus right back on the absence of adequate security protocols again, [PBS].

No doubt, securing the ADS-B data stream with encryption would add a new degree of technological overhead to the monitoring and management of air traffic. Not every nation is able or willing to implement such provisions in their airspace without assistance. A similar sentiment is expressed in the view that effective encryption to secure the ADS-B data could imply the export of cryptographic hardware or software to nation-states that are prohibited from receiving such technology. While this is a real restriction, it is also true that not every nation is currently scheduled to convert to ADS-B. So, many international flights will utilize traditional radar systems for a long time to come.

2.3. Selected Objectives

The goal of this project is to develop a dual frequency (1090 MHz and 978 MHz) Mode-S receiver with a programmable decoding system to obtain an ADS-B data stream. From the data stream, we will develop additional software for the Android platform that will display the information in near real-time. However, more can be gained than just a personal view of local airspace. The benefit to be demonstrated is an encryption of the information (where law and treaty permit) to protect the location data of such aircraft from being used by criminals, terrorists and enemy combatants to target the aircraft.

Conceptually, this is the same type of benefit offered by Mode-5 for military aircraft, but extended to civilian flights and general aviation. Unlike Mode-5, which encrypts the entire communication channel, this project will encrypt/decrypt the fields within the ADS-B messages while preserving the unencrypted framework of the message. To this end, we plan to simulate the broadcast of an encrypted ADS-B transponder in software, as an actual ADS-B broadcast would be forbidden. Bypassing our Mode-S receiver, our enhanced software would then decrypt the signal and display the information in near real-time on an Android smartphone. Meanwhile a standard ADS-B decoder and display software would not be able to determine the aircraft's location or related information.

Other objectives:

- Low Cost (less than $500). Although, the initial prototype(s) may be of higher cost.
- Portable (battery-powered with a mobile antenna system). A telescoping mast arrangement is envisioned.
- Accurate (100km detection range and position resolution to 25m). This also depends on the type of transponder with which the signaling aircraft is equipped.
- Responsive (decoded/decrypted position and altitude information with no more than a 5-second delay). The ADS-B source information may be delayed more than this; however the latency which our system introduces will be restricted by this objective.

2.4. Project Requirements and Specifications

After researching the ADS-B signal it is clear that to receive signal and have any usability for the received data we must detect signals originating from a range of 100 km along with a position resolution of 25 m. With a range of less than that defined the user would have no need for the device and could easily locate the aircraft via visible contact. Any lack of range would further debilitate the usefulness by not allowing the user enough time to prepare and perform any tasks that they wish to do with the gathered data and location of the aircraft.

2.4.1. Signal Reception

The important aspect of the signal reception is for the device to receive any ADS-B data streams via the of the 978 and 1080 MHz radio waves. To do this the antenna will need to be designed to maximize reception while still allowing for maintaining of the portability requirement and power management. Signal reception will need to be constant throughout the user's use of the device in order to maintain a live tracking of the aircraft; if the signal were to no longer be received, then outdated information would be displayed and accuracy would obviously be lost. Reception strength of the signal will vary largely upon the location that the device is set up and the abilities of the antenna.

2.4.2. Signal Conversion and Decoding

Unrelated to encryption / decryption is the method used to propagate the ADS-B data in analog form over the Mode-S carrier wave. In this case the method in use is called Manchester encoding / decoding. This is a widely used and standardized scheme where both the data and the clock can be recovered from the same signal. It is considered a binary form of phase-shift keying, [Tanenbaum]. In our design, an Analog to Digital Converter (ADC) performs this function, (or in other designs, the portion of a Software Defined Radio (SDR) which simulates ADC operation.)

Every message begins with an 8-microsecond preamble of four pairs of on-pulse/off-pulse with specific inter-pulse timing followed by 56 or 112 microseconds of Manchester coded data, (which translates to 56 or 112 bits of data due to the fixed time nature of this scheme.) The 56-bit portion is a datagram and is formatted using figure-3:

Message Type	Surveillance-Control	ICAO (Aircraft) Code + Parity
5-bits	27-bits (Format depends on which airspace authority controls the aircraft).	24-bits

Figure 3 – Short Squitter Format

The 112-bit portion is also a datagram and is formatted with an additional 56-bit field inserted in the datagram as shown in figure-4:

Message Type	Surveillance-Control	Extended Data	ICAO Code +
5-bits	27-bits	56-bits (Format depends on message type).	24-bits

Figure 4 – Extended Squitter Format

Other datagram formats also exist but are not used in this project.

Our requirement in this project is to correctly comply with this Manchester coding scheme for ADS-B messages.

Specifically, the following ADS-B message types, [Köllne], [Radar] will be decoded for further processing:

- DF0 (56-bit) Generally referred to as the ACAS message.
- DF4 (56-bit) Rollcall reply: Altitude - resolution to 100ft.
- DF17 (112-bit) Extended Squitter: Contains ADS-B data (position, heading, etc.)
- DF18 (112-bit) Extended Squitter: Same as DF17 but from ground traffic.
- DF20 (112-bit) Rollcall reply: Altitude - resolution to 25ft. Uses EHS/BDS registers.
- DF21 (112-bit) Rollcall reply: Identity. Uses EHS/BDS registers.

2.4.3. Android Mobile Application

Users will be able to view the ADS-B data via an Android Application. This application shall display information from the aircraft by a graphical user interface. This interface will display markers of aircrafts on a map that are picked up by the receiver and are within the current GPS position of the Android device. From this map, the user can view information for each individual aircraft from traveling speed, current altitude, to the type of plane by selecting its marker. As this process occurs, all aircraft information shall be updated in real-time. In addition, the application shall be able to decode any encrypted information that is sent to the mobile device. As the application is being developed, additional features will be added and the main features will be improved.

2.4.4. Signal Cryptography

Naturally, signal cryptography is not applicable to current Mode-S transmissions of ADS-B messages. The absence of such a secure protocol is the fundamental problem being addressed in this project. The intent is to add cryptographic capabilities to a Mode-S receiver and decoder without breaking ADS-B message integrity. Fortunately, the Air Force Institute of Technology (AFIT) has already made a preliminary assessment of this issue. AFIT proposes that format-preserving encryption (FPE) could be employed using an algorithm named FFX to protect key data within the ADS-B messages without changing the overall structure of the messages.

As specified earlier in figure-4 of the Signal Conversion and Decoding requirement, the ADS-B extended squitter message types follow a known format which includes an extended data field of 56-bits. That figure is expanded here in figure-5 to illustrate the proposed scope of the encryption:

<table>
<tr><td>Messag e Type</td><td colspan="2">Surveillance-Control</td><td>Extended Data</td><td>ICAO Code +</td></tr>
<tr><td>5-bits</td><td colspan="2">27-bits</td><td>56-bits
(Format depends on message type).</td><td>24-bits</td></tr>
<tr><td></td><td>Capabilit y
3-bits</td><td colspan="3">Encryption Target
104-bits</td></tr>
</table>

Figure 5 – Partial Encryption of the Extended Squitter Format

Our requirement in this regard is to correctly apply a cryptographic scheme to only targeted portions of the ADS-B messages.

Consequently, the following ADS-B message types, will be impacted by the new FPE processing:

- DF17 (112-bit) Extended Squitter: Contains ADS-B data (position, heading, etc.)
- DF18 (112-bit) Extended Squitter: Same as DF17 but from ground traffic.
- DF20 (112-bit) Rollcall reply: Altitude - resolution to 25ft. Uses EHS/BDS registers.
- DF21 (112-bit) Rollcall reply: Identity. Uses EHS/BDS registers.

Ultimately, other ADS-B message types which use the extended squitter format could also be affected, but these initial message types can serve as a adequate proof-of-concept.

The FFX algorithm specifies a variable number of rounds of encryption (and therefore also decryption) using other known encryption algorithms, (Bellare.) The FFX documentation illustrates its operation using the Advanced Encryption Standard (AES) as this underlying algorithm. However, FFX itself could use another encryption algorithm depending on the need at hand.

2.5. Initial Budget

The initial budget (figure-6) was created at the very beginning of the project with only a minimum of research having been completed. As research progressed, the additional circuits and parts that have become part of the project are generally offset by our more precise estimates of part costs and by locating vendors with favorable pricing.

Component	**Estimated Costs**
Battery	$ 32.00
Power Supply Circuits	$ 25.00
Antenna Hardware	$ 65.00
RF Receiver Circuits	$ 45.00
ADC w/eval board	$ 102.00
Decoder FPGA w/eval board	$ 75.00
USB or Wireless Interface	$ 45.00
Custom Printed Circuit Board	$ 105.00
Total	$ 494.00

Figure 6 – Initial Budget

The significant changes include; Antenna hardware costs are climbing with the costs of the PVC housings. The use of evaluation boards for the ADC and FPGA components were avoided with the use of minimal CPLD packaging. But, the number of ADC and FPGA devices has increased – offsetting these gains. Please refer to the Bill of Materials section for the detailed breakdown.

3. Research

3.1. Existing and Similar Projects

3.1.1. DVB-T SDR

The Digital Video Broadcast – Terrestrial, Software Defined Radio, or DVB-T SDR, is the most widely used digital television protocol in the world. The close proximity of the frequencies used for DVB-T and ADS-B is what first stimulated a few engineers to wonder if the devices being used for DVB-T could be modified and repurposed to receive ADS-B. Now it is commonplace to see the familiar USB dongle (figure-6) and tiny antenna of an ADS-B receiver attached to a traveler’s laptop computer. The same SDR is used as the basis of both devices.

Today, that SDR is the Realtek RTL2832U. This chip is paired with a “tuner” chip that depends on the application to which the radio is being put. For both DVB-T and ADS-B, the most common tuner is the R820T from Rafael Microelectronic, Inc. This tuner is well matched to both the 1090 MHz and 978 MHz frequencies used for ADS-B messages. Now, as a commodity item, the typical cost of these USB dongles (figure-7) is less than $20 USD.

Figure 7 – Low Cost ADS-B Receiver Dongle

While no other ADS-B receiver can compete with the low cost of this approach, it fails to meet our needs in two specific areas. The device is not able to be customized sufficiently to accommodate the programmability we require to merge ADS-B from the reception of both frequencies. And, the device expects to interface with a computer running Windows or Linux to process the datastream it has received. However, we did consider using the Realtek RTL2832U SDR and Rafael Micro R820T tuner directly as a set of chips in our design. Unfortunately, the need for FPGA merging of the data could not be avoided and using actual ADCs rather than an SDR allows us to achieve greater sensitivity for the receiver.

3.1.2. miniADSB

Popular with hobbyists and do-it-yourself types is the miniADSB receiver kit from jetvision.de (miniADSB). This device takes a minimalist approach to offer a simple 1090 MHz receiver with no SDR, and no FPGA or other signal processing. It is an open project, and all schematics, data sheets, PCB layout and related design documents are freely available. Extensive instructions and assembly guidance are available online for the hobbyist.

However, the obvious absence of programmability limits the use of this approach to a front-end only receiver. We considered this approach as a possible reference design for the Mode-S receiver portion of the project.

3.1.3. Köllner Project

Günter Köllner documented his contributions to the research and development of the ADS-B receiver and signal processor now known in alternate forms as Radarcape, (figure-8,) and the Mode-S Beast. His outstanding analysis and project notes were very useful, and that device most closely met our expectations for a reference design for this project, (Köllner.) In general, the Mode-S Beast combines a 1090 MHz receiver with a fast ADC and then performs signal-processing to decode the digital data stream. The result is packaged for USB transfer (and Bluetooth in some of his design variations) for further processing by a companion computer. The Radarcape absorbs this companion computer requirement by embedding a small Linux-based computer in the device.

Figure 8 – The Radarcape rebranded for use with the flightradar24 network
by Nubifier, reprinted under the Creative Commons Attribution-Share Alike 3.0 license

This design approach delivers much greater possibilities for programmability and customization of the ADS-B data stream. The substantial differences in our design are the inclusion of the 978 MHz UAT frequency in the receiver, an even larger signal processing capability, the use of an Android smartphone platform rather than an embedded Linux platform, and FPE-based encryption and decryption.

3.2. Relevant Technology

3.2.1. Radio Frequency Design

Data that is to be received will travel on radio frequencies of 978-MHz and 1090-MHz. The source locations of the signals will be originating on aircraft that are traveling in the airspace located within the range of the antenna's reception abilities of 100 km. The data frames that are collected are referred to as "squitters."

3.2.1.1. Mode-S Antenna

Using an antenna or antennas optimized for the desired frequency would significantly reduce the noise associated with the incoming signals. Frequency filters would further decrease noise causing interference of surrounding frequencies. To further enhance the reception of the antenna it should be placed as high as possible to ovoid in interference located near the ground due to trees and buildings. To view a greater number and variety of aircraft the placement of the antenna and decoding device should be within range of either high flight path location or an airport, preferably one with international flights.

Range of signals received can depend on antenna design in addition to placement. An antenna can be designed for large area detection via Omni-directional reception but usually with the tradeoff of a shorter detection range, or to design to enhance the range but at the cost of detection in a targeted direction from the source. Optimization can be performed with a combination of two antennas working in tandem. One Omni-directional antenna working in conjunction with a directional antenna merging their received signals via a signal combiner. This method will only work as long as the incoming signals from the separate antennas are in the same phase as each other to avoid signals being canceled out.

Once the mode-S frame is received by the antenna it will travel to the receiver where it will be filtered and distributed to the necessary decoder.

3.2.1.2. Mode-S Receiver

The receiver for this system accepts the signals from the antenna and prepares the signal for the next stage of the system. The preparation includes tuning to the specific frequency desired and decoding the signal. In the case with ADS-B, it will be tuned to receive the desired signals, these will be either 1090-MHz or 978-MHz frequencies or if costs allow perform both alongside each other. Due to the antenna's or antennas' design of accepting both frequencies the receiver has the assignment of distinguishing the two signals and distributing them to the decoder. A subsystem of the receiver is to convert the analog signal of the radio waves into a digital signal.

Implementation of the receiver will be through the use of a software-defined radio discussed in section 3.2.2. The receiver and the decoder will work in close conjunction with each other to convert the signal to data streams ready for input to software that the user can use to gather information via a user interface.

In addition to tuning to a signal range and converting the signal, this will be the point in the system where any amplification to the signal will be performed to it if needed. Amplification will need to occur in the receiver system prior to the conversion subsystem to allow for the weak signals to have their information read. Due to the nature of weak signal strength it is expected that amplification will be undesirable. Amplification of signals have the possible effect of introducing noise into the signal, this can cause erroneous results. Though if it proves more beneficial to the system and the budget allows, then amplification will be introduced as needed.

3.2.2. Software-Defined Radio

A software defined radio allows for the radio to be controlled, accepting and processing signals with desired traits such as frequency and data rates. While still a new technology that is continually being implemented into new uses. The basis of the SDR is that while an antenna can be used to pick up any number of signals the physical layer of the Open Systems Interconnection (OSI) model is controlled by software rather than specialized hardware as in a standard radio.

Implementation of a SDR (figure-9) can be performed via FPGA, DSP, and performed on GPPs. Tasks that will need to be performed by the radio will include filtering, amplification if necessary, analog to digital conversion. We plan to implement the SDR with a FPGA that will be controlled via an Android device running an application with a user interface to control the radio and to view the resulting decoded signal. FPGAs will be discussed in a later section. Our PCB will house our SDR receiver along with other required hardware.

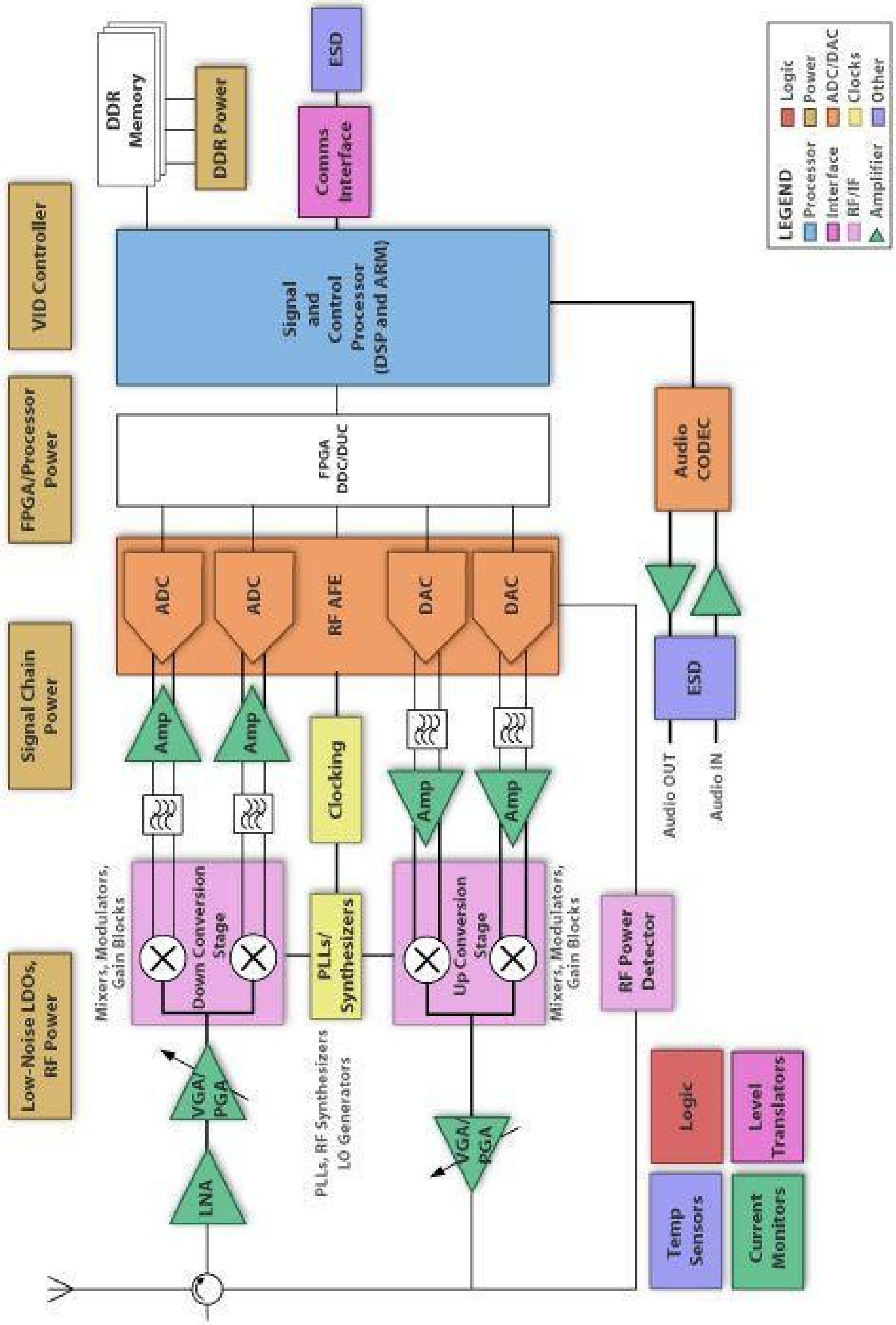


Figure 9 - Implementation of a SDR
Courtesy of Texas Instruments

http://www.ti.com/solution/software-defined-radio-sdr-diagram

The advantage of a SDR is the ability to reprogram the software to accept different signals of parameters without changing any hardware. While a software-defined radio is not needed for the basis of this project, it does allow, if needed in the future more frequencies for data link to be added to the radio. This becomes a cost saving effect by not having to remanufacture hardware to accept the new signal; a simple software update should suffice for any future needs of the user.

The two chips that we are looking at using to implement in our design are the Realtek RTL 2832U and the TI TMS320C6657. The TI chip holds the benefit of using an arm architecture allowing for a wider implementation users. The Realtek chip has already show it usability in other projects of a similar nature with extensive documentation in with DIY hobbyists. Because of its proven results we would choose the Realtek chip over the TI.

Increased complexity of the programming design architecture led us to seek the option of programming FPGA chips to decode the incoming analog signals into digital signals.

3.2.3. Signal Decoding

This section discusses a special research goal related to our eventual encryption / decryption of a targeted portion of the ADS-B data stream. Naturally, it is important to remember that decoding and decryption must not be confused. In our usage, decoding refers to interpreting the transmission scheme used to package the data in the signal. In contrast, decryption refers to the secrecy scheme used to protect the data in the signal from being put to unintended uses. They are related in this project in that the first decoding step is the first opportunity to consider whether to also begin the decryption steps.

The research has revealed that it is possible to explore opportunities to perform a hardware-based decryption, but only after the Manchester decode has been completed for a given frequency. Consequently, two-tiered logic architecture is warranted. The Manchester decoder represents the first tier, and the merge of the decoded digital data streams is a second tier where decryption can be considered. This implies that there would be a first tier of compact logic device(s) (FPGA) performing just decode and a second tier of more capable logic device(s) that could eventually perform much more than a merge of the data.

To keep this option open, and in line with the programmability objective of this project, our design will include a large capability to add additional functionality to the second tier decoder step in the form of decryption logic. Considering common

AES implementations on Xilinx FPGAs can consume 250,000 logic gates, and the FFX algorithm could exceed this complexity, then an additional 300,000 logic gates should be the *starting* point for this reserve capability.

3.2.4. ADS-B

A detailed breakdown of the internal message formats used in the ADS-B protocol is needed for the project. As an international standard, such formats are publicly available; however they are under the control of regulatory bodies that choose not to disclose them without payment. To avoid the $262 USD cost of acquiring the information through such regular channels, I have collected this information by less formal inquiries and discussion forums over the internet. While this increases the possibility that this information may be error-prone, it at least gives us a reasonable starting point to program and test against.

As previously discussed in section 2.4.2, the ADS-B extended squitter message format includes an extended data field of 56-bits. This field is further subdivided (figure-10) depending on the content of the message:

Messag e Type	Surveillance-Control	Extended Data	ICAO Code +
5-bits	27-bits	56-bits	24-bits

Airborne Position Squitter	
Type code “05”	5-bits
Surveillence status	2-bits
Single antenna flag	1-bits
Altitude	12-bits
UTC time flag	1-bits
Compact Position Reporting (CPR) code	1-bits
CPR encoded latitude	17-bits
CPR encoded longitude	17-bits
	56-bits

Figure 10 – Airborne Position Squitter

Alternately, these same 56-bits could contain surface position, airborne velocity, aircraft identification, or the state and status message known as DO-260A. Those additional layouts for the extended data field follow in figures 11-14:

Surface Position Squitter	
Type code “06”	5-bits
Movement	7-bits
Valid status flag	1-bits
Ground track	7-bits
UTC time flag	1-bits
Compact Position Reporting (CPR) code	1-bits
CPR encoded latitude	17-bits
CPR encoded longitude	17-bits
	56-bits

Figure 11 – Surface Position Squitter

Airborne Velocity Squitter	
Type code “09”	5-bits
Ground or airspeed flag	3-bits
Intent change flag	1-bits
IFR capability flag	1-bits
Velocity uncertainty	3-bits
East-West velocity	10-bits
East-West velocity sign	1-bits
North-South velocity	10-bits
North-South velocity sign	1-bits
Turn indicator	2-bits
Geometric altitude less barometric altitude	7-bits
Geometric altitude less barometric altitude sign	1-bits
	56-bits

Figure 12 – Airborne Velocity Squitter

Aircraft Identification Squitter	
Type code “08”	5-bits
Aircraft category (sm/med/lg, vehicle/glider/etc..)	3-bits
Callsign character-1	6-bits
Callsign character-2	6-bits
Callsign character-3	6-bits
Callsign character-4	6-bits
Callsign character-5	6-bits
Callsign character-6	6-bits
Callsign character-7	6-bits
Callsign character-8	6-bits
	56-bits

Figure 13 – Aircraft Identification Squitter

DO-260A State and Status	
Type code “29”	5-bits
Subtype “0”	2-bits
Target altitude and flags	18-bits
Target heading or track	14-bits
NACp / NICb / SIL indicator	7-bits
Reserved	5-bits
ACAS status and RA status	2-bits
Emergency or priority status	3-bits
	56-bits

Figure 14 – State and Status Squitter

Other content is also possible; however this is the only information to be included in the project at this time.

3.2.5. Power System

3.2.5.1. Battery Types

When designing a device for portability, battery power often comes to mind for a power source for being compact and transportable. Upon selecting a battery type to use, there is a large variety to choose from each being favored for certain applications from something like a watch to an automobile. When actually choosing a battery, we have to take into account the voltage, current, and power that is being supplied. In addition, size will play an important matter as using a bulky battery is quite undesirable. These attributes will dictate which battery can be applied to the receiver. The following are some of those battery types that we have selected for our project.

We will begin with one of the most used battery today, alkaline. Alkaline batteries were introduced in the 1960s and to this day are still popular as ever [4]. The applications of these batteries include household electronics like MP3 players, toys and radios. Compared to most batteries, alkaline has one of the longest shelf life with a self-discharge rate of 0.3% per month and does not require any maintenance to retain its charge [20]. This is extremely useful if the device is not used often. In exchange, most alkaline batteries have a short cycle life from being a single use for non-rechargeable to about 50 recharge cycles for the reusable variant. In addition, alkaline batteries, when dead, are prone to leak potassium hydroxide, which can cause eye and skin irritation [4]. In doing so, it is somewhat a risk if we forget to replace the battery. The average nominal cell voltage for an alkaline battery is about 1.5V [20].

The next battery on the list is the lithium ion battery. Lithium ion batteries are a common battery that has been gaining popularity as the technology for it improves. Applications of lithium ion batteries can vary from things like smartphones and MP3 players to large scale items like cars and aircrafts, for example the Toyota Prius and the Boeing 787. The normal cycle life for a lithium ion battery is about 500 to 1000 recharge cycles before the battery hits 80% of its max capacity [20]. The self-discharge rate is about 10% per month with a nominal cell voltage of 3.6V [13]. Some advantages of lithium ion batteries include high energy density, such as the Tesla Model S uses a large lithium ion battery to run the vehicle, and requires low maintenance to retain its charge. A disadvantage, however, is that lithium ion batteries requires a circuit to limit voltage and current to prevent the battery from overcharging and discharging [20].

Now we move on the nickel metal hydride battery (NiMH). NiMH is a popular commercial rechargeable battery. This type was essentially an interim step before lithium ion. Even with the popularity of lithium ion batteries, NiMH are still used for many devices with very few of the negative side effects from its predecessor, the nickel cadmium battery. Compared to most batteries, NiMH batteries are much more environmental friendly being much simpler to recycle. However, NiMH has a high discharge rate of about 30% per month and reduced performance under high temperatures. These traits makes it less ideal should the

receiver is shelf for a period of time. The average cycle life for this type of battery is around 300 to 500 recharges with a nominal cell voltage of 1.25V.
Finally, we have the lead acid battery. Lead acid batteries were developed in 1859 and were the first battery produced for commercial use [15]. Prominent uses of lead acid batteries include automobiles and uninterruptible power supplies. Some advantages includes being easy and inexpensive to produce, low self-discharge rate compared to the other rechargeable batteries, and low maintenance. In exchange, lead acid batteries have low energy density and must be disposed property due to the lead content. The average cycle life is around 200 to 300 recharges with a nominal cell voltage of 2V [13]. The nominal cell voltage appears to be comparable to NiMH and lithium ion however, lead acid battery cell size are much greater than those two. The self-discharge rate is about 5% per month [13].

3.2.5.2. Comparison

With the assortment of batteries, we have to select the one that will fit our application. To deal with having to continuously remove the battery, using a rechargeable battery would be more efficient. The benefits of rechargeable batteries include being reusable and cheaper as buying replacement batteries will slowly rack up. Because of this, we are down to lead acid, NiMH, and lithium ion. Since the receiver will be portable and equipped with a mobile antenna, the battery must have a good energy density to size ratio. This fact cuts lead acid batteries off the list and leaves NiMH and lithium ion. Table-15 compares the differences between NiMH and lithium ion batteries.

Battery Type	**NiMH**	**Lithium Ion**
Energy Density (Wh/kg)	60 - 120	150 -190
Cycle Life	300 - 500	500 - 1000
Self-Discharge Rate	30%	10%
Charge Time	2-4 Hours	2-4 Hours
Overcharge Tolerance	Low	Low, Cannot not handle tickle charge
Safety Requirement	Thermally Stable	Protection Circuit
Nominal Cell Voltage	1.25V	3.6V

Table 15 - NiMH vs Lithium Ion [4][13][20]

As we can see, lithium ion essentially defeats NiMH in almost every category except for tolerance and safety requirement. NiMH can handle tickle charges and just have to be stable at high temperatures. Lithium ion, on the other hand, is not as tolerant to overcharges and needs a circuit to protect itself. In doing so, NiMH, in general, is cheaper than lithium ion batteries. Even with these differences, both batteries will work well with the receiver.

3.3.1. Analog to Digital Conversion

Upon receiving an ADS-B signal from the antenna, it became important that we translate the analog signal into a format that we can use. In doing so, an analog digital converter was needed to convert the signal to digital. An analog digital converter (ADC), as the name conveniently suggests, it is an electrical circuit that is used to convert an analog signal to a digital signal that is in terms of bits. In the scope of this project, the receiver will use an analog digital converter microchip. Even though we no longer have to design the converter, it was important to understand the types and properties of ADC so that would choose the correct type for our application.

3.3.1.1. Conversion Types

The primarily factor in choosing an ADC for the receiver was how quickly, how simple and how efficiently a signal is converted into bits. Since we are planning to use the data in real-time, speed played a major factor in choosing the converter. Other factors for the converter include power consumption and accuracy. The following will cover the types of conversion, and the pros and cons of each. In terms of conversion, there are three methods that we have considered. The methods are flash, successive approximation, and dual-slope.

The first method that we will consider is flash conversion. For flash conversion, the converter uses a combination of resistors, comparators, with a priority encoder. A comparator is a component that will output either a 0 (low) or 1 (high) depending on if one input is greater than other. While a priority encoder will output a bit sequence depending on which input goes high first. Figure-16 is an example of the interworking of the converter in the simplest form. From the figure, there is a resistive ladder that is connecting to a ground at one end and a reference voltage at the other. With the ladder, there is an even voltage drop across each resistor. In between each resistor, one input from a comparator is connected. The other input for each comparator is connected to the input signal. When the input signal causes the comparators to go high, the encoder will use the first high input and output a sequence. A benefit of this method is that it is faster than the other methods; however it has higher power consumption and uses more hardware than the others [5].

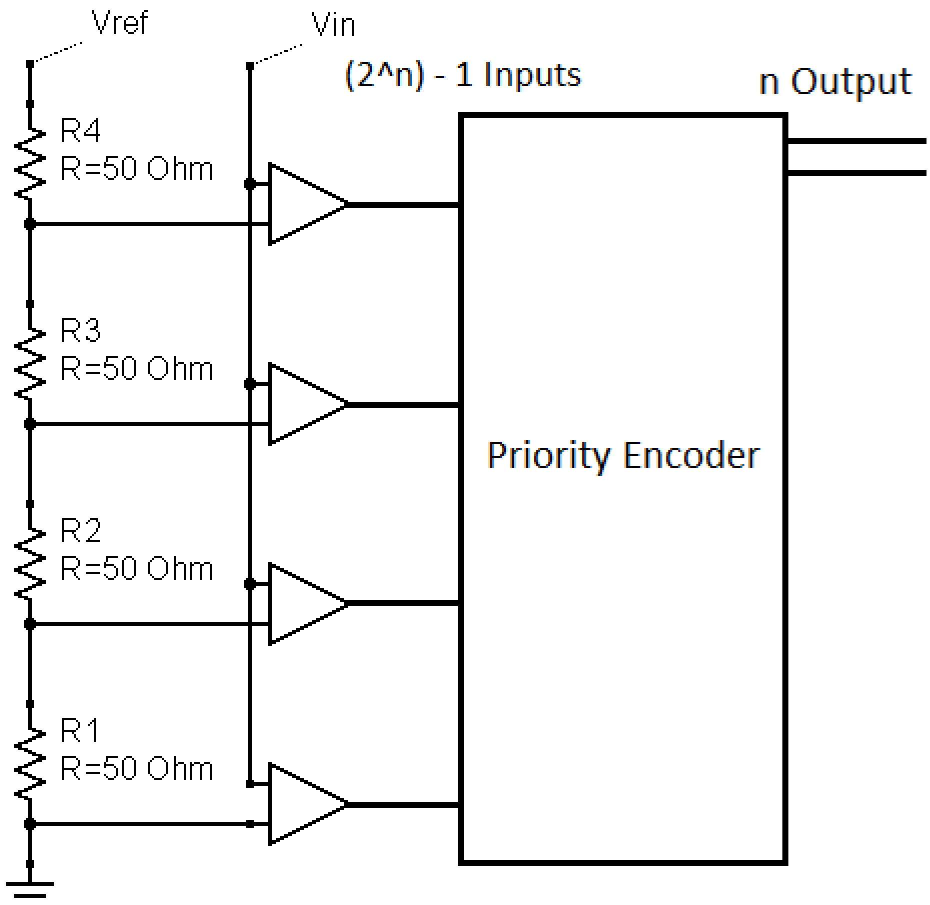


Figure 16 Flash Conversion Circuit [19]

The second method that we will consider is the successive approximation conversion. For successive approximation conversion, this method uses a digital analog converter (DAC) that is connected to a comparator, and a successive approximation register. With that, the input signal is connected to the comparator. Figure 17 is an example of how a general successive approximation converter would appear. The converter works by setting all the bits in the register to zero. From there, the register will set the most significant bit to one and send the bits to the DAC which will translate it into a voltage for the comparator. If the comparator goes low then the register set that bit to zero and repeats. If the comparator goes high, the register will save that bit and move to the next bit. The converter will continue with the same pattern until it reaches the least significant bit. Afterwards, the bit stream is sent to the decoder and the whole process is repeated for the next signal. This method is not as quick as the flash conversion but, the power consumption is lower. An issue with this method is that as the number of bits increase, the longer it will take for the converter to output a digital value [5].

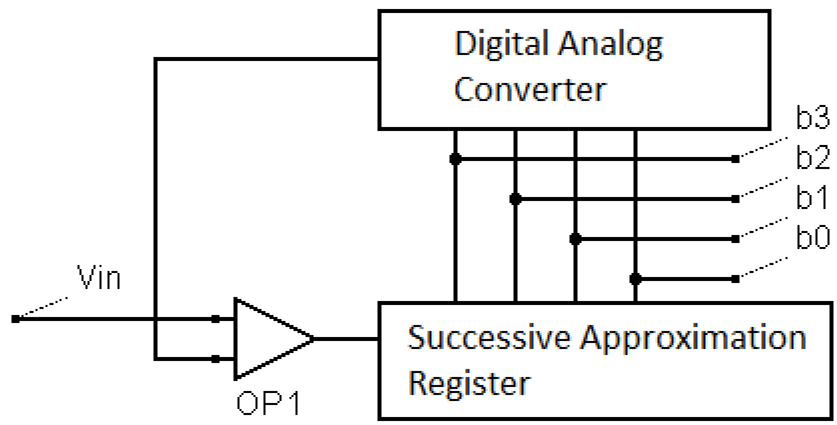


Figure 17 Successive Approximation Circuit [19]

The third method that was suggested for the converter was dual-slope conversion. This conversion method uses an assortment of components. These include resistors, a switch, a register and a control unit. Figure 18 below is an example of this converter. Compared to flash and successive approximation, this method is a bit more complicated in the way it converts the signal. In simplifying the method, Dual-slope begins with the control changing the switch to the input voltage. The input voltage then charges the capacitor for a constant time set by the control. After the constant time, the control changes the switch back to the reference voltage. The capacitor will then discharge at a constant rate. During which, the control uses a counter to see the number of clock cycles it takes to complete discharge. The value that is stored in the counter is use to output the bit sequence. In terms of accuracy, this method is very precise. The primarily disadvantage is that it takes a time for the capacitor to discharge.

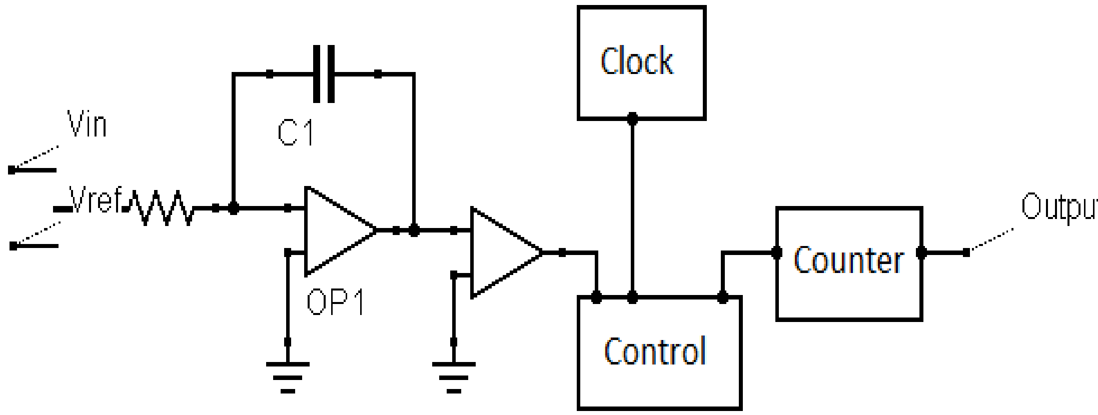


Figure 18 Dual-Slope Circuit [19]

The fourth and final architecture design is pipeline. This conversion method uses comparators, adders and multipliers. Figure 19 below is an example of the pipeline architecture. This design is a hybrid of both flash and successive approximation. The pipeline method begins with the input voltage being compared to half the reference voltage. If the input is higher, subtract half of the reference voltage from the input and set that bit to one. If the input is lower, it will just set the bit to 0. Both way, the input voltage is double and the process is repeated for the next bits. The location between the comparator and selector is where the value will be taken. The advantage of this method is that it can be as fast as the flash conversion it does not require as much as flash when the number of bits increases. The disadvantages include high latency and possible errors from the mathematical operation [5].

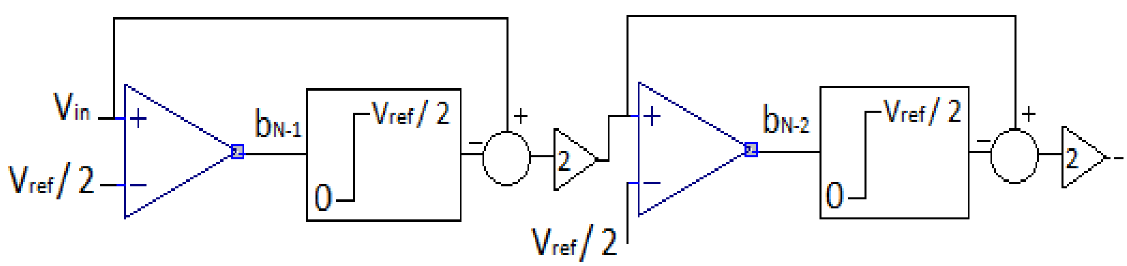


Figure 19 Pipeline Circuit [5]

3.3.1.2. Additional Properties

In addition to conversion type, other properties to take into account are resolution, linearity, and sampling rate. Resolution, in terms of an ADC, is the number of discrete values or bits that a converter can return over a range of possible analog values [6]. Linearity deals with how gradual the change in bit value versus the change in analog input. With that, high linearity means improve accuracy. Finally, the sampling rate is the rate at which the converter is sampling the analog signal for new digital values [6]. As with conversion type, these attributes have to be analyzed for the selection of an ADC.

3.3.2. Wireless Interface

Wireless connections play an important role in the transmission of data. For this project, transferring data from the receiver to a mobile device will become crucial when it comes to displaying aircraft locations and updating the aircraft information in real-time. The aim is choosing an interface with least latency, fast data transfer, and have low power consumption. Due to the large selection of wireless interfaces, the two that were most suggested and will be discussed are Wi-Fi and Bluetooth. Since most mobile device comes equip to handle both Wi-Fi and Bluetooth signals, these were the most logical choices. In addition to Bluetooth, Bluetooth Low Energy (BLE) will be discussed as well.

3.3.2.1. Wi-Fi

To begin, we will start with one of the most widely used wireless interface around, Wi-Fi. Wi-Fi is a wireless local area technology that has been implemented and used in multiple devices from smartphone to things like smart TVs. The benefits of Wi-Fi include efficiency, low cost, bandwidth, and transmission speed. However, some of the disadvantages include power consumption and security. All Wi-Fi devices use a set of standards called the IEEE 802.11. The IEEE 802.11 standard is used for implementing wireless local area networks. There are multiple versions of this standard as they are being updated continuously over the years. These standards include 802.11a, 802.11b, 802.11g, and 802.11n being the latest to be developed and released. In general, all IEEE 802.11 standards were designed to function on radio frequencies at 2.4GHz and 5GHz [22].

In terms of speed and range, each iteration of the IEEE 802.11 differs from each other in those aspects. With IEEE 802.11a, the standard allowed for speeds up to 54 Mbit/s but the tradeoff was that the high speed meant a shorter wavelength so it could not penetrate obstructions thus limiting its range [23]. Next is the IEEE 802.11b, this standard has a lower speed than 802.11a, at a rate of 11 Mbit/s, but unlike 802.11a the signal was able to travel a much farther distance [23]. Afterwards we have the 802.11g; this standard had the benefits of both 802.11a and 802.11b in speed and range, respectively [23]. Finally we have 802.11n, this standard has improve over the 802.11g by allows speeds up to 600 Mbit/s and being able to perform on both 2.4 GHz and 5 GHz [15]. This differs from the previous iterations where each standard could only work on either frequency but not both.

Security plays an important role when it comes to connections and Wi-Fi implements an assortment of wireless encryption types. These types of encryptions include the Wired Equivalent Privacy (WEP) and the Wi-Fi Protected Access (WPA and WPA2) [22]. Since the data being transferred must be secure, the risk of information being obtained is a constant threat. In terms of power consumption, Wi-Fi uses more energy than other wireless networks due to being a wireless local area network that has to cater to multiple devices.

3.3.2.2. Bluetooth

The other wireless interface that was suggested was Bluetooth. Bluetooth is a wireless personal area network designed for transferring data between devices at a small distance. The benefits of Bluetooth includes being robust, having a low power consumption, being low cost, and improve security. Some disadvantages include having a low transfer rate and having a short range. Like Wi-Fi, Bluetooth follows its own set of standards. These standards are developed by the Bluetooth SIG (Special Interest Group) [9]. This organization deals with the creation and development of standards and the management of the programs used to qualify devices.

As with Wi-Fi, Bluetooth operates on its own set of frequencies ranging from 2.4 GHz to 2.485 GHz. With range, there are multiple classes of radios that Bluetooth devices can support with Class 1 going up to 3 feet, Class 2 going up to 33 feet, and Class 3 that can reach distances of up to 330 feet [9]. When it comes to max transfer rate, Bluetooth has gone through multiple iterations with version 4.0 being able to reach speeds at about 24 Mbit/s [9]. In general, Bluetooth has low power consumption with about 2.5mW with Class 2 radios. In terms of security, Bluetooth is designed to use algorithms based on SAFER+ block cipher to protect from attacks and infiltration [9]. However no matter what the security, all wireless networks are vulnerable in some form to another. Table 20 summarizes the difference between the two interfaces.

Wireless Interface	**Wi-Fi**	**Bluetooth**
Range	Depending on Standard 200 feet (Indoors) – 820 feet (Outdoors)	Depending on Radio 3 – 330 feet
Frequencies	2.5GHz and/or 5GHz	2.4GHz – 2.485GHz
Data Transfer (Max)	Depending on Standard 54 Mbit/s – 600 Mbit/s	24 Mbit/s
Standards	IEEE 802.11	SIG
Security	WEP, WPA, and WPA2	Based on SAFER+ Block Cipher
Power Consumption	High	Low

Table 20 Wi-Fi vs Bluetooth [9][11][21]

Since Bluetooth is a personal area network compared to Wi-Fi, each caters to deal with certain issues. Wi-Fi is primarily used to deal with multiple devices in a large area where all devices can communicate to each other and to the Internet. Often, there is a single of entry and there is a primary node that deals with the information transfer. Bluetooth, however, is made to allow communication for devices in close proximity. With Bluetooth, there is a direct connection between both devices.

3.3.2.3. Low Power Options

In addition to the wireless interfaces mentioned, Bluetooth has a low power option called Bluetooth Smart, also known as Bluetooth Low Energy. With this version of Bluetooth, power consumption is reduced and the latency of data transmitted is reduced as well. In exchange, the range and data transfer rate is reduced. This reduction is only minor, thus this makes Bluetooth Smart more appealing. However, Bluetooth Smart is not backwards compatible with classic Bluetooth. Even with this disadvantage, multiple devices are transitioning to support both classic Bluetooth and Bluetooth Smart with the goal of replacing classic with smart over the next few years. Table 21 shows the differences between classic Bluetooth and Smart.

Bluetooth	Classic	Smart
Range	330 feet	<330 feet
Power Consumption	1W as a Reference	Around 0.5W
Latency	100ms	6ms
Data Transfer (Max)	1 – 3 Mbit/s	1 Mbit/s

Table 21 Bluetooth Classic vs Bluetooth Smart [10]

3.3.3. Android

With the data obtained and decoded from the receiver, there is a need to display the information to the user. Our choice to display this information is to develop an Android application that would run side by side with the device. This mobile application will be developed in Java using Eclipse with the Android Development Tools. Plus, a set number of Application Programming Interfaces (APIs) will be integrated into the app.

The goal of this application is to allow the user to view the location and information of all aircrafts found by the receiver. The data on each aircraft will be sent via Bluetooth from the receiver to the Android device where it will be converted into visual information. The graphical user interface will make it simple for a user to select an individual aircraft and track it on a map. During which, the information for each aircraft will be updated in real-time for the user. There are a number of features that we would like to achieve. However due to time constraints, we are limited to a set number of those features that we will implement in the Android application. These features to be implemented in this application includes: accessing the application, viewing a list of possible aircrafts that are seen by the receiver, viewing the information for each of those aircrafts, updating the information for each in real-time, displaying the location of the aircraft that the user have selected, and finally decoding any encrypted data streams that are sent to application.

3.3.3.1. Bluetooth API

When it comes to how the application is going to obtain the data from the receiver, Android has tools to deal with this situation. Since the receiver will be sending aircraft information by Bluetooth, our application will include the Bluetooth API. This API allows our application to locate and receive information from the Bluetooth adapter that will be built into the receiver. However, the application can only be run on devices that have a built-in Bluetooth adapter.

Table 22 display the various functions from the Bluetooth API that will be used for the application.

Name	Function
BluetoothAdapter Class	This class is used to perform the fundamental tasks for Bluetooth. For example to discover other Bluetooth enable devices
getDefaultAdapter()	Returns an object that represents the Bluetooth adapter built into the device
startDiscovery()	Used to locate other Bluetooth devices
getBondedDevices()	Return a list of devices connected to the current device
BluetoothSocket Class	This class is used to manage the Bluetooth connection.
connect()	Used to connect to a Bluetooth device
getInputStream()	Used to locate the input stream of the connected device
getOutputStream()	Used to locate the output stream of the connected device

Table 22 Bluetooth API [8]

3.3.3.2. Google Maps API

In order to display the locations of any aircraft, the application will be implementing the Google Maps Android API v2. This API allows the application to uses maps based on Google Maps. In addition, it will allow our application to display and update markers from the data being received from the receiver. Table 23 display the various functions from the Google Maps Android API v2 that may be used in the application.

Name	Function
GoogleMap Class	This class is used to gain access to all the methods that can be used to deal with the maps.
addMarker()	Used to add a marker to a map
animateCamera()	Used to animate the movement of the camera on the map
MapFragment Class	This class is used to add maps to an application.
getMap()	Used to obtain the GoogleMap for the fragment
newInstance()	Used to create a new Map fragment

Table 23 Google Map Android v2 API [14]

3.3.3.2. Graphical User Interface

The graphical user interface is going to be used to display the information obtained from the receiver. The general plan is for a simple and clear interface. The general function of the interface is when opening the application it will display a title screen. Afterwards, there will be a menu screen where the user can select an option. If the user selects to view aircraft, the application will transition to a screen where a map is displayed and markers for all aircrafts within range will appear. From there, the user can select an aircraft and view its statistics.

3.3.3.3. Related Interface Issues

When designing an application for the mobile market the need for a user interface that is both compact and uncluttered becomes an obvious challenge due to the screen size of most mobile devices. Other challenges will be to implement a design that will suit the variety of aspect ratios that the application could encounter upon release.

Other thoughts on layout design include portrait versus landscape modes and how the layout will adjust itself upon the user's rotation of the device. Touch feedback is an option that is available but for the type of application that we are developing it does not seem necessary to implement. Touch feedback is more reserved for application where the user will be rushed and will desire feedback to know the desired selection has been implemented. It may be an implementation that could appear in later updates of the software.

When placing menus and buttons in the main viewing location they should not be overly interceptive of the main index of the application, in our case this would be the map that will give a visual representation of the location and heading of the planes in the airspace.

Menus should have very definite indexes that will lead directly to the desired and predictable selections. Anything that requires traversing through more than three lists should be reconsidered for alternative placement in list traversals.

Android's development suite has many built in building blocks that developers are able to build off of by either expanding upon or modifying to suit their specific needs or preferences for a task.

- Tabs
- Lists
- Grid Lists
- Scrolling
- Spinners
- Buttons
- Text Fields
- Seek Bars
- Progress & Activity
- Switches
- Dialogs
- Pickers

3.3.3.3. Security

In terms of security, Android was designed to protect the information of the users, the applications, the device, and any network connected to the device. In doing so, the application is inherently secure. This is important because the data being transmitted from the receiver will often be open and not encrypted. As an added benefit, we have access to an encryption package which will make it easier for us to decrypt any aircraft information should the application have to decode a data stream.

3.3.4. Format-Preserving Encryption

Format-Preserving Encryption (FPE) was previously discussed in section 2.4.4 as a method to encrypt ADS-B message data without requiring vast changes in the already established air traffic control (ATC) infrastructure. One of the Air

Force Institute of Technology (AFIT) proposals involved encrypting only the last 104-bits of the ADS-B extended squitter message format. Furthermore, AFIT proposed using AES as the underlying encryption algorithm within the larger scheme of FFX encryption as illustrated in the FFX documentation.

The acronyms FPE and FFX are closely related, and require more explanation to clearly delineate their meaning. FFX stands for Format-preserving, Feistel-based encryption. (Feistel-based is a reference to the high symmetry in the algorithm between encryption and decryption operations). FFX is a proposed NIST standard encryption scheme, and FPE is what FFX accomplishes.

It is of critical importance to recognize that this combination of new schemes and protocols with existing ADS-B technologies is a completely untested idea. It is very likely that unforeseen problems will arise in regard to the scope of the partial message encryptions. The actual bits of the message format targeted for encryption may need to be shifted into alignment with other field boundaries within the message as the project proceeds. This should not be viewed as a deviation from the recommended or planned approach, but rather as a needed adjustment made apparent by testing. FPE permits us to encrypt what we need to encrypt without regard to a minimum block size or number of bytes.

Likewise, the anticipated use of AES as the underlying encryption algorithm is also likely to trigger performance issues in the timely processing of the ADS-B data streams. The actual underlying encryption algorithm used may need to be one that is less computationally intensive. Particularly during unit testing, a simpler encryption scheme would aid in debugging unexpected results. Again, this is not a deviation that would render the project as unsuccessful. It is an adjustment shown to be necessary during testing. FFX permits the use of other underlying algorithms.

Regarding the FFX algorithm, it is not yet an accepted NIST standard. It is a proposed NIST standard. This further highlights the experimental nature of the project. However, this does not mean that FFX lacks definition. FFX is parameter driven. From the proposed standard, (Bellara), the parameters are shown in figure-24:

parameter	description
radix	The *radix*, a number radix ≥ 2 that determines the alphabet Chars $= \{0, \ldots, \text{radix} - 1\}$. Plaintexts and ciphertexts are strings of characters from Chars.
Lengths	The set of *permitted message lengths*. For a plaintext to be encrypted, or for a ciphertext to be decrypted, its length must be in this set.
Keys	The *key space*, a finite nonempty set of binary strings.
Tweaks	The *tweak space*, a nonempty set of strings. Conceptually, different tweaks name unrelated encryption mappings.
addition	The *addition operator*, either 0 (characterewise addition) or 1 (blockwise addition). Determines the meaning of the operators $X \boxplus Y$ and $X \boxminus Y$ that add or subtract equal-length strings over the alphabet Chars $= \{0, 1, \ldots, \text{radix} - 1\}$.
method	The *Feistel method*, either 1 or 2. The value determines which of the two prominent Feistel variants will be used.
split (n)	The *imbalance*, a function that takes a permitted length $n \in$ Lengths and returns a number $1 \leq \text{split}(n) \leq n/2$.
rnds (n)	The *number of rounds*, a function that takes a permitted length $n \in$ Lengths and returns an even number rnds(n).
F	The *round function*, a function that takes in a key $K \in$ Keys, a permitted length $n \in$ Lengths, a tweak $T \in$ Tweaks, a round number $i \in \{0, \ldots, \text{rnds}(n) - 1\}$, and a string $B \in \text{Chars}^*$. It returns a string $\text{F}_K(n, T, i, B) \in \text{Chars}^*$. If method $= 1$ or i is even then $\lvert B \rvert = n - \text{split}(n)$ and $\lvert \text{F}_K(n, T, i, B) \rvert = \text{split}(n)$. If method $= 2$ and i is odd then $\lvert B \rvert = \text{split}(n)$ and $\lvert \text{F}_K(n, T, i, B) \rvert = n - \text{split}(n)$.

Figure 24 – FFX Parameters

This approach suggests a possible bias toward procedural languages, but is certainly not exclusive to that orientation. What is more important is how it works. Typical of symmetric encryption schemes, the data is obscured progressively through round after round of substitution and shuffling. What is different in FFX, is that it employs a second encryption algorithm within each round – essentially encrypting encrypted data.

The following example, (figure-25,) also from the proposed standard, [Bellara] illustrates four rounds of FFX using what is termed the 'left' method. This is what occurs when the method parameter is set to 1. The input data, shown as the block labeled A_0 followed by B_0 is an unspecified fixed length n. The B portion is encrypted with algorithm F and recombined with the A portion to create a new encrypted A portion. This is prefixed with the prior B portion to complete one round. Note that the input data and the output data are exactly the same size after every round. This would be an exploitable weakness in most symmetric algorithms – but not in FFX. Each new round is using variations introduced according to the parameters. All this is undone in reverse for the decryption side of the operation.

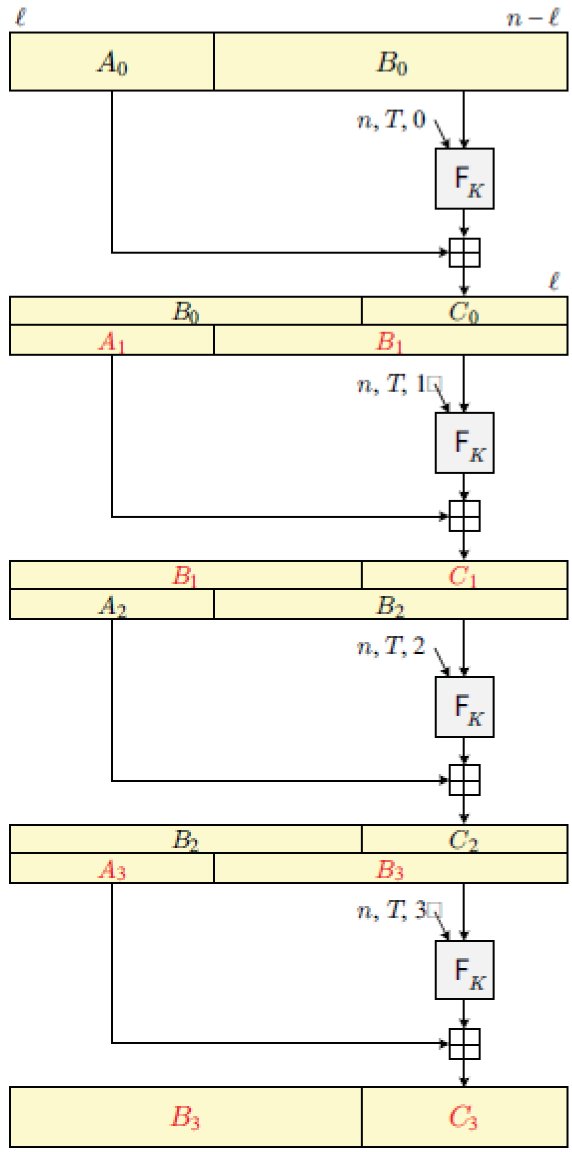


Figure 25 – Four rounds of FFX using method=1

4. Project Hardware and Software Design

4.1. Initial Design Architecture

The design of the project was the result of the specifications, requirements and research from each of the team members. In general, the design of the project will contain a hardware and software component. The hardware component is made of an antenna and a receiver. With the software component, this would be the Android application. Figure 26 gives a guided idea of the desired design for the project. The following sections will expand on this diagram by giving an in depth look of each block.

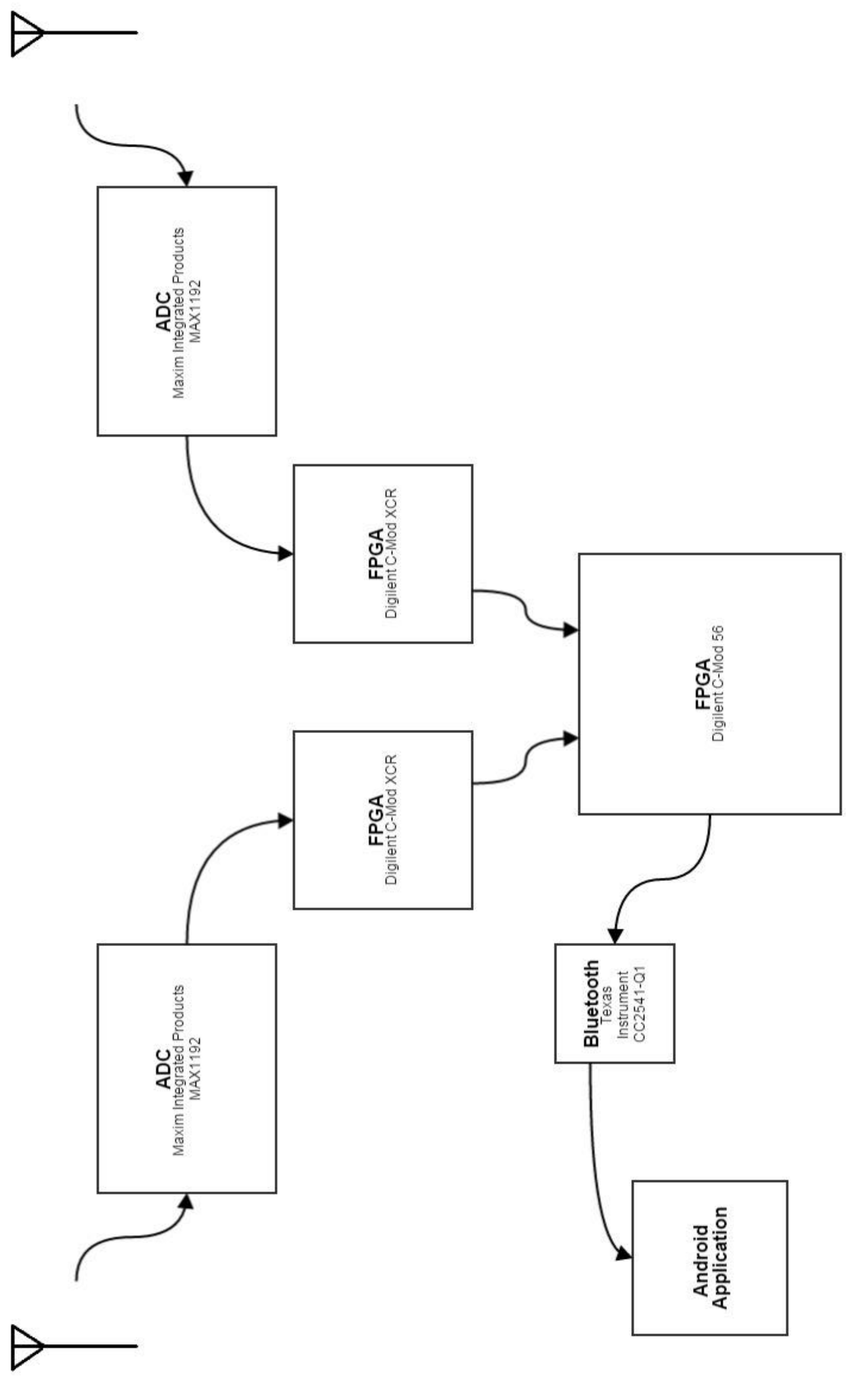


Figure 26 Initial Design Block Diagram

4.2. Antenna Design

4.2.1. Signal Polarization

Radio signals travel in a linearly polarized manner linearly to the plane in which the antenna is facing. While signals that are refracted will have their polarization shifted, as long as line of sight is maintained with the source, then the expected polarization will be maintained and can be designed specifically for it.

To receive the vertically polarized signals an antenna will be designed with a vertically mounted element (figure-27) or through multiple elements working together. Another benefit to mounting the elements vertically will help ensure that any erroneous signals that have a horizontal polarization, either because of some refraction that may have occurred or in some other way, do not insert themselves into the receiver. Eliminating unnecessary signals to the receiver alleviates unnecessary filtering and/or processing that must be performed.

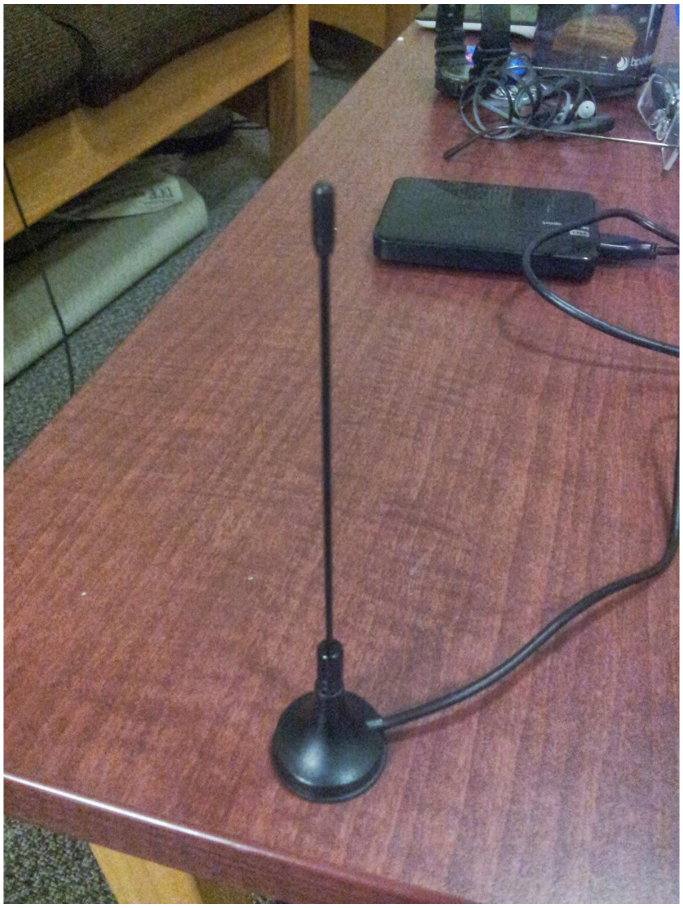


Figure 27 - Simple Vertically mounted dipole antenna

There are multiple things to take into consideration when designing an antenna that will be receiving signals that are expected to ideally be of a specific polarization but due to environmental reasons cannot be transmitted ideally. To combat this effect, the antenna will have to be designed to still accept polarized signals that are not of the ideal transmission state.

The formula for determining the polarization mismatch due to the misalignment of the polarity from the signal and the receiving antenna(s) is as follows.

$$m_p = cos^2(\theta / 2)$$

θ is the angle between the receiving antenna and the direction that the incoming wave is originating from.

The power that is received when the polarization has no polarization mismatch will ideally be equivalent to the strength of the signal. I there is polarization mismatch due to the signal not being received at the ideal angle then the strength of the signal will be weak and the power associated with it will be reduced. The equation for the power received is as follows:

$$P_r = S_{av} A_e m_p(a, w) = P m_p(a, w)$$

S_{av} is the incoming waves average flux density,
A_e is the effective aperture of the antenna.

Therefore the maximum power that the antenna can receive is equivalent to:

$$S_{av} A_e$$

The incoming angle of the ADS-B signal will be out of our control due to the originating location being on a plane that will have to pitch and yaw and it navigates its intended flight path. To make sure that the receiver has enough gain to be capable of making use of the signal the antenna will have to be able to maximize its efficiency to minimize missed signals due to the polarization mismatch.

4.2.2. Coaxial Collinear Options

Options for improving signal gain without implementation of an amplifier includes the use of arranging multiple identical antennas in connected together to act as one larger antenna. This arrangement of antenna design is referred to as a collinear antenna array. An importance with the collinear design is that all the sections must have equal lengths so that they receive the same frequencies.

Another benefit for an array of antennas is an improved Signal to Interference Plus Noise Ratio. This allows for a clearer signal by allowing the antenna to recognize erroneous signals and eliminating them from the signal that is meant to be received.

What this allows for is the reception of the desired signal at multiple locations that when the signals are combined will hopefully amplify the signal received and make the received signal enhanced by reducing signal noise.

Our arrangement for a collinear antenna array design (figure-28) will include a series of two or more dipole antennas arranged vertically using a simple but highly effective design that utilizes basic coaxial cable similar or exactly the same as the coax cable used for transferring television signal.

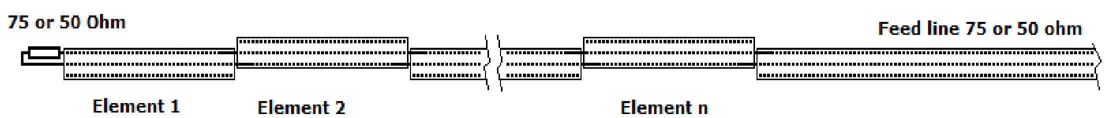


Figure 28
Diagram of how elements will be connected together
Reprinted with permission of Balarad.net

When designing a coaxial collinear antenna it is important to take into account the admittance of the antenna and at each of its sections. The formula for the admittance at a particular section is denoted as:

$$y_n = y_{rn} + y_J$$

y_{rn} is the radiation admittance
y_J is the shunt admittance

The admittance of the antenna will determine the ease that the signal will be received by the element. If the admittance is too low then the signal strength received will be weak in comparison to an antenna with elements of a high admittance.

4.2.3. Pre-Amplification

Amplification for the system will occur in the receiver which is to be implemented by the SDR. Due to the weak nature of the signal that is to be received it is unlikely that it will be implemented in our system. This is because of the fact that the signal receive will be too weak to amplify without introducing noise into the system. Any noise that we introduce to the receiver would cause erroneous results, and therefore it would be better if the signals that are too weak to be accurately received are just ignored by the receiver rather than attempt to include them.

Building a pre-amplifier of the necessary signal-to-noise ratio would result in a significant increase in the cost associated with the receiver system. As we do not anticipate the need to increase the amplification of the signals received do to the fact that the signals that would be needed to be amplified will be signals originating from distances outside of our range of requirements. This coupled with the cost requirement has eliminated the need for a pre-amplification circuit.

4.2.4. Multilateration

Multilateration is the utilization of the received time of a signal compared between multiple locations and then uses that information to locate the source of the signal. In order for this technique to work, the times of the location must be synchronized to provide accurate time differences between the reception locations.

This technique would be used as a way of verifying the general location of the source of a transmission. In the case of a plane transmitting a GPS coordinate that is then triangulated via multilateration to be in a different location than the source of the signal’s origination then there could be errors with the GPS system on board the plane or someone could be attempting to spoof the the FAA by broadcasting a nonexistent plan. In the case of the latter issue then greater security measures would have to be taken to prevent future occurrences and remove the false signal or ensure that planes in the airspace know to ignore the false signal.

4.2.5. Coupling

Antenna coupling is the designing of a series of antennas so that signals from one antenna reach the receiver even if the energy signature does not reach any of the other antennas in the series. This is a design that will be implemented if in order to capture signals of both ADS-B signals if a single antenna design is incapable of such a task. By utilizing a coupling circuit to merge the received signals from both antennas eliminates the need to switch between the two antennas and causing the receiver to receive only signals from one antenna. This option will increase the expense of our design due to the added design and build costs over a single design but it should greatly enhance reception of signals as long as the two antennas are kept at a distance that does not allow for them to interfere with each other.

Due to the portability requirement of this project the antennas will have to be within a few feet of each other located within close range to the receiver. If possible they will be mounted together on ballast.

Things to be aware of when designing include impedance and current distributions. Formulas for calculating these are as follows.

$$Z = -(V \div I)$$

Z is the received impedance with a received current at the second antenna
V is the coupled voltage at the first antenna
I is the received current at the second antenna

This formula is equivalent to the following formula.

$$Z = -((V - U) \div I)$$

U is the voltage at the first antenna when isolated from the circuit.

4.3.1. Bandpass Filters

Implementing a bandpass filter prior to the receiver and following the antenna(s) will help to remove extraneous signals that will have been accepted by the antenna. Even though the SDR will be performing filtering of signals, having the physically filter will aid the SDR in that it will hopefully remove a large portion of the signals that are undesirable. This is more for a precaution rather than a necessity due to the filtering aspect of the SDR requiring the most processing. Because the SDR will still be performing filtering on the signal and to save cost, a low complexity, low cost, dual wide band filter will be implemented that allows a range of frequencies originating from the 978-MHz and the 1090-MHz frequencies. Or if it turns out that the two frequencies are too close together and two separate bands cannot be distinguished, then a single bandpass filter with a lower Q value will be implemented.

4.3.1.1. Placement of Filter

Placing the filter after the merging of the signals received by the antenna would allow for increased gain if it so happens that the one antenna happens to pick up signals that were originally intended to be received by the other. However, due to the nature of the increased complexity and therefore cost of such a design it will most likely be practical to filter the signals immediately following each of the antennas. This implementation of the filtering of our signals will be a more practical implementation even though we would now need to design two separate bandpass filters though they can be of a simpler design. It is believed though that the decreased complexity of two separate simpler designs will offset the cost of having to do one design of a higher complexity.

Unfortunately, the need of the FPGAs to each need to receive signals of independent frequencies. So the need to two separate filters will be needed, one for each frequency to be converted, thus requiring one filter prior to each FPGA.

4.3.1.2. Component Filter

The resistive load of the filter circuits will be the resistance of each antenna. Designing around the experimental resistance of the antenna and then fine tuning the the filter design will pose the best way of matching the ideal resonant frequency of the antenna with that of the ADS-B signals' frequency. Each filter would most likely be a first-order bandpass filter, this maintains the simplicity and effectiveness that is required. The increased cost and complexity of a second-order bandpass filter would only increase the steepness of the frequency slope; this does not increase the effectiveness of the filter for our needs even though it would make the cutoff frequencies more accurate.

4.3.1.3. Ribbon Filter

Designing a filter for the very high frequencies that this project will operate at become very difficult when designing will typical components used for filters such as resistors, capacitors, and inductors. Because of this difficulty we will instead implement a distributed element filter. The design for the filter will be composed of a distributed element filter also referred to as a hairpin filter due to the hairpin turns of the design. Figure-29 is the preferred choice because of the high frequencies of the signals and its ability to be etched directly onto a PCB board. While this will add more designing to the PCB is will greater simplify the effectiveness of the circuit.

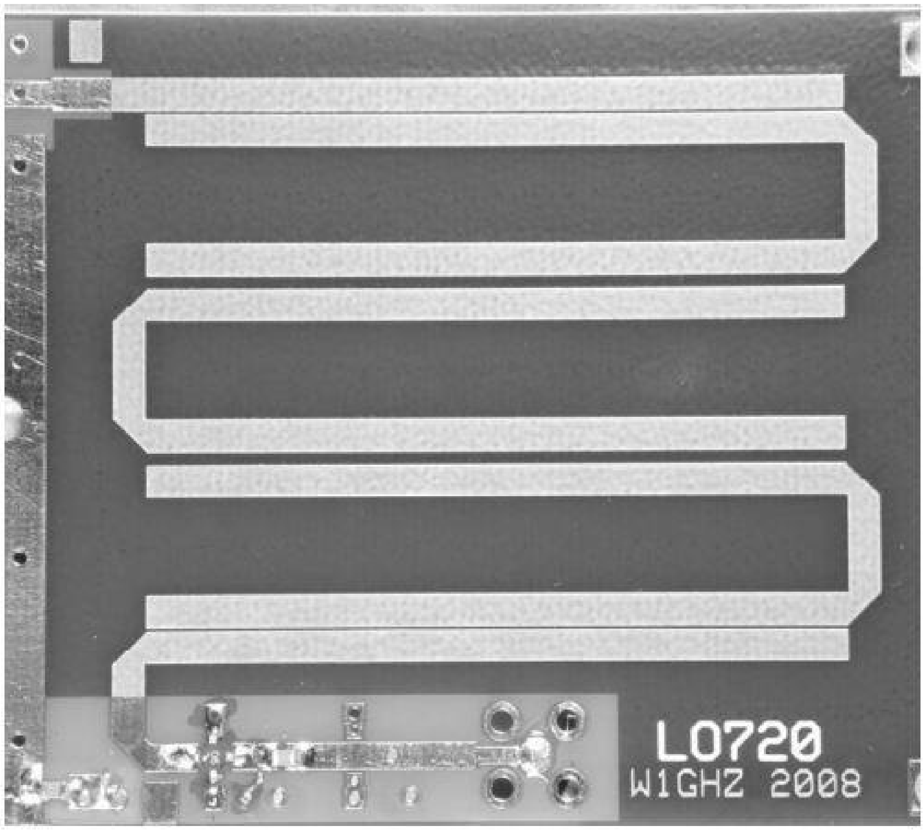


Figure 29
Hairpin Filter on a PCB
Permission granted by Paul Wade

While upon first looking at the circuit design it is noticed that this is an open circuit, but signals are transmitted via induction onto the neighboring copper trace.

When designing the filter the dielectric constant of the PCB board itself must be taken into consideration as it can affect the way signals are transferred. The length of the ribbons is a major consideration (figure-30) as it is the major factor that dictates the center frequency of the filter.

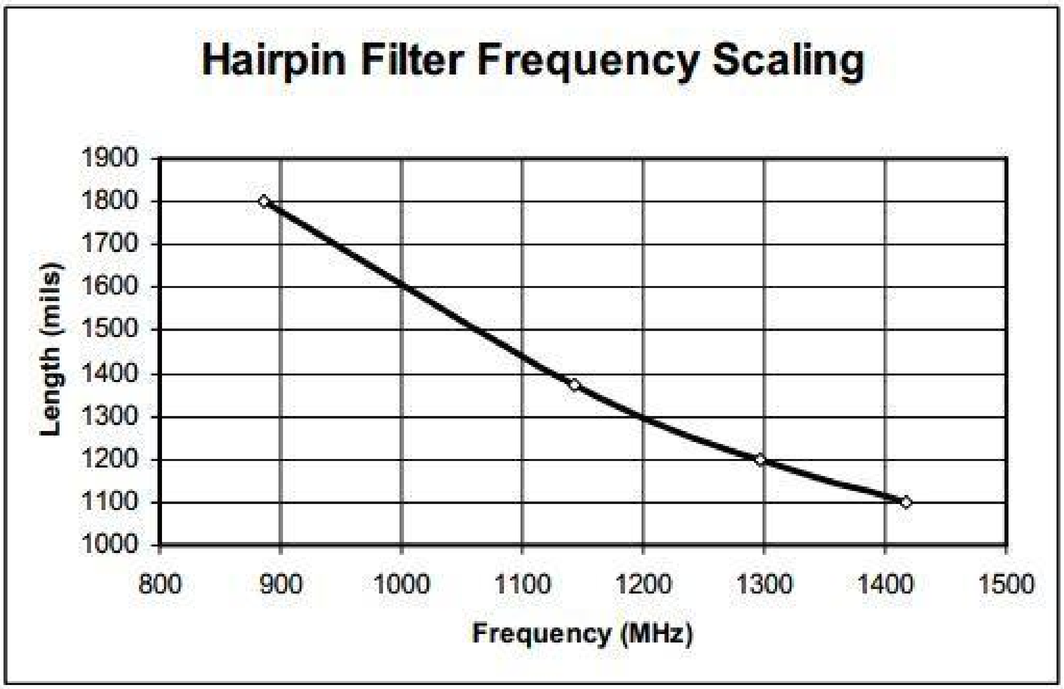


Figure 30
Center frequency of filter based on dimensions
Permission granted by Paul Wade

4.3.2. Log Detector/Controller

From the filters, the signal will have to go through one more step before being converted to a digital signal. At this point, the signal has a wide range that will be problematic for the analog digital converter to deal with. To mediate this issue, a log detector/controller will be used. The purpose of the log detector/controller is to convert this wide range, low power RF signal to a smaller-scale DC output. This output will then be fed to the ADC for additional processing. To complete this task, we have decided to use Analog Devices AD8319 Log Detector/Controller. How the AD8319 achieves this task is that the chip is comprised of a series of Op-Amp that continuously test the input signal. The values obtained from the detector from each Op-Amp are then summed together and then filtered. From there, the corresponding output voltage is sent. Figure 31 shows the block diagram for the AD8319.

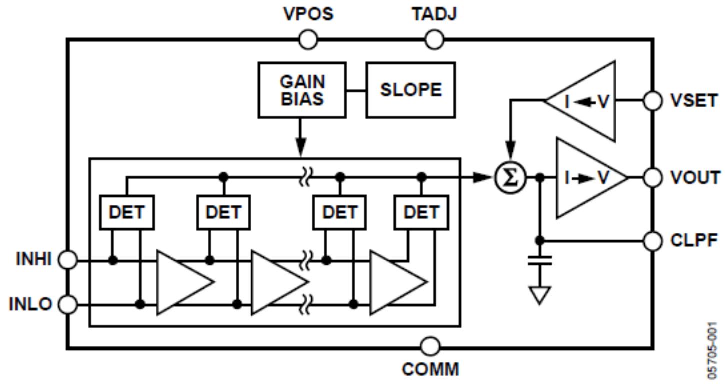


Figure 31 AD8319 [54]
Copyright Analog Devices (Permission Pending)

4.3.3. Analog Digital Converter

4.3.3.1. Selected Analog Digital Converters

After receiving the signal, there are a number of steps that have to be taken in order to actually use the data. One of the more important steps includes converting that analog signal to a digital signal where we can process that data. In order to simplify the task, we have decided to select a prefabricated analog digital converter instead of having to construct one by ourselves. The benefits of using one is that they have improve precision and have a number of features that we can use compared if we were to start from scratch. When it comes to selecting an analog digital converter, there are a set of properties that we are looking for. These properties includes: resolution, sampling rate, power consumption, and etc. With this in mind, the analog digital converters we have selected are the MAX1192 by Maxim, ADS930 from Texas Instruments, and AD9281 from Analog Devices.

For the design of the receiver, we will begin with the first ADC, the MAX1192 from Maxim Integrated. The MAX1192 uses a pipeline architecture for its conversion method and have two channels for inputs. The ADC features three power saving modes that can really help when it comes to conserving power. Under normal operations, the converter has a sampling rate of 22 mega-samples per second (MSPS) with a power consumption of 27.3mV [17]. This is quite impressive to get this kind of performance for so little power. The signal to noise ratio and signal to noise and distortion ratio is 48.6 dB which is high enough to support this application [17]. The converter outputs the analog signal as an 8-bit digital signal by a parallel CMOS interface which allows fast transfer of data to the decoder [17]. When not in use, the MAX1192 can consume as little 1.8µV when in shutdown mode. The operating voltage for the MAX1192 is from 2.7V to

3.6V [17]. In addition, the reference voltage can be set internally or externally. Figure 32 shows the functional diagram for this ADC.

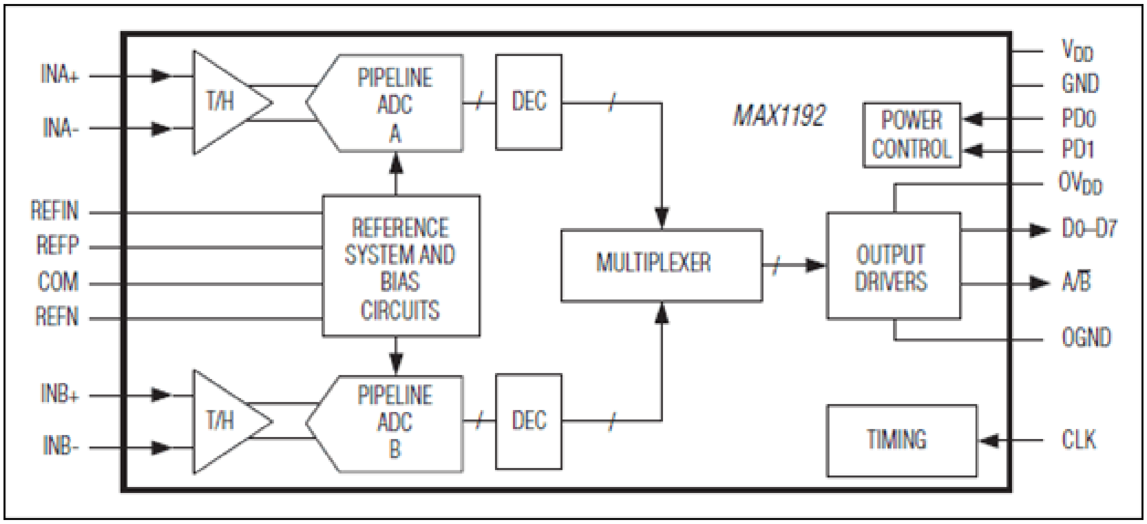


Figure 32 MAX1192 Copyright Maxim Integrated Products [17] (http://www.maximintegrated.com). Used by permission.

The analog digital converter next on the list is the ADS930 from Texas Instruments. The ADS930, like the MAX1192, uses a pipeline architecture for its conversion method. It has a single channel for inputs with a sampling rate of 30 mega-samples per second (MSPS) [3]. This gives this analog digital converter the highest sampling rate on the list. A useful feature in the ADS930 is a built-in error correction module to improve linearity. The signal to noise ratio is 46 dB and signal to noise and distortion ratio is 45 dB [3]. This converter outputs the 8-bit digital signal via a parallel CMOS interface. When it comes to power consumption, the device uses 66mV [3]. Like the MAX1192, the reference voltage can be set internally or externally. Figure 33 shows the functional diagram for this ADC.

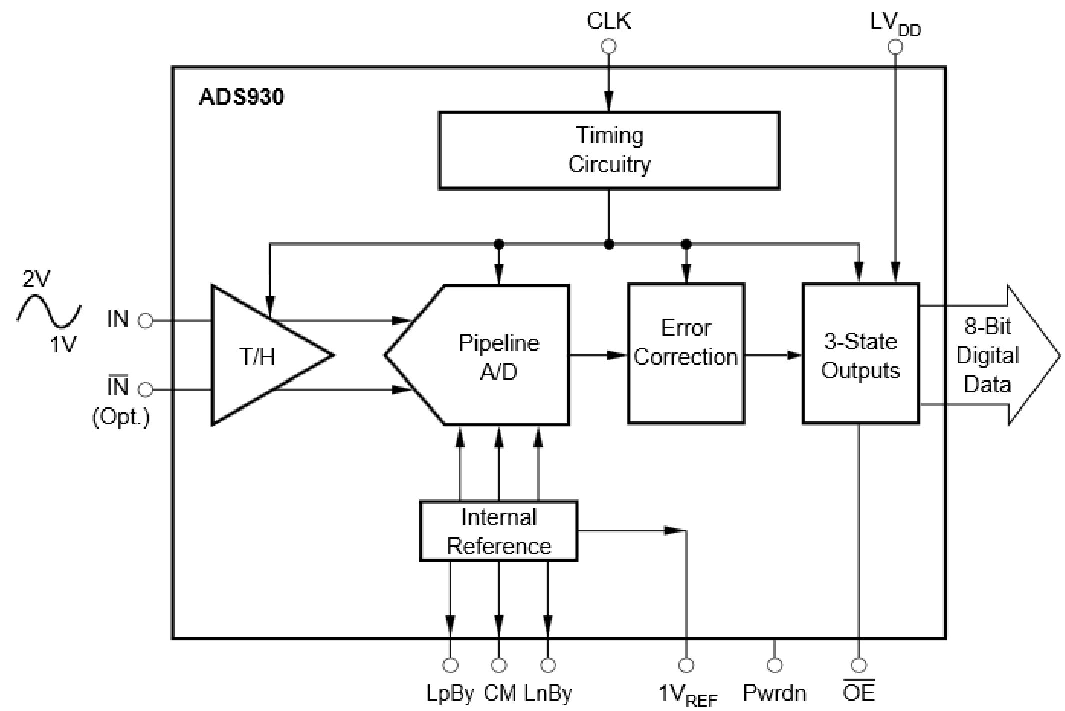


Figure 33 ADS930 [3]
Courtesy of Texas Instruments

Finally, we have the AD9281 from Analog Devices. The AD9281 uses a pipeline architecture for the analog digital conversion. It has two channels for inputs with a sampling rate of 28 mega-samples per second (MSPS) [2]. The signal to noise ratio is 48.5 dB and signal to noise and distortion ratio is 48 dB [2]. The output is an 8-bit digital signal using a parallel CMOS interface. When it comes to power consumption, the device uses 225mV with an operating voltage of 2.7V to 5.5V [2]. In addition, the reference voltage can be set internally or externally. Figure 34 shows the functional diagram for this ADC.

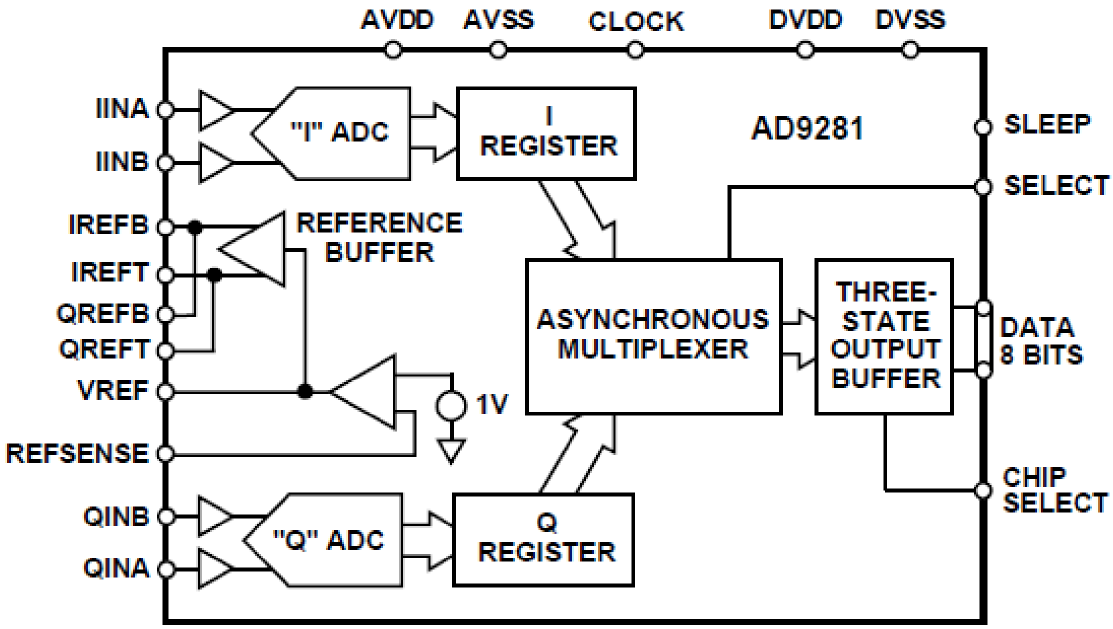


Figure 34 AD9281 [2]
Copyright Analog Devices (Permission Pending)

4.3.3.2. Comparison

Table-35 below gives a better comparison between each of the selected analog digital converter.

	MAX1192	**ADS930**	**AD9281**
Architecture	Pipeline	Pipeline	Pipeline
Resolution (bits)	8	8	8
# of Input Channels	2	1	2
Sample Rate (MSPS)	22	30	28
SNR (dB)	48.6	46	48.5
SINAD (dB)	48.6	45	48
Output Interface	Parallel	Parallel	Parallel
Operating Voltage	2.7V to 3.6V	3V to 5V	2.7V to 5.5V
Power Consumption	27.3mV	66mV	225mV

Table 35 Comparisons of Analog Digital Converters [2][3][17]

As the table shows, each converter is quite similar to each other with a handful of differences in-between. They all are using the pipeline architecture with the same number of output bits and output interface. That is where the similarity ends; the ADS930 has the highest sampling rate of the three. In exchange, it only has one channel for inputs compared to the other two. The MAX1192 has the lowest power consumption compared to the three. The AD9281 is essentially the middle of the road between the MAX1192 and ADS930 with sampling rate and operating voltage. However, the high power consumption makes it less desirable for the receiver. In the end, we have selected the MAX1192 as the best fit for the receiver.

When it comes to the design of the receiver, the analog digital converter will interact with the antenna and convert the signal so that the decoder can translate and process the data. To start, we are looking for a resolution of 16 bits for the decoder. In this case, we will need two converters with each external reference voltage set to a difference value. From the figure the antenna will be connected to either INA+ or INB+ with D0 to D7 connected to FPGA. V_{DD} is connected to the power source to enable operation and A/B will be set to 0 or 1 depending on the channel. Reference voltage will be set on the REFIN pin with CLK as the input for the clock.

4.3.4. FPGA Decoder

As first discussed in section 3.2.3, a two tier FPGA architecture will be employed to create a large reserve capacity for experimentation with hardware-based decryption and to separate the essential Manchester decoding from later functions like the merger of the ADS-B data streams from the two frequencies being received. To explain this in more detail, one must recognize that our receiver is a “dual” receiver; meaning that both the 1090 MHz and the 978 MHz frequencies are received simultaneously. Simple filters are used to suppress any other frequencies. There is no tuner or “frequency-hopping” circuit to limit the inbound data to a single channel at a time. The signals remain separate and simultaneous thru the RF portion of the design. Two ADC circuits sample the signals at the same time.

Consequently, the first tier is comprised of two small FPGAs; one for each Maxim 1192 ADC. The 8-bit output of each ADC dictates these FPGAs accept 8-bits. Manchester decoding then takes place. Ideally, these two FPGAs will run at eight times the clock speed of the ADCs, but half that speed can still be made to work efficiently. Specifically, faster than five times the incoming bit rate to correctly detect the next bit transition, and less than twelve times the incoming bit rate to suppress any between-bit transition. Fortunately, this phase of the decoding can be accomplished in less than 1500 gates, (i.e. approximately three complex logic blocks.) A reference design for the decoder and corresponding test fixture from Xilinx to perform this function on their very inexpensive CoolRunner-II series of FPGAs.

The second tier of this FPGA cascade takes both decoded ADS-B data streams and merges them together. Unlike the first two FPGAs doing the Manchester decode, this one needs at least 16 GPIO and enough logic space to emulate a small buffer in which to combine and order the results. The Xilinx Spartan-6 FPGA meets this need with enough reserve capacity to address our objective to experiment with the location of the decryption functions.

An advantage here is that in the case of both types of FPGAs, they are available in socket able DIP packaging to simplify assembly and facilitate easy reprogramming. These DIP packages are compatible with the standard JTAG programming interface cable for embedded devices. A disadvantage is that there are outstanding questions about the clock speed requirements in the second tier. In time, with experimentation, it may be possible to combine all these logical functions in a single FPGA to reduce costs further. However, the two tier approach allows us to limit our risk to just the second tier.

4.4. Mobile Application Design

4.4.1. Architecture and Constraints

The mobile application plays an important role to the project. The application is what will give us a visual representation of the ADS-B signal from the antenna and will allow us to decode any encrypted signals. To begin, the software architecture for this application will be pipe and filter. The pipes would include the state of the map and aircrafts, the updated information for those aircrafts and the user's inputs. The filters would be the changes to the graphical user interface and the aircraft information. The software development model that will be used is the agile development model. The reason for this choice is that it will allow us to have a useable application after iteration. By following this model, we can focus on making sure that the program is working correctly before moving on to optimizations and improvements. With the application, there are a set of requirements that will have to be fulfilled for it to work properly. Table 37 shows the list of requirement that we hope achieve.

Software Requirements
The application shall perform on Android device running on Android version 4.2 and higher.
The application shall display all aircrafts within the phone's GPS location that the decoder received
The application shall be able to process the data that was obtained from Bluetooth.
The application shall display aircraft information when selected by the user.
The application shall allow the user to select an aircraft that is displayed.
The application shall update aircraft data in real-time.
The application shall recognize an encrypted data stream from the receiver.
The application shall be able to decode an encrypted data stream.

Table 38 Mobile Application Requirements

When it comes to constraints, there are a few situations we have to deal with to allow the application to be effective. First, the application can only be ran on Android devices that are using at least Android 4.2 Jellybean. Next, the Android device that is being used must be able to receive Bluetooth signals. Finally, the device will have to be close enough to the receiver for the application to work. When it comes to designing the application, models and diagram can help us simplify the work. The class diagram, shown in Figure 38, is one of those diagrams.

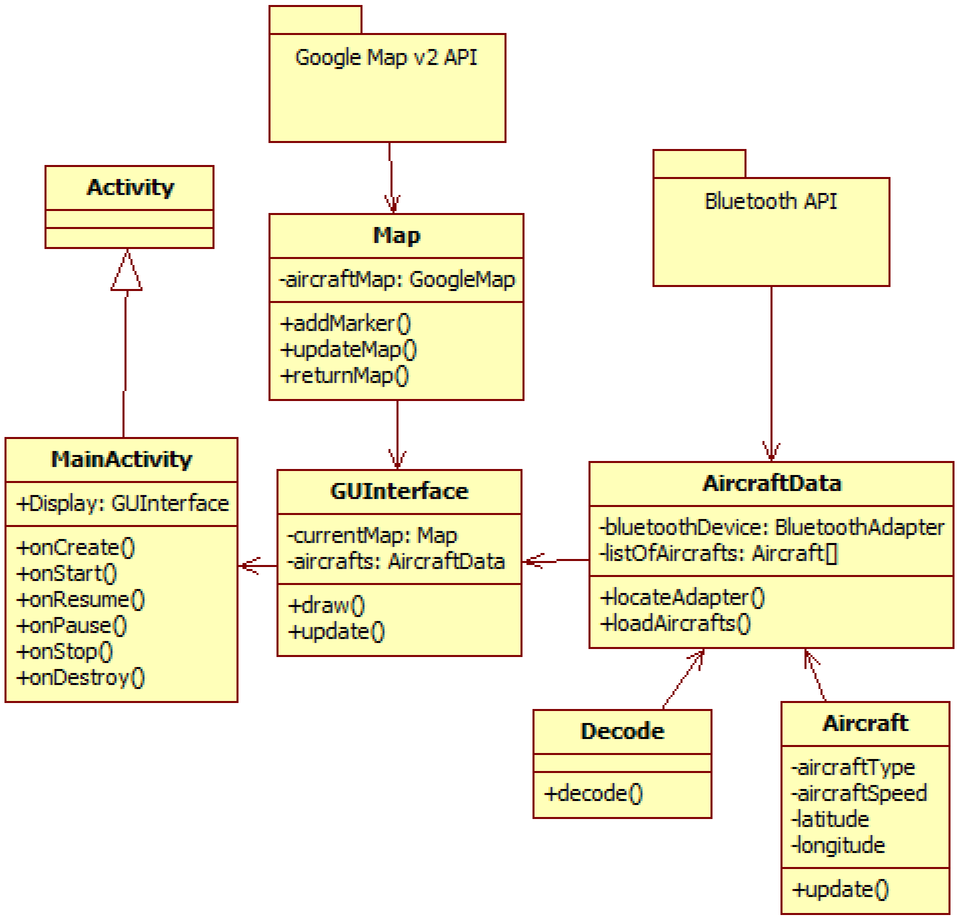


Figure 39 Class Diagram

MainActivity - The MainActivity class implements the Android Activity class. MainActivity is responsible for managing the application's lifecycle from where it starts to when it ends. This class also deals with the displaying the user interface and the all of the user's inputs.

Aircraft - The aircraft class is used to store the information for each individual aircraft. After the data is processed, a new aircraft object is created and is updated periodically by the AircraftData class.

Map - The Map class implements the Google Map android v2 API. This class is used to display the aircraft markers and to update the map. User inputs from the MainActivity and the data from the AircraftData will be sent here.

GUInterface - The GUInterface class manages the graphical user interface that will display to the user. This includes menus, buttons, and the images from the AircraftData class.

Decode - The Decode class deals with the decrypting an encrypted message. When an encrypted signal is sent to the AircraftData class, this class will be called to decode the message.

AircraftData - The AircraftData class is used to process the data from the Bluetooth adapter. This class will use the Bluetooth API to locate the Bluetooth microcontroller on the receiver and process the signal obtained.

The class diagram give us a useful view on how the application will be developed. The activity diagram, shown in Figure 39, shows the processes and situations that will occur as the program is running.

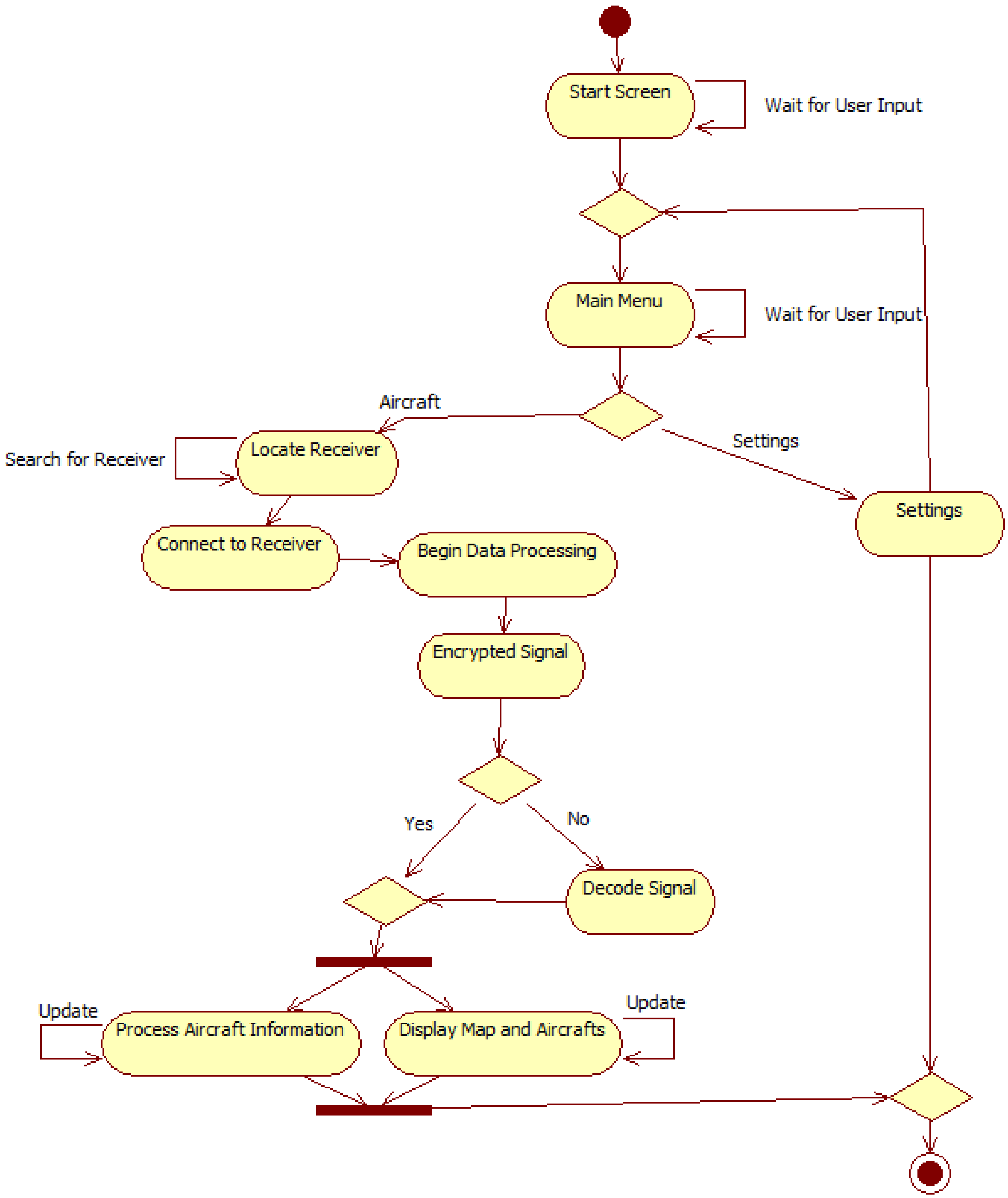


Figure 40 Activity Diagram

When the applications begins, user will see a start screen. During which, the application will wait for an input from the user to proceed. This process will repeat after the start screen with the main menu. When the user select the setting option, the user will be able to modify some of the settings to the application like locating and selecting a different receiver and changing the looks of the icons. If the user was to select view the aircrafts, the application will begin locating a Bluetooth signal emitting from the receiver. After connecting, the application will begin to process data. If the signal is encrypted, decode the data and then display it to the screen. If not, display the data straight to the screen for the user to view. As the user view the aircraft information, the program will continuously update the information while updating the user's view. From each of the options, the user can exit from the application.

4.4.2. Bluetooth Communication

Bluetooth communication plays an important role for the application. If the signal from the receiver to the app is loss, the application will not function correctly. In doing so, selecting the correct Bluetooth microcontroller unit play a vital part. With that, we have decide to use either the CC2541-Q1 from Texas Instruments or the PmodBT2 from Diligent, Inc. for this application.

To start, we will begin with the CC2541-Q1 from Texas Instruments. Since this part will play an important role to the project as a whole, the following is a simple overview of the unit. Figure 41 shows the block diagram for this microcontroller.

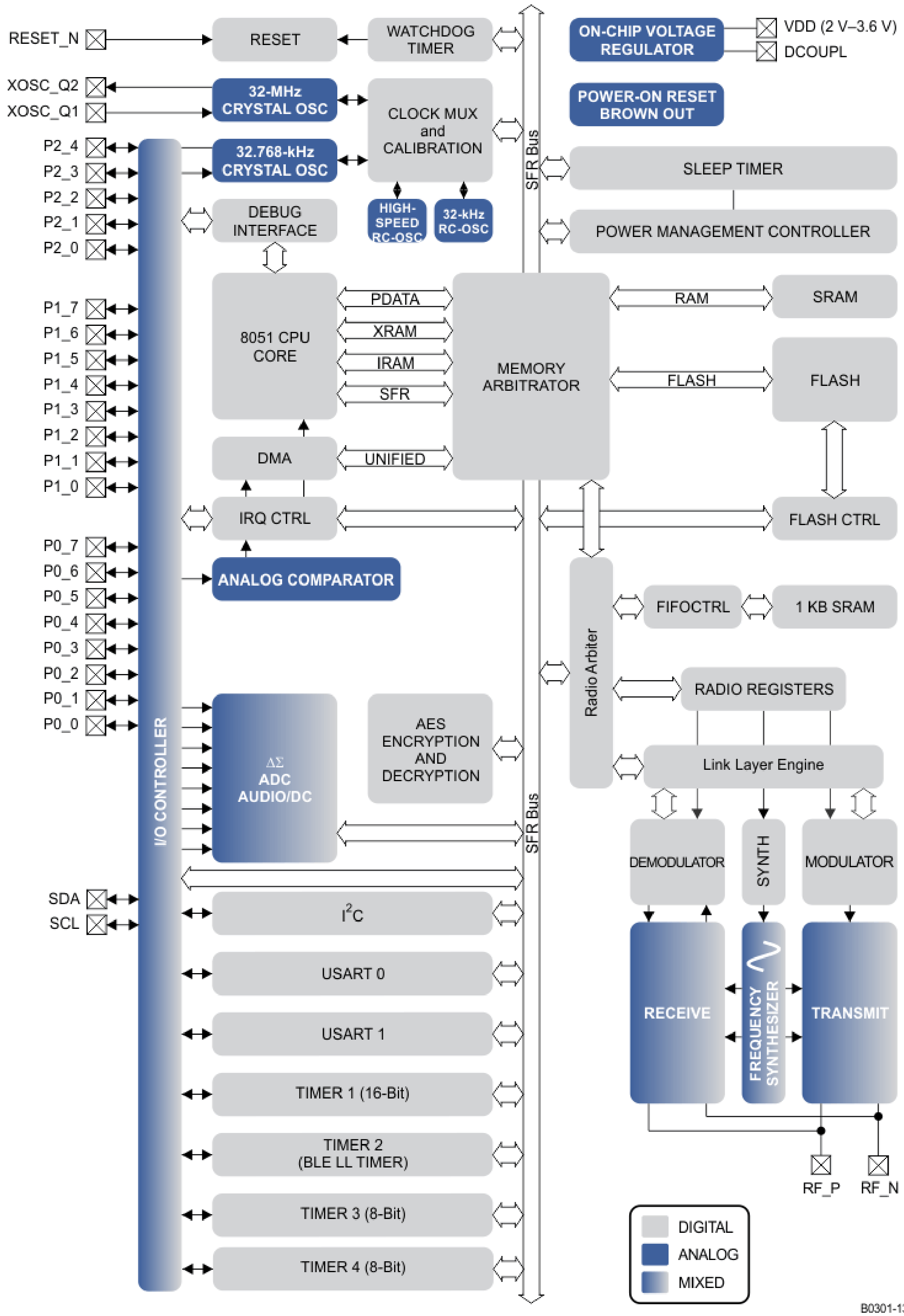


Figure 41 Bluetooth Microcontroller Block Diagram [12]
Courtesy of Texas Instruments

In terms of the FPGA, pins P0_0 to P1_7 is where the output of the FPGA is going to be connected. This Bluetooth unit uses a parallel interface to receive its data. It will be up to the FPGA to control what will be sent from the Bluetooth module to the Android app. From Figure 40 we can see the Bluetooth module has many benefits that will be useful to the application. These benefits include a built in RF module, low power options, a dedicated CPU, timers and its own memory. The device requires an operating power supply of 2V to 3.6V for normal conditions [12]. VDD is where the power source will be connected.

Next is the PmodBT2 from Diligent, Inc. Figure 42 shows the block diagram of the module.

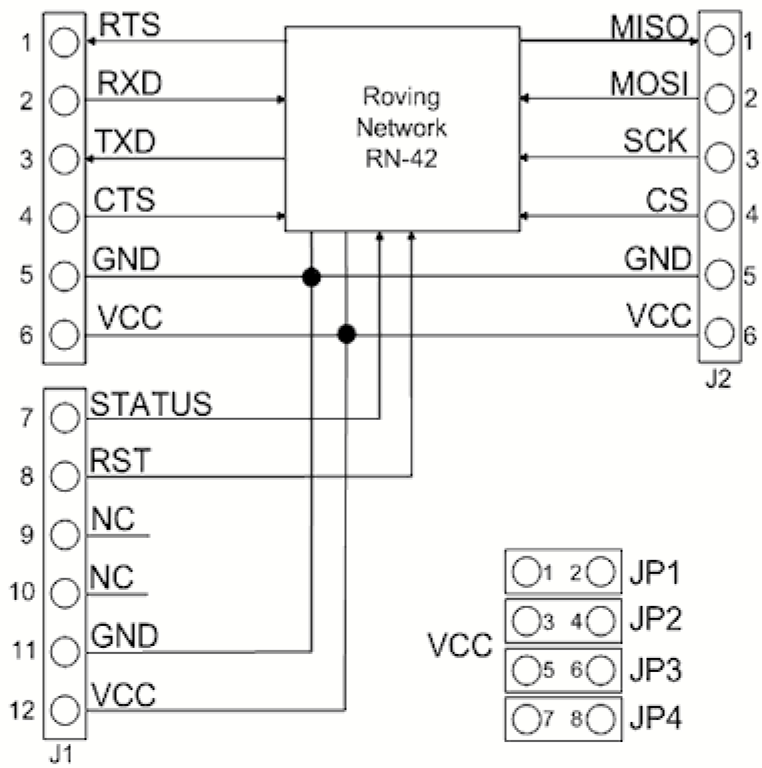


Figure 42 PmodBT2 Block Diagram [55]
Copyright Diligent, Inc. (Permission Pending)

Unlike the CC2541-Q1, this module uses the Roving Network RN-42 to handle message processing. In addition, this module contains a built-in antenna and a command interface to control the device. Next, the module uses an UART interface for data transferring and is compatible with the Spartan 6 FPGA that is being used for the project. This one fact alone makes this unit a very good selection for this position. In the conclusion, the PmodBT2 was selected for this project.

4.4.3. Interface

When designing an application for the mobile market the need for a user interface that is both compact and uncluttered becomes an obvious challenge due to the screen size of most mobile devices. Other challenges will be to implement a design that will suit the variety of aspect ratios that the application could encounter upon release.

The interface for the application must be contain all the required features at the forefront of our requirements, yet still allow the user easy access to find so that they do not get lost in menus and are unable to find a feature that they desire.

The layout design will be built upon a portrait orientation with no reorganization or rescaling of the design to accommodate a landscape layout if the user rotates their device.

Required interface features include:

- Menu Button
- Menu
- Map with location overlaid upon it.
- Selection to switch between the two ADS-B frequencies
- Show a connection status

Statuses that must be shown on the interface include:

- Connection Status
- Flight Statuses
- Selected Detailed Flight Status

Interaction with the application will be of a natural and seamless design similar to how the most mobile applications on the market operate today. Some multi-touch gestures will be included to some features though most actions will be performed through the use of a single-touch gestures that may include basic hold and point and single tap operations.

The design will be segmenting primarily into two sections; one section will contain the map section of the interface and all that are associated with it, the other section will be much smaller and contain only statuses and a menu button. When the menu is pressed the current interface will be replaced entirely with a full screen menu that the user may or may not need to scroll through. This menu will contain all of the various selections to the various options that the user has control over. The status section will include information pertaining to the status of the connection with the receiver and decoder. Pressing upon this section will replace the current interface with an interface pertaining entirely to the connections of the application with its data stream. This interface will include such things as those pertaining to whether or not a connection has been established, shortcuts to the in device connection settings, and any other connection requirements deemed necessary.

The map segment being the primary focus for the user to interpret the information being received and decoded is the most important aspect for this application. Providing the information clearly and without confusion as well as in as naturally perceivable manner as possible is vital to this segment of the interface. To perform this we will overlay images or icons in the shape of planes or other aircraft on top of the map in addition to the information already provided by the map and its orientation tools such as the compass rose. The locations of the aircraft icons on the map will correlate with the GPS coordinates received from the incoming data stream of that particular aircraft. The orientation of the icons on the map will rotate the perceived cockpit of the image to be facing in the direction of the actual aircrafts current flight direction. The combination of these two requirements of the overlaid icons will provide the user an instantly recognizable perception of the location and orientation of the actual aircraft with respect to the orientation of the map (Figure 43).

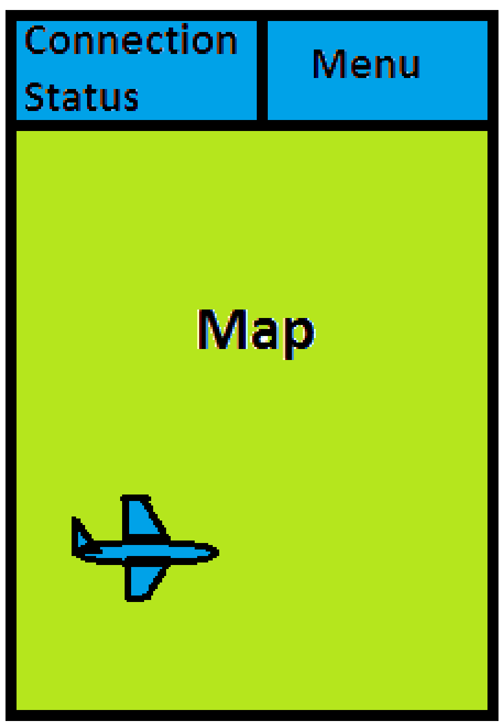


Figure 43 Basic Layout of the Application Interface

5. Design Summary of Hardware and Software

5.1. Hardware Overview

Signal is received by the antennas and passed to the appropriate analog to digital converter. The log detector/controller selected for this project is the Analog Devices AD8319. The converter selected for this project is the Maxim Integrated Products MAX 1192 for reasons discussed prior. The converters sole intention is to convert the analog signal received from the antenna and to convert it to a digital format for the FPGA. The FPGA selected for signal decoding is the Digilent C-Mod XCR. To merge the two decoded signals will be another Digilent FPGA chip, the C-Mod 56. Once the signals have been merged they are now prepared to be passed to the Android application. This task is performed by the low power Bluetooth chip from Diligent Inc, the PmodBT2, selected for its power efficiency and range.

Bill of Materials
Part-1

Electrical Procurement BOM

Manufacturer	Part Number	Description	Quantity	Budgetary Price	Footprint (mm^2)
AVX	08053C104KAT2A	0805	1	$0.01	7
Kemet	C0805C106K8PACTU	0805	3	$0.04	7
Samsung Electro-Mechanics	CL10A106MQ8NNNC	0603	3	$0.02	14
Vishay-Dale	CRCW0402100KFKED	0402	1	$0.01	3
Vishay-Dale	CRCW040210K0FKED	0402	1	$0.01	3
Vishay-Dale	CRCW040254K9FKED	0402	1	$0.01	3
Vishay-Dale	CRCW0603100KFKEA	0603	1	$0.01	5
Vishay-Dale	CRCW0603182KFKEA	0603	1	$0.01	5
Vishay-Dale	CRCW0603499KFKEA	0603	1	$0.01	5
MuRata	GRM155R71E472KA01D	0402	1	$0.01	3
MuRata	GRM219R61E106KA12	0805	1	$0.05	7
MuRata	GRM31CR70J226KE19L	1206	1	$0.30	11
MuRata	GRM32ER61E226KE15L	1210	1	$0.28	15
MuRata	LQM2HPN1R5MG0L	1008	1	$0.12	10
Bourns	SDR0403-4R7ML	SDR0403	1	$0.18	28
Texas Instruments	TPS562209DDCR	DDC0006A	1	$0.58	10
Texas Instruments	TPS62172DSGR	S-PWSON-N8	1	$0.65	10
Texas Instruments	TPS63050YFFT	YFF0012AFAF	1	$1.05	6
TDK	VLF252015MT-2R2M	VLF252015MT	1	$0.45	12
Total			**23**	**$3.92**	**162**

Part-2

Antenna Parts list

File Edit View Insert Format Data Tools Add-ons Help Last edit was made yesterday at 8:08 PM by Sean Koceski

Part	Amount	Price Per Unit	Total Cost				
PVC Tube3/4 in x 10ft	1	2.58	2.58	Home Depot			
PVC Tube 1.5 in x 10ft	3	6.06	18.18	Home Depot			
Eye bolts	6	3.28	19.68	Home Depot			
Paracord	50ft	7.98	7.98	http://www.amazon.com/Rothco-550lb-Type-			
RG-6 Cable	50ft	14.97	14.97	Home Depot			
Connectors	50	14.99	14.99	http://www.cablesforless.com/Ridgeloc-360-High-Performance-			
PVC End caps	2	0.99	1.98	Home Depot			
PVC 90degree	1	0.99	0.99	Home Depot			
PVC T joint	1	0.99	0.99	Home Depot			
			82.34				

Part-3

Products | Support | Classroom | Services | Showcase | About Us | My Cart

ints
ards
s
ucts
itions

Your Shopping Cart

Quantity	Item #	Description	Unit Price	Sub Total
1	CMOD-S6	Cmod S6	$69.00	$69.00
2	CMOD	Cmod *(XC2 CPLD)*	$25.00	$50.00
2	DIP-SOCKET	40-pin DIP Socket	$1.49	$2.98
			Subtotal:	**$121.98**

Part-4

2x Maxim 1192 ADC	***$7.00 ea***	***$14.00***
PCB manufacture		***$115.00***
Battery		***$34.00***
Total		***$371.24***

5.2. Software Overview

As with the hardware, the software side of the project will play an important role when it comes to having a successful project. After all of the coding has been completed, the goal is to have the FPGA be able to communication with the Android application. Figure 42 shows a broad overview of how this will be achieved and how each section will relate to each other.

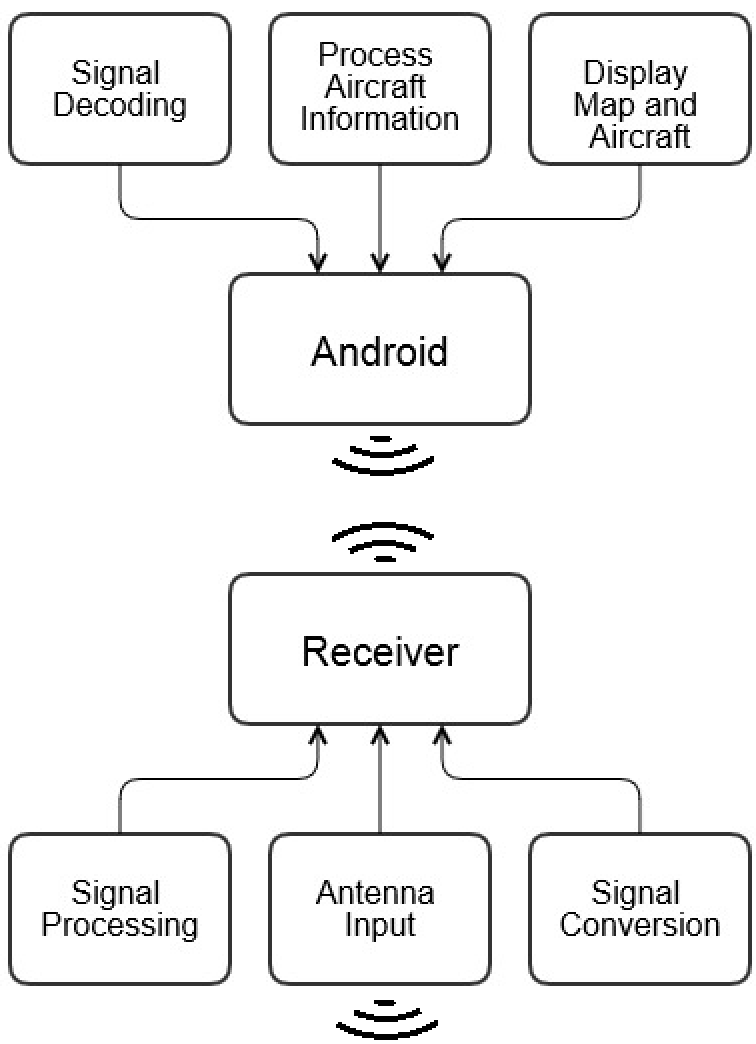


Figure 44 General Overview of Software Design

Starting with the receiver, from Figure 2, we can see that the receiver will obtain and input from the antenna. After the conversion from an analog to digital, the receiver is in charge of processing the signal to be sent to the Android device. The goal is that the receiver will transmit a signal to locate a compatible Bluetooth device. Afterward when it connects to a device, the data will be transmitted and the Android application is to take over from there. The Android application will then process that information to a visual representation of that data for the user to see. During which regular updates will occur in real-time. If the receiver sends an encrypted signal, the application shall notice it and decode it for actual use.

6.1. Antenna Design

6.1.1. Repeaters

Signal repeaters (Figure 45) are simply an antenna that receives an incoming transmission and repeats it by retransmitting the signal back out into the airspace, commonly preceded by amplification of the signal. It is an effective way of allowing a signal to traverse large distances with minimal signal degradations upon receipt by the final source.

This is not a method that we plan to implement in our project due to the requirements of the expansive network of receivers and transmitters needed as well as the permissions needed from the FCC to retransmit the desired signals that were received. While this would be an excellent implementation for usage surrounding and within a city with tall buildings that could block the signal, placing a repeater in an ideal location or locations and having a tight beam radio signal or laser communication system focusing the repeated signal to the final source would allow for both an optimal reception of the transmission and secure forwarding to the final destination either via directly or through a network of repeaters.

6.1.2. Coupling/Merging of Signals

Signal coupling of the signals received by the two separate antennas will not be performed. The reason for this is that to design a decoder that would accept both signals and then decode the two signals simultaneously would require a decoder with a much faster rate which would be too great of a cost. The signals will instead be decoded separately and the resulting digital signals will then be merged as according to their requirements.

6.1.3. Cable Selection and Connections

Cable to be used for transporting the received signals and for the antenna elements will be RG-6 75 Ohm coaxial cable. The reasons for this selection are for its coaxial design, its reliable availability to be purchased with minimal modifications to its design for the intended use, as well as its wide availability, low expense, minimal design flaws.

Connections will be performed with F-Type compression connectors specifically designed for a secure, weatherproof termination. The compression design will not only create a more secure connection but will also ensure a connection with a greater signal reception. This connection will be used to connect the antennas to the transmission line and to the decoder device. Connections of the segments will be performed by soldering the connections.

6.1.4. Antenna Segments

The collinear design is made up of a number of segments of cable that together make up the entire antenna design. The lengths of these segments are determined by the frequency at which the user desires to make the antenna most receptive to.

Segments will be connected soldering the center conductor to the shielding of the cable. In the case of the cable that is selected the center conductor is copper while the shielding is a braided aluminium.

At the opposite end of the antenna's transmission line the center conductor will be soldered to its own segments shielding, forming the completion of the circuit. (Figure 46)

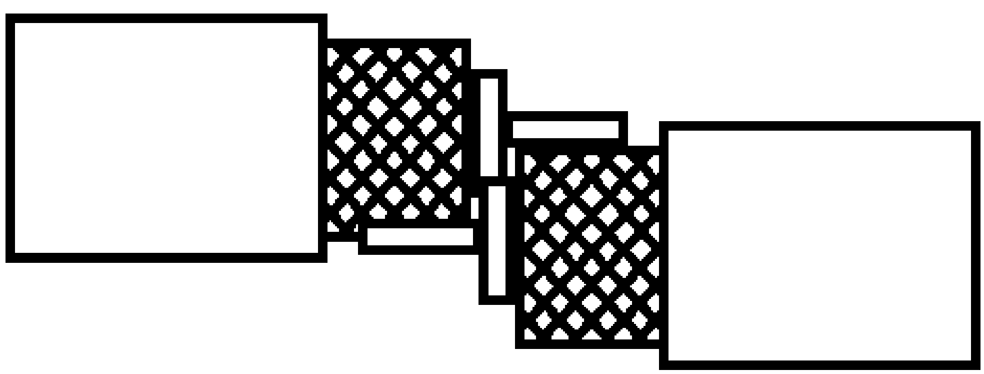


Figure 46
Showing the connection of the segments/elements

Due care must be taken when making the soldering connections because of the heat level that is required to wet the metals being merged will melt both the inner and outer insulation layers if not performed quickly enough.

Formulas for determining the lengths required for different lengths of cable are as follows:

$$(((492\times K)\div f)\div 4)\times 12 = \textit{Quarter wave length in inches}$$

$$(((492\times K)\div f)\div 2)\times 12 = \textit{Half wave length in inches}$$

Or

$$(c\div f)\div 2\times K = \textit{Quarter wave in metric}$$

Where: *K* is the velocity constant of the cable, commonly .83,
c is the speed of light
f is the desired frequency of attenuation

The lengths of our elements will be 114mm for the antenna tuned to receive the 1090 MHz frequency and 127.3mm for the antenna tuned to receive the 978 MHz frequency.

Our antenna designs will utilize an eight element design. This requires roughly one meter of cable for each antenna.

The transmission line must also be taken into account when forming connection. This is because that any difference in the length of the cable used for transmitting the received signal will cause a phase misalignment. Phase misalignment will cause the signals to not match up when they reach the decoder.

6.1.5. Transmission Line

Transmission of the signal from the antenna to the decoder will utilize the same RG-6 cable as the antenna elements. The required length of the cable will depend upon the distance of the antenna from the decoder device.

It is important to note that the length of both transmission lines must be the same length to prevent any phase cancellation.

6.1.6. Antenna Alignment

Reception of the signal is vital so minimizing the interference that is created by the antennas themselves that affect themselves is an issue that is cause for concern. Alignment of the antennas also becomes an issue that needs to be addressed. Placing the antennas in a horizontal alignment they must them be place parallel to each other such that both are an equal distance from the source transmission. This would require constant manipulation of the antennas so that they face the correct direction to prevent any phase cancellation that may occur. The alternative is to have one antenna mount above the other such that any radiation is not passed onto the other.

6.1.7. Antenna Boom and Housing

To ensure that adequate signal strength is capable of being received from the user's location raising the antenna above a majority of the obstacles limit interference received. To achieve this task a boom built out of a non-electrically inductive material. The two choices that would be best suited to both cost restraints and minimizing electrical interference would be wood and PVC. Do to the portability requirement we selected PVC over wood do to it wider range of sizes and shape selection and lower weight and greater portability. The greater portability is in the design of the PVC tubing and its interlocking connections that do not require any additional tools or assembly parts. This allows for quick and simple assembly when out in the field. The hollow tubing of the pipes creates a protection housing to run the transmission lines in addition to raising the antenna. Guide wires will be connected to both the peak of the boom as well as the in the middle of the boom is the height the boom is erected to requires it. Eye bolts will be bolted to the mast to form the connection points for the guide wires.

Large diameter PVC will be selected for the mast of the boom; this is due to the increased strength that is needed in order to support the weight of both mast segments and the antenna plus housing. More or less segments can be added to achieve a desirable height within location constraints and logical reasoning. The transmission line will run through these segments to reach the decoder on the ground level.

A casing to protect the antenna elements will also be built to permanently house the antenna. A smaller diameter tubing will be used for this section due to the shorter length required will also be used to store the collinear antenna segments as well. This is beneficial because it will protect the delicate elements from being bent or damaged in any other fashion.

The design of the PVC housing to store the elements will place the two antennas in a vertical alignment (Figure 47) with each other so as to reduce the phase cancellation effects to an almost eliminated level.

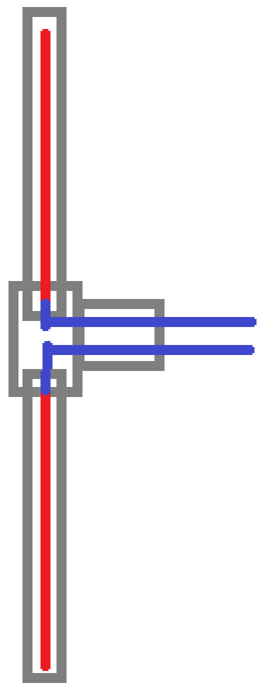


Figure 47

Antenna Housing. Red lines represent antennas. Blue lines represent transmission lines. Grey represents PVC tubing.

6.2. Receiver

6.2.1. PCB Fabrication

The two PCB vendors that we considered were PCB123 also known as Sunstone Circuits and OSH Park. Both vendors have received positive reviews from past users. OSH Park provides $5 dollar per square inch board which provides for a total of 3 boards of the design. This benefit would allow for use to have backup boards at no extra cost. They however do not provide as quick of a return as PCB123, 12 days versus a build time of 2 days and a guarantee of the board being free if it is not shipped on time.

PCB123 does provide their own software to design with that includes a range of 750,000 parts that they have pre designed. They will also accept a larger number of file types including OrCAD, National Instruments as well as the popular Cadsoft EAGLE files in addition to their own file type.

While the decreased cost of OSH Park is a strong selling point for a project that includes a requirement of needing to be inexpensive we will select PCB123 because of their shorter turnaround time and guarantee. We feel that if it is needed having a board delivered when we need it will be more important for this phase of the project versus saving money.

6.2.2. PCB Assembly

Placement and mounting of the chips onto the board will be all performed by hand as the tools used for mass manufacturing will not be an option to us. This creates the issue of soldering large pin count surface mount chips by hand. Requirements for this task will require careful heat setting so as not to burn the board and/or damage the chip being mounted or other chips in close proximity. To accomplish we will place a strip of solder across one edge of the chip with pins and ensure the use of adequate flux for efficient heat transfer and place the iron tip on one end of the solder strand and have the entire strand wet with the pins and mounts as they heat to the necessary temperature.

Placement of any through-hole chips will pose fewer hazards due to the greater heat diffusion and further distance from the chip. The same would go for the connection ports that are to be mounted as well.

6.3 Final Coding Plan

With our software design, there are two major portions to the project. These portions are the Android application and the code that is to be ran on the FPGA. In doing so, we will need repositories to store and manage all the code that will be used for the project. To do so, we will used Github and Dropbox. The primary storage location will be Github where any up to the minute changes will occur. Dropbox will act as the secondary storage which will receive periodic updates when a new stable version of the application the FPGA code is completed. As mention before, Eclipse with the Android Development Tools will be used for the application. With the FPGA, we are going to be using a Xilinx Platform Cable USB device with Xilinx Lab Tools software for programming the logic to the FPGA.

When it comes to the FPGA, the FPGA will be required to convert the incoming ADS-B signal into a data stream where it will be sent to the Bluetooth module which will be received by an Android device. We will use the Manchester encoding - decoding method to process the signal. Following the decoding of the signal a third FPGA will be used to merge the digital signals into a single data stream ready for output to the Bluetooth transmitter (Figure 48).

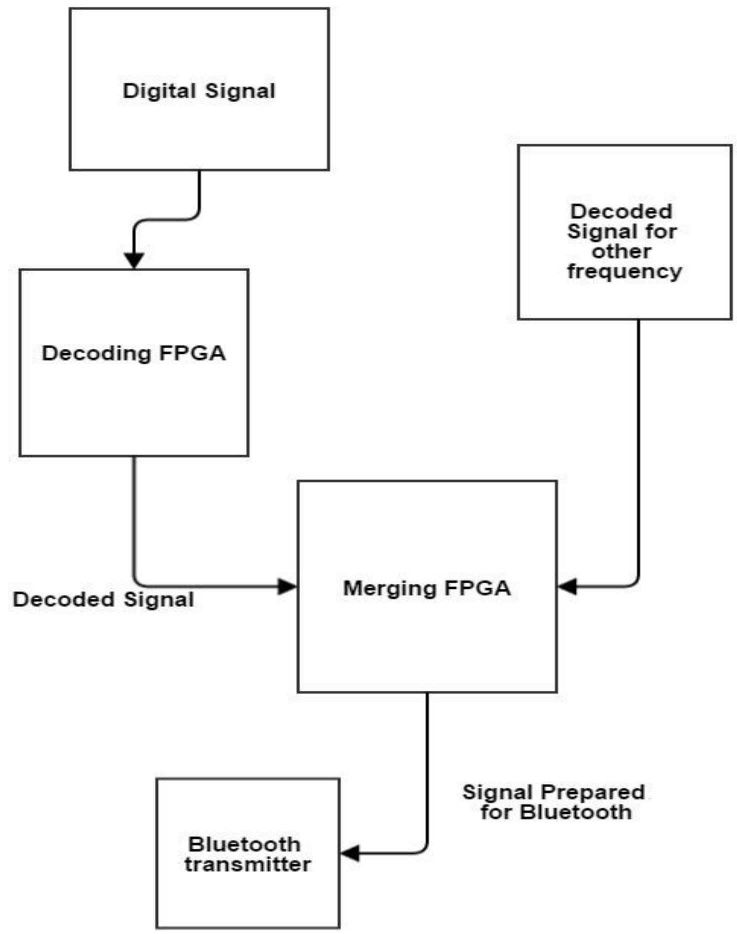


Figure 48

7.2. Hardware Specific Testing

Test Objective: Antennas' reception
Test Description: Test each antenna separately when connected to the RealTek SDR and verify valid results compared to plane tracking websites and visible contact.
Test Conditions: Ensure adequate line of sight with airspace to prevent false negatives caused by failed signal reception
Expected Results: • With aircraft in the vicinity locations compared with various websites should prove same results • No results should be visible if no planes are present

Table 39 Testing Antenna

Test Objective: Filter's frequency bandpass range
Test Description: Test the range of the frequencies that blocked from passing onto the rest of the receiver by connecting a frequency generator to the input and an oscilloscope to the output and noting the cutoff frequency is within tolerances and not eliminating the desired ADS-B frequency carrier waves
Test Conditions: Disconnect the rest of the receiver from the output of the filter to prevent any possible damage that may occur from erroneous procedures
Expected Results: • Cutoff frequencies will be span equally around the desire center frequency with minimal noise introduced.

Table 40 Testing filter

Test Objective: Bluetooth testing
Test Description: Ensure that a bluetooth enabled device is present and set up to detect an open bluetooth signal. Upon receiving of signal initiate pairing of devices.
Test Conditions: No actual data needs to be passed just a connection or verification that bluetooth device is outputting signal
Expected Results: • Signal is outcast and connections could be established if desired

Table 41: Testing Bluetooth connection

Test Objective: Decoding of Signal
Test Description: To ensure that the signal being based to the bluetooth module is valid and not unusable by a bluetooth receiving device, the signal will be intercepted through various testing procedures
Test Conditions: No bluetooth receiving device will be necessary, but connection to the receiver via USB connection will be established
Expected Results: • Signal about to be accepting by bluetooth chip is of the appropriate format

Table 42: Testing FPGA

Test Objective: Test antenna faults
Test Description: Disconnect antennas and via frequency generator input a false signal into the receiver
Test Conditions: Antenna must be disconnected to ensure signals that cannot be verified for accuracy are not inserted into the receiver.
Expected Results: • If the antennas are faulty then the results output from the receiver should match to an expected result if results prior were invalid. • If the receiver is faulty then the results of the output from the receiver will not match to an expected result

Table 43: Testing Ensuring RF boards reception

Test Objective: Test antenna faults
Test Description: Disconnect antennas from receiver and connect output to an oscilloscope. and measure amplitude of received signals
Test Conditions: Antenna must be disconnected to ensure signals that cannot be verified for accuracy are not inserted into the receiver.
Expected Results: • If the antennas are faulty then signals received will have low amplitude • If the antennas are faulty then the signals received will be of an incorrect frequency • If the antennas are providing adequate gain and correct frequencies then the receiver could be faulty.

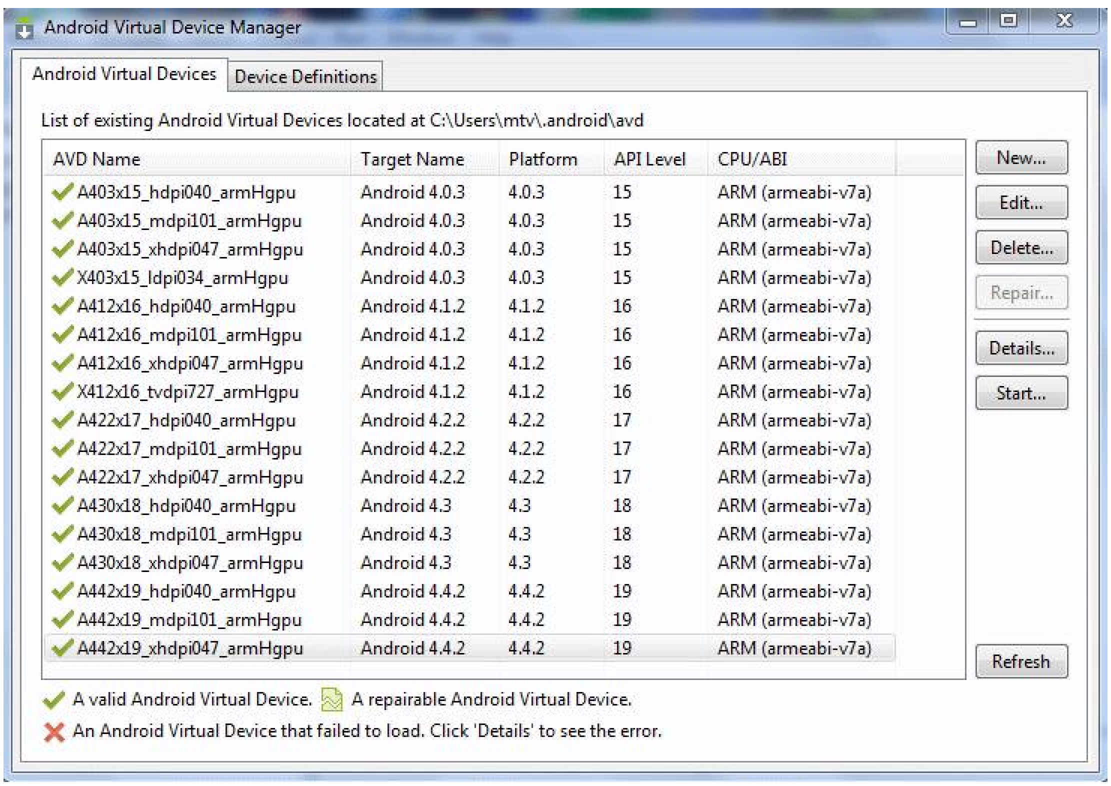
Table 44: Testing receiver connection status

7.3. Software Test Environment

Individual members of the Development team will perform unit testing using the Android Virtual Device facility in the ADT. Integration testing and performance testing across Android 4.0.3 to Android 4.4.2 will be performed using both the Android Virtual Device facility in the ADT and actual external hardware including a Samsung Galaxy Avant Smartphone running Android 4.4.2, and a Nabi 2

Tablet running Android 4.1.1. These specific hardware devices were selected due to the range of versions of Android with which they were operating. In one case, the test environment is older than the environment in which the software is expected to normally operate. In the other case, the device is selected due to it being a current version of Android. This is the environment in which the software is expected to normally operate.

In regard to the Android Virtual Device facility, for integration and performance testing, a battery of devices in each of the three principle resolutions (xh,h,m) were created across the five Android API levels from 15 to 19 to provide comprehensive coverage. Two exceptional resolutions (tv, l) were created on one level each to explore compatibility concerns even though such concerns were not documented requirements. During QA testing, the application is exercised against every A-prefixed AVD in the battery shown in Figure 49:



AVD Name	Target Name	Platform	API Level	CPU/ABI
A403x15_hdpi040_armHgpu	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
A403x15_mdpi101_armHgpu	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
A403x15_xhdpi047_armHgpu	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
X403x15_ldpi034_armHgpu	Android 4.0.3	4.0.3	15	ARM (armeabi-v7a)
A412x16_hdpi040_armHgpu	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
A412x16_mdpi101_armHgpu	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
A412x16_xhdpi047_armHgpu	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
X412x16_tvdpi727_armHgpu	Android 4.1.2	4.1.2	16	ARM (armeabi-v7a)
A422x17_hdpi040_armHgpu	Android 4.2.2	4.2.2	17	ARM (armeabi-v7a)
A422x17_mdpi101_armHgpu	Android 4.2.2	4.2.2	17	ARM (armeabi-v7a)
A422x17_xhdpi047_armHgpu	Android 4.2.2	4.2.2	17	ARM (armeabi-v7a)
A430x18_hdpi040_armHgpu	Android 4.3	4.3	18	ARM (armeabi-v7a)
A430x18_mdpi101_armHgpu	Android 4.3	4.3	18	ARM (armeabi-v7a)
A430x18_xhdpi047_armHgpu	Android 4.3	4.3	18	ARM (armeabi-v7a)
A442x19_hdpi040_armHgpu	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
A442x19_mdpi101_armHgpu	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)
A442x19_xhdpi047_armHgpu	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)

Figure 49

7.4. Software Specific Testing

Test Objective: Confirm that the application starts operation on an Android device running version 4.2
Test Description: After installing application start procedures to begin operation
Test Conditions: If the application begins operation without unexpected results
Expected Results: ● Application begins and reaches the next level of its operation, Success. ● Application fails to start, Fail. ● Device begins to perform to erroneously and must be restarted, Fail.

Table 45: Confirm Startup of Application

Test Objective: The application shall display all aircraft within the phone's GPS location that the device received.
Test Description: Via USB interface of the device, false data will be provided as input
Test Conditions: The application will be reviewed for interpreted results
Expected Results: ● Application displays the data expected. Success ● Application display incorrect data. Fail ● Application displays various amounts of correct and incorrect data. Fail

Table 46: Confirm aircraft location

Test Objective: The application shall be able to process the data that was obtained from bluetooth
Test Description: Via USB interface of the receiver, false data will be provided as input
Test Conditions: The application will be reviewed for interpreted results
Expected Results: ● Application displays the data as expected. Success ● Application displays incorrect data. Fail ● Application displays various amounts of correct and incorrect data. Fail

Table 47: Verify bluetooth datafeed

Test Objective: The application shall display aircraft information when selected by the user.
Test Description: Via USB interface with the receiver, false data will be provided as input
Test Conditions: Upon selection of an aircraft by the user the displayed data will be reviewed for accuracy.
Expected Results: ● Application displays the data as expected. Success ● Application displays incorrect data. Fail ● Application displays various amounts of correct and incorrect data. Fail

Table 48: Verify data stream

Test Objective: The application shall allow the user to select an aircraft that is displayed
Test Description: Via USB interface with the receiver, false data will be provided as input.
Test Conditions: Upon display of an aircraft on the display window of the application the user will attempt to select the aircraft as the applications focus
Expected Results: ● Application makes the selection the target focus. Success ● Application does not make the target the focus. Fail ● Application makes another aircraft the focus. Fail

Table 49: Verify user input selection

Test Objective: Application shall update aircraft data in real-time
Test Description: Via USB interface with the receiver, false data will be provided as input.
Test Conditions: Upon display of an aircraft on the display window the user will view the changes that the aircraft performs.
Expected Results: ● The changes the user views match with the changes that the receiver are fed. Success ● The changes the user views do not match with the changes that the receiver are fed. Fail

Table 50: Verify application update

Test Objective: The application shall recognize an encrypted data stream from the receiver.
Test Description: Via USB interface with the receiver, false data will be provided as input in an encrypted manner.
Test Conditions: Upon display of an aircraft on the display window the user will perform necessary tasks for handling the data stream
Expected Results: ● Application performs the necessary changes to accept the encryption. Success ● Application fails to perform the necessary changes to accept the encryption. Fail

Table 51: Application will decode data

Test Objective: The application shall be able to decode an encrypted data stream
Test Description: Via USB interface with the receiver, false data will be provided as input in an encrypted manner.
Test Conditions: All prior tests will be performed to verify that they will also perform in accordance with their test conditions when performed with the encrypted data stream
Expected Results: ● All prior test objectives perform without change when compared with the unencrypted tests. Success ● Any number of test objectives fail when compared to the unencrypted data test. Fail

Table 52: Application will decode data stream

Test Objective: Confirm that the state which the application was left in is the same and in the process of updating when application is brought back into focus
Test Description: Upon leaving the application and returning as a user would do if they brought the devices focus to another part of the OS or to another application. When focus is returned to the application the data is updated within 10 seconds
Test Conditions: The application is brought to a usable state and upon return performs as is was prior to loss of focus
Expected Results: ● Application is brought back into focus and data displayed is updated within the required time limit. Success ● Application is brought back into focus but data fails to update within the required time limit. Fail ● Application is brought back into focus and fails to update entirely. Fail ● Application is brought back into focus but fails to update all data. Fail

Test 53: Verify application memory validity

Test Objective: Interface will provide an Exit button to leave the application
Test Description: Upon selection and verification of exiting the application the application is removed from focus and ceases resource utilization
Test Conditions: The application is closed and the device remains in a stable and usable state
Expected Results: ● Application closes and device remains in a stable and usable state. Success ● Application closes and the device crashes. Fail ● Application fails to close. Fail

Table 54: Exiting application via exit button

Test Objective: The program shall reboot should any abnormal behavior occurs.
Test Description: Upon erroneous activity the application closes itself and begins anew
Test Conditions: Introduce a known activity that will crash the program and determine if the application recovered as designed
Expected Results: ● Application recovers as designed. Success ● Application crashes and never recovers. Fail ● Device crashes and cannot recover. Fail

Table 55: Reboot upon error

Test Objective: Maximum 1-second response time to interface commands and game events on Android devices with 512Mb of RAM. Maximum 2-second response time to interface commands and game events on Android devices with 256Mb of RAM.
Test Description: Press each of the selection and time response. Application should be running with other applications
Test Conditions: None
Expected Results: ● Actions execute within the time requirements. Success ● Actions fail to execute. Fail ● Actions execute outside of the time requirements. Fail

Table 56: Response time test

7.5. Integration Testing

Once all of the various parts of the system have been tested individually as per their test procedures then testing can begin on the entire system all together.

Test Objective: Confirm location will have aircraft flying within half of the required max receiving distance
Test Description: Setup the system by erecting the antenna boom and antenna and connecting all cabling connections ensuring proper antennas output is connected to the proper receiver inputs.
Test Conditions: Only test the system with weather permitting condition, i.e. no chance of lightning strikes. Setup in a location away from any power lines or other electrically conductive or electrically powered objects
Expected Results: • Aircraft within half the max receiving distance outlined in the requirements are detected and tracked.

Table 57: Location verification

Test Objective: Power supply performance
Test Description: Ensure that the system is able to perform to adequate levels and no subsystem is drawing more power that it is designed to
Test Conditions: Ensure that the only power system in operation is that that was designed for the system as a whole
Expected Results: System is able to perform its intended functionalities

Table 58: Power supply levels

Test Objective: Length of functional operation
Test Description: Run the system with the intended power system and not the length of time from when the system is powered on to when noticeable depreciation in functionality occurs do to low power levels
Test Conditions: Portable power source should be at its optimal levels for the intended use
Expected Results: System is able to perform for an adequate amount of time to view data.

Table 59: Total usable time

Test Objective: Confirm location will have aircraft flying within the full required range
Test Description: Setup the system by erecting the antenna boom and antenna and connecting all cabling connections ensuring proper antennas output is connected to the proper receiver inputs.
Test Conditions: Only test the system with weather permitting condition, i.e. no chance of lightning strikes. Setup in a location away from any power lines or other electrically conductive or electrically powered objects
Expected Results: • Aircraft within max receiving distance outlined in the requirements are detected and tracked.

Table 60: Verify maximum received distance

Test Objective: Confirm that the system can handle a higher than average number of aircraft
Test Description: Setup the system in a location of higher than average air traffic density
Test Conditions: Only test the system with weather permitting condition, i.e. no chance of lightning strikes. Setup in a location away from any power lines or other electrically conductive or electrically powered objects
Expected Results: • All aircraft in the vicinity should be perceived on the user’s device.

Table 61: Confirm total aircraft visible

8. Administrative Content

8.1 Milestones

ID	Task_Name	Start_Date	Finish_Date
1	**Group24 Project Scope**	Thu 8/28/14	Mon 9/8/14
2	Research Sofware Defined Radio	Thu 8/28/14	Fri 8/29/14
3	Research Mode-S Receivers	Fri 8/29/14	Mon 9/1/14
4	Research ADS-B Topics	Tue 9/2/14	Wed 9/3/14
5	Preliminary Costs Estimate	Wed 9/3/14	Wed 9/3/14
6	Develop Initial Project Plan	Wed 9/3/14	Mon 9/8/14
7	Scope complete	Mon 9/8/14	Mon 9/8/14
8	**Detailed Analysis**	Tue 9/9/14	Fri 9/19/14
9	Review specifications/budget with team	Tue 9/9/14	Tue 9/9/14
10	Incorporate feedback on specifications	Tue 9/9/14	Wed 9/10/14
11	Draft detailed hardware specifications	Wed 9/10/14	Fri 9/12/14
12	Draft detailed software specifications	Fri 9/12/14	Thu 9/18/14
13	Improve milestone timeline	Thu 9/18/14	Thu 9/18/14
14	Obtain approvals to proceed (concept, timeline, budget)	Fri 9/19/14	Fri 9/19/14
15	Analysis complete	Fri 9/19/14	Fri 9/19/14
16	**Design Signal-to-USB**	Tue 9/23/14	Thu 10/16/14
17	Develop Schematics	Tue 9/23/14	Wed 10/1/14
18	Outline Verilog Approach	Thu 10/2/14	Wed 10/8/14
19	Outline Software Classes/Modules	Thu 10/9/14	Thu 10/9/14
20	Review specifications	Fri 10/10/14	Fri 10/10/14
21	Incorporate feedback into specifications	Fri 10/10/14	Tue 10/14/14
22	Revise and expand project documents	Tue 10/14/14	Thu 10/16/14
23	Design Signal-to-USB complete	Thu 10/16/14	Thu 10/16/14
24	**Design USB-to-Display**	Tue 9/23/14	Wed 10/1/14
25	Outline Software Classes/Modules	Tue 9/23/14	Wed 9/24/14
26	Review specifications	Thu 9/25/14	Thu 9/25/14
27	Incorporate feedback into specifications	Thu 9/25/14	Fri 9/26/14
28	Revise and expand project documents	Fri 9/26/14	Wed 10/1/14
29	Design USB-to-Display complete	Wed 10/1/14	Wed 10/1/14
30	**Design ADS-B Encryption**	Wed 10/1/14	Fri 10/10/14
31	Evaluate streaming vs block modes	Wed 10/1/14	Fri 10/3/14
32	Outline Software Classes/Modules	Fri 10/3/14	Wed 10/8/14

33	Review specifications	Wed 10/8/14	Wed 10/8/14
34	Incorporate feedback into specifications	Thu 10/9/14	Thu 10/9/14
35	Revise and expand project documents	Fri 10/10/14	Fri 10/10/14
36	Design ADS-B Encryption complete	Fri 10/10/14	Fri 10/10/14
37	**Prototype Signal-to-USB**	Thu 10/16/14	Thu 11/13/14
38	Breadboard solution	Thu 10/16/14	Wed 10/22/14
39	Develop Verilog Decoder	Wed 10/22/14	Fri 10/31/14
40	Developer testing (primary debugging)	Fri 10/31/14	Fri 11/7/14
41	Revise and expand project documents	Fri 11/7/14	Thu 11/13/14
42	Development Signal-to-USB complete	Thu 11/13/14	Thu 11/13/14
43	**Testing Signal-to-USB**	Thu 11/13/14	Tue 1/13/15
44	**Unit Testing**	Thu 11/13/14	Wed 11/26/14
45	Test component modules to product specifications	Thu 11/13/14	Tue 11/18/14
46	Identify anomalies to product specifications	Tue 11/18/14	Wed 11/19/14
47	Modify Verilog	Wed 11/19/14	Thu 11/20/14
48	Re-test modified code	Thu 11/20/14	Fri 11/21/14
49	Revise and expand project documents	Fri 11/21/14	Wed 11/26/14
50	Unit testing Signal-to-USB complete	Wed 11/26/14	Wed 11/26/14
51	**Integration Testing**	Wed 11/26/14	Tue 1/13/15
52	Module integration	Wed 11/26/14	Thu 12/4/14
53	Identify anomalies to specifications	Thu 12/4/14	Tue 1/6/15
54	Modify code	Tue 1/6/15	Wed 1/7/15
55	Re-test modified code	Wed 1/7/15	Thu 1/8/15
56	Revise and expand project documents	Thu 1/8/15	Tue 1/13/15
57	Integration testing Signal-to-USB complete	Tue 1/13/15	Tue 1/13/15
58	**Develop USB-to-Display**	Thu 10/16/14	Wed 11/5/14
59	Coding	Thu 10/16/14	Thu 10/23/14
60	Developer testing (primary debugging)	Thu 10/23/14	Fri 10/31/14
61	Revise and expand project documents	Fri 10/31/14	Wed 11/5/14
62	Development USB-to-Display complete	Wed 11/5/14	Wed 11/5/14
63	**Testing USB-to-Display**	Wed 11/5/14	Thu 12/4/14
64	**Unit Testing**	Wed 11/5/14	Tue 11/18/14
65	Test component modules to product specifications	Wed 11/5/14	Thu 11/6/14
66	Modify code	Thu 11/6/14	Fri 11/7/14
67	Re-test modified code	Fri 11/7/14	Wed 11/12/14
68	Revise and expand project documents	Wed 11/12/14	Tue 11/18/14
69	Unit testing USB-to-Display complete	Tue 11/18/14	Tue 11/18/14
70	**Integration Testing**	Tue 11/18/14	Thu 12/4/14

71	Module integration	Tue 11/18/14	Wed 11/19/14
72	Identify anomalies to specifications	Wed 11/19/14	Thu 11/20/14
73	Modify code	Thu 11/20/14	Fri 11/21/14
74	Re-test modified code	Fri 11/21/14	Tue 11/25/14
75	Revise and expand project documents	Tue 11/25/14	Thu 12/4/14
76	Integration testing USB-to-Display complete	Thu 12/4/14	Thu 12/4/14
77	**Develop ADS-B Encryption**	Wed 11/5/14	Wed 11/26/14
78	Coding	Wed 11/5/14	Thu 11/13/14
79	Developer testing (primary debugging)	Thu 11/13/14	Fri 11/21/14
80	Revise and expand project documents	Fri 11/21/14	Wed 11/26/14
81	Development ADS-B Encryption complete	Wed 11/26/14	Wed 11/26/14
82	**Testing ADS-B Encryption**	Wed 11/26/14	Wed 1/21/15
83	**Unit Testing**	Wed 11/26/14	Fri 1/9/15
84	Test component modules to product specifications	Wed 11/26/14	Thu 12/4/14
85	Modify code	Thu 12/4/14	Tue 1/6/15
86	Re-test modified code	Tue 1/6/15	Wed 1/7/15
87	Revise and expand project documents	Wed 1/7/15	Fri 1/9/15
88	Unit testing ADS-B Encryption complete	Fri 1/9/15	Fri 1/9/15
89	**Integration Testing**	Fri 1/9/15	Wed 1/21/15
90	Module integration	Fri 1/9/15	Tue 1/13/15
91	Identify anomalies to specifications	Tue 1/13/15	Wed 1/14/15
92	Modify code	Wed 1/14/15	Thu 1/15/15
93	Re-test modified code	Thu 1/15/15	Fri 1/16/15
94	Revise and expand project documents	Fri 1/16/15	Wed 1/21/15
95	Integration testing ADS-B Encryption complete	Wed 1/21/15	Wed 1/21/15

Table 62: Milestones

References:

[1] "A Look at the Basics of Bluetooth Technology." *Basics | Bluetooth Technology Website*. N.p., n.d. Web. 21 Oct. 2014. <http://www.bluetooth.com/Pages/Basics.aspx>.

[2] "AD9281: Dual Channel 8-Bit Resolution CMOS ADC." *Analog Devices*. N.p., n.d. Web. 15 Nov. 2014. <http://www.analog.com/en/analog-to-digital-converters/ad-converters/ad9281/products/product.html>.

[3] "ADS930." *Texas Instruments*. Texas Instruments Incorporated, n.d. Web. 15 Nov. 2014. <http://www.ti.com/product/ADS930/description>.

[4] "Alkaline Battery." *Wikipedia*. Wikimedia Foundation, 27 Oct. 2014. Web. 30 Oct. 2014. <http://en.wikipedia.org/wiki/Alkaline_battery>.

[5] "Analog-Digital Converters." *OnMyPHD*. N.p., n.d. Web. 16 Oct. 2014.

[6] "Analog-to-digital Converter." *Wikipedia*. Wikimedia Foundation, 12 Feb. 2014. Web. 02 Dec. 2014. <http://en.wikipedia.org/wiki/Analog-to-digital_converter>.

[7] "Android Security Overview." *Android Developers*. N.p., n.d. Web. 15 Oct. 2014.

[8] "Bluetooth." *Android Developers*. N.p., n.d. Web. 13 Oct. 2014.

[9] "Bluetooth." *Wikipedia*. Wikimedia Foundation, 28 Oct. 2014. Web. 28 Oct. 2014. <http://en.wikipedia.org/wiki/Bluetooth>.

[10] "Bluetooth Low Energy." *Wikipedia*. Wikimedia Foundation, 29 Nov. 2014. Web. 02 Dec. 2014. <http://en.wikipedia.org/wiki/Bluetooth_low_energy>.

[11] "Bluetooth vs. Wi-Fi." *Diffen*. N.p., n.d. Web. 21 Oct. 2014. <http://www.diffen.com/difference/Bluetooth_vs_Wifi>.

[12] "CC2541-Q1." *Texas Instruments*. Texas Instruments Incorporated, n.d. Web. 02 Dec. 2014. <http://www.ti.com/product/CC2541-Q1/datasheet>.

[13] "Comparison Table of Secondary Batteries." *Battery University*. N.p., n.d. Web. 05 Nov. 2014. <http://batteryuniversity.com/learn/article/secondary_batteries>.

[14] "Google Maps Android API V2." — *Google Developers*. N.p., n.d. Web. 13 Oct. 2014.

[15] "IEEE 802.11n-2009." *Wikipedia*. Wikimedia Foundation, 25 Oct. 2014. Web. 28 Oct. 2014. <http://en.wikipedia.org/wiki/IEEE_802.11n-2009>.

[16] "Lead–acid Battery." *Wikipedia*. Wikimedia Foundation, 12 Jan. 2014. Web. 02 Dec. 2014. <http://en.wikipedia.org/wiki/Lead%E2%80%93acid_battery>.

[17] "MAX1192 Ultra-Low-Power, 22Msps, Dual 8-Bit ADC." *Maxim Integrated*. N.p., n.d. Web. 15 Nov. 2014. <http://www.maximintegrated.com/en/products/analog/data-converters/analog-to-digital-converters/MAX1192.html/tb_tab0>.

[18] "The Pros & Cons of Using Wireless Internet Services." *EHow*. Demand Media, 26 Aug. 2009. Web. 28 Oct. 2014. <http://www.ehow.com/about_5348495_pros-using-wireless-internet-services.html>.

[19] "Types and Techniques of Analogue to Digital Conversion." *PC Control Learning Zone*. N.p., n.d. Web. 17 Oct. 2014. <https://www.pc-control.co.uk/atod_types.htm>.

[20] "What's the Best Battery?" *Advantages and Limitations of the Different Types of Batteries*. N.p., n.d. Web. 01 Nov. 2014. http://batteryuniversity.com/learn/article/whats_the_best_battery

[21] "Wi-Fi." *Wikipedia*. Wikimedia Foundation, 26 Oct. 2014. Web. 26 Oct. 2014. <http://en.wikipedia.org/wiki/Wi-Fi>.

[22] "Wi-Fi Basics: A Beginner's Guide to the Wireless Web." Maximum PC. N.p., n.d. Web. 21 Oct. 2014. <http://www.maximumpc.com/article/features/wi-fi_basics_beginners_guide_wireless_web>.

[23] AIAA, "A Framework for Aviation Cybersecurity." The American Institute of Aeronautics and Astronautics, 1 Aug. 2013. Web. 1 Dec. 2014. <https://www.aiaa.org/uploadedFiles/Issues_and_Advocacy/AIAA-Cyber-Framework-Final.pdf>.

[24] Bellare, M., Rogaway, and P., Stegers, T. (2010) The FFX mode of operation for format-preserving encryption. <http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ffx/ffx-spec.pdf>.

[25] Finke, Cindy, Jonathan Butts, Robert Mills, and Michael Grimaila. "Evaluation of a Cryptographic Security Scheme for Air Traffic Control's Next Generation Upgrade." Air Force Institute of Technology. Web. 2 Dec. 2014. <http://www.usafa.edu/df/dfe/dfer/centers/accr/docs/ICIWedit2.pdf>.

[26] Federal Aviation Administration, "Automatic Dependent Surveillance--Broadcast (ADS-B) Out Performance Requirements To Support Air Traffic Control (ATC) Service." Federal Register vol. 72(193), Pages 56947-56972. 5 Oct. 2007. Web. 1 Dec. 2014. <http://www.gpo.gov/fdsys/pkg/FR-2007-10-05/html/07-4938.htm>.

[27] Federal Aviation Administration, "Automatic Dependent Surveillance-Broadcast (ADS-B) Out Performance Requirements to Support Air Traffic Control (ATC) Service; Final Rule, 14 CFR Part 91," Federal Register, vol. 75(103). 28 May. 2010.

[28] Flightradar24 Live Air Traffic. Flightradar24. Web. 1 Dec. 2014. <http://www.flightradar24.com/>.

[29] FlightAware Live Flight Tracking. FlightAware. Web. 1 Dec. 2014. <http://flightaware.com/>.

[30] ICAO, "SECURITY ISSUES OF ADS-B OPERATIONS." International Civil Aviation Organization, 1 Apr. 2014. Web. 1 Dec. 2014. <http://www.icao.int/APAC/Meetings/2014 ADSBSITF13/WP19_India AI.7 - Security Issues of ADS-B operations.pdf>.

[31] Köllner, Günter. "Project “ADS-B Receiver and Decoder”." Mode-S Beast. Web. 2 Dec. 2014. <http://www.modesbeast.com/resources/Koellner_Projekt-ADSB3.pdf>.

[32] McCallie, Donald. "EXPLORING POTENTIAL ADS-B VULNERABILITIES IN THE FAA’s NEXTGEN AIR TRANSPORTATION SYSTEM." AIR FORCE INSTITUTE OF TECHNOLOGY, 1 June 2011. Web. 10 Nov. 2014. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ada545599>.

[33] "MiniADSB 1090 MHz Receiver." MiniADSB 1090 MHz Receiver. Web. 3 Dec. 2014. <http://miniadsb.web99.de/>.

[34] Nickels, Robert. "Virtual Radar from a Digital TV Dongle." ARRL, The National Association for Amateur Radio. American Radio Relay League, 1 Jan. 2014. Web. 1 Dec. 2014. <http://www.arrl.org/files/file/QST/This Month in QST/January 2014/VirtualRadarJan2013QST.pdf>.

[35] Nubifier. "Figure 8." Wikimedia. Web. 3 Dec. 2014. <http://commons.wikimedia.org/wiki/File:Flightradar24_ADS-B_receiver.png>.

[36] Obama, Barack. "President Barack Obama's State of the Union Address -- As Prepared for Delivery." The White House. Office of the Press Secretary, 12 Feb. 2013. Web. 1 Dec. 2014.

<http://www.whitehouse.gov/the-press-office/2013/02/12/president-barack-obamas-state-union-address>.

[37] PBS, "Watch Now: NOVA | Why Planes Vanish | PBS Video." PBS Video. Web. 2 Dec. 2014. <http://video.pbs.org/video/2365334815/?Utm_source=youtube&utm_medium=pbsofficial&utm_campaign=nova_covefullprogram>.

[38] Tanenbaum, Andrew. (2002). Computer Networks (4th Edition). Prentice Hall. pp. 274–275.

[39] Wolff, Christian. "Radar Basics." Radar Basics. Web. 2 Dec. 2014. <http://www.radartutorial.eu/13.ssr/sr24.en.html>.

[40] " The 10 principles of mobile interface design" Creative Bloq. Apr 15 2012. Web. Nov 6 2014. <http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910>

[41] "Applying Fitts' Law To Mobile Interface Design". tuts+. May 9 2012. Web Nov 6 2014. <http://webdesign.tutsplus.com/articles/applying-fitts-law-to-mobile-interface-design--webdesign-6919>

[42] "Applications for RTL based Software Defined Radios". ieee. Mar 2014. Web. Oct 28 2014. <http://sites.ieee.org/wnc/files/2014/04/IEEE_Frady_SDR_Mar_2014.pdf>

[43] "Software Defined Radio". Texas Instruments. Feb 2014. Web. Oct 28 2014.. <http://www.ti.com/solution/software-defined-radio-sdr-diagram>

[44] "Understanding mode S technology". defense electronics. Dec 5 2005 Web. Nov 2014. <http://defenseelectronicsmag.com/systems-amp-subsystems/understanding-mode-s-technology>

[45] "Why Mode S". radartutorial. Web Oct 14 2014 <http://www.radartutorial.eu/13.ssr/sr20.en.htm>

[46] "Combining Two TV Antennas". Stallions Satellite and Antenna. Web. Nov 12 2014. <http://www.tvantenna.com/support/tutorials/combining.html>

[47] "ADS-B Decoder and Software with PIC18F2550". sprut. Oct 14 2014. <http://www.sprut.de/electronic/pic/projekte/adsb/adsb_en.html>

[48] "OSH Park Pricing and Specifications". OSH Park. Web Dec 2 2014. <https://oshpark.com/pricing>

[49] “Recipes for Printed Hairpin Filters” w1ghz. 2011. Web Dec 1 2014 <2011 Web Dec 1 2014 <http://www.w1ghz.org/filter/Recipes_for_Printed_Hairpin_Filters.pdf >

[50] “Coaxial Collinear Antenna for ADS-B Receiver”. balarad.net Web Oct 12 2014 <http://www.balarad.net/>

[51] “Antennas”. Lamont V. Blake. 1966. John Wiley & Sons, Inc

[52] “Antenna Engineering Handbook”. Richard C. Johnson. 1961. McGraw-Hill.

[53] “Antenna Theory and Design”. H. Paul Williams. 1966. Sir Isaac Pitman & Sons Ltd.

[54] "AD8319 (Rev. C)." Analog Devices, Inc. (n.d.): n. pag. Web. <http://www.analog.com/media/en/technical-documentation/evaluation-documentation/AD8319.pdf>.

[55] "PmodBT2 - Bluetooth Interface.", Diligent, Inc., n.d. Web. <http://www.digilentinc.com/Products/Detail.cfm?Prod=PMOD-BT2>.

Appendix A: Copyright Permissions

Coaxial Collinear Diagram

Hello Sean,

yes, please use diagrams from my web site http://www.balarad.net in your report.
Would you please send me the link to this report if it would be possible?

Dusan Balara
balarad@balarad.net

Ribbon Filter Image

you have my permission, as long as credit is given

paul

Texas Instruments

Mark as unread

Bassuk, Larry <l-bassuk@ti.com>
Tue 12/2/2014 11:39 AM

To: ☐llam1;

Action Items | + Get more apps

Thank you for your interest in Texas Instruments. We grant the permission you request in your email below.

On each copy, please provide the following credit:

Courtesy Texas Instruments

Regards,

Larry Bassuk
Deputy General Patent Counsel &
Copyright Counsel
Texas Instruments Incorporated
214-479-1152

Maxim Integrated

AUTHORIZATION FOR LIMITED USE OF MAXIM MATERIAL
Copy and Use Restrictions

Upon Maxim's receipt of a copy of this Authorization form signed by You, Maxim hereby grants You permission to download, reproduce, display and distribute the Materials posted on its site solely for informational and non-commercial or personal use subject to the following terms and conditions:

1. No modifications may be made to the Materials.
2. All copyright and proprietary notices as they appear in such Material shall remain in such Material.
3. The following credit line must appear in such Material: "Copyright Maxim Integrated Products (http://www.maximintegrated.com). Used by permission."
4. Any unauthorized reproduction, disclosure, distribution or use of any Maxim Application Notes, Reference Designs and other such Design Materials shall be reported promptly to Maxim.
5. Use of Maxim trademarks in literature must attribute ownership to Maxim by use of a footnote indicating the word, name, symbol or device and applicable ownership by Maxim.

MAXIM DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING YOUR USE OR REPRODUCTION OF THE MAXIM DATA SHEETS OR IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE.

By signing this Authorization, You agree to the terms and conditions herein and agree to defend and indemnify Maxim, its officers, directors, employees and agents from any and all damages, liability or causes of action, whether known or unknown incurred in connection with any claim, lawsuit or action that arises or results from Your use or reproduction of any Maxim material.

Please complete the information below, sign where indicated, and return the original to:
Maxim Integrated Products, Inc.
Attn: MarCom
160 Rio Robles
San Jose, CA 95134
memcenter@maximintegrated.com

Analog Devices

TERMS OF USE

Please carefully read these terms of use before using this Analog Devices, Inc. ("ADI") website (the "ADI Site"). By using the ADI Site you indicate your agreement to these terms. If you do not accept these terms, please do not use the ADI Site. These terms are intended to be supplemented by other license agreements and specific terms governing:

- Software downloaded from the ADI Site
- Analog Devices Standard Terms and Conditions of Sale
- Online Seminar / Webcast Terms and Conditions of Use Agreement
- Community Forum Terms of Use
- Evaluation Board License Agreement

PROPRIETARY RIGHTS AND USE RESTRICTIONS

The ADI Site is the property of ADI. Copyright © 1995-2014 by Analog Devices, Inc. All copyright, trademark, and other intellectual property and proprietary rights in the ADI Site and in the software, text, graphics, design elements, audio, and all other materials originated or used by ADI at its site (the "ADI Information") are reserved to ADI and its licensors. Except as explicitly provided herein, the ADI Information may not be reproduced, published, adapted, modified, displayed, distributed or sold in any manner, in any form or media, without the prior written permission of ADI.

ADI grants you a revocable, non-transferable, non-exclusive license to view, print out or download a single copy of the ADI Information, solely for internal non-commercial or informational use provided that you do not remove any copyright, trademark or other proprietary notices. Other than the foregoing limited single copy license, no transfer of any other rights is made or intended with respect to the ADI Information and the ADI Site. ADI may terminate this single copy license at any time for any reason, including for any breach of these terms of use, or breach of any other ADI agreement or policy. Upon termination you will immediately destroy all ADI Information in your possession or control.

You will not post on or transmit to the ADI Site any defamatory, libelous, obscene, profane, pornographic, threatening, or unlawful materials or any materials that could constitute or encourage conduct that would be considered a criminal offense or give rise to civil liability, whether under statute, common law or equitable principles, or otherwise violate any law. You agree to defend, indemnify, and hold ADI, its subsidiaries, employees, directors and officers harmless from and against any and all claims, damages, costs and expenses, including attorneys' fees, arising from or related to such postings or transmissions.

Diligent, Inc.

Appendix B: Data Sheets