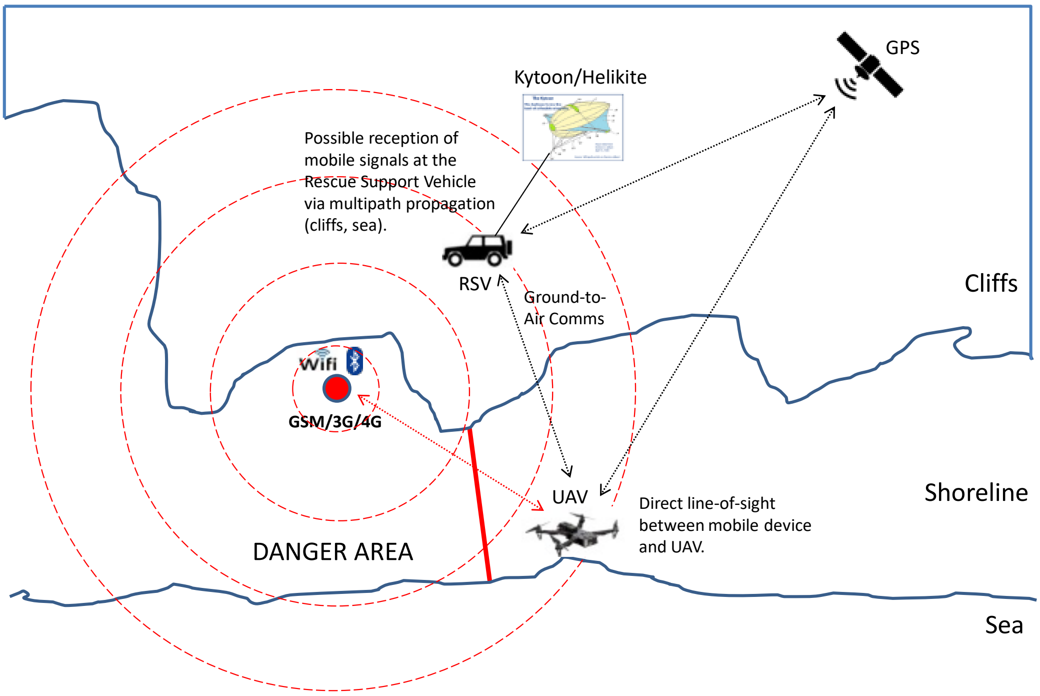
EESW Project Report

Adroddiad Prosiect EESW

2016-17



Monmouth Comprehensive School

in association with

General Dynamics UK

Project: **RADIAL**

(Radio Identification and Location)

Team members:

Bethan Arnell
Reuben Blakebrough
Joshua Williams
William Steer-Rose
Jago Strong-Wright

Contact teacher:

Colin Brown

Company Engineers:

Ellie Bettinson
Kieran Williams

EESW STEM Cymru

Monmouth Comprehensive School | Ysgol Gyfun Trefynwy

GENERAL DYNAMICS United Kingdom Limited

Table of Contents

Table of Figures

1. Executive Summary

Every day, hundreds of people around the world find themselves in difficulty, in isolated locations and unable to call for assistance. These people are sometimes victims of accidents or natural disasters but often they are partly responsible for their predicament through lack of preparedness and excessive risk-taking. Many people who find themselves in trouble are carrying with them electronic communications devices that they cannot use to call for help because they are out of range of the necessary communications infrastructure.

The Radio Identification and Location (RADIAL) project being executed by Monmouth Comprehensive School and General Dynamics United Kingdom Limited (GDUK) has evaluated options for a rapidly deployable system to detect and locate the radio signals transmitted by modern personal electronic devices e.g. mobile phones, tablets. A prototype of the preferred option has been designed and implemented, taking into account the legislation for the use of such a system, in particular safety and radio transmissions. The prototype shall be trialled on location with GDUK in March 2017.

Before settling on the final scope of the RADIAL project the school team explored a wide range of ideas that might meet the needs of different types of Emergency Services users. A number of these ideas were discussed when the school team visited GDUK for a site tour. GDUK then decided whether they could adapt any of them to meet their customers' needs. Some ideas were considered too impractical at the current time e.g. a system that could automatically call the emergency services if a swimmer strayed into a dangerous position in the sea. After reviewing these alternatives it was decided that a system that could detect and precisely locate people in areas of coastline or countryside where they may be at risk would be a useful capability for GDUK to offer to customers.

The school team then worked to further understand the project scope and did a lot of research. It was decided to develop a radio detection system with airborne sensors to minimise the blocking of radio signals by the environment. It was then decided that an Unmanned Aerial Vehicle (UAV or 'drone') operated from a Rescue Support Vehicle (RSV) on the ground would provide the best way to increase coverage and improve detection of radio signals. The team then produced a list of questions that it needed GDUK's advice on. With GDUK's assistance the team selected a Software Defined Radio (HackRF One SDR) that could detect the range of radio frequencies used by mobile phones and similar devices. A small lightweight processor that could be packaged with the SDR was required to run the SDR software (called GNU radio). The Raspberry Pi 3 (RPi3) was selected from a number of alternative processors. A Global Positioning System (GPS) hardware module was connected to the RPi3 to allow the location of the UAV to be reported back to the RSV.

A considerable effort was made to learn how to get the SDR software to detect the required radio frequencies and power levels. The software was written using the Python language. The software loaded on the RPi3 can process frequencies and power levels and associate these with GPS coordinates. This data is then transmitted to a base computer located in the RSV. A visual display was developed for the base computer to enable users to locate a source of radio signals and direct the UAV. A device that transmits at 27MHz (an unlicensed frequency band used with remote control toys) was acquired to test the system in places where there were already Wi Fi and mobile phone signals.

At the workshop in Cardiff Metropolitan University it was agreed exactly which parts of the overall system would be designed and built by the school team and which parts would be provided by GDUK. It was decided that the team would design and make a case suitable to protect the RPi3 and fix it to the UAV, using sacrificial connectors and cabling between the case and the RPi3. This would then be fitted to the UAV by GDUK alongside the HackRF One SDR which would be in a special screened case. The team sent the specifications for the case and the cabling to GDUK for them to design the rest of the system to be used in the trial.

2.Introduction

2.1 Context of the Project

A list of potential project ideas was presented by GDUK to the students during an EESW introductory visit to Monmouth in September 2016. From these the group elected to investigate doing a project that would support public safety, possibly involving radio communication and/or drones, as these were areas of interest for GDUK.

2.2 School Profile

Monmouth Comprehensive School is a state secondary school situated in Monmouth, a small town on the English and Welsh borders. It is a mixed school with students varying from ages 11 to 18 and is supported by the Monmouthshire County Council. The KS3 and GCSE students amount to approximately 1300 students, with around 300 students in the Sixth Form. Over 20 students choose Physics each year for A Level, many of these continuing with degrees in Engineering and Physics.

2.3 Company Profile

General Dynamics United Kingdom Limited (GDUK) is a leading prime contractor and complex systems integrator working in partnership with government, military and private companies around the world. The company has had operations in the UK for over 50 years and now consists of two main businesses: Mission Systems and Land Systems. GDUK provides highly integrated systems, innovative products and support services to a range of markets including mission critical communications, armoured fighting vehicles, avionics systems and deployable infrastructure.

General Dynamics is currently investing significant effort into research and development (R&D) for Public Safety and Security markets. This is work is called the SHIELD Ecosystem. One aspect of this R&D is investigation into systems and technologies that attempt to detect and hopefully prevent detect occurrences of errant behaviours which have a detrimental impact on safety and security.

2.4 Student Profiles

Bethan Arnell is studying Mathematics, Further Mathematics, Physics and Chemistry at A Level. She hopes to study Pure Mathematics at university.

Reuben Blakebrough is studying Mathematics, Physics and Computer Sciences at A Level. He intends to either do a BSc in Computer Sciences at a university or a software apprenticeship.

Joshua Williams is studying Mathematics, Further Mathematics, Physics and Computer Sciences at A Level. He hopes to either study Physics or Computer Sciences at university.

Jago Strong-Wright is studying Mathematics, Further Mathematics, Physics and Chemistry at A Level. He hopes to study Physics at university. He sails competitively with a Welsh National Squad.

William Steer-Rose is studying Mathematics, Further Mathematics and Physics at A Level. He has an Arkwright Engineering Scholarship and he intends to study Structural Engineering at university.

Name	Role within Team
Jago	Software for on board processor and SDR
Will	Case for the Hack One RF and Pi.
Beth	Problem analysis, research and report writing.
Josh	Software for visuals on ground computer.
Reuben	Problem analysis and report writing.

2.5 Key to Acronyms and Abbreviations

ABS Acrylonitrile Butadiene Styrene

FM Frequency Modulated

GDUK General Dynamics United Kingdom Limited

GPS Global Positioning System

HAT Hardware Attached on Top

HIPs High Impact polystyrene

IRS Interface Requirement Specification

NIC Network Interface Card

RPi3 Raspberry Pi 3

RF Radio Frequency

RSV Rescue Support Vehicle

SMA Sub –minature coaxial connector version A

SDR Software Defined Radio

UAV Unmanned Aerial Vehicle

USB Universal Serial Bus

3. Project and Brief Selection

Initial meetings were held in school as follows:

1. *(Sept 23rd*) The group generally explored the whole area of "drones", "surveillance", "public safety" and related topics, trying to define the need they wanted to address and whether it might be possible to demonstrate a working model of a solution within their time, skill and finance limits. They were wary of duplicating already available systems such as a manually controlled drone with camera. They were aware of the limits to flight time with anything they could afford and the need for a person to be constantly watching screens. They considered perimeter surveillance with fixed sensors, but again this is already tried and tested. The team considered the recent drownings off UK beaches with no lifeguard, and of migrants in the Mediterranean, and wondered if they could offer something. Team members each went away to research an aspect.

2. (Sept 29th) The team decided to concentrate on this as a potential scenario. They explored options for knowing that someone is in trouble, identifying where they are and getting eyes on them and potential help before lifeboat/helicopter arrives.

Idea	**Pros**	**Cons**
To search the coast with a drone to pick up heat signatures.	It would cover the whole coast and does not require the swimmer to have any special device.	Expensive and not reliable due to water waves and the body cools down in water meaning body may not be detectable. Would not be able to fly for very long, due to battery life. May get confused if it was to be used in a built up area of the coast.
To have a perimeter with lasers so that if anyone breaks a connection it sends a message to base.	Almost instantaneous knowledge of where they are.	Lasers could be triggered by waves, non-human animals, or debris; hence this is unlikely to work effectively. Expensive if used extensively. Possible danger to sight.
To equip everyone with a distress beacon and use triangulation to detect their position.	Signals could be detected by fixed detectors or patrolling drone.	People usually in danger are not well equipped, therefore this would not work for the masses, unless routinely and cheaply incorporated into swimwear or it became the norm to wear a wristband.

The victim could perhaps trigger a detector automatically, either through a cheap passive device or just the presence of a live human body, when going beyond the boundary of a "safe" space. As the above table shows, there were no obvious ways of doing this, but it was decided to research the devices carried by skiers to aid detection when buried in an avalanche. Some of these are active transponders incorporated into clothing; perhaps it could become routine for every swimmer to have such device on a wristband or as part of clothing? Crossing a particular boundary between detectors would locate position; alternatively there would have to be a system for locating the source of the transmission. A drone, or other vehicle, could then be sent to give a visual image for rescuers and if possible offer some practical lifesaving assistance. It was confirmed by research that it was impossible to keep any affordable drone in the air continuously.

3. (October 7th) A project proposal was formulated and forwarded to GDUK:

Radio transmitters for beach water users to locate them by triangulation when they are in distress.

Water users in distress press a button on their device and bases stations forward the intensity of the signal to a main base, which triangulates location.

A fixed wing drone is launched which flies over location, a human checks from video that they are in distress. A flotation device is dropped to aid the person in distress and the lifeguard is summoned. This is cheaper, faster, and has more carrying capacity than an affordable conventional drone. GDUK had independently alerted us to the development of new lighter than air devices; this might give more flotation? An approximate price per drone was suggested as follows:

Radio receivers ≈ £40
Drone + launcher ≈ £150
Other parts ≈ £10

This seems quite cost-effective though it was not possible to cost the fixed infrastructure at this stage.

4. GDUK contact engineers provided detailed feedback on this proposal and provided the team with a revised scope to the project that was a good match to the needs of some of their Emergency Services customers:

I. The majority of people who come to harm in isolated coastal locations usually do so because they are: unfamiliar with their surroundings, and ill prepared for the activity they are undertaking

   The people most likely to get themselves into trouble are the ones least likely to have a transmitter, because they: didn't know the service was available to them, wouldn't pay for it, assuming the government would not fund it to be free to use might think the device is used to track their whereabouts even when they aren't in trouble, were too busy or could not be bothered.

II. RF propagation in and around water is troublesome. There are many problems inherent to the medium such as reflections, refraction and energy dispersion. Any transmitting device will have to take this into account. *(The team's research into avalanche detectors had also raised this issue; the cheap passive devices in clothing explicitly work less well the wetter the snow is)*

III. The chances of the UAV dropping the flotation aid such that it lands right next to the person (or persons) in difficulty is slim. If, for example, they're become stranded on a sandbar or rocks then the floatation aid could well land in a place that will require someone getting themselves into greater peril trying to retrieve it.

IV. Coastal locations, particularly those with cliffs, present very challenging environmental conditions. The UAV is going to have to fly in high winds (target: Beaufort Scale 7, moderate gale) and be resistant to sand particles and moisture. Controlling a UAV from ground level is probably very difficult in many coastal wind conditions; it may be better to let the UAV adapt autonomously (see below).

V. The mechanics inside the UAV are also going to have to survive crashing from, say, 50 m above ground level without significant damage to the UAV or payload (sensors).

Revised Proposal

It was pointed out that people in danger at the coast are not just swimmers. Many people find themselves stranded by rising tide. These people are likely to be carrying phones or other devices that will transmit radio waves. Indeed, their first response to perceiving the danger of their situation may well be to try to make a phone call, but they may find that there is "no phone signal" at the base of a cliff or in a remote spot. Kayakers, small boat users and walkers in mountainous areas may be similarly equipped and try unsuccessfully to do the same. It was suggested that the team try to detect the radio signals generated by such people; some devices radiate continuously eg Wi Fi. The revised proposal could involve a UAV 'swarm'.

Advantages of the revised proposal are:

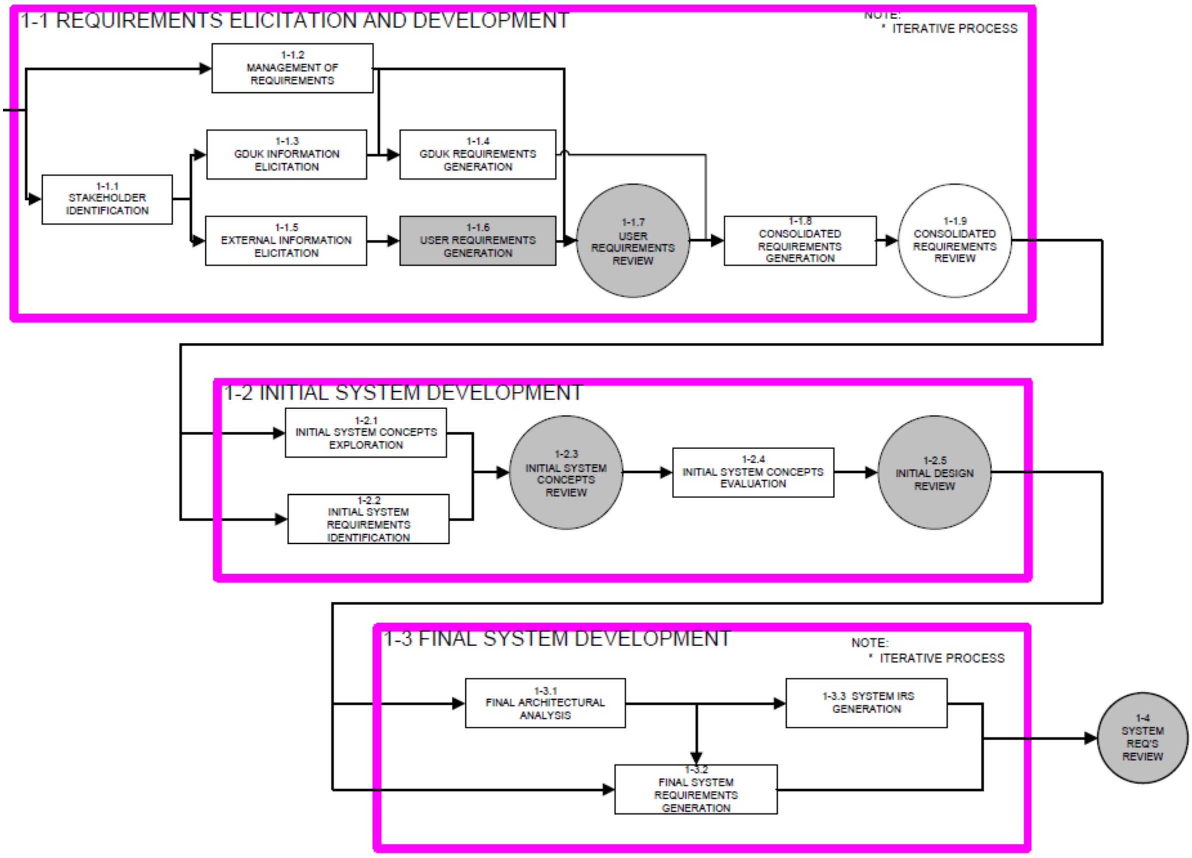
- As there is no need to point a sensor (e.g. camera) accurately at a specific location whilst maintaining good stability any UAVs do not have to be very precisely located i.e. being blown about by the wind isn't so much of an issue.
- The reduction in payload means the team could consider introducing complementary sensors e.g. thermal infrared.
- The solution is now much less domain specific. Removing the floatation aid aspect and making it purely a location finder means the solution could be applied to many different public safety and military scenarios.
- There is less need to worry about payload and power management. RF sensors and transmitters will be much lighter and less power hungry than a decent camera and a floatation aid with launch mechanism.

After visiting GDUK on 20th October for the "Requirements Review" (see 4.1), the team agreed to pursue this avenue. There was discussion of various possible ways to give a detector height so that it had good coverage: a tethered balloon (e.g. a Kytoon or Helikite), perhaps from a vehicle or lifeboat, a manually piloted UAV, or a UAV that would automatically follow its operator without needing to be actively piloted. The final brief was agreed. This is summarised in the Executive Summary and also included as Appendix 1.

4. Project Management

4.1 Process advised by GDUK

During the visit to GDUK the team were advised to follow a process that GDUK uses for many of their own projects:



This material has been licensed to General Dynamics United Kingdom Limited

Figure 1 GDUK Concept Exploration Process

This diagram shows the part of GDUK's process called Concept Exploration. GDUK said this was suitable for the RADIAL project since they are still working on the concepts for products for Emergency Services customers. There are three phases in Concept Exploration; Requirements Elicitation and Development, Initial System Development and Final System Development. Not all the stages of the process would be relevant to the RADIAL project as GDUK were also going to be the user of the system.

The first phase of Concept Exploration is where the scope of the project and the needs of users (requirements) are agreed. Requirements would come from GDUK and other sources e.g. legislation that limits how drones can be used in the UK. The requirements that the team would work with were to be agreed at a Requirements Review with GDUK. The Initial System Development phase is where different solutions to the requirements are worked through and the capabilities, costs and risks of these solutions are considered. The Initial System Concepts Review is a review of all the ideas and concepts and how well they meet the project scope. The Initial Design Review is a second meeting where it's agreed which of the alternative solutions is the best balance between what the users want from the system and how much they want to pay for it. At this point GDUK would decide what was the best way forward for the RADIAL project.

The Final System Development phase stage is where more detailed requirements and architecture design are agreed so that models and prototypes can be built to test out whether the concepts are good and that the requirements can be implemented in a real system. The work in this phase includes writing an Interface Requirements Specification (IRS) that details the connections between different parts of the system. (See section 5.71). It also deals with how the user should interface to the system, and whether system could survive in a real world environment. GDUK said that they would like the school to help them build a prototype of the RADIAL system before having their own System Requirements Review (SRR) to decide whether they had all the requirements they needed to take the project forward.

4.2 Allocation of Time and Resources

Meetings were held weekly in school, to monitor progress and set tasks, with extra time as needed. Members of the team were allocated tasks according to their strengths and interests. (See section 2.3).

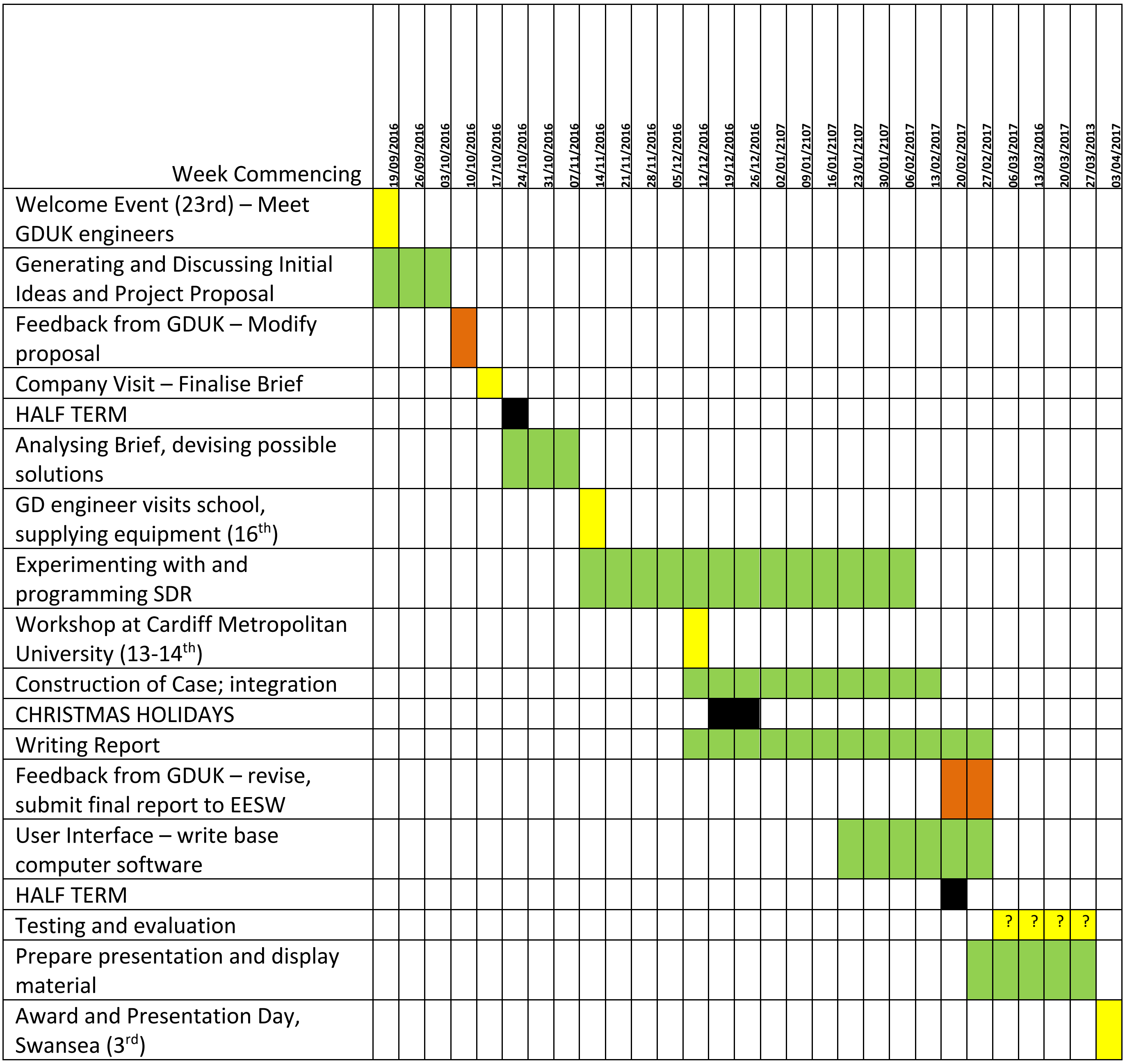
The Gantt chart overleaf shows the progress of the project.

Green indicates work in school

Yellow indicates input in person from GDUK engineers

Orange indicates feedback by email from GDUK engineers

Gantt Chart



? At time of writing, precise date of the test is not finalised.

Figure 2: Gantt Chart

5. Development of Design

5.1 Overall System

This diagram shows the receiver, mounted on a UAV, detecting signals from an elevated position and sending them to the ground PC, possibly mounted on a rescue vehicle or boat. GPS is used so that the position of the UAV is known. Received signal strength can be mapped against position to help locate the signal source. The receiver could alternatively be mounted on a tethered balloon.

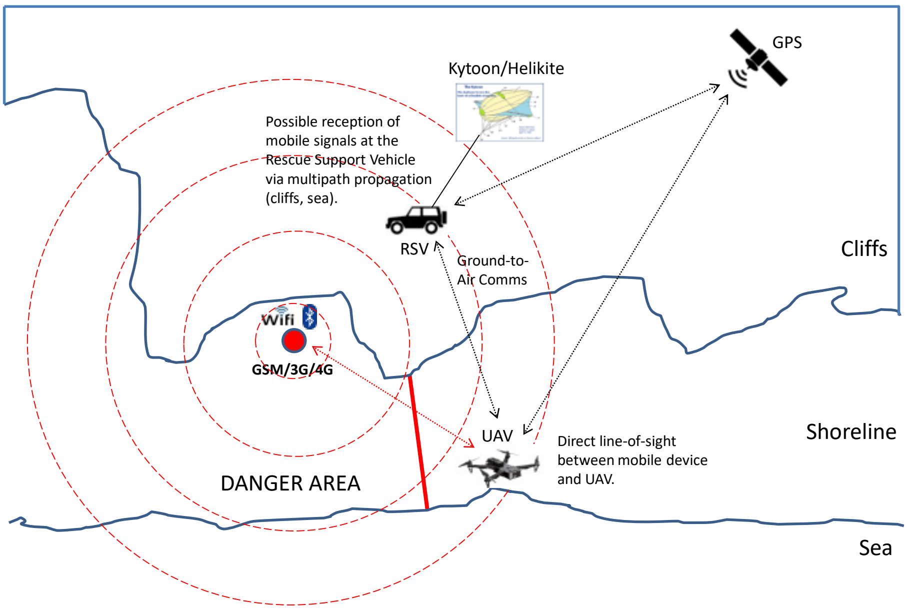


Figure3 Overall System

5.2 Problem analysis

A more detailed analysis at school on 3rd November raised these questions:

1. How are height and stable platform to be achieved?
2. Which components are to be being flown? (Antenna, amplifier, receiver, processor, power source...?)
3. What are they contained in?
4. What is the weight of above (+ tether, if a balloon)? -and hence lift required?
5. Length of tether – max height? (Range of receiver– power of typical phone, sensitivity ?)
6. Communications to ground – radio/Wi Fi, electrical cable, optic fibre?
7. What is best radio receiver to use; which frequencies should be scanned; how would the receiver be controlled to do this?
8. Is a test frequency required for indoor testing and demonstration with interference present?
9. What kind of demonstration/test of overall function is practical?
10. How could position be determined; do signals from more than 1 detector need to be processed?
11. What is the most useful final output display?

5.3 Airborne Platform for Test

The receiver would have to be flown using a drone or lighter than air balloon. Research indicated that a vehicle capable of lifting the system, even made as light as possible, would be beyond the budget. It may require a licence in order to fly. It was agreed with GDUK that the team should just build the detecting system and that it would either be tested on a high structure or GDUK might be able to provide a suitable vehicle and fly it. This still gave the requirement for the system to be as light as possible.

5.4 Selection and operation of Radio Receiver

5.4.1 Required Frequency Range

GDUK engineers advised that a software defined radio product (SDR) should be considered rather than a conventional receiver, set for a particular frequency range, especially as access to a very wide range of frequencies up to 5Ghz would allow many different devices to be detected. This also means that it is straightforward to adapt the solution for other purposes, such as military base protection or casualty recovery, where different radio frequencies, security functions, and UAVs are in use. The ability to detect very weak signals is also required.

75% of the people (61 out of 81) of the people who died of water-related causes at UK coast, shore or beach locations in 2015 were in the age group 10 to 69. This age group is the most common group to have an electronic device such as a Smartphone or a Smartwatch. However, there are now many other devices that emit RF signals that might be detected; the 'Internet of Things':
(See https://www.ofcom.org.uk/manage-your-licence/radiocommunication-licences/internet-of-things):

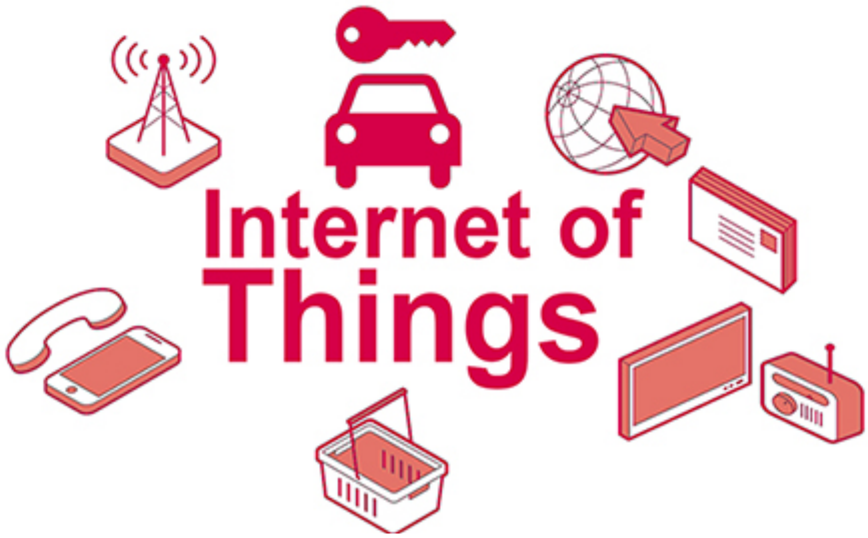


Figure 4 Internet of Things

Definition
The interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data.

Internet of Things frequency ranges

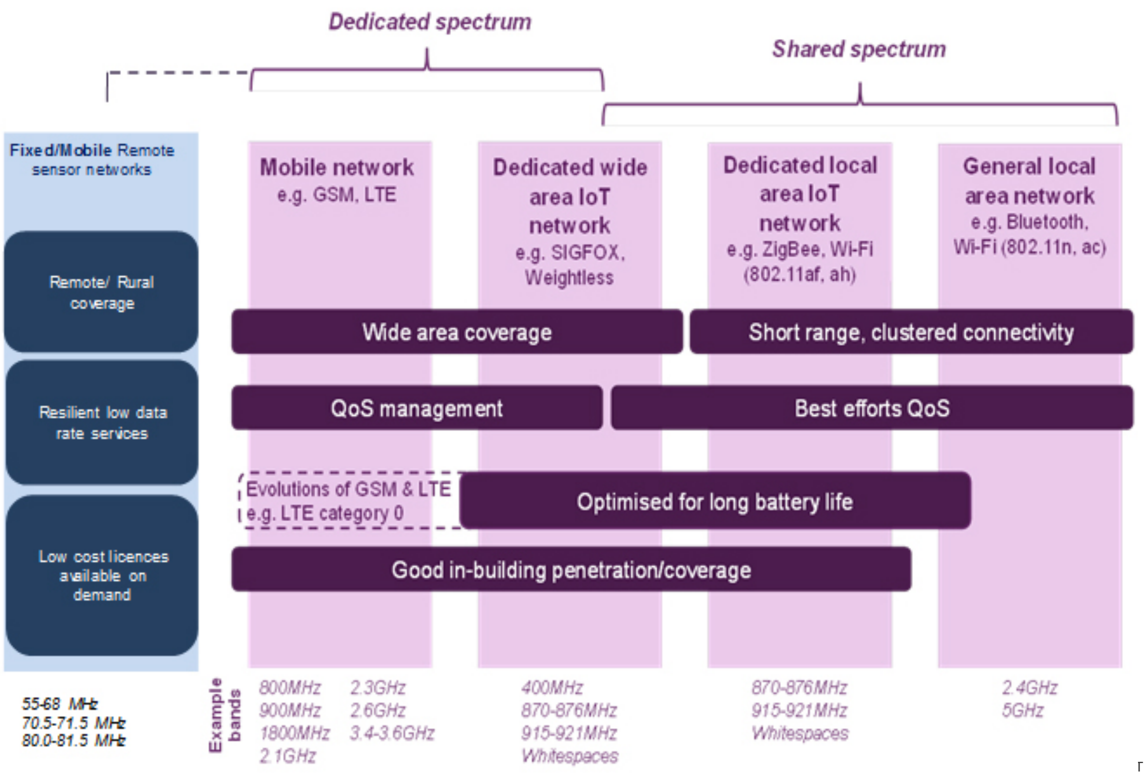


Figure 5 Internet of Things Frequency Ranges

The team decided that the ability to pick up all of the following would be desirable so that people in distress could be detected if they had any of a wide variety of devices that emit RF signals, including, but not limited to mobile phones:

1. Mobile Network Operators (800Mhz, 900MHz, 1800Mhz, 1900Mhz, 2100Mhz, 2600Mhz)
2. Wi Fi (2.4GHz, 5GHz)
3. Bluetooth (2.4GHz)

5.4.2 Selection of Receiver

The following link gives a huge list of available SDRs:

https://en.wikipedia.org/wiki/List_of_software-defined_radios

It is clear from this that most SDRs that can detect up to or beyond 5MHz cost nearly $1000 (£800) or more. However the “HackRF One” from “Great Scott Gadgets” has a frequency range of 1MHz to 6GHz and costs less than £300. It is recommended in several reviews as well as many social media outlets such as “reddit” forums full of enthusiasts.

It is advertised as having the following features:

- *1 MHz to 6 GHz operating frequency*
- *Half-duplex transceiver*
- *Up to 20 million samples per second*
- *8-bit quadrature samples (8-bit I and 8-bit Q)*
- *Compatible with GNU Radio, SDR#, and more*
- *Software-configurable RX and TX gain and baseband filter*
- *Software-controlled antenna port power (50 mA at 3.3 V)*
- *SMA female antenna connector*
- *SMA female clock input and output for synchronization*
- *Convenient buttons for programming*
- *Internal pin headers for expansion*
- *Hi-Speed USB 2.0*
- *USB-powered*
- *Open source hardware*

(https://greatscottgadgets.com/hackrf/)

5.4.3 Using Wi Fi and Phone Networks

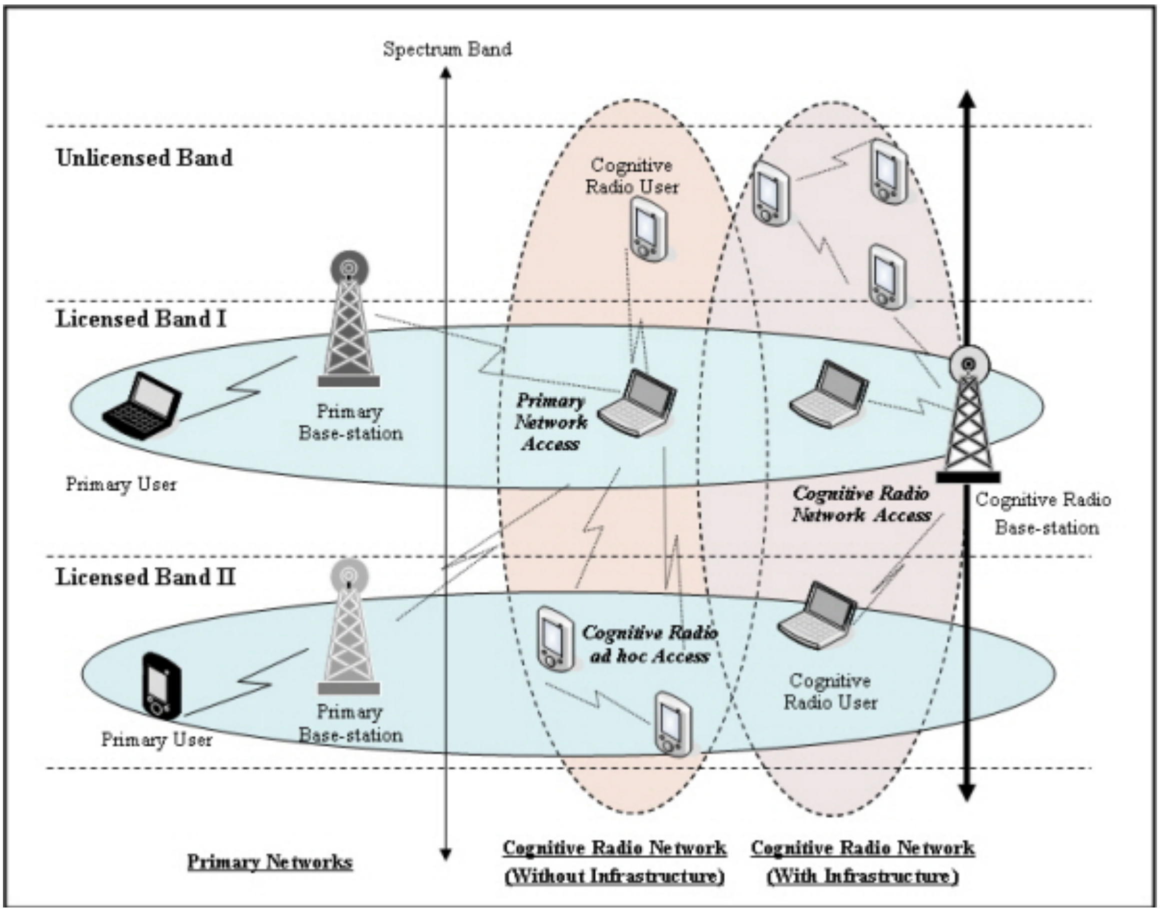


Figure 6 Wi Fi Networks

Wi Fi networks work differently to cellular networks with regard to establishing network connections.

A cellular device searches for a mobile network to connect to. Once the device identifies its desired network, it sends a message to the network to indicate that it has entered into an idle state. The mobile network then checks its database to determine whether this device should be allowed to connect to the network. This is called the "Attach" procedure and details held on the device and the SIM card are used to provide the necessary credentials.

In contrast, Wi Fi access points in home routers or "personal hot spots" send out beacon frames that announce the availability of the network. The Wi Fi network interface card (NIC) in user devices scan all Wi Fi radio channels searching for beacons. When a radio receives a beacon frame from a network, it receives information about the capabilities and configuration of that network and is able to sort those networks by received signal strength. This how Wi Fi scanning apps that can be downloaded onto an iOS or Android device work.

The Wi Fi NIC in a device continues to scan for beacons even after it has connected to a network. By continuing to scan for other networks, the device has options for alternative networks if the current access point's signal becomes too weak.

5.4.4 Legal Constraints applying to working with radio frequency transmitters

It was made clear that any RF transmitters for the purposes of developing and testing the prototype must operate within the licence-exempt frequency bands stated in the legislation; the project does not have time to engage with Ofcom for licences to transmit in different ranges.

The Ofcom document OfW 311(26 July 2010) pertaining to radio controlled models states:

"The use of the specific frequencies used for model control is exempt from the requirement to hold a Wireless Telegraphy Act 2006 (WT Act) providing that the apparatus meets the conditions of the IR2030. In addition the equipment must be lawfully CE marked and comply with all relevant EU directives. Apparatus that complies with the totality of the regulations do not require a licence to operate."

https://www.ofcom.org.uk/spectrum/radio-spectrum-and-the-law/licence-exempt-radio-use/licence-exempt-devices/ofw311

The team decided to use a 27MHz radio-controlled model car transmitter, should they need a controllable source of signals for testing.

5.4.5 The SDR System

A Software Defined Radio (SDR) is a radio that receives a wide frequency range and samples each waveform to give an array of digital data. Digital Signal Processing software is then used to process the data otherwise done by hardware such as filters, mixers, amps, and modulators. This means that different software can be used to decode signals of different frequencies and protocols.
https://www.sparkfun.com/products/1300

There are limitations with the system such as having to fly it with a separate power supply and processor. These are addressed in sections 5.5.1 and 5.6.

Learning how to process the signals was the major part of the software development.

5.5 Software

5.5.1 Onboard Processing

The SDR requires a processor. The options were a PC or various headless computers such the different models of “Rasberry Pi” , “Beaglebone”, “Black” or “Pine64”. The RPi3 was selected the as it has sufficient processing power, was readily available and there is big supporting community producing software. It is also light and small, bearing in mind it needs to be airborne. GNU Radio was preinstalled on the operating system as the software that would process the output from the SDR. Considerable experimentation with the software was undertaken. The first stage was to register that an FM signal was being received. It was discovered how to plot a waveform. The first attempts to extract the power data from the data and write it to file were unsuccessful. The engineer from GDUK suggested using Log FFT Power which samples the wave amplitude and gives useful output in dBm (Decibel-milliwatts). GNU Radio has a visual user interface to write code. Using this, bandpass filters were introduced, allowing selection of particular bandwidths eg Wi Fi at 2.4-2.5 GHz.

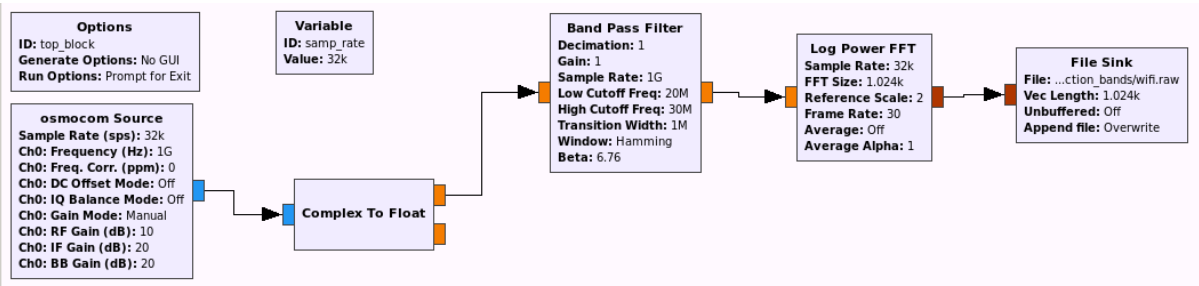


Figure 7 GNU Radio Flowchart

Figure 8 Jago working on software

Figure 9 Software Development by the Team

The initial design for the overall system was as follows:

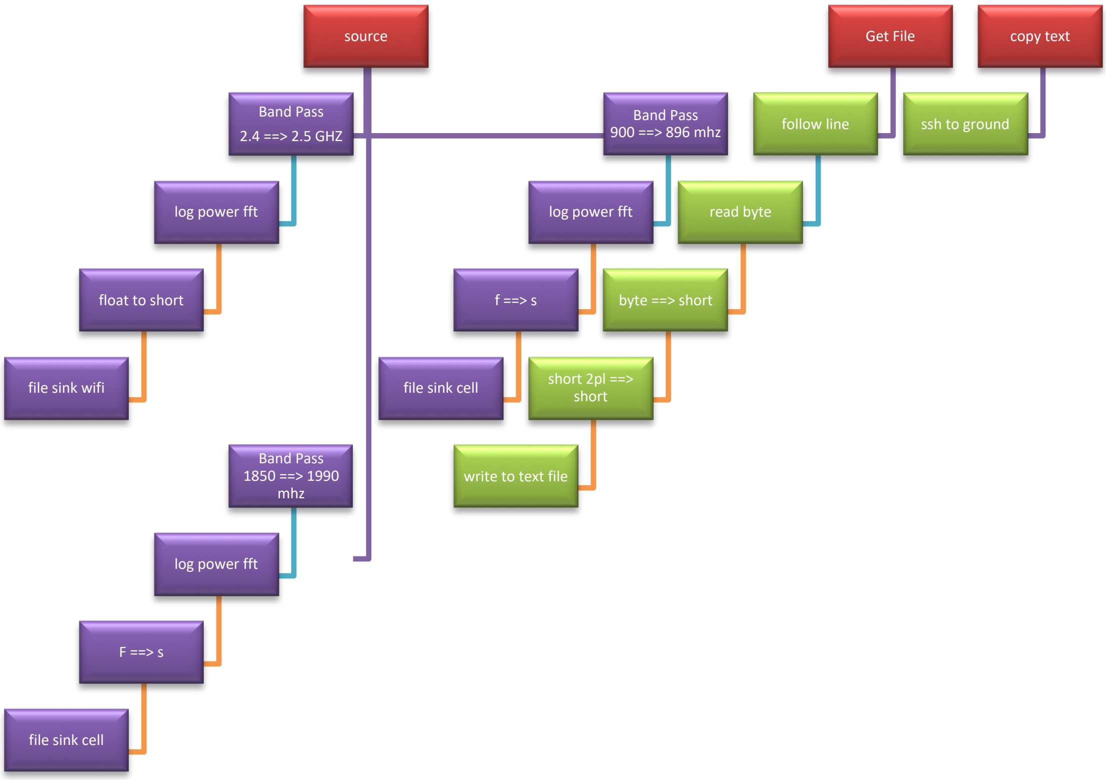


Figure 10 Software Flowchart

In practice only one band pass filter was used with the frequency altered to suit the source being used. "Short" numbers proved not to be sufficiently accurate and so all data is being handled as "floating point" numbers. The "Copy Text" script was not necessary because requests the file.

This output data stream gives a received power value every second in binary. Script was written in *Python* (the "get file" line on the flow chart) to turn this into a decimal number and associate each value with GPS coordinates and heading coming from a GPS HAT attached to the RPi3, and also a timestamp. (See appendix for sample script)

5.5.2 Ground based processing and output display

A PC will be used as the base computer. This will run on Ubuntu because team members know how to use Linux for remote file transfer. (Windows would be cumbersome) There are 2 scripts (in Python):

1. This starts the scripts on the airborne RPi3 and calls the processing script
2. This asks the RPi3 for the file and downloads the output. It reads each line, separating the power level, GPS string and timestamp. It calculates a rolling average as a background, discounting individual readings that differ by more than a predetermined amount, but storing these as the signals of interest.
3. These figures, indicating power level will be plotted on a graph in the position set by the GPS coordinates. Care will be taken to maintain readability in the case of several signals in the same location or close proximity. They will be updated in real-time (about once per second).

5.6 Agreeing the system boundaries for the system trial

As explained in section 5.3, it was decided early on that the team would not be responsible for the airborne platform. The positioning of the two antennae (Signal Detection and GPS) is critical; dependent on size, shape of the UAV and position of UAV control and power components that might give spurious interference signals. Likewise, the UAV will also already have its own air-to-ground communications system. The system does not need its own independent power supply if it can draw from the power supply already on board. Therefore, system boundaries were agreed with GDUK, at the Cardiff workshop, as shown in Figure 11. The team is responsible for the aspects within boxes A and B. GDUK undertook to supply, as far as possible, all the aspects outside these boxes. The team had to supply GDUK with clear dimensions, weights, mounting points, cable connectivity and power requirements, (the "IRS") if the system was to be integrated into their platform successfully.

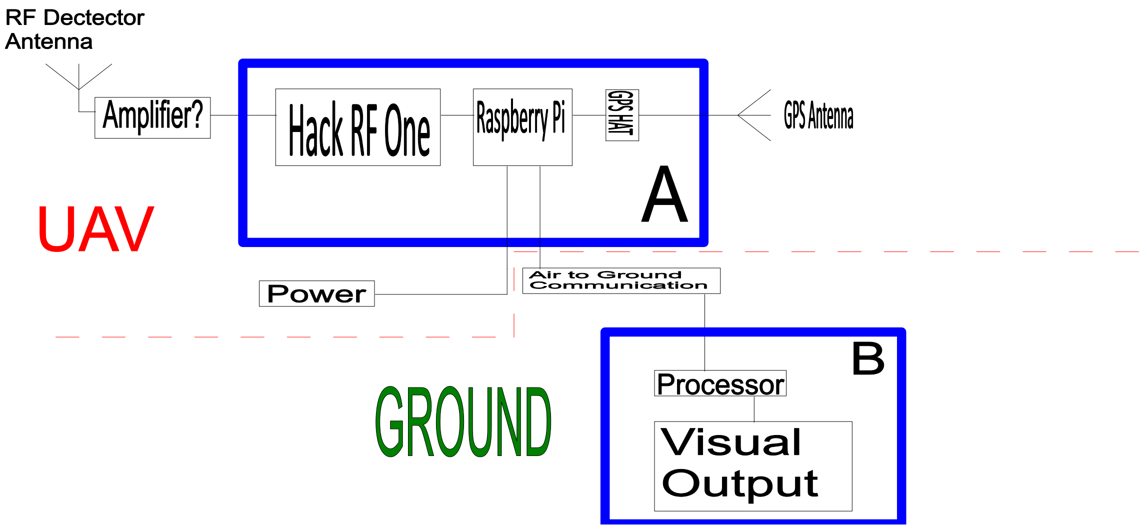


Figure 11 System Boundary Diagram

5.7 Encasing and connecting the products to be flown

The RP3i required a weatherproof and protective case. The Hack1 was already in a fairly tough case, but there was potentially a need to give it further protection, and electromagnetic shielding. The team was advised that they should open the Hack1 only with professional assistance as there was great risk of damaging the sensitive components inside if it was not done properly. For this reason, and to give more versatility when mounting onto the UAV, it was decided to have the 2 different components in different boxes.

5.7.1 Layout and Connectivity of the Rasberry Pi Case

At the Cardiff workshop the team decided to concentrate on designing and constructing a case for the R Pi3 processor with its GPS HAT. It needed a number of different connections for power, data in and out, and GPS RF signal. The picture below shows members of the team working out the layout and type of all the required ports. We were advised to have sacrificial connectors; use short leads to duplicate sockets to avoid damaging the ones on the main boards with over-use. The design has to include room for it to be possible to bend the cables and move them around inside for optimal fit and durability. Sourcing the connectors proved quite difficult with confusions over SMA male/female connectors, internal/external threads etc and ideal cable lengths or bulkhead mountable ends were not readily available. Therefore it was decided to have cables running out of the RPi3 case as shown in figure 13, which, together with figure 16, and the mass of all the components, 390g, formed the IRS given to GDUK.

Figure 12 Initial Case and Connectivity Design

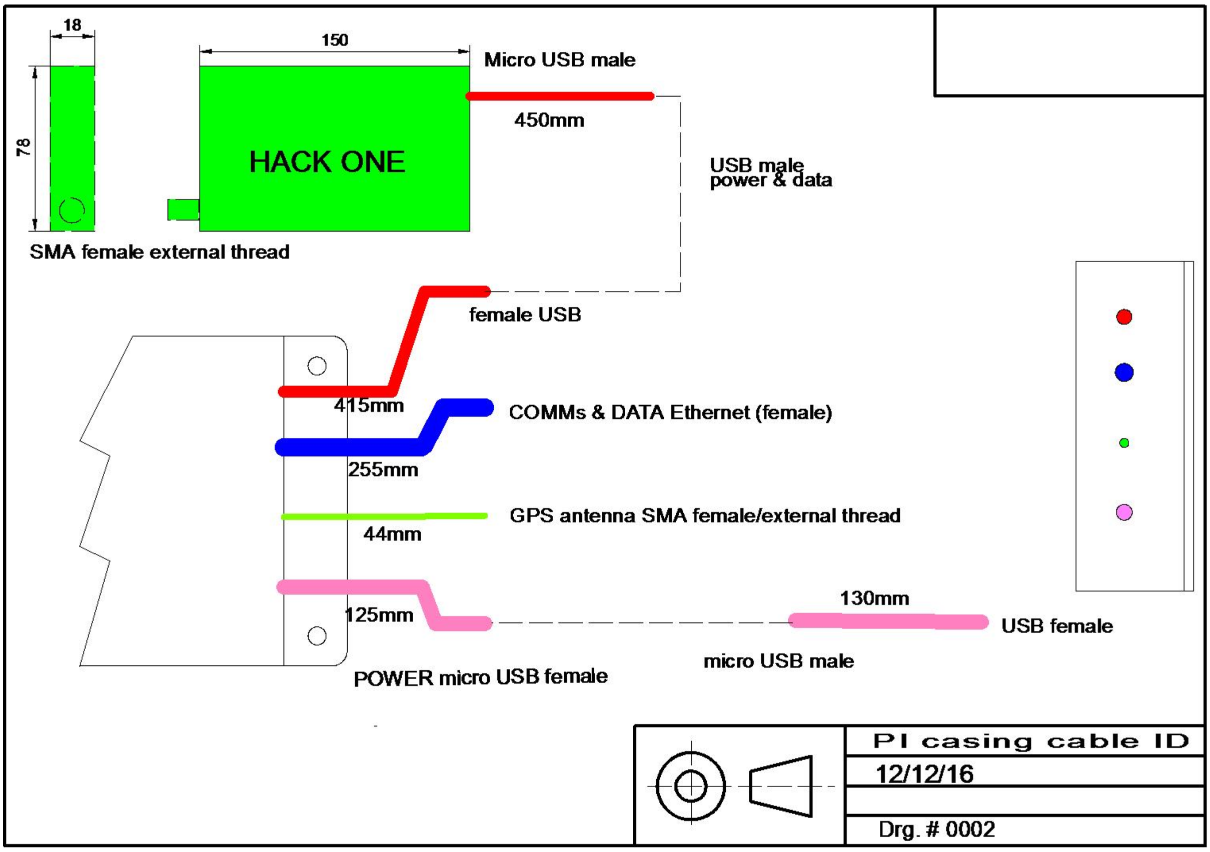


Figure 13 Cable ID Diagram

5.7.2 Choice of materials and construction method

The options considered were:

	Aluminium	ABS	Acrylic	HIPs		bad=	1
Strength to Weight	5	4	3	3		good=	5
Durability	5	4	4	3			
Impact Resistance	5	3	3	2			
Ease of process	2	2	4	2	Low Volume Only		
Total	17	13	14	10			

From the table above it can be seen that aluminium is very much suited to the job but if chosen, a ready-made box would have to be purchased and modified it suit the needs of the project. The ABS and HIPs plastics are a good in large volume. However for a prototype acrylic is the sensible option because the material is already in stock at school, together with a laser cutter; several iterations can be made if necessary until the optimal solution is produced.

5.7.3 Design and Manufacturing of the Case

The case had to be protective, yet easy to get in to for maintenance, with space for all components and wires with everything connected in and out of the system, so a model was first made out of card to test possible layouts. It was soon realised that a lot more space was needed for all the cables and to include mounting points for the drone. After making changes to the design, it was drawn up on 2D Design, and then cut from 3mm acrylic by laser. Holes were drilled in the bottom for the RPi3 to be attached to. The last thing was to add the slots and holes for all of the cables to the HacK RF1, air-to-ground radio, power supply and GPS antenna.

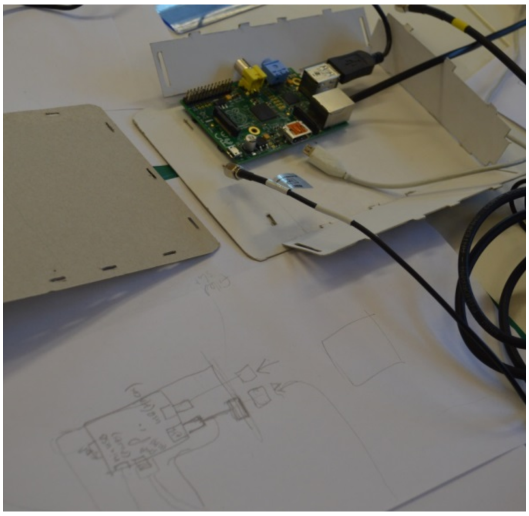


Figure 14 Designing the case

Figure 15 Cardboard Mock-up

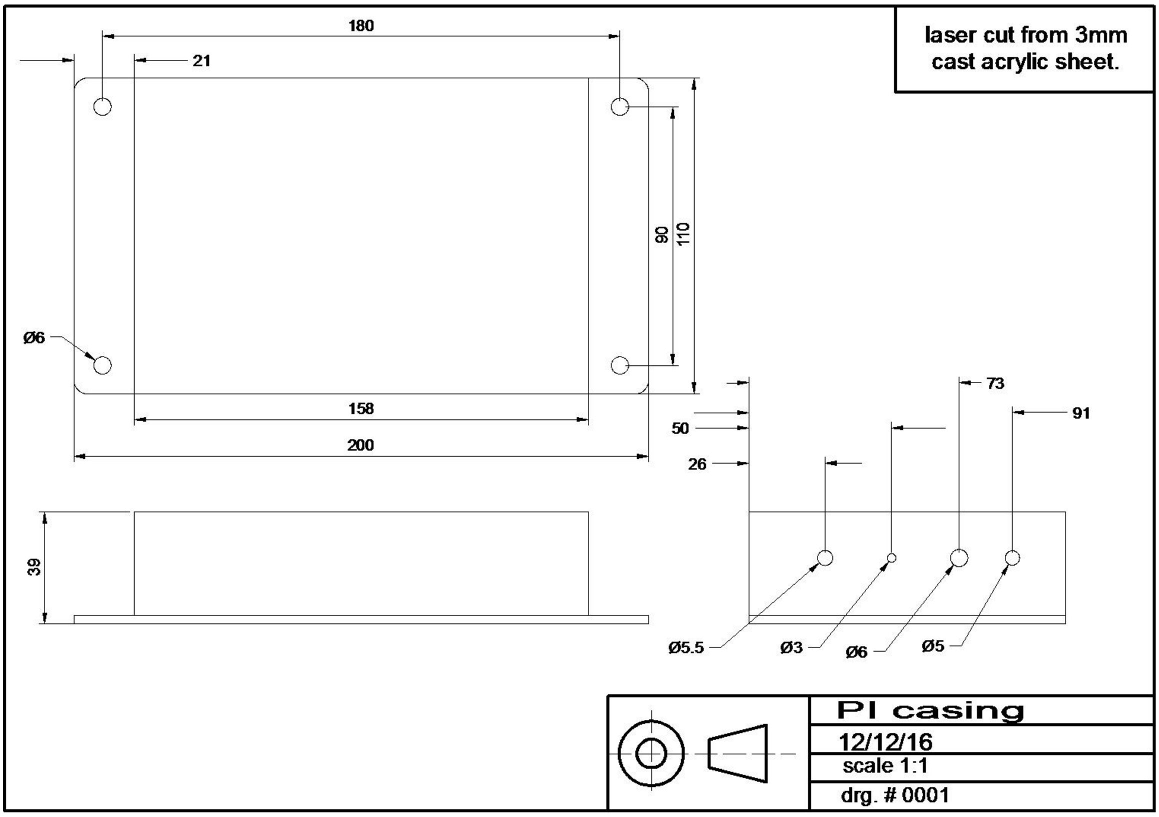


Figure 16 RPi3 Case Drawings

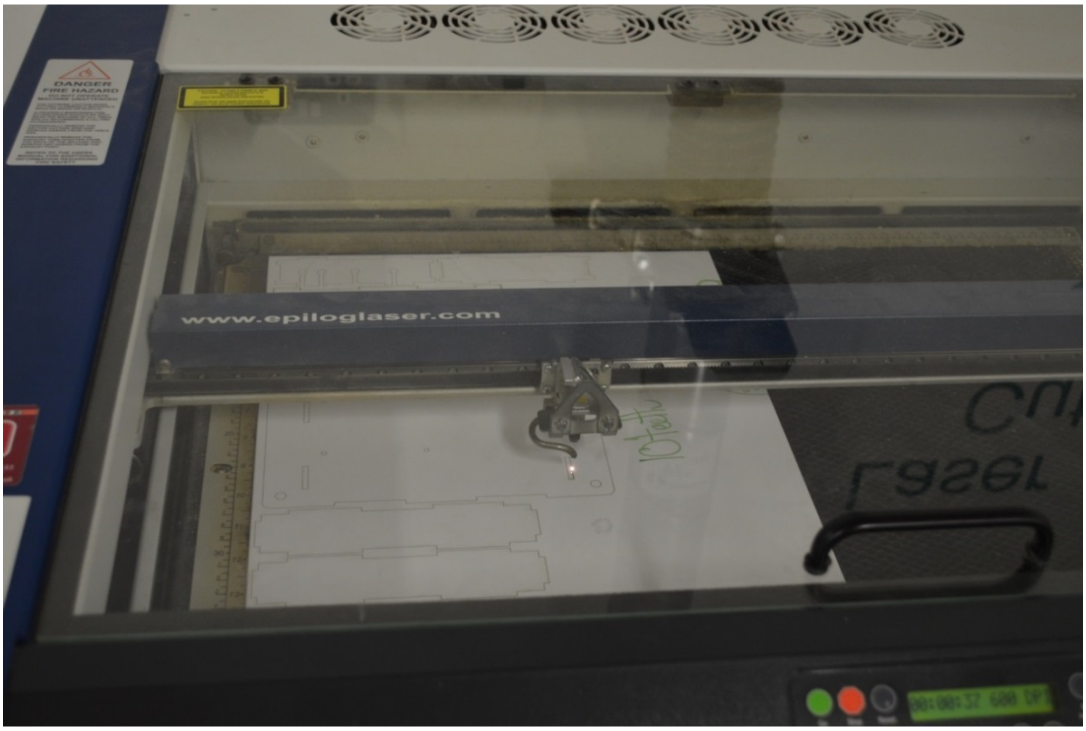


Figure 17 Laser Cutting the RPi3 Case

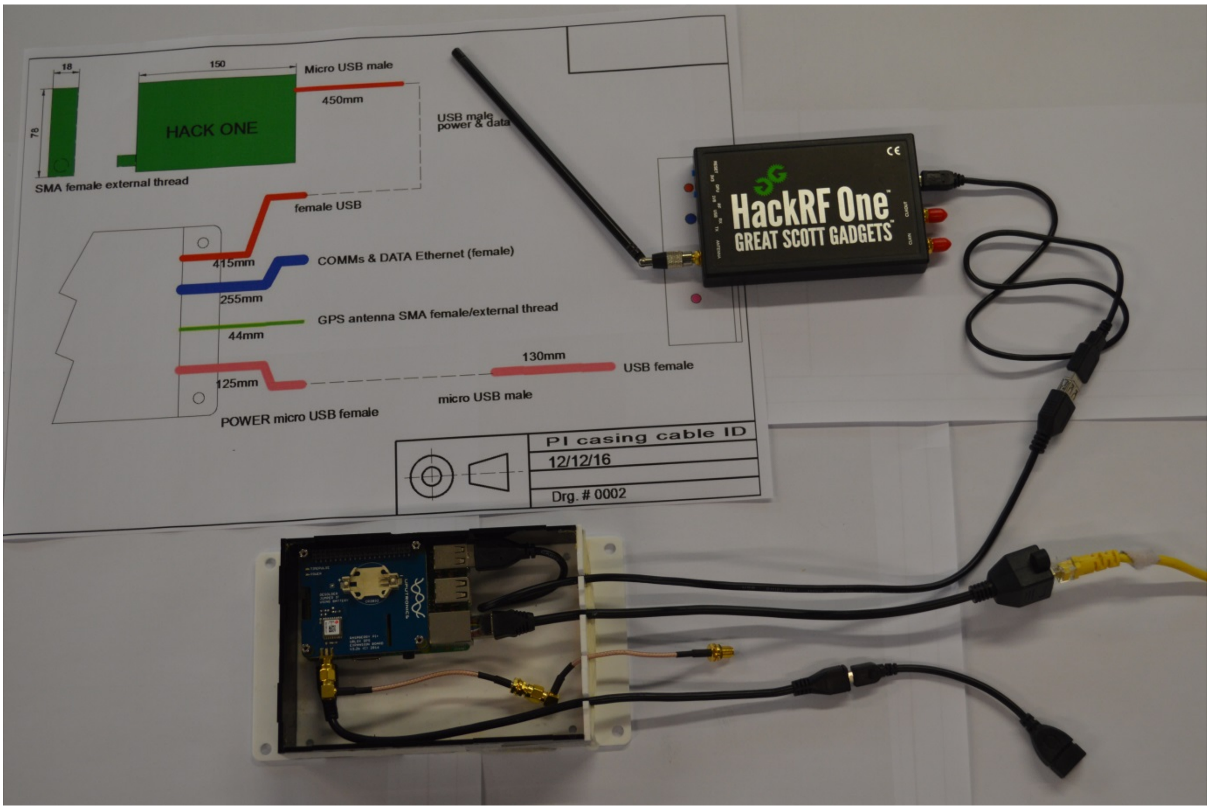


Figure 18 Connections between the Hack RF1 and RPi3

5.8 Position Location

It will be helpful to the rescue services to have as much information as possible about the location of the source of the signal being received. Several options were explored.

5.8.1 Using Received Power Level

"RSS (Received Signal Strength)" methods

With one omnidirectional receiver, measuring the power level of the received signal might give some indication of the distance to the signal source, but not direction. It might be assumed that a weak signal is coming from a distant source, whilst a stronger signal is coming from a closer source. However nothing is known about the transmitting power of the source: a signal might be weak because it is a weak source rather than because it is far away.

For a given power, transmitted uniformly in all directions (isotropically), the power received, **P**, falls off with distance **r** according to the inverse square law.

$\mathbf{P = K/r^2}$ where K depends on output power of source and wavelength

If P is measured in dB: $\mathbf{10\log P = 10\log K - 20\log r}$

Comparing the signal power received at two or more locations might allow some fixing of position. If the distance from the source to receiver 1 is r1, (power P1 in dB) and to receiver 2 is r2, (power P2 in dB), then:

P2 –P1 = [10logK -20logr2] – [10logK – 20logr1]
=20 log (r1/r2)

So the *difference* in dB received gives the *ratio* of the distances. If the difference is zero then the locus of possible positions for the source is the straight line perpendicular bisector of the line joining the two receivers. A non-zero difference gives curved loci as on the diagram.

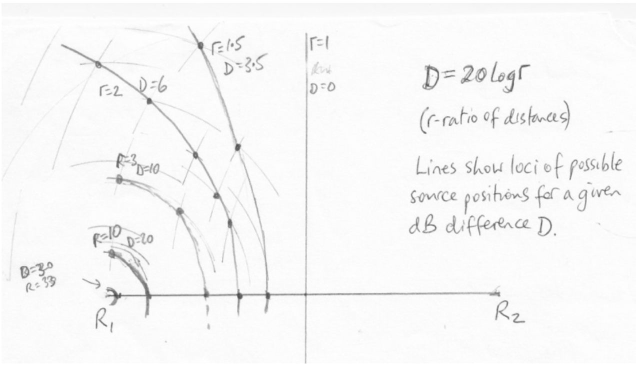


Figure 19 Plot of Possible Source Positions when comparing Power Level Received in two Positions (Hand drawn by link teacher)

It is clear that for small dB differences, there is huge uncertainty in possible position. A third receiver would allow 3 loci to be drawn and position fixed more reliably by intersection of the lines.

It was decided that, for this project, that using data from two or more receivers like this was impractical, for these reasons:

1. The resources are available for only one receiver to be constructed for the trial
2. The calculations involved, and software to display the results usefully, are complex
3. It is unlikely to be as accurate as having the receiver on a UAV that could be flown in the direction of increasing signal strength; a much simpler solution. Having a directional receiver could aid the hunt for the source.

5.8.2 Using information other than received power

Wi Fi signals include accurate timestamp information. If this could be decoded, and the time of transmission and time of reception accurately calculated, then the distance from the source is merely **time-of-flight x speed of light.** This gives a circular locus of possible positions. A measured distance relative to another receiver location allows two intersecting circles to be drawn. A third narrows the position further.

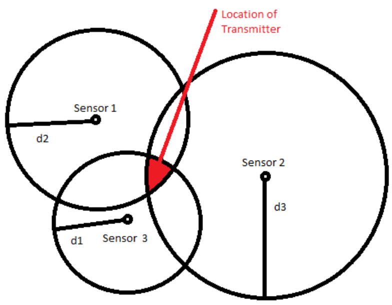


Figure 20 Locating Position by Triangulation using Time of Flight

Alternatively, the difference in time of arrival of the same packet of information (if it could be clearly identified) at several receivers could be used to similarly triangulate position. Both methods require all the receivers to have extremely accurately synchronised clocks, as the time intervals are of the order of 10µs (for a 3km distance). Time delays in hardware can cause errors.

A review of possible methods is contained in:

https://web.wpi.edu/Pubs/E-project/.../NRL_MQP_Final_Report.

The technical challenges of these methods are such that it was decided not to pursue them further for this prototype and demonstration.

5.8.3 Chosen solution for demonstration

The system will have one detector, mounted on a UAV.   Data on power level received is linked to the position of the UAV. If this information can be displayed in real-time in a way that gives the pilot feedback as he flies the UAV around (and points it in different directions if the antenna is directional and heading data is included) then he can try to move towards the source of the signal and gradually "home in" on its position.

5.9 Plan for Practical Demonstration Test

The system will be trialled in as realistic a situation as possible, outdoors, as far away as practical from other RF sources. There will be a "hidden" source of signals (details to be determined) and the processor with HackRF1 receiver, that the team has programmed, will hopefully be deployed on a UAV. The first test will be whether the signal is detected at all, above background. If it is, the system will send data regarding signal strength received and UAV position to a PC on the ground. The team's software on this PC should give a suitable display of this information, updated in real-time, that enables the UAV pilot to "hunt" for the signal source and give some indication of its position.

GDUK have selected a suitable drone (see appendix 8), designed a case to rehouse the HackRF1 so that it is better protected and RF screened, (see appendix 9) and designed mounting brackets to match our specifications (see appendix 10). The team looks forward to integrating their devices onto the GDUK platform and hopes to report on the trial at the Awards Ceremony.

6 Conclusion

To satisfy the brief the device had to be able to detect radio signals coming from mobile devices, from a quickly deployable platform. Team members have learnt to successfully programme a Raspberry Pi3 processor so that using the HackRF One SDR the system can gauge the average background RF noise in an area and then pick out anomalous signals. The system associates the power level received with the GPS coordinates of the system at that time. The source of the anomalous signal could then be tracked down by moving in the direction that the received signal gets stronger. The package devised, with the SDR, the processor in a protective case and all required connectors, could be mounted to numerous types of UAV such as a quadcopter, fixed-wing aircraft or a helikite giving it good deployability and mobility.

7. Evaluation

7.1 Conduct of project

Throughout the project, the team worked well with each other. In the ideas stage of the project there was a lot of thorough research and discussion carried out, so that they had a good range of options open to them. Once the final brief was agreed, and a suitable SDR and processor selected and procured, the main challenge was programming the radio, because it was difficult to get it to pick out the right frequency, remove back ground noise and measure its power level. These problems were overcome by seeking advice from GD UK professional engineers. Whilst some team members worked on software, others worked on a case for the processor and the required connectivity; time was used wisely, with the team working to self-imposed deadlines at each stage of the project and with roles allocated to suit each team member's expertise and interests.

7.2 Quality of the solution

The problem being addressed is a real one and it is felt that using RF signals that a person emits offers rescue services a powerful tool beyond just searching visually.

The Hack RF1 was a good choice of SDR as it covers all the required frequency ranges, successfully detects weak signals and is much cheaper than the alternatives. The Pi3 proved to have sufficient processing power, whilst being light and compact. The casings of the HackRF1 (already supplied) and of the Pi3 (designed and manufactured by us) were sufficiently robust and light for the system to be flight tested in prototype form. Sacrificial connectors protect the vulnerable and hard to replace on-board connectors, and are worth the small extra weight.

In the laboratory, the software successfully processes data from the SDR and GPS receiver to give an output on the base computer. It is hoped that the effectiveness of the system, and the usefulness of this output in assisting location of a person in distress, (who is transmitting RF signals) will be demonstrated in the March trial, with GDUK engineers present.

The system, as it stands, requires a power supply and air-to-ground communications to be provided separately. As both of these are already present on a UAV, it is good not to duplicate those parts of the system. Screening, extra mounting brackets, leads and antennae will also have to be supplied.
Having two cases gives flexibility with mounting, but is heavier than having everything integrated in one case. The various leads of different lengths are untidy, and maybe longer than needed. All of these aspects would be improved in a production version.

7.3 Cost Effectiveness

The cost of the prototype was:

HackRF1		£290
Raspberry Pi3		£ 55
GPS "HAT"		£ 35
Connectors		£ 20
Case for Pi3	(materials supplied free from stock)	
TOTAL for Materials		**£400**

The team feels that this is a cost effective part of the overall system, in which the major expense is likely to be the UAV (with the required carrying capacity and range) and its associated control and communications equipment. £400 is also not a large cost compared to that of the base PC.

7.4 Social and Environmental Benefits

This system should help save lives and resources when searching for people in distress in areas where they cannot contact emergency services direct by mobile phone. The need for extensive search parties that endanger life, and might have a damaging effect on a fragile wild environment, will be reduced. Without the availability of a system like this it might be decided to extend the area of mobile phone network coverage further into wild hilly and coastal areas, and into areas currently in the "shadow" of cliffs and hills from current masts, which would involve the construction of new masts and other infrastructure in remote places, damaging the environment.

The device itself can be reused many times, uses few raw materials, and these can be recycled at the end of its life.

8. Further Developments and Recommendations

As explained in section 5.9, GDUK are investing in a UAV and constructing an installation kit to integrate the team's prototype onto their platform for testing. They are considering commercial development of systems similar to RADIAL.

A production model would be made smaller and lighter, probably integrating the radio and processor into one unit as far as possible without compromising RF screening for the receiver. A lighter and sturdier material such as resin or thin aluminium would be used for the case. Sacrificial connectors would be mounted flush with the outer case.

Despite the rescue and armed services already having systems which have similar functionality to that of RADIAL, we believe it has excellent potential for commercial development for the following reasons:

- RADIAL is a lot cheaper to produce and maintain than the pre-existing alternatives.
- RADIAL is considerably lighter than its competitors.
- RADIAL requires a less expensive aircraft in order to be able to operate.

The system could be extended to have more than one airborne receiver, with more sophisticated software on the base computer that uses this extra data to narrow down the position of the source more quickly. Mounting a receiver on a lighter-than-air tethered balloon, or steerable "helikite" or "kytoon" would give longer time in the air than a conventional quadcopter or fixed wing UAV, though it might take longer to deploy.

Devices on permanently tethered balloons could give permanent coverage to detect the presence of people emitting radio signals, even if they were not trying to call for help. This could be used for security purposes, monitoring a border or property boundary, as well as for public safety.

RADIAL could also provide a service in locating trapped or stranded civilians following natural disasters.

The device could be used to monitor whether radio beacons in remote locations were working as intended.

9. Acknowledgements

We would like to thank everyone who has helped complete our project. This includes:

General Dynamics, for supplying us with materials and helping to develop our ideas.
Cardiff Metropolitan University, for the use of their facilities and technical help.
EESW, for creating this competition and giving us a chance to experience working together and creating our project.
Mr Brown for his support, inspiration and guidance throughout the course of the project.

10. References

Radio frequencies: regulations and internet of things

https://www.ofcom.org.uk/manage-your-licence/radiocommunication-licences/internet-of-things

https://www.ofcom.org.uk/spectrum/radio-spectrum-and-the-law/licence-exempt-radio-use/licence-exempt-devices/ofw311

SDR

https://en.wikipedia.org/wiki/List_of_software-defined_radios

https://greatscottgadgets.com/hackrf/

https://www.sparkfun.com/products/1300

Locating position of an RF source

https://en.wikipedia.org/wiki/Wi-Fi_positioning_system#Problem_statement_and_basic_concepts

https://en.wikipedia.org/wiki/Beacon_frame

https://web.wpi.edu/Pubs/E-project/.../NRL_MQP_Final_Report

RNLI Rescue Support Vehicle

https://www.linkedin.com/pulse/rnli-launch-first-rescue-support-vehicle-rob-inett-1?forceNoSplash=true

Kytoon

https://en.wikipedia.org/wiki/Kytoon

11. Appendices

Appendix 1 The Final Brief

Every day hundreds of people around the world find themselves in difficulties in isolated locations and unable to call for assistance. These people are sometimes victims of accidents or natural disasters but often they are partly responsible for their predicament through lack of preparedness and excessive risk-taking. Many of these individuals are carrying with them electronic communications devices that they cannot use because they are out of range of the necessary communications infrastructure, either because it sparsely deployed or severely disrupted.
Emergency services and public safety organisations are constantly looking for innovative systems that can help them in their mission to help people in isolated locations both before and after they find themselves in trouble. This project is to evaluate options for a rapidly deployable system to detect and locate the radio signals transmitted by modern personal electronic devices e.g. mobile phones, tablets. A prototype of the preferred option shall be designed and implemented, taking into account all relevant legislation for the use of such a system, in particular safety and radio transmissions. The prototype shall be trialled on location with GDUK in March 2017.

Appendix 2 – top_block.py

```python
"""This file is produced by GNU radio from the flow chart found in the software
section of report
 """
#!/usr/bin/env python2
##################################################
# GNU Radio Python Flow Graph
# Title: Top Block
# Generated: Tue Dec 13 05:36:56 2016
##################################################

from gnuradio import blocks
from gnuradio import eng_notation
from gnuradio import filter
from gnuradio import gr
from gnuradio.eng_option import eng_option
from gnuradio.fft import logpwrfft
from gnuradio.filter import firdes
from optparse import OptionParser
import osmosdr
import time

prefix = "/home/pi"

class top_block(gr.top_block):

    def __init__(self):
        gr.top_block.__init__(self, "Top Block")

        ##################################################
        # Variables
        ##################################################
        self.samp_rate = samp_rate = 32000

        ##################################################
        # Blocks
        ##################################################
        self.osmosdr_source_0 = osmosdr.source( args="numchan=" + str(1) + " " + "" )
        self.osmosdr_source_0.set_sample_rate(samp_rate)
        self.osmosdr_source_0.set_center_freq(1e9, 0)
        self.osmosdr_source_0.set_freq_corr(0, 0)
        self.osmosdr_source_0.set_dc_offset_mode(0, 0)
        self.osmosdr_source_0.set_iq_balance_mode(0, 0)
        self.osmosdr_source_0.set_gain_mode(False, 0)
        self.osmosdr_source_0.set_gain(10, 0)
        self.osmosdr_source_0.set_if_gain(20, 0)
        self.osmosdr_source_0.set_bb_gain(20, 0)
        self.osmosdr_source_0.set_antenna("", 0)
        self.osmosdr_source_0.set_bandwidth(0, 0)

        self.logpwrfft_x_0_1 = logpwrfft.logpwrfft_f(
            sample_rate=samp_rate,
            fft_size=1024,
            ref_scale=2,
            frame_rate=30,
            avg_alpha=1.0,
            average=False,
        )
        self.logpwrfft_x_0_0 = logpwrfft.logpwrfft_f(
            sample_rate=samp_rate,
            fft_size=1024,
            ref_scale=2,
            frame_rate=30,
            avg_alpha=1.0,
            average=False,
        )
        self.logpwrfft_x_0 = logpwrfft.logpwrfft_f(
            sample_rate=samp_rate,
            fft_size=1024,
            ref_scale=2,
            frame_rate=30,
            avg_alpha=1.0,
            average=False,
        )
        self.blocks_float_to_short_0_1 = blocks.float_to_short(1024, 1)
        self.blocks_float_to_short_0_0 = blocks.float_to_short(1024, 1)
        self.blocks_float_to_short_0 = blocks.float_to_short(1024, 1)
        self.blocks_file_sink_1_0 = blocks.file_sink(gr.sizeof_short*1024, prefix+"/Power_detection_bands/cell2.raw", False)
        self.blocks_file_sink_1_0.set_unbuffered(False)
        self.blocks_file_sink_1 = blocks.file_sink(gr.sizeof_short*1024, prefix+"/Power_detection_bands/cell1.raw", False)
        self.blocks_file_sink_1.set_unbuffered(False)
        self.blocks_file_sink_0 = blocks.file_sink(gr.sizeof_short*1024, prefix+"/Power_detection_bands/wifi.raw", False)
        self.blocks_file_sink_0.set_unbuffered(False)
        self.blocks_complex_to_float_0 = blocks.complex_to_float(1)
        self.band_pass_filter_2 = filter.fir_filter_fff(1, firdes.band_pass(
            1, 100e6, 20e6, 40e6, 1e6, firdes.WIN_HAMMING, 6.76))
        self.band_pass_filter_1 = filter.fir_filter_fff(1, firdes.band_pass(
            1, 1800e6, 896e6, 900e6, 1e6, firdes.WIN_HAMMING, 6.76))
        self.band_pass_filter_0 = filter.fir_filter_fff(1, firdes.band_pass(
            1, 4000e6, 1850e6, 1990e6, 1e6, firdes.WIN_HAMMING, 6.76))

        ##################################################
        # Connections
        ##################################################
        self.connect((self.band_pass_filter_0, 0), (self.logpwrfft_x_0_0, 0))
        self.connect((self.band_pass_filter_1, 0), (self.logpwrfft_x_0_1, 0))
        self.connect((self.band_pass_filter_2, 0), (self.logpwrfft_x_0, 0))
        self.connect((self.blocks_complex_to_float_0, 0), (self.band_pass_filter_0, 0))
        self.connect((self.blocks_complex_to_float_0, 0), (self.band_pass_filter_1, 0))
        self.connect((self.blocks_complex_to_float_0, 0), (self.band_pass_filter_2, 0))
        self.connect((self.blocks_float_to_short_0, 0), (self.blocks_file_sink_0, 0))
        self.connect((self.blocks_float_to_short_0_0, 0), (self.blocks_file_sink_1_0, 0))
        self.connect((self.blocks_float_to_short_0_1, 0), (self.blocks_file_sink_1, 0))
        self.connect((self.logpwrfft_x_0, 0), (self.blocks_float_to_short_0, 0))
        self.connect((self.logpwrfft_x_0_0, 0), (self.blocks_float_to_short_0_0, 0))
        self.connect((self.logpwrfft_x_0_1, 0), (self.blocks_float_to_short_0_1, 0))
        self.connect((self.osmosdr_source_0, 0), (self.blocks_complex_to_float_0, 0))


    def get_samp_rate(self):
        return self.samp_rate

    def set_samp_rate(self, samp_rate):
        self.samp_rate = samp_rate
        self.logpwrfft_x_0.set_sample_rate(self.samp_rate)
        self.logpwrfft_x_0_0.set_sample_rate(self.samp_rate)
        self.logpwrfft_x_0_1.set_sample_rate(self.samp_rate)
        self.osmosdr_source_0.set_sample_rate(self.samp_rate)


if __name__ == '__main__':
    parser = OptionParser(option_class=eng_option, usage="%prog: [options]")
    (options, args) = parser.parse_args()
    tb = top_block()
    tb.start()
    try:
        raw_input('Press Enter to quit: ')
    except EOFError:
        pass
    tb.stop()
    tb.wait()
```

Appendix 3 – simple_process.py

```
"""Version 2.1
GPS issue fixed, read file was truncated after new line written casuing
	0 to be returned as next value, this was GPS related as GPS slowed down
	script allowing the new line to be written before truncation

This version handels all power data as intergers and so is in the process of
upgrading to improve the precision of results by using floating point numbers
This is the latest working version at present
"""

import struct
import time
import gps

def follow(file):
	"""Constantly follows file reading latest/newest lines"""
	file.seek(0,2)
	while True:
		line = file.readline()
		if not line:
			time.sleep(0.1)
			continue
		yield line

def prepend_file(filename,line):
	"""Adds new comment to end of file (rarther than start)"""
	with open(filename, "r+") as fil:
		content=fil.read()
		fil.seek(0,0)
		fil.write(line.rstrip("\r\n")+"\n"+content)

def truncate_file(filename, trunc_pos):
	"""Quick call for truncation of file"""
	with open(filename, "r+") as fil:
		fil.truncate(trunc_pos)

read_file = open("/home/pi/Power_detection_bands/wifi.raw", "r+")

truncate_len =[0,0,0,0,0,0,0,0,0,0]
truncate_counter = 0

read_lines = follow(read_file)

for line in read_lines:
	"""Handles each line of the raw file

	After each line is read (it will be latest line due to follow)
	the file is truncated

	The power is them decoded, timestamp and gps location produced
	and results added to txt file

	txt file is truncated at the length of the last 10 resuts to
	prevent the file from becoming too large
	"""
	power = struct.unpack_from('h', line)
	power = str(power[0])
	time_read = str(time.time())

	gps_loc = gps.loc()
	write_data = power+ ", " + time_read+", "+gps_loc

	prepend_file("/home/pi/Power_detection_bands/wifi.txt", write_data)

	print write_data

	if truncate_counter<=9:
		truncate_len[truncate_counter]=len(write_data)+1
		truncate_counter+=1
	else:
		truncate_counter=0
		truncate_len[truncate_counter]=len(write_data)+1

	truncate_pos = sum(truncate_len)
	truncate_file("/home/pi/Power_detection_bands/wifi.txt", truncate_pos)
```

Appendix 4 – start_script.py

```
"""Starts scripts on pi, and calls download/processing file simultaniously"""
from multiprocessing import Process
import os
import time

print("\n\nStarting RADIAL\n\nThis may take a few seconds\n\n")

air_ip = ip = open("radial_air.ip","r").readline()[:-1]

os.system('ssh pi@'+air_ip+' sudo killall python')
os.system('ssh pi@'+air_ip+' sudo /etc/init.d/ssh restart')

time.sleep(5)

def start_topblock():
        """Starts top_block.py (GNU radio file) on pi"""
        os.system("ssh pi@"+air_ip+" 'python /home/pi/Power_detection_bands/top_block.py'")

def start_process():
        """Starts simple_process.py (data processing file) on pi"""
        os.system("ssh pi@"+air_ip+" 'python /home/pi/Power_detection_bands/simple_process.py'")

def start_download():
        """Starts download script on ground"""
        import download_file

if __name__=="__main__":
        """Starts processes"""
        p1 = Process(target=start_topblock)
        p1.start()
        p2 = Process(target=start_process)
        p2.start()
        time.sleep(20)
        p3 = Process(target=start_download)
        p3.start()
```

Appendix 5 – download_file.py

```
"""This version handels all power data as intergers and so is being upgraded to use floating point
numbers, improving precision. This is still the latest working version.

This file:
    Downloads data from RADIAL AIR

    Splits the string recieved - into power, time and location

    Checks that the time isn't the same as previous strings (string is new)

    Checks that the power is not 0, if so then there is an error on the Pi as no data is being
    collected

    A rolling average is collected from the first 5 results

    Otherwise the data is checked to see if it is more than 5% different from the rolling average
        If it is then the data is flagged as an anomoly
        Otherwise it is added to the list as part of the rolling average
"""

import os
import time as t

local_root = open("local.root","r").readline()[:-1]
air_ip = ip = open("radial_air.ip","r").readline()[:-1]


def prepend_file(filename, line):
    """Allows data to be prepended to file (rarther than appended)"""
    with open(filename, "r+") as fil:
        content = fil.read()
        fil.seek(0, 0)
        fil.write(line.rstrip("\r\n")+"\n"+content)

local = local_root+"data_dump/wifi.txt"
average = [0, 0, 0, 0, 0]
list_counter = 4
test_counter = 0
last_time = int(0)
log_file = local_root+"log.txt"

while True:
    os.system("scp pi@"+air_ip+":/home/pi/Power_detection_bands/wifi.txt "+local_root+"data_dump")
    with open(local) as f:
        try:
            content= f.readlines()
            content.reverse()
            for i in content:
                list=i.split(",")
                if len(list)==3:
                    power=int(list[0])
                    time=list[1]
                    time=int(time.split(".")[0])
                    location=list[2]
                    if time>last_time:
                        last_time=time
                        time = str(time)
                        disp_line = str(power)+","+time+","+location
                        #prepend_file(log_file,disp_line)
                        if power == 0:
                            print("Error on pi, restart may be required")
                        elif test_counter <= 4 and power!=0:
                            print("List conter = ", list_counter)
                            average[list_counter] = power
                            list_counter += 1
                            if list_counter >= 4:
                                list_counter = 0
                            test_counter += 1
                            print disp_line

                        else:
                            average_real = (average[0]+average[1]+average[2]+average[3]+average[4])/5
                            disp_line = disp_line[:-1]
                            disp_line += ',' + str(average_real)
                            if power <= 1.05*average_real or power >= 0.95*average_real:
                                print 'Anomalous Data'
                                print disp_line
                                prepend_file(log_file, disp_line)
                            else:
                                average[list_counter] = power
                                list_counter += 1
                                if list_counter >= 5:
                                    list_counter = 0
                                print disp_line
                #Else data is not needed

        except Exception, error:
            print error
            print("An error occured, continuing anyway")

    t.sleep(1)
```

Appendix 6 – Wi Fi.txt sample

```
-158, 1485103038.13, 5145.60545 N 00248.45228 W
-158, 1485103037.12, 5145.60548 N 00248.45226 W
-158, 1485103036.13, 5145.60553 N 00248.45224 W
-158, 1485103035.12, 5145.60556 N 00248.45226 W
-158, 1485103034.13, 5145.60558 N 00248.45229 W
-158, 1485103033.12, 5145.60563 N 00248.45221 W
-158, 1485103032.13, 5145.60566 N 00248.45213 W
-158, 1485103031.11, 5145.60572 N 00248.45199 W
-158, 1485103030.13, 5145.60577 N 00248.45184 W
-158, 1485103029.12, 5145.60582 N 00248.45174 W
```

Appendix 7 – gps.py

```python
"""This version uses the GNGGA String passed from GPS but will be updated to read GNRMC
strings as they include heading essential for position location
"""

import serial

def process_gngga(str):
        """Reads GNGGA String and splits to elements, assigns each element a name in
        dictionary and returns dictionary
        """

        list = str.split(",")

        """list format - id, time, lat, NS, long, EW, quality, num sats,
        horizontal dilution of precision, alt, units alt, Geoid separation, units sep,
        Age of differential corrections, ID of station providing differential corrections, check sum
        """

        list[14:15]=list[14].split("*")
        dict = {
                "time":list[1],
                "lat":list[2],
                "NS":list[3],
                "long":list[4],
                "EW":list[5],
                "quality":list[6],
                "numSV":list[7],
                "HDOP":list[8],
                "alt":list[9],
                "uAlt":list[10],
                "sep":list[11],
                "uSep":list[12],
                "diffAge":list[13],
                "diffStation":list[14],
                "cs":list[15]
        }
        return dict

def loc():
        """Reads NMEA strings from serial port, checks to see if it is a GNGGA string and passes
        string to process_gngga

        Otherwise sets gps message to be error message

        Turns dictionary to string in this instance and returned it
        """

        serialPort = serial.Serial("/dev/ttyAMA0", 9600, timeout=0.5)
        gps_success = 0

        while gps_success != 1:
                gps_str = serialPort.readline()
                if gps_str.find('GGA')>0:
                        gps_msg = process_gngga(gps_str)
                        gps_success = 1
                else:
                        gps_success = 0

        gps_loc = str(gps_msg['lat']+" "+gps_msg['NS']+" "+gps_msg['long']+" "+gps_msg['EW']+" "+gps_msg['alt']+
                " "+gps_msg['uAlt'])

        return gps_loc
```

Appendix 8 UAV to be used by GDUK as aerial platform

Specification for GDU Byrd Advanced 1.0

Foldable

Large-capacity battery (14.8V 6700mAh 99.4Wh)

Maximum take-off weight of 3.8kg

Maximum load weight of 2.0kg

Delivery Box

Modular design

Support infrared camera

Support Compact camera

Support DSLR camera

Support optical zoom camera

Level 5 wind resistance

Materials tested for high temperatures (PC chassis with plastic)

Resistance at 40 ℃ temperature and humidity of 95%

Salt spray test temperature (35 ± 2 ℃, mass concentration of salt solution percentage formula 5 ± 0.1%, PH value of 6.5 to 7.2 within)

Double protection design for propellers performance

Multi-screen sharing support

Dual remote control

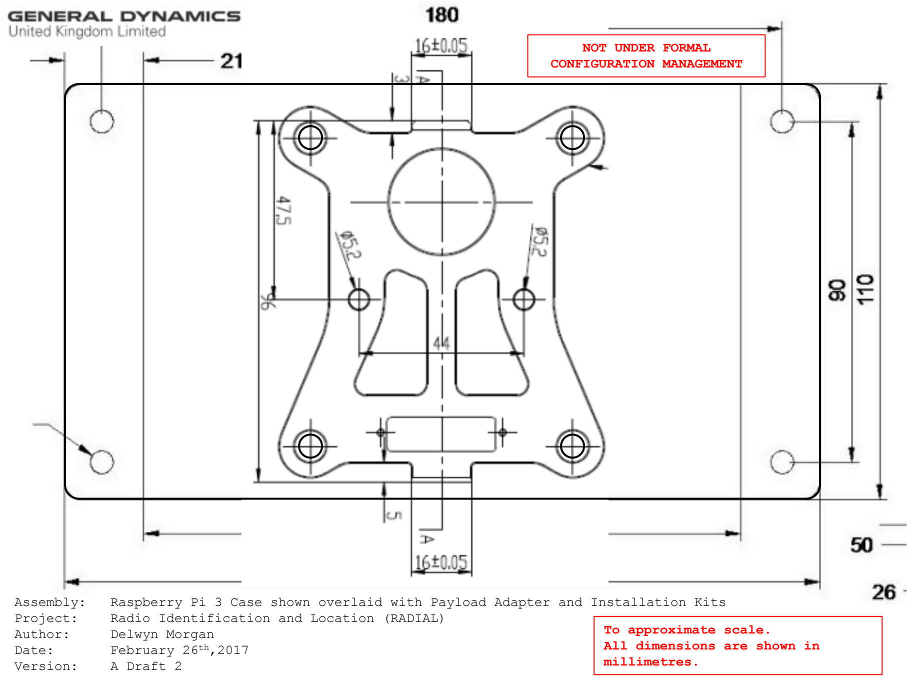
Resistant materials tested for maximum endurance

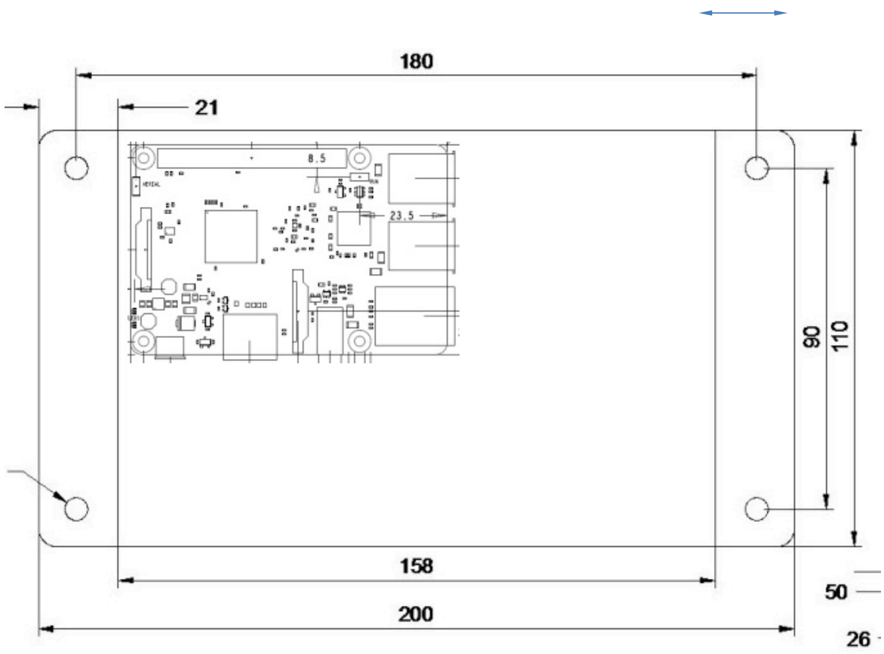
AIRCRAFT

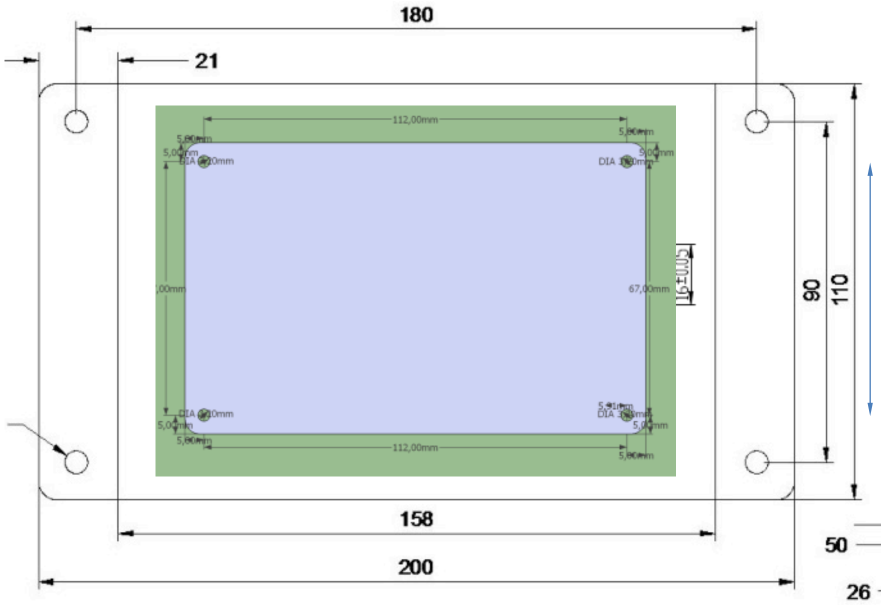
Dimension (folded)	**273 × 223 × 107 mm**
Diagonal Size (unfolded)	**622 mm**
Weight (Including battery and propellers)	**1890 g**
Battery	**14.2V-6700mAh-99.4Wh-LIPO 4S**
Charger	**Support controller and battery, two-in-one charger**
Flight time	**29 mins**
Maximum speed	**15 m/s**

Appendix 9 Case for Hack1 RF to be supplied by GDUK

Appendix 10 Sample drawings from GDUK for RADIAL installation kit




180
21
8.5
23.5
90
110
158
50
200
26


180
21
112,00mm
DIA 3,00mm
67,00mm
90
110
158
50
200
26