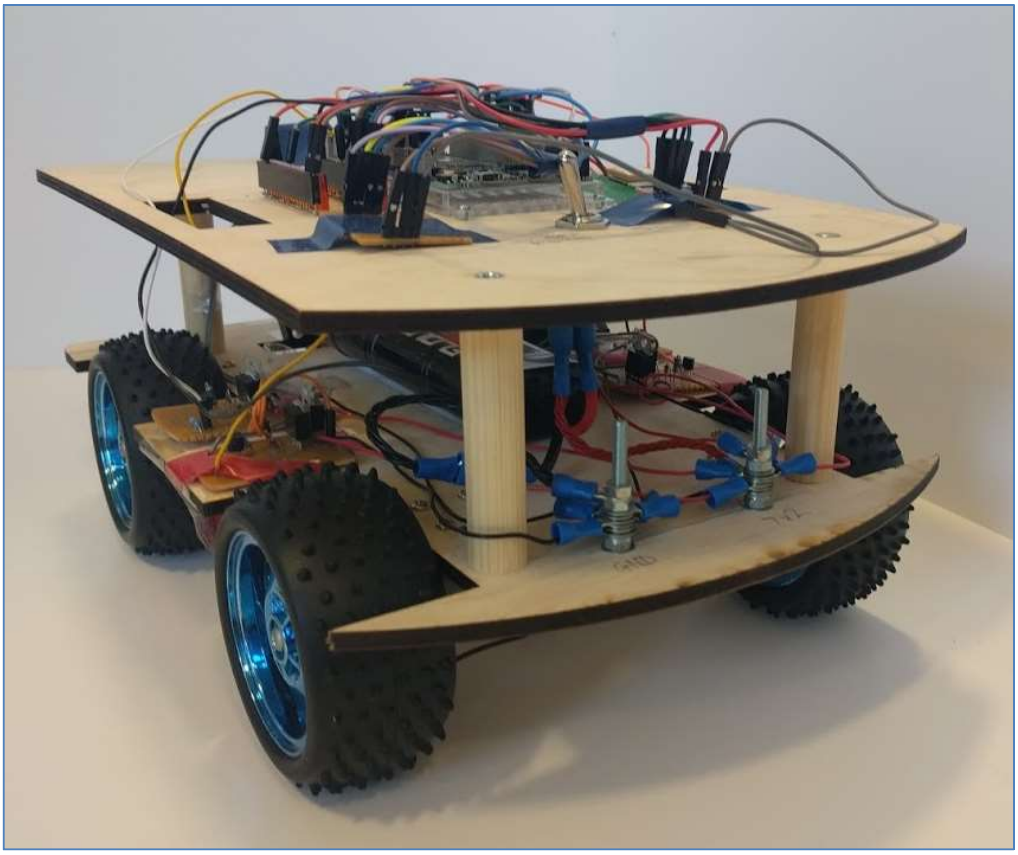
SID

The Safe IED Detector



Watford Grammar School for Boys

Table of Contents

1 Abstract

This report summarises and explains a seven-month long project which ran from October 2016 to April 2017. The project was undertaken by six students from Watford Grammar School for Boys in partnership with EDT and Leonardo.

We begin by introducing the project and the brief, detailing our preliminary interpretation and analysis. We proceed to summarise the findings of the research component of the project, and identify the key requirements we have defined for our solution.

We then include description and justification for the design we developed, followed by an account of the construction and testing of the solution. We conclude with an evaluation of our solution and some of our personal reflections on the project. Also included, as appendices, are our initial project brief, our Gantt chart, 3D CAD drawings, and photographs of the final product.

2 Acknowledgements

We would like to begin by deeply thanking the all the people who made this project possible.

First and foremost to our teacher, Dr Buckley, who held us accountable for our progress and whose past experience in running the scheme ensured we were always on the right track. Dr Buckley has been a truly invaluable source of support and advice, and without his dedication we would never have had the opportunity to take part in this project.

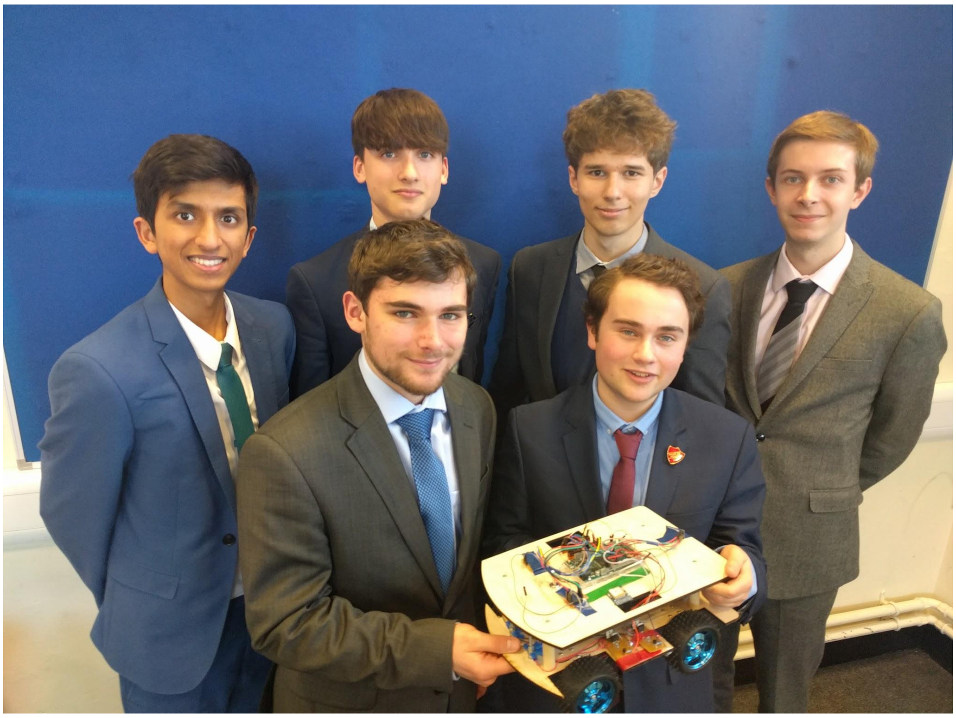
Secondly to our mentors Mark, Harry, and Rob, whose guidance helped us develop the project professionally and efficiently. They have provided a wealth of expertise and assistance, and throughout the project they have been exceptionally friendly, funny, and caring.

We would like to extend our gratitude to Leonardo for generously sponsoring our participation in the scheme and offering us the valuable time of the mentors. Leonardo warmly hosted us for a site visit and tour of their facilities, and we are very grateful for the valuable insight they gave us into the engineering profession.

We would also like to thank Cambridge University's Department of Engineering and St Catharine's College for the hospitality, resources and advice offered to us at their workshop in December. It proved, thanks to the labour of Cambridge staff behind the scenes, to be a highly productive and entertaining two days.

Finally, we wish to thank EDT for the hard work that has gone into making the EES programme. It has provided the six of us, and many hundreds of others, with an amazing experience of the world of engineering and the opportunity for a tremendously enjoyable team project. We truly appreciate all the work they do in inspiring our generation to pursue a career in engineering.

3 Team Profiles



Srivatsa, Ralph, Jeremy, Luke
George, Ethan

3.1 Srivatsa Garg

I have always enjoyed maths and problem-solving, and the distinctive opportunity for a creative yet methodical approach to problem-solving offered by life as an engineer draws me to an engineering degree at university. I currently study Maths, Further Maths, Chemistry, and Physics, and am thoroughly enjoying my A-level courses. I applied to take part in the Young Engineers programme in order to gain greater clarity into the path I wish to pursue in the future, as well as to gain the valuable experience of an electronics and robotics project. In my free time, I enjoy running, cycling, and playing badminton, and have a keen interest in history and politics.

3.2 Ralph Curwen

I have chosen to take part in the Young Engineers scheme as a career in engineering is something that interests me greatly and I very much enjoy working on projects in a team. The A Levels I am taking are Maths, Further Maths, Physics and Economics which are all subjects that I am passionate about. Furthermore, they are subjects that would give me a solid platform to study engineering in the future. In my spare time, I represent both club and school teams at cricket and hockey, whilst also playing in multiple school bands.

3.3 Jeremy Zolnai-Lucas

As a child I wanted to be ‘an inventor’. As my life progressed this motivation has developed into a passion for all things engineering, and a love for problem-solving. I study Maths, Further Maths, Physics and Computing, and in my spare time I enjoy creating electronics projects, programming and playing music. I applied for Young Engineers as it seemed like the ideal way to move forward into an engineering career, as well as being an invaluable experience. I hope to study engineering at university, and plan to work in engineering – robotics in particular.

3.4 Luke Ashford

I have been passionate about engineering as a degree and a career for a while: finding solutions to real problems is something I am keen to engage in. I applied for the Young Engineers programme in order to gain greater experience in engineering, and to decide whether I wished to truly dedicate my life to such a career. At school I study Maths, Further Maths, Physics and History at A Level; I hope to pursue Maths, Physics or Electrical Engineering as a degree. I am a keen follower of news, politics and current affairs; and I also enjoy reading fantasy and literary fiction. I also play badminton, tennis, and enjoy cycling.

3.5 George Wright

I am interested in a studying computer science or robotics and going into a career in one of these fields. Knowing we would be doing an electronics project with Leonardo, I decided to take part in the Young Engineers scheme to gain more of an understanding in robotics, and as I felt I could contribute to the team because I enjoy working in a group. I am taking Maths, Further Maths, Physics and Computer Science for my A Level subjects. In my spare time I attend Explorer Scouts, I play rugby for the school team, and thoroughly enjoy programming and making games.

3.6 Ethan Honey

I have always been interested in science and maths, and ever since taking part in the Go4Set project in Year 9, I have wanted to pursue a career in engineering, a discipline which unites both maths and physics. In particular, I find creating real world solutions to hidden problems the most intriguing part of engineering; this is one of the main reasons I applied for Young Engineers. I study Maths, Further Maths, Physics and Latin at A-Level, and hope to study engineering at university. Outside of school I enjoy playing piano and trumpet, debating and sailing. I also hold an active leadership role in my community.

4 Introduction

Our project is a counter-IED system in which our Raspberry Pi-based prototype detects mobile phone signals and uses these to determine the location of the device. We named it the Safe IED Detector (SID) as it allows for unmanned - therefore safe - detection of bomb threats. The project has been carried out in partnership with Leonardo, a global aerospace, defence, and security company. Leonardo provided us with a number of project choices and we decided to pick this one because it appeared to be the most interesting whilst also seeming highly relevant to the modern world. We also decided that this seemed to be the most challenging project that we felt capable of carrying out. A copy of the project brief may be found in Appendix A.

The primary applications we envision for the SID are related to counter-IED and counter-insurgency. Mobile phone-triggered IEDs are a modern bomb threat used increasingly by terrorist organisations for their simplicity and low cost. Such devices have been used in South-East Asia[1] to devastating effect, and are also a popular choice for insurgents on the front line, threatening soldiers and civilians alike. When the phone is called, a relay system connected to a blasting cap is triggered, activating an explosive device. Mobile phones are ideal for adapting into bombs because their power is optimal for the relay system to function and it is easy to modify their circuits. Clearly, being able to locate phones offers valuable capability for the protection of military and civilian life.

On the front line in counter-insurgency operations, determining the location of the enemy is also vital. The ability to detect phone or radio signals can therefore be of tremendous strategic value, as rival fighters will almost always carry communication devices. The SID is thus multipurpose; it also helps protect soldiers from the threat of nearby adversaries by pinpointing their location.

Our project was supported by fortnightly meetings with two engineers from Leonardo, who gave us invaluable advice and guidance. We ourselves met each week, where we collated work, shared ideas, and allocated tasks for the week ahead. The project was completed over a seven-month period, including a two-day workshop in Cambridge University's Department of Engineering in December. Our total budget for the project was £200, half each from Leonardo and school.

[1] http://www.oss.net/dynamaster/file_archive/051011/04e118e543b29b06a494b03a0e192cc8/OSS%20Remote%20IED%20Initiation%20Updated%20Final.pdf

5 Initial Project Analysis

Having received the project brief on the Launch Day, we brainstormed our first thoughts about how we would tackle the project, and made a number of key decisions about the project.

From the project brief we knew that we would be investigating particular scenarios which involved detecting IEDs that would be detonated using a mobile phone. Therefore our first thoughts were about how we intended to detect the mobile phone. From the possible methods of detection[2], which were radio frequency or ferromagnetic, we decided that radio frequency was the most achievable method. It offers long-range detection whereas ferromagnetic would need to be short-range, and it is therefore more useful in a real-world scenario

Additionally, we quickly decided that our solution must be a mobile unit - either handheld or robot-mounted – because its applications in a warzone demand manoeuvrability. In light of the dangerous nature of the situations in which this system is to be used, we concluded that a robot-mounted device would be most appropriate.

One requirement given to us by Leonardo was that our system had to use a Raspberry Pi. Having decided to produce a robot to detect the mobile phones, we elected to use the Raspberry Pi as a controlling base-station rather than on the robot itself. We made this decision as we wanted to use the Raspberry Pi's computing power to drive the software used by the operator whereas a less capable device would suffice for the robot itself.

Having made these initial decisions, we began to conduct detailed research.

[2] http://www.pnl.gov/main/publications/external/technical_reports/PNNL-17734.pdf
https://www.bvsystems.com/what-locates-cell-phones-better-rf-or-ferromagnetic-detectors/

6 Research

6.1 Existing Technology

6.1.1 Unmanned Systems

6.1.1.1 Ground Vehicles

Unmanned Ground Vehicles (UGVs) are vehicles with multi-sensor systems that can – under supervision – locate the rough position of an IED without relying on GPS or another vehicle. For example, Leonardo has produced a UGV called TRP2 HD[3] which is a remotely controlled vehicle that has been manufactured to withstand hostile environments and obstacles. It can travel with speeds up to 10 km/h whilst having the adaptability to perform a wide variety of missions.

6.1.1.2 Aerostats

Aerostats are large balloons that can hover above a certain area for long periods of time and use infrared and radar sensors to provide high definition images of the surrounding area[4]. The imagery that the aerostats relay can be constantly manned externally, so if an IED is planted in the area under surveillance it can be easily tracked and disarmed. Aerostats can also detect weapon fire and blasts using sensors on the ground which are directly linked with it. This technology is part of the counter-IED and counter-insurgent arsenal of modern militaries. A notable example is the Persistent Threat Detection System (PTDS), an Aerostat which has been deployed by the US army in Afghanistan and Iraq. The PTDS can stay in the air for up to 20 days at a time, and features numerous types of sensors, as well as a multispectral imaging and targeting system[5].

6.1.2 Electronic Countermeasures and Jammers

A number of vehicle-mounted jammer systems are on the market, such as the Duke V3, CREW Vehicle Receiver Jammer and the Vehicle Jammer System STAR. These are all automated electronic models that use radio frequencies to jam signals which might result in the detonation of IEDs.

The Thor III dismounted CREW system consists of 3 subsystems. Each subsystem contains a remote control unit, pack frame, antenna, GPS antenna, cables, and software. The Thor III is designed specifically so that it can be worn by a soldier and it has multiple jamming systems which enable it to disable multiple threats simultaneously. This is a significant feature in warzones, where the threat of IEDs is constant.

3 http://www.leonardocompany.com/en/-/trp1
4 http://lockheedmartin.com/us/products/lighter-than-air-vehicles.html
5 http://lockheedmartin.com/content/dam/lockheed/data/ms2/documents/PTDS_LTA_Brochure.pdf

Electromagnets are also used to disable sensor triggered IEDs either by setting them off or by preventing their detonation through a radio or mobile system. It can be used to create a zone that prevents IEDs from exploding so that vehicles can travel safely in potentially dangerous areas.

These technologies could be vehicle-mounted, and thus represent possible further functionalities of the SID, though cannot feature in our time-restricted prototype.

6.2 Communication

There are three technologies that could be used for communication between the robot and the base station: Wi-Fi, Bluetooth, and radio-frequency (RF) communications. Wi-Fi can use either an existing network, or the Pi can be configured to act as a wireless access point. There are also two major frequencies for RF communications: 433MHz and 2.4GHz.

Wi-Fi using existing network infrastructure is less suitable for the project than using the Pi as an access point - if this robot is to operate in the field, it must be an entire system that should not depend on the presence of other infrastructure such as a Wi-Fi connection. Even with the Pi as an access point, it has at best a range of around 35m[6]. Factoring for interference or other less than ideal conditions, and this estimate for the range becomes unacceptably low. Bluetooth has similar issues only worse - most estimates at the range of a Bluetooth connection are around 10m[7].

RF communication has much better ranges - many 433MHz transmitters predict ranges of 50m and the nRF24L01 chip from Nordic Semiconductor boasts a 100m range[8]. Ultimately, we decided to use the nRF24L01 modules over 433MHz modules for one main reason: ease of use. Reading online from the experiences of hobbyists gave the impression that 433MHz communication was temperamental at best, and near impossible at worst. Many reported problems of ranges below 1m, which was clearly unacceptable. The nRF24L01, by contrast, has extensive documentation available on the retailer Sparkfun's website, which encouraged us that it would be simpler to configure, and any problems easier to solve, than with the 433MHz modules.

When transmitting data between the Arduino and the Raspberry Pi, or vice versa, we decided we needed to protect the data from corruption which might result from a poor signal. We thought of three potential methods: hash the data payload; transmit every message with along with its bitwise-NOT value; or use an exclusive-or (XOR) bitwise duplication to detect corruption.

To use hash protection, the transmitting device calculates the hash (for instance using the MD5 algorithm) of the payload and transmits this hash alongside the payload. The receiver recalculates the hash of the received payload and checks it against the received hash. If the two match, then the data is not corrupted. If transmitting every message twice with the NOT value, then the receiver checks the two messages add to give, in binary, a string of 1s. Corruption - itself already unlikely - would have to affect the same bit or bits twice in order for this method to fail. The third method, the XOR bitwise duplication,

[6] http://litepoint.com/whitepaper/80211ac_Whitepaper.pdf
[7] http://www.bluair.pl/bluetooth-range
[8] https://www.sparkfun.com/products/691

takes every bit in the payload and replaces all '0's with '01' and all '1's with '10'. If any individual bit in these pairs becomes corrupted then it will be easy to detect. Two neighbouring bits would have to both become corrupted in opposite manners for a corrupted message to not be detected.

We decided to transmit every message with along with its NOT to detect corruption. Hashing is a computationally expensive process that is impractical on the Arduino so is unsuitable, and the XOR method means sending a 64-bit integer, which we are unable to do with the nRF24L01 chip. Furthermore it is easier to program and check using this method over the other ones.

6.3 Locating Phones

The key aspect of our project is the ability to detect mobile phones. Commercial technology to this effect is prohibitively expensive, and often relatively ineffective. Most phone detector devices are little more than metal detectors, as found at airports and event venues. To reliably detect phones from a distance, one must be able to detect transmissions the phone makes.

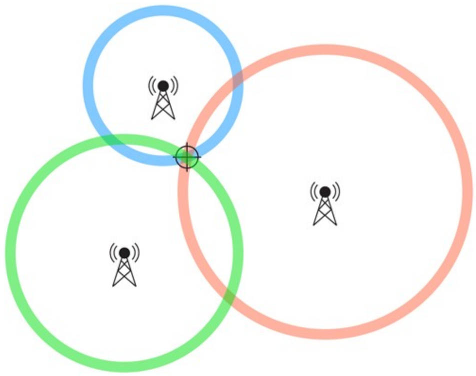
Mobile phones 'check in' with their nearest tower every few minutes or so[9], but other than that do not transmit unless they actively making a call, sending a text or using data. In the project scenario - the target phone is attached to an IED - it is entirely conceivable that the only communications it makes are the sporadic 'check ins' and the terminal phone call that detonates it. Given the frequency and duration of the check in signals - rare and short - it is near impossible to differentiate between them and random noise.

Upon further research, we discovered that it is firstly possible for a cell tower to initiate the 'check in' signal, and secondly possible to build a virtual base transceiver station - a spoof phone tower[10]. However, development of such technology is illegal in Britain due to the potential for malicious use in intercepting mobile phone traffic and tracking the movement of mobile phone users[11].

In response to this problem, we chose to create a 'dummy phone' that used other transmission technology. We made this decision in light of the illegality of pursuing further the fake tower approach, while still meeting our requirements of the proof of concept, as such concerns would not be an issue in a real counter-terrorism situation. As the nRF24L01 technology was reserved for communications of the robot, we chose to use a 433MHz transmitter as the fake device, and 433MHz receivers on our robot to detect this signal. We used a microcontroller to broadcast a known signal on this frequency for our robot to detect and track. We felt this was a suitable compromise, as we still detect radio frequency signals and use them for location, thus fulfilling our brief, whilst avoiding any legal concerns.

[9] http://news.bbc.co.uk/1/hi/technology/4738219.stm
[10] European Patent No. EP1051053
https://worldwide.espacenet.com/publicationDetails/biblio?CC=EP&NR=1051053&KC=&FT=E&locale=en_EP
[11] http://www.legislation.gov.uk/ukpga/2006/36/section/8

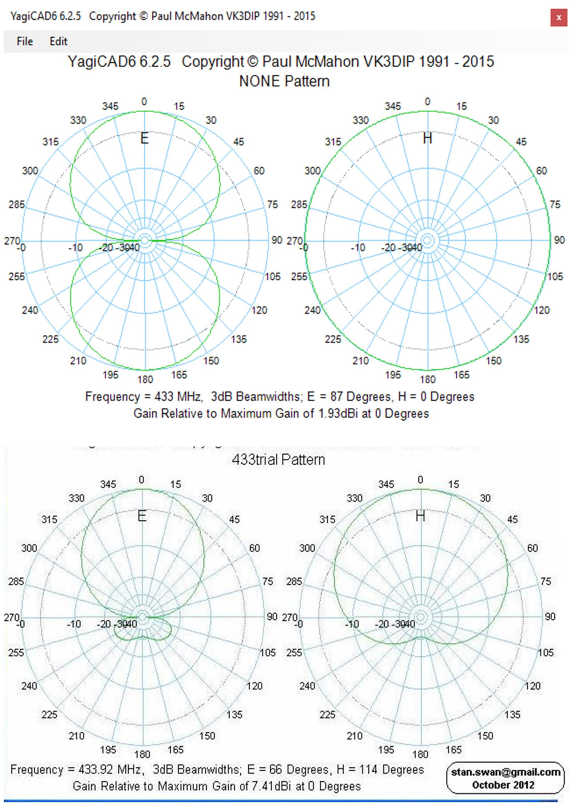
Determining the location of the phone could be done in one of two ways. Firstly, we could use the signal strength of the 433MHz signal to determine the distance of the phone from the robot. Geolocation data, provided by a GPS module, could be used in conjunction with these measurements taken from multiple points in order to triangulate the position of the phone, as shown in the adjacent diagram. Alternatively, we could have two 433MHz receivers with directional antennae pointed away from each other, and then always turn the robot in the direction that receives a stronger signal. We would then move the robot towards the phone and repeat, until it reaches the phone.



We concluded that our preferred method would be using directional antennae, as this would likely be more accurate. The calculation of the distance of the phone from the robot would have a significant uncertainty, and so too would the GPS data. Therefore we felt the triangulated location would have an unreasonably high uncertainty. Additionally, the directional antennae method could more easily allow us to pursue an autonomous system, as we could simply code the robot to move in the direction of the stronger signal. Accordingly, we searched for suitable antennae for the robot.

6.4 Antennae

Directional antennae rely on either a reflector to enhance signals from the desired direction or some sort of blocking to reduce signals from the wrong direction. Of the available types, ‘rubber ducky’ and ‘Yagi’ antennae are the smallest viable options we could find. However, despite rubber ducky antennae being smaller than Yagi antennae, they require a large metal plane behind the antenna to prevent signals from behind, which the robot could not have. Thus the Yagi antenna was the only option.



However, Yagi antennae are typically either the same size as the wavelength, or half the size. So they would be either 692mm or 346mm for 433MHz radio waves, both of which are far too large for our robot. In theory, dividing the wavelength by integer ratios should still enable directionality in the antenna, so using a piece of software called YagiCAD, We set about designing a Yagi antenna with quarter wavelength size, based on a half wavelength antenna from the internet. The tool has automatic optimisation features, and even with these, we were not able to create an antenna that had any discernible directionality, as can be seen on the antenna pickup pattern on the top half of the diagram. The bottom half shows the pattern from a far more capable antenna for comparison.

We thus opted to switch to the triangulation method of finding the phone.

6.5 Motor Control

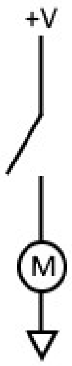


Fig. 1
Monodirectional control

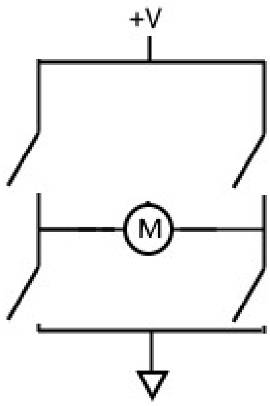


Fig. 2
Bidirectional control

If only monodirectional 'on-off' control of a motor is desired, then a circuit as in Fig. 1 above may be used, either switching high-side (as shown) or low-side (with the switch located between motor and ground). The direction of the motor is then determined entirely by its physical orientation, and will turn when the switch is closed.

Alternatively, if bidirectional control is desired, then an 'H-bridge' circuit as in Fig. 2 may be constructed. Diagonally opposite switches are concurrently closed in order for the motor to turn, and by changing which pair are closed the direction is reversed because the polarity of the motor's terminals is swapped. It is important to realise that with such a design, both switches on the same side must not be simultaneously closed, else there will be a short circuit.

For our projected robot, it is important that its motors are capable of bidirectional control, so it can rotate without changing its position, and so it can reverse. Therefore, an H-bridge design is needed. The next important consideration is the choice of switch. Manually actuated switches are impractical if the robot is to be controlled remotely, so electrically controlled switches must be used: either relays or transistors.

Relays use electromagnets to allow a small input current to switch a much larger current. The two circuits are isolated: the current to switch the electromagnet is completely separate from the current being switched. This is of no particular use in our application as there is no requirement for isolation. A key upside of relays is their typically very low resistance and consequently very large voltage and current ratings. However, they are slow to switch, bulky and they are more expensive than transistor devices.

Transistors are semiconductor devices, most commonly of either the BJT or FET varieties. BJTs are incredibly simple to use and very fast to switch. One key downside is the large power dissipation, especially when in the 'Darlington pair' configuration which allows for a larger current capacity. FETs can be more complicated to drive due to the gate capacitance which must be charged or discharged to switch the transistor, however, they are often more efficient than BJTs when switching large currents. When driven properly, they can be nearly as fast as BJTs.

In our project, transistors are most appropriate because the fast switching speed enables pulse-width modulation (PWM) control of the motors' speed. Controlling the speed of the motors allows software correction of any differences between motors, and also allows finer control of the robot. We chose to use FETs instead of BJTs because they are more efficient and produce less heat. This eliminates the need for a heatsink and decreases the weight of the robot.

For the motors themselves, we will reuse the motors from a previous project, because we realised as long as the motors were DC and moderately powerful, the exact model didn't matter. This helped us keep costs down, is a more environmentally friendly approach to robotics, and saved time, as we already knew the motors worked and were reliable.

7 Requirements

Having conducted our initial research we finalised the ultimate goal of the project: a robotic device housing receivers that can detect the dummy phone and relay this information back to a Raspberry Pi-powered base station. The robot is controlled from, and the data processed, on the Pi. We then defined a set of requirements to which we would work. The purpose of these requirements is to prioritise the functionality of our solution, and use as a measure of our success at the end of the project. Our ultimate requirements were as follows, alongside our requirements for the individual sections of the system:

- **The system must detect a phone within 50m of the phone**
- **The system must locate phone to within ±5m**

7.1.1 Robot Requirements

- The robot must detect the dummy phone
- The robot must be able to move
- The robot must be able to receive location data
- The robot should be controlled remotely
- The robot could move autonomously based on its proximity to the dummy phone

7.1.2 Base Station Requirements

- The base station must process data sent to it from the robot in order to determine the location of the robot
- The base station should process location data from the robot to display the robot graphically to the operator
- The base station should use received data in order to triangulate the location of the dummy phone
- The base station could triangulate the location of the dummy phone in real-time and display this to the operator

7.1.3 Robot-Base Communications System Requirements

- The communications system must enable data transfer between the robot and the base station
- The communications system should enable real-time, two-way data transfer between the robot and the base station

7.1.4 Dummy Phone Requirements

- The dummy phone must transmit a signal on a known frequency
- The dummy phone must transmit this signal in a periodic and repetitive fashion
- The dummy phone’s signal must be such that it can easily be detected
- The dummy phone must be detected at a 15m range
- The dummy phone should be detected at a 50m range

8 Product Research

8.1 GPS

A GPS module is required on the robot to determine the location of the robot, and thus enable the base station to track its path. In research for a GPS module, we considered a number of possible options, including the Microstack L80, uBlox neo-7M, and neo-6M, but in the end we opted for the Adafruit Ultimate. The Adafruit module was considerably more expensive than all the others; however it boasted a number of good reviews from hobbyists, as well as coming with clear instructions and libraries. The company is well known for the reliability its products, and we felt this was particularly essential in the case of the GPS. Other possibilities, whilst being cheaper, had poor or no reviews and offered few or no instructions or software. Given that we intended to use the module with an Arduino, it also made sense to purchase a module designed particularly for that purpose. Crucially, the Adafruit module was the most sensitive option in our budget, with reviewers telling us that it functioned well even indoors, achieving up to 1.8m accuracy in good conditions. Given the importance of GPS accuracy in triangulation, we therefore opted for this – the most accurate we could justifiably afford. The various modules we considered are listed and detailed in the table below:

GPS Module	Sensitivity /dBm	Arduino instructions?	Reviews?	Cost
Adafruit Ultimate[12]	165	Yes	Good	£39.95
Microstack L80[13]	165	No	None	£19.94
uBlox Neo-7M	161	No	None	£8.01
uBlox Neo-6M	161	No	None	£6.16

8.2 Arduino

Arduinos are low-power microcontrollers used in applications where the computational power and graphical abilities of a computer, such as a Raspberry Pi or BeagleBone Black, are unnecessary or overly expensive. The Arduino's low power consumption means that it is ideal to act as a microcontroller on-board the robot as the tasks it will be doing are not computationally intensive. Moreover, the integrated development environment used while coding for the Arduino is an invaluable tool in debugging, and one that many alternatives lack. Documentation and support is widely available online also. Below is a table of different Arduino types, and a comparison of their properties[14]. We elected for the Due, for its price compared to the Mega and number of pins compared to the Uno.

Type	Cost	Hardware Serial Ports	Digital I/O pins	Analogue Pins	CPU Speed /MHz	Typical Current Draw[15]
Uno	£15.00	-	14	6	16	50mA
Mega	£35.00	3	54	16	16	25mA
Due	£30.00	3	54	12	84	150mA

[12] https://www.amazon.co.uk/Ultimate-GPS-Breakout-channel-updates/dp/B008FZIZUE
[13] http://uk.farnell.com/microstack/microstack-gps/add-on-board-l80-gps-raspberry/dp/2434228
[14] www.arduino.cc and its related forums
[15] Varies depending on CPU load

8.3 Battery

A battery is required to power both the robot and the dummy phone. On the phone, the power consumption is low and ordinary AA or AAA batteries are sufficient. The robot, however, needs to power many more components. Ideally the battery on the robot should last as long as possible before it needs recharging, favouring a large capacity battery. Also, the battery must either be rechargeable or easily replaced, in order to minimise the disruption when it runs out of power. The greatest voltage required by any component on the robot is 6V, meaning the battery needs to be at least 6V, but ideally not more than 9V as power would be wasted in the built-in converters on the Arduino.

The various battery types available are compared in accordance to our needs in the table below using information about products we considered.

Battery	Size /mm	Weight /g	Voltage /V	Capacity / mAh	Cost	Rechargeable
Lead Acid[16]	151x25x94	1120	6	7000	£18.99	Yes
Nickel-Metal Hydride (NiMH)[17]	129x24x49	370	7.2	3500	£14.50	Yes
Lithium Polymer (LiPo)[18]	96x44x14	140	7.4	3500	£17.39	Yes
Alkaline (AA)[19]	50x14x14	20	1.5	2100	£2.00	No

Although the AA batteries have a very good capacity for their price, they drain at a much faster rate than other types of batteries so are unsuitable for use on the robot. On the other hand they are cheap, small and light making them a perfect power source for the 'dummy phone'.

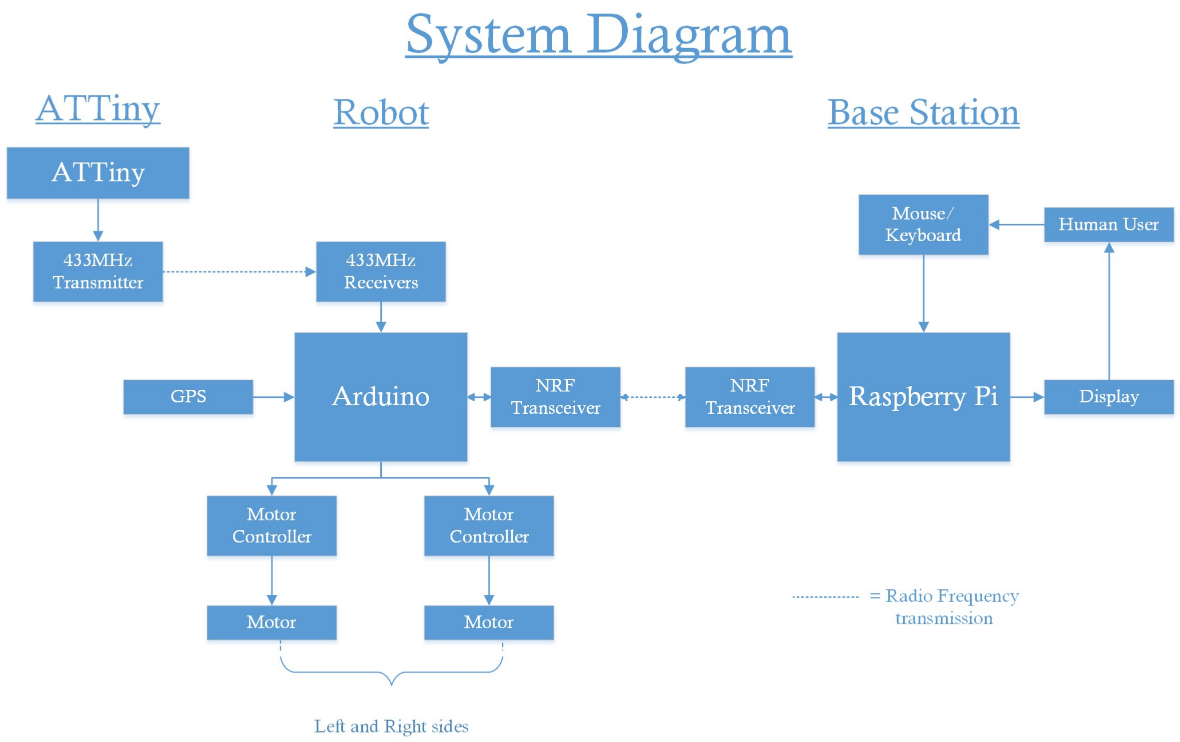
Lead acid batteries are a very large volume, heavy and expensive. This made them less suitable for the robot. Ni-MH was slightly less expensive, making it a better choice for the project. Furthermore, it was difficult to find both LiPo batteries and a suitable charger that would deliver by the time we went to the workshop in Cambridge where we planned to begin work on constructing the project, thus making LiPo unavailable for use.

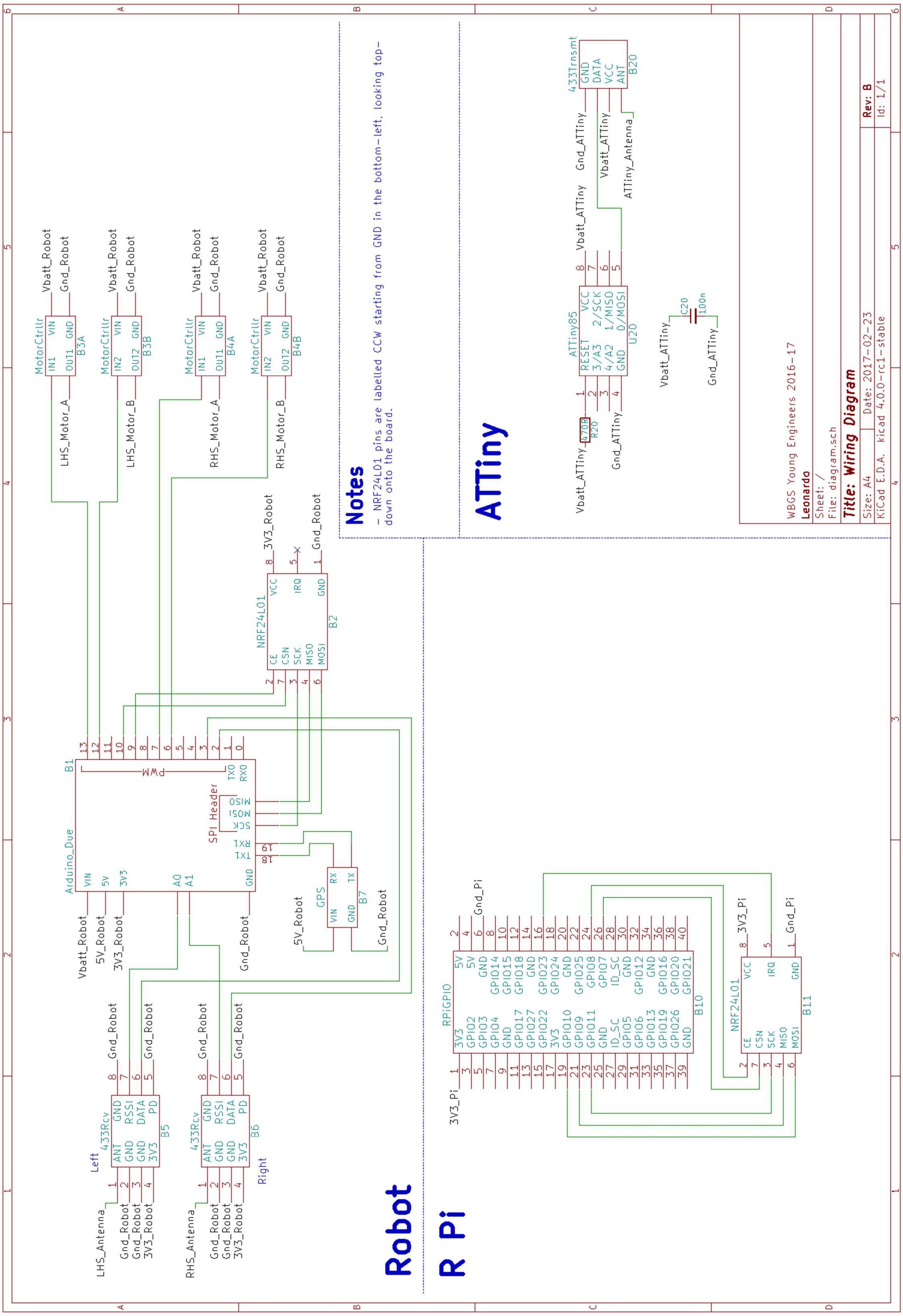
[16] http://www.maplin.co.uk/p/6v-sealed-lead-acid-battery-7ah-1-pack-n33fr
[17] https://www.amazon.co.uk/dp/B00Q3TP584
[18] https://www.amazon.co.uk/dp/B01AHJRLXW/
[19] http://www.alliedelec.com/duracell-mn1500bkd/70149224/

9 Design

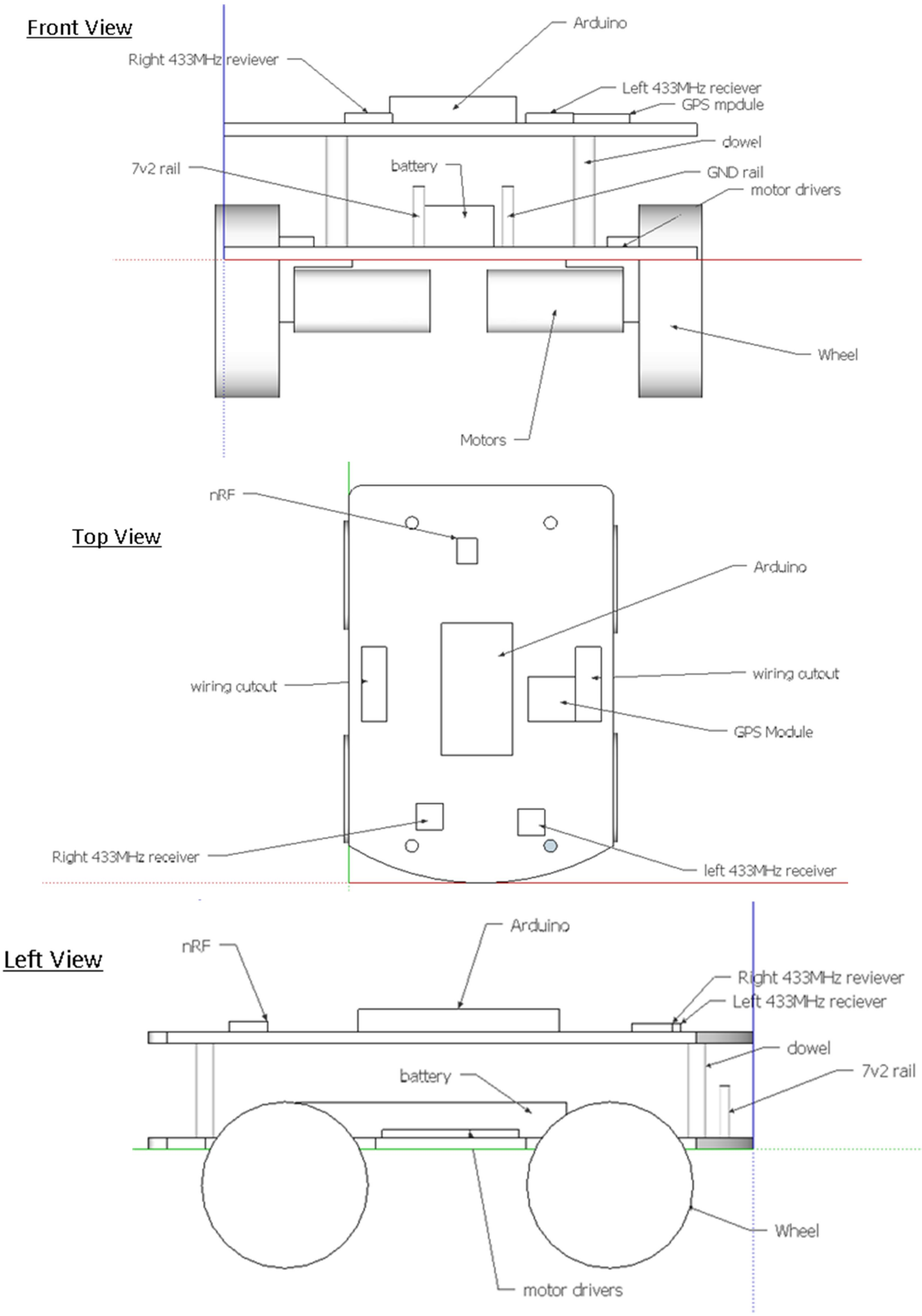
Having completed our research into the different components and considerations necessary to fulfil our requirements, we began to design our system. The system is split into two main project areas: the robot and the base station. The dummy phone required only a small amount of work. Based on our past experience and knowledge, different members of the team worked on different parts of the design. For instance, three members of the team had significantly more experience and familiarity with coding so the software work was left to them. Below our final design is analysed, component by component, and our methodology and decisions relevant to each component are explained.

Here is our final system diagram. It shows the flow of information and the structure of the whole solution. On the following page is the circuit schematic for the whole project.




Robot
R Pi
ATTiny
Notes
- NRF24L01 pins are labelled CCW starting from GND in the bottom-left, looking top-down onto the board.
WBGS Young Engineers 2016-17
Leonardo
Sheet: /
File: diagram.sch
Title: Wiring Diagram
Size: A4
Date: 2017-02-23
Rev: B
KiCad E.D.A. kicad 4.0.0-rc1-stable
Id: 1/1

9.1 Robot



2D CAD drawings of the robot's chassis, and how components will be laid out upon it
A 3D render of our design is in Appendix C

9.1.1 Arduino

The robot needed a central controller; we chose an Arduino for this as opposed to a Raspberry Pi as the Arduino uses far less power whilst still being computationally adequate, meaning the batteries are able go much further. Within the Arduino category we chose to use a Due as it has hardware serial communications channels needed for the GPS module, as well as being more powerful than a Mega and having more I/O pins, needed for all peripherals, than an Uno.

The Arduino controls the motors through the motor controllers, takes readings from the 433MHz receivers and the GPS module, and communicates this information to, and takes instructions from, the Raspberry Pi through the nRF module.

The Arduino is placed on the upper deck since it needs to be in relatively close proximity to the communications devices, which all benefit from being on the upper deck due to potential blocking of signals were they on the lower deck. It also makes programming the Arduino easier, as connecting the cables is less fiddly.

9.1.2 Batteries and Power

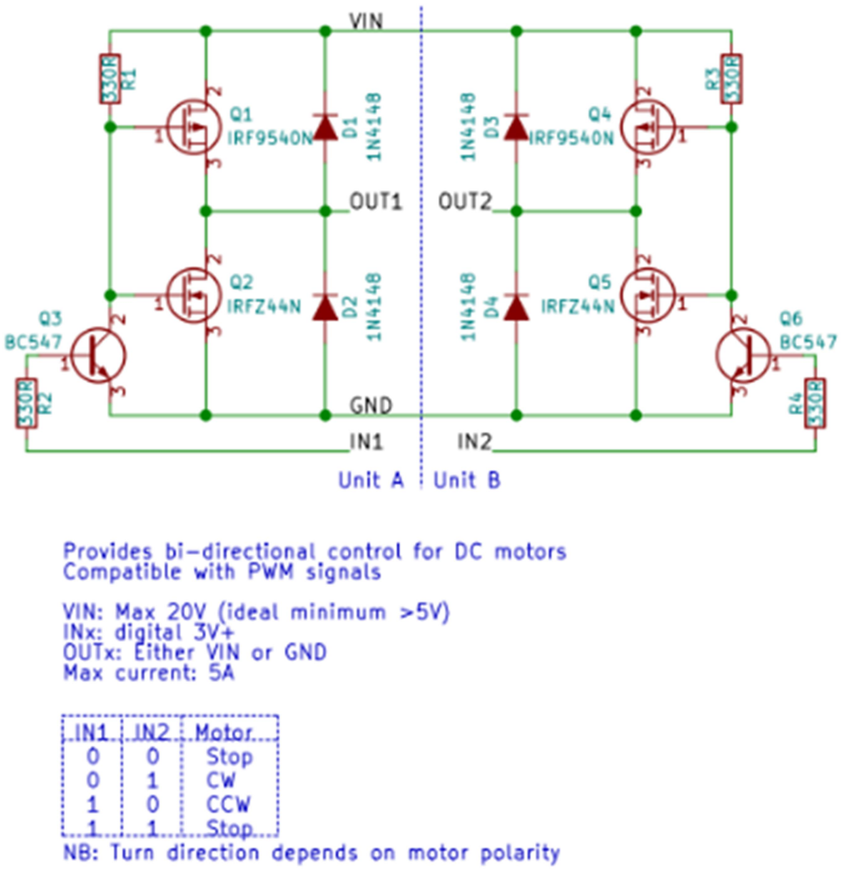
A power source is needed to power the robot. It needed a large capacity to power the robot for extended periods of time between charging. The battery also needed to be compact enough to fit on the robot. As detailed previously, we chose the Ni-MH batteries as they best fulfilled the needs of the robot's power source.

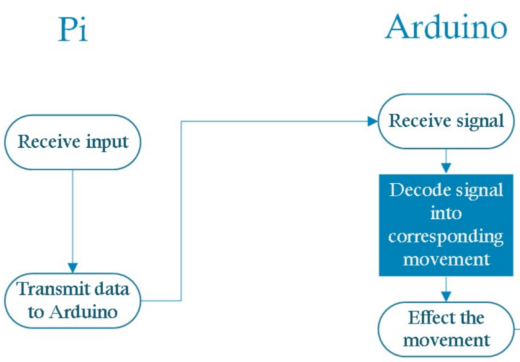
The battery is rechargeable so it has a Velcro strip on the underside to hold it in place, while also being removable from the chassis. Velcro is a weak way of attaching it, but in this case suitable as the robot will not be traveling very fast, so there is little chance for components to dislodge. The battery needs to be wired up to the other circuitry so the original, non-standard connectors are replaced by spade connectors.

9.1.3 Motors

We decided to use motors already owned by the school, which had been used by the previous year's Young Engineers team. Nevertheless, we still required a motor controller to allow the low-current Arduino outputs to drive the motors, which together could require over 1A of current. Additionally, they run best off the full battery voltage rather than the 3.3V digital output of the Due. We elected to produce our own motor controller design, because it would give us the chance to design circuits from scratch rather than just reusing a commercial design.

To this end, we created a simple H-Bridge design using N- and P-channel MOSFETs. PFETs were chosen for the high side as, despite their typically worse operating characteristics, they are easier to drive on the robot as no higher voltage than the battery voltage is available. The IRFZ44N and IRF9540N MOSFETs were used, because they were cheap, had a sufficiently low $R_{ds(on)}$ and could handle the power required easily. Finally, a small BJT transistor was also required in order to handle the potentially large peak currents that could occur when the motor controller switches state. A transistor protects the Due from potentially being damaged and the BC547 was chosen, again for its price. The overleaf circuit schematic shows our final design for the motor controller. Two copies were needed, one each to cover each side. The subsequent flowchart shows how the Arduino controls the movement, following instructions transmitted from the Raspberry Pi.





Movement	Left side	Right side
Forward	Anti-clockwise	Clockwise
Turn left	Anti-clockwise	Anti-clockwise
Turn right	Clockwise	Clockwise
Backwards	Clockwise	Anti-clockwise

9.1.4 Chassis

We decided to use plywood for the robot chassis, as it was relatively strong whilst being very cheap. This ensured that, if we had to redesign at any point, producing a different chassis would be inxpensive. We decided it would be best to laser-cut the plywood as would ensure an accurate cut out of the design, as opposed to the uneven cut, and therefore imprecise layout for component holes, which may have resulted from cutting the wood by hand. Using a laser cutter also made it possible to produce a new chassis very quickly, meaning that changing our design could be fast and easy. Another alternative was to buy a premade chassis, however we quickly ruled this out as it would prevent us from being able to remodel our design, as valuable time and money would be wasted in ordering new ones.

The chassis was designed to be as easily maintainable as possible, with ease of use in mind. The robot needed a large enough wheelbase to make use of the 90mm wheels, hence the size of the chassis. We quickly realised that the components would not all fit on a single deck, so we opted for a two-tiered design rather than hanging electronics from the bottom of the chassis and having to rely on the attachment method used to hold the component. The shape also lends itself nicely to having a protective shell fitted around it were it to be used in a military situation, which would provide resistance to the elements, or even possibly protection against explosives.

9.1.5 Communications

Communications needed to be reliable and fast, as a robot which is too slow to respond to an operator's command or which has 'gone rogue' is inherently dangerous. To combat this, we ensured tight error checking rules and implemented a failsafe system of defaulting to transmitting a stop command from the base station, should the robot miss the first stop command. The command protocol for the communications we used is explained in detail in the Base Station design section below.

We decided communication would be achieved through nRF24L01 chips, which are compatible with both Raspberry Pi and Arduino, and as mentioned in the research section, are cheap and easy to use. They do not require an antenna, as the range we plan to use them for is well within that of the antenna built into the device.

9.1.6 Firmware

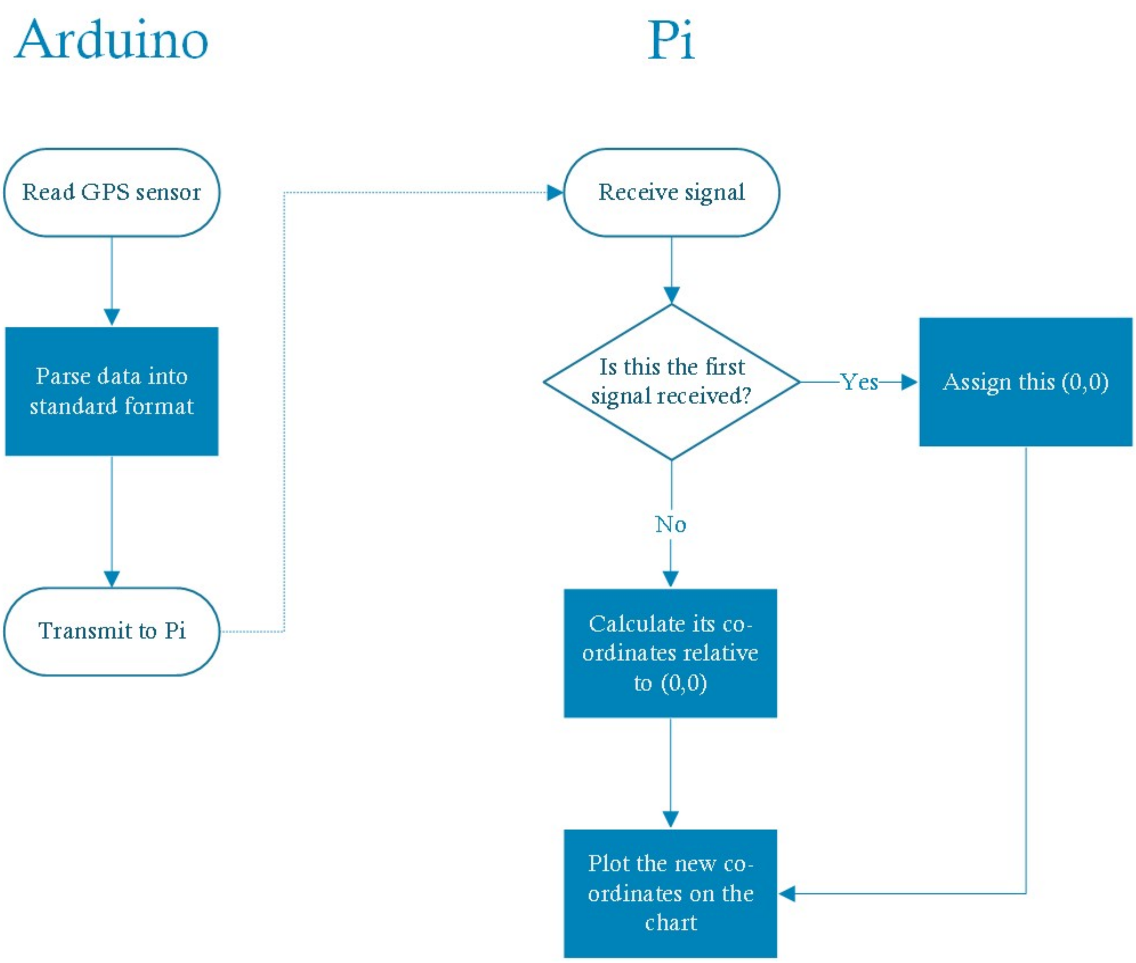
We decided that the Arduino on the robot should not be doing any computational tasks, since the Pi is more powerful and can perform tasks quicker. Furthermore, it means that algorithms and code can be altered without having to reprogram the firmware of the robot.

This meant all that the Arduino should be doing was waiting for messages from the Pi via the communications module. To this end, the firmware is programmed to be constantly trying to read messages from the nRF module and then execute any commands received. If data is requested from the Pi, the Arduino will collect the readings then send the data back to the Pi.

9.1.7 GPS

We realised that in order to obtain the precise location of the phone we would need a GPS module on the robot. We decided that the robot would send back periodic GPS data so that the path it takes can be plotted and mapped. This enables 'blind' driving, so the SID can be controlled in a situation where the operator doesn't have direct line of sight on the robot, and maximises the amount of information available to the operator should they need to make any informed decisions.

We positioned the GPS on the top level so that it could have the maximum possible signal strength from the satellites it communicates with. This also meant it was easier to physically connect to the Arduino. The following flowchart maps how the GPS data is used in the SID system.

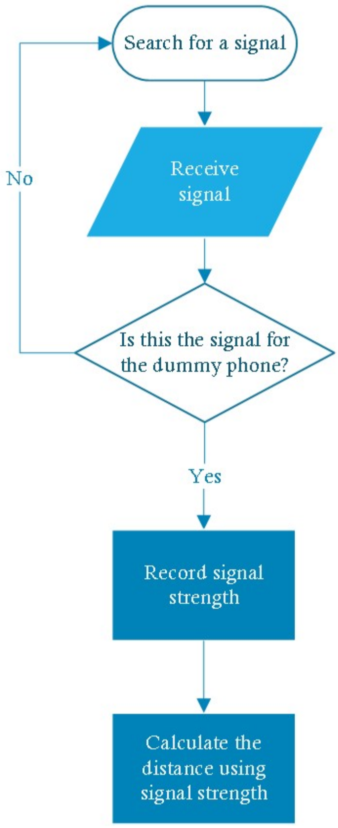


9.1.8 Signal Receivers

The SID proof-of-concept is centred on the detection of 433MHz radio signals which are emitted from the dummy phone (see the Dummy Phone design section below). In order to do this, we have 433MHz signal receivers positioned on the robot.

We initially planned to have two receivers pointing in opposite directions, and move the robot in the direction of the stronger signal. However, we soon found that it would be nearly impossible to achieve directional signal reception, as detailed in the research section, and so switched to a triangulation method. Strictly this requires only one receiver, but we kept both for redundancy.

The 433MHz receivers are positioned on the top level, because this allows us to wire them easily into the Arduino, and also it reduces the shielding of the signal. Below is a flowchart for the detection of the dummy phone, using the 433MHz receivers.



9.2 Base Station

9.2.1 Communications

A command protocol is the best way to achieve reliable and fast communication between the Pi and the Arduino. We decided that each command would consist of a command followed by any data that may accompany it as a payload.

To minimise the data sent, commands are limited to single bytes. The command values are all chosen to have at least two bits different to any another value, to give further protection against corruption. To add to this, all data sent along with a command, for instance the GPS position following a 'GPS Data' command, is put through an error detection algorithm to ensure data is not corrupted. This is done by sending the message, then sending a bitwise-NOT version of this afterwards. If the two binary messages do not add to make a full message of 1s then corruption must have occurred and the message is discarded.

We chose not to include distance to travel as a parameter of forward movement, and instead send a forward command, which the robot obeys until a stop or another direction command is received. We did this because we felt that the distance was never going to be a meaningful number as the robot would move different distances depending on how much traction it has on its current surface. Furthermore sending single direction commands was more compatible the GUI design and functionality. The Pi does, however, send a stop command once every half second when the robot is not meant to be moving, in case a stop is ever missed, preventing the robot from trailing off.

Below is a table of commands and their respective bytes and values:

Command	Binary	Integer Value
Stop	0000 0000	0
Move Forward	0000 0011	3
Move Backward	0000 0101	5
Turn Left	0000 0110	6
Turn Right	0000 1001	9
Request GPS Data	0000 1010	10
GPS Data	0000 1100	12
Request Scan	0000 1111	15
Scan Data	0001 0001	17

9.2.2 User Experience

We decided to create a graphical user interface (GUI) on the base station side, as it makes the program easier to use and requires less training. Rather than typing commands, the operator can click buttons on the screen or press buttons on the keyboard to control the actions of the robot. Additionally, it enables the Pi to graphically plot the GPS coordinates of the robot and the corresponding distance of the dummy phone, so the operator can make a judgement on the location of the phone. Thus, the phone's location need not be computed mathematically. This also is a more useful way of displaying the information, as it is significantly easier for the operator to interpret a map than an array of data.

9.3 Dummy Phone

In line with our research into the illegality of detecting actual phones, we needed to produce a dummy phone that would send a signal we could detect in place of a real phone. It needed to transmit a signal that is easily identifiable as the dummy phone and can be detected at a 50m range.

To accomplish this, we used an ATTiny85 microcontroller. We selected this device because it supports the Arduino framework and therefore the libraries used on the receiving robot can also be used on the transmitting phone. It also makes programming simpler, as we had experience with the Arduino framework. The electrical design of the dummy phone is included on the main wiring schematic at the start of this section.

Finally, we had access to an old Nokia mobile phone. We decided that the dummy phone would act as a better demonstration of our project if it were concealed within the Nokia's case. We planned to remove the battery and other circuitry from the old Nokia, and place the dummy phone's circuit board in their place. To those unfamiliar with the project, the purpose of the robot (to locate mobile phones) immediately becomes apparent in an intuitive manner.

10 Procurement

Having designed our project and its three separate aspects, we produced a Bill of Materials (BOM) for the robot, the base station and the dummy phone. Shown below, it includes the cost and source for each component in our final design. The total cost of all the components was £134.28. This was funded from our budget of £100 from Leonardo, supplemented with £100 from school.

10.1 Robot

Component	Source	Quantity	Cost per Unit
Arduino Due	Leonardo - RS	1	£25.85
Adafruit Ultimate GPS	School - Amazon	1	£39.95
nRF24L01 (Pack of 5)	School - Amazon	1	£5.25
Floureon 7.2V Battery (Pack of 2)	School - Amazon	1	£28.99
Ansmann Battery Charger	School - Amazon	1	£6.98
DC Motor	School	4	Free
Wheel	School	4	Free
Plywood Chassis	School	1	Free
Quasar Telemetry 433MHz Receiver QAM-RX1-433	Leonardo - RS	2	£5.17
Motor Controllers			
IRF9540N P-Channel MOSFET	Leonardo - RS	4	£0.86
IRFZ44N N-Channel MOSFET	Leonardo - RS	4	£0.84
1N4148 Diode	Leonardo - RS	8	£0.06
BC547 NPN Transistor	Leonardo - RS	4	£0.14
330R Resistor 0.6W	Leonardo - RS	8	£0.01
100x160mm Stripboard	Leonardo - RS	1	£4.09

10.2 Base Station

Component	Source	Quantity	Cost per Unit
Raspberry Pi Model 2	Leonardo	1	Loaned
nRF24L01 - *See Robot BOM*	-	-	-
Monitor	School	1	Free
HDMI-VGA Adapter	School	1	Free
Keyboard	School	1	Free
Mouse	School	1	Free

10.3 Dummy Phone

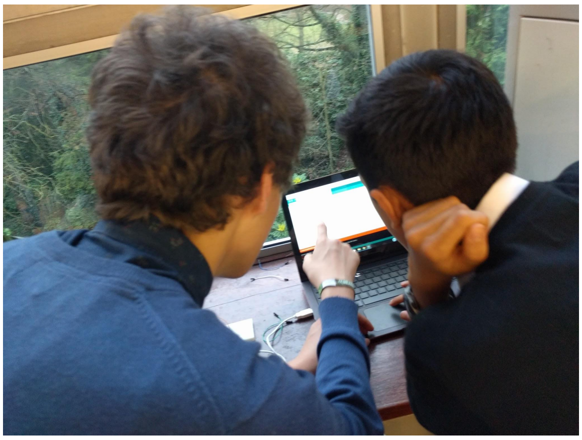
Component	Source	Quantity	Cost per Unit
ATTiny85 Microcontroller	Leonardo - RS	1	£1.78
Quasar Telemetry 433MHz Transmitter QAM-TX1	Leonardo - RS	1	£3.13
Nokia Phone Case	School	1	Free
Green LED 5mm	School	1	Free
470R Resistor 0.25W	School	1	Free
3K Resistor 0.25W	School	1	Free
100nF Capacitor	School	1	Free
Perfboard	School	1	Free

11 Construction and Testing

11.1 Cambridge Workshop

In December we attended a workshop at the Department of Engineering in Cambridge University for two days. We drew upon the expertise available and utilised the quality resources in the laboratory to make progress on a number of key deliverables.

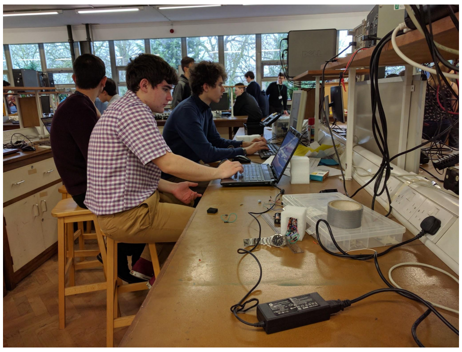
11.1.1 GPS Coding and Debugging



Using information about the GPS module we found online, we wired it up to the Arduino ready to receive a signal. We conducted some research online about other projects that used GPS and looked at examples of documentation. Following this, we wrote up some of our own code and trialled it with the GPS. Initially we were unsuccessful, encountering problems with the software, syntax of the code and wiring of the GPS component, but by the end of the workshop we managed to run a successful test of the GPS within error margins of ±10m.

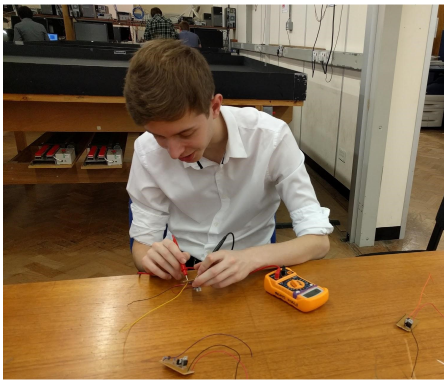
11.1.2 Communications Coding and Debugging

Similar to the GPS, we researched the component datasheets for the 433MHz receiver and transmitter components, and found the wiring diagrams online. We then set up the wiring as required and connected the components to run the code for them. This became slightly more complex than anticipated because we had to program both the receiver and the transmitter separately and test they were running successfully - this was hard to do for each of the components independently. We had to go through several trials and debugging attempts but we made good progress and at the end of the workshop had a reliable set up to imitate a phone signal.



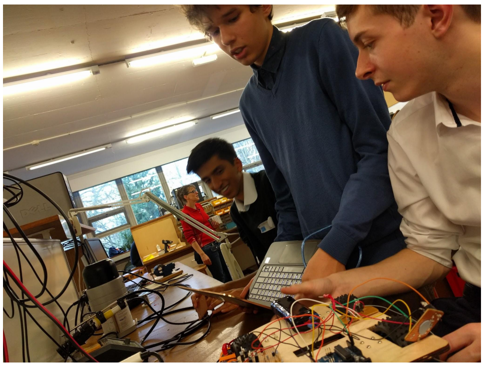
11.1.3 Electronics for Motor Controllers

We made a big step in terms of electronics during the workshop. The engineering department at Cambridge had an excellent supply of tools and equipment on offer, as well as technicians who were happy to assist or answer questions. We were able to make all four of the motor controllers required to allow the robot to move. They consist of wires and transistors that had to be soldered onto pieces of circuit board with high precision. After the soldering process, we cut the circuit board down to size and mounted it on the robot. We also tested that each one was working. An issue we found during testing was that the solder on one motor control had accidentally short circuited the power to ground, which resulted in a damaged transistor. However, we were able to replace it and fix the motor controller.



11.1.4 Chassis Layout Design

With the robot in front of us together with all the other components, we were able to refine the layout of everything that had to be mounted onto the robot. Using paper sketches that we compared with the prototype chassis, we experimented with different chassis shapes and checked that they could hold the necessary components. We decided that the motor controls, when cut to size, could fit snugly at either side of the chassis in between the wheels, and that the best place for the wires to come up through would be at the back of the chassis.

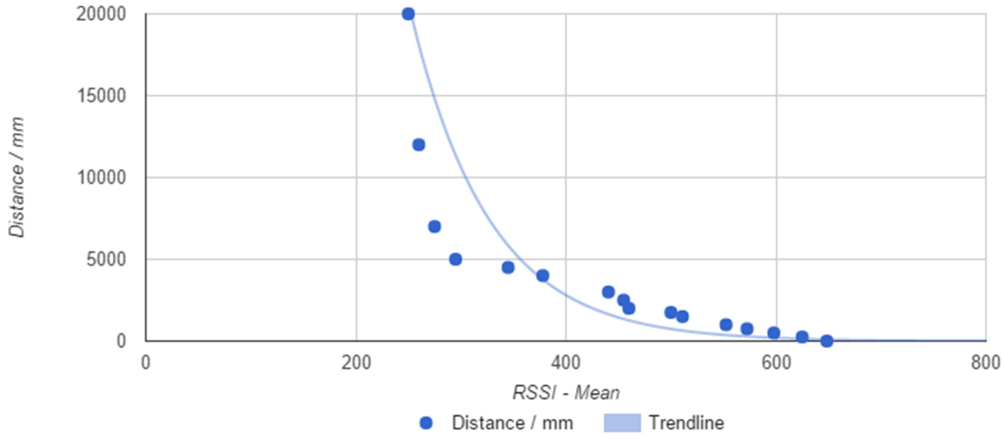


Overall, we found the Cambridge workshop useful, insightful, and a fantastic experience. We were very glad to have had that opportunity, and it was a central element of the project's development.

11.2 Calculating Distance

We needed to calculate the distance of the fake phone from the received signal strength (RSSI). We did not initially have much of an idea of how this would be done, but we assumed we would be able to model the behaviour of the RSSI depending on the distance. A test was conducted where we would measure a distance and record the average RSSI for that distance, ultimately allowing us to plot the distance against the RSSI.

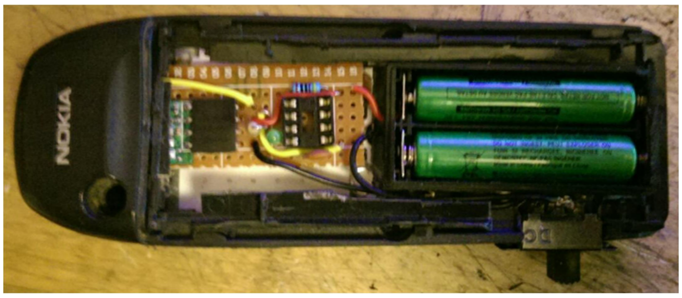
The graph overleaf shows the results from this experiment. Evidently there is not a very clear trend in the data, but it appeared to be exponential in nature. Using regression, we were able to obtain an equation for the trend line, thus enabling us to convert RSSI values into distance measurements.



After further testing, the closer readings tended to be less accurate and very similar in value, as shown by the trend line. For close distances, we spotted that a linear line modelled the data better so we decided to use a linear equation for high RSSIs.

11.3 Dummy Phone

Creating the dummy phone initially on a breadboard was relatively easy, only requiring minor modifications in the code for the switch between using a temporary Arduino Uno, used during testing of the 433MHz communication, to the ATTiny. However, when we assembled the circuit with the ATTiny on a breadboard, nothing appeared to be working. After having the idea to put an LED in parallel with the data line running to the 433MHz transmitter from the ATTiny to see if the ATTiny was sending any data at all, we realised the ATTiny was no longer functioning, and replaced it to find everything worked perfectly.

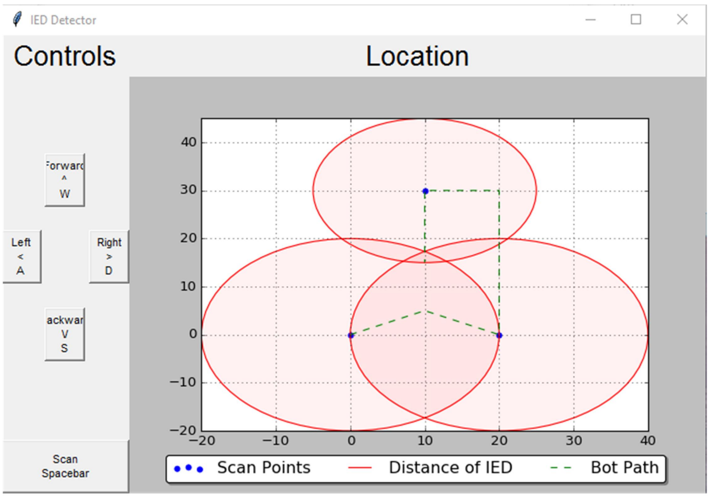


We then transferred the circuit from the breadboard to a permanent PCB with a battery pack. The case was made from an old Nokia 6310i, whose own circuitry was removed with a rotary tool.

11.4 Base Station Software

The base station required a GUI in order for the operator to communicate with the robot effectively. This is needed to allow input for movement control, as well as output any relevant data such as the location of the potential IED.

Using TkInter, a simple GUI was created with buttons that could be pressed, which sent signals to the robot on what to do. Additionally, event handlers were attached to the pressing of keyboard buttons. This allowed for the robot to be controlled using the keyboard, in addition to clicking with the mouse, as we felt that this was an easier and more useable system. Using matplotlib, we were also able to draw a graph. This allowed us to mark the location of the robot with its latitude and longitude as well as draw a circle whose circumference is at the distance corresponding to the RSSI value at that location. Subsequently the smallest area bounded by the circles (which is shaded the darkest automatically) is the region in which the IED can be, which provides an easy visual cue to the operator of the whereabouts of the IED. From experience, roughly 4 to 5 measurements are enough to get the position of the phone to within 5m.



11.5 Code

The code was continually developed as the project advanced. Each section could be developed on separate Arduinos and then all the parts combined at the end of the product, allowing for the development to be streamlined. It was developed using GitHub, which allowed for everyone to work on the coding at the same time. The code is available at https://github.com/wrightg42/young-engineers-16-17.

The following libraries were used in the development of the code:

11.5.1 Raspberry Pi

- TkInter to construct the GUI
- Matplotlib to plot the location of the robot and the dummy phone signals
- GPIO and SPI libraries to connect to the nRF module
- A Python port of the nrf24 library in order to talk to the nRF module

11.5.2 Arduino

- Adafruit GPS library to interface with the GPS module
- SPI library to interface with the SPI pins on the Arduino to work with the nRF module
- nrf24 library to interface with the nRF modules to read/write data
- RCSwitch to interface with the 433MHz transmitter and receiver modules

12 Challenges Faced

12.1 nRF Communications

Communicating via NRF chips is a trivial task when between two Arduinos and we managed to achieve this fairly swiftly. We had to perform this initial testing as the chips we originally wanted were shipped from America and therefore would take too long to arrive, so we bought cheaper ones from the UK whose quality was less reliable.

When it came to using the nRF chip with a Raspberry Pi it became more difficult. Our solution was being coded in Python 3, and it is very difficult to find a Python port of the C library used on the Arduino. After looking online we found a port but there were many required dependencies which had not been mentioned, and the examples threw errors as we had the wrong version of some of these dependencies.

Eventually, after some minor adjustments, we managed to make the library work, enabling the Arduino and Raspberry Pi to communicate. This being said, sometimes the code would stop sending messages and meant the program needed to be restarted. After some testing, the bug disappeared once both chips had tried to read some data, so we made sure it did this during initialisation so the problem did not occur again.

12.2 Radio Library

With the 433MHz library we encountered a problem of the library involving many static types, meaning we could not load two instances of the library to read data from both of the 433MHz receivers. No library or fix could be found online so we were forced to make a second copy of the library under a different name so that we could read from both devices simultaneously.

12.3 Motor Control

The majority of our testing was done in Cambridge, however, once we brought the robot back and changed the wiring of the motors from their temporary positions to their final places on the Arduino, we experienced strange behaviour on start-up of the Arduino – the robot would jump roughly a centimetre forward. This turned out to be because certain pins on the Arduino are set to be powered on by default, whereas the motor controllers required either all of the pins or none of the pins to be on for the motors to stay still. This meant that whenever the robot was turned on, the wheels would turn for a fraction of a second and cause the robot to ‘jump’ forward. This issue was inadvertently resolved when we moved the motor control wires to PWM pins in order to be able to calibrate the motors with respect to each other. Using the PWM pins allowed us to drive one side slightly faster than the other in order to compensate for slight differences in the motors’ torque.

13 Recommendations

The SID is a proof-of-concept development whose functionality has been limited by time, financial, and legal constraints. Our recommendations for an actual solution expand on the current implementation in a number of ways.

Firstly, the mobility of the vehicle is inadequate for hostile environments, and stronger motors with a caterpillar track would be required to enable it to navigate uneven terrain. This would need to be accompanied with a stronger, bigger, and more durable chassis, as well as greater capacity batteries, or perhaps a combustion engine. We decided, in light of cost and time constraints, that this functionality was not necessary for a proof-of-concept, and as such is limited in our project.

The sensitivity of the 433MHz receivers is another limitation in the functionality of the SID. The accuracy and range of our detection is largely determined by the receivers, and financial constraints demanded that we opt for cheaper ones whose sensitivity would be insufficient in a real counter-IED or counter-insurgency situation. Again, however, we felt that this did not affect the integrity of our proof-of-concept, as the range and accuracy of receivers are dramatically improved at higher price bands, so this limitation is purely financial.

The SID is reliant upon a human operator to guide it by eye, which would of course be unfeasible in a real-world situation. Depending on client requirements, this could be resolved in one of two ways. Firstly, FPV cameras mounted on the robot could relay live video data to a remote operator which would ensure human safety whilst enabling direct control of the device. Alternatively, an infrared or ultrasonic sensing system could be used whereby the device is able to autonomously avoid obstacles using sensors positioned around the robot. This would enable fully unmanned operation, liberating valuable manpower. Both of these options were considered, but considered ultimately unfeasible early in our project given time, cost, and expertise constraints. A camera system would have required some form of wireless communication with greater bandwidth and would have been itself expensive. Whilst an IR mechanism could have been economically viable, time constraints proved too great.

An important change we have had to make to the SID is its detection of 433MHz signals rather than phone signals. Discussed earlier, this was due to strict legal constraints surrounding the manipulation of phone signals in the UK. In a genuine counter-IED or counter-insurgent situation, where such concerns would not apply, a ‘cell site simulator’ - a fake phone tower - could be used. Such technology already exists and is in widespread use in military and law enforcement contexts[20]. This would enable the device to force phones in the area to transmit radio signals, and also force them to increase the signal strength. Subsequently, RF receivers on the robot would be able process the signals as they do in our current implementation.

[20] https://www.aclu.org/map/stingray-tracking-devices-whos-got-them
https://www.nytimes.com/2016/02/12/nyregion/new-york-police-dept-cellphone-tracking-stingrays.html

14 Evaluation

14.1 Robot

The robot itself has met most of our requirements: it houses the 433MHz receivers in suitable positions on the front of the top tier and it can be moved remotely from our Raspberry Pi base station. The robot can move at a reasonable speed and can traverse all flat surfaces. The battery mounted on the robot is easily able to power the device for hours at a time and can easily be switched out with a replacement.

However, the robot does not function autonomously. As detailed in the recommendations, an autonomous system could have been developed using infrared sensors. However, the complex coding required for such a system could not feasibly have been perfected in the given time period. Autonomy would have been a nice feature for the SID, but we felt that not having it would not affect our proof-of-concept as it is not detrimental its functionality. The key detection and processing systems are independent of way the robot moves. This is why we assigned it a low priority, and focused on other elements of the system. Given more time and a greater budget, an autonomously-driven system would certainly be possible as an additional feature to the SID.

14.2 Base Station

The Raspberry Pi - as the base station – is able to process all the data sent to it by the Arduino. It is able to use the location data from the GPS to determine the location of the robot, and maps the path of the robot graphically. The base station also processes the 433MHz data from the robot into distance values, and graphically displays this on the map to the operator. This enables triangulation of the location of the dummy phone, but relies on the human operator to select the location of the device.

Thus, the base station meets most of our requirements of it, but falls short of computing the triangulated location in real-time. This would have been a useful functionality for the SID, as it would have further demonstrated the ability of the system to liberate manpower. However, developing triangulation software was too complex given our time constraints. One reason for the complexity is that if multiple devices are present then the data may not all be centred on one point. For a human operator this poses no challenge but for an automated system this introduces a significant complexity that we did not have the time or expertise to solve.

Therefore, again, we assigned it a low-priority as we felt that human-operated triangulation was sufficient to demonstrate the concept behind the SID and automated triangulation would not have added enough value to justify prioritising it over other elements of the system. Given more time, and were the autonomous movement of the robot to be developed, we could develop this software in order to further save manpower.

14.3 Communication sSystem

The communications system is perhaps the most important aspect of SID, as it allows the robot to be remotely controlled. This is what makes the 'Safe IED Detector' safe, and is vital in a warzone. Despite our initial problems with getting the Pi and Arduino to communicate with the nRF modules, the communication system now successfully enables real-time, two-way data transfer between the robot and the base station, and therefore fulfils our requirements of it.

14.4 Locating the Phone

The GPS, as promised on the data sheet, provides the location of the SID to within ±10m. By averaging multiple readings of the phone distance with respect to GPS coordinates and triangulating the position of the phone, the operator can determine the location of the phone to within ±5m. However, the phone cannot be found within the entire range stated in our requirements. We can only detect the phone within 30m of the robot, not the specified 50m. This is most likely because of signal interference and smaller than ideal antennae.

14.5 Project Management

Our project was driven by weekly meetings, where we discussed the progress made, shared thoughts, and planned for the week ahead. Every other week, two engineers from Leonardo came to the meeting to offer their expert advice and assistance. These regular meetings proved to be essential as we encountered many problems as the project developed, and regular meetings enabled us to collectively work out solutions and adjust the direction of the project if needed.

We knew that the pragmatic delegation of tasks would be a decisive factor in the success of our project, so our weekly meetings ensured that work was allocated to the right people in appropriate quantities in order to meet project deadlines. We recognised quickly that communication was absolutely essential given how interconnected the different elements of the system were. We set up a joint Google Drive to collaborate on tasks and the report, and a WhatsApp group to discuss issues. This proved to be a very useful way to ensure that we maintained a high degree of communication and shared the workload.

We developed a Gantt chart early on in the project, a copy of which can be found in Appendix B. The Gantt chart helped create deadlines for different elements of the project, helping us to keep on track by constantly drawing our attention to the bigger picture. We found that it was a highly effective way of managing time; when we occasionally fell behind, it was clear how it would affect other areas of the project, and so we were able to readjust the direction if needed.

Using GitHub for code management meant that we could work on the code concurrently. We made a separate branch for the different aspects of the system: one each for the robot, the Pi and the dummy phone. It also meant that we could bring the SID into school or take it home and still have access to all the code.

Overall, we worked very well as a team. We each had an understanding of each other’s strengths and weaknesses, and so were able to delegate tasks effectively and efficiently. Combined with the support and guidance of Dr Buckley; our mentors, Harry, Mark and Rob; and with our collaboration and communication systems in place, we functioned as an exceptionally productive unit.

15 Personal Reflections

15.1 Srivatsa Garg

Over the past six months, working on the SID as part of the Young Engineers project has been both incredibly enjoyable and rewarding. I have gained a huge amount of invaluable experience taking part in this programme, from learning how a long-term team project takes shape and develops over a period of time, to getting into the nitty-gritty of hands-on electronics. This has been particularly valuable for me as I entered into the project with no real experience of coding or electronics, and leave with significantly more confidence and understanding of how programming and circuitry work.

Throughout the course of the project, I have endeavoured to get 'stuck in' and contribute in whatever way I can. I have taken joy in conducting extensive research around the project, particularly surrounding GPS systems and modern IED functionality, have been actively involved in the procurement process and design of the robot, and have also genuinely enjoyed working on the report. In hindsight, the one thing which I wish I had done is broaden my programming skills beforehand, which would have enabled me to support the software development more.

The project has really deepened my interest in the world of engineering, and I feel more confident in making an informed decision about my university course for having been involved. I would like to profoundly thank Dr Buckley for providing us with this opportunity, and also Harry, Mark, and Rob for all their support and guidance. It has been tremendously satisfying to see the project and report come to life, and I can only hope that whatever I do in the future will be as enjoyable as the Young Engineers project has been.

15.2 Ralph Curwen

The EES Young Engineers scheme has been a highly challenging yet rewarding experience that has given me valuable insights in what it would be like if I were to pursue engineering at University. Initially, the project seemed daunting for me as the majority of it was based around electrical engineering, which is field that I did not have much prior knowledge in. Participating in this programme has meant that I have gained a stronger understanding of electronics and the coding necessary in a project like this. I have developed a basic knowledge how the Raspberry Pi and Arduino function but also how signals are transmitted, received and then processed. This means that if I were to tackle something like this in the future I would know what steps would be required, as I now know how to structure an electronics-based project, the components required and also how to create working circuits.

To begin with, my role within the team was researching, writing and designating tasks for the report whilst planning our project schedule with the Gantt chart. As the construction phase began I helped with some parts of the wiring and the layout of the components which was mainly during the Cambridge workshop. Also, participating in this project has helped my teamwork skills to develop, as I have learned how best to allocate roles and also how to communicate closely with other team members to ensure deadlines are reached. If I were to do the project again I would try and provide myself with a greater understanding of electronics and programming before beginning the project to ultimately save time and strengthen the team further. Furthermore, I would try to allocate roles closer to the start of the project so everyone knows what they have to do and the report and model could have been produced much faster. The most rewarding aspect to our project was producing a working model, and it is the satisfaction I gained from this that it is main reason why I would want to study engineering in the future and why I would recommend this scheme to anybody interested in a career in engineering.

15.3 Jeremy Zolnai-Lucas

The EDT Young Engineers programme has been an amazing experience. I have gained an incredible amount of knowledge in the field of electronic engineering, despite previously considering myself adept in this subject. Getting the budget to build something on a larger scale was an opportunity I have never had before, so the project was both daunting and enticing, and seeing it progress over the past six months has been very rewarding.

Having already had lots of experience with programming short scripts or small-scale programs, the challenge of creating something larger, with more than one code developer was a welcome one. I learnt about code management, and what it is like to write code as part of a team. On the physical side of things the scale, again, was a novelty for me. Getting used to the idea of not having a complete understanding of the entire system, only the parts that I was responsible for, was at first difficult but ultimately an important lesson in trusting teammates to do their jobs. In fact, the importance of delegation and clear roles roles in a team have become very apparent to me, as we found we made much more efficient progress once we assigned roles than when we worked in a round-robin style.

Though I already wanted to be an electrical engineer, this project has only made me surer about myself. It was, however, a wakeup call for the amount of work one must do documenting work whenever working as part of a team or in a corporate environment. The Cambridge workshop was a better insight into the world of engineering at Cambridge than any open day could possibly be, and I hope to be there again in two years.

15.4 Luke Ashford

The Young Engineers project has been a fantastic experience. I could definitely see myself pursuing a career in engineering having undertaken this project, and the knowledge and understanding gained of how to carry a project through from a brief description to a fully-operational prototype has been rewarding. In particular, the most important skill I have developed is time-management and working to deadlines. I had a tendency to leave things to the last minute but as part of this project I have seen the benefits of getting things done in good time.

My role in the project was focussed on the hardware, in particular designing and constructing the motor controllers. All of us helped to contribute to the general direction of the project and I am proud to look back and see how my ideas have either made it into the final product (for instance using screws and rails to distribute power), or have shaped the solution that was eventually decided upon (having a two-tiered chassis). If I were to go through the project again, I would try to be more involved with writing the software as well.

The project has been immensely enjoyable, but also informative. Through Leonardo, I have seen the many different types of varied careers available within the umbrella of 'engineer', such as performing R&D work in cutting-edge labs or working with a team of people to design a physical product. Though perhaps more questions have been raised than answers, this can only be a beneficial outcome as I feel more thoroughly informed about my career options. I would absolutely recommend the Young Engineers scheme to anyone interested in maths, science or engineering!

15.5 George Wright

Participating in the Young Engineers scheme has been highly rewarding and insightful into the world of engineering. Over the course of the scheme I feel I have gained an incredible practical experience on how development of a product occurs which I will benefit from later in life. It has strengthened me personally, especially my ability to work to deadlines and to work efficiently with a group, as previously I tended not work plan my work so struggled with extended projects. The Young Engineers project has forced me to develop this skill as it is critical to plan what each member is to do every week for the project, thus making me plan my work and set short deadlines.

I have found it very satisfying to help progress the project from a brainstorm once a brief was selected, into a complex prototype. Although at times the work was extremely challenging, resulting in many late nights of coding and debugging, it was satisfying to complete each problem and feel like a valuable member of the group. I did a fair amount of research for the project initially which I enjoyed as it was intriguing to learn about new topics that I had little knowledge in. After this phase in the project, I mainly focused on the programming tasks on the project as this is where I felt I had more knowledge. I hadn't worked much with hardware before so it was fascinating to work with the different components. It was enjoyable learning how to program the hardware, especially setting up the communications between the robot and Pi as it was a long and difficult task.

The engineering process was enjoyable and I can definitely see myself working in a team using electronics or computers in this way in the future, if not working with both and doing more robotics. I have enjoyed every part of the scheme and would highly recommend it to anyone looking into a career in any form of engineering.

15.6 Ethan Honey

The Young Engineers programme has been an incredibly informative and enjoyable experience. Going into this project, I had little prior knowledge in advanced electronics and coding, but now I have an insight into not only the theory behind these topics, but also their applications in the real world, and for that I am very grateful.

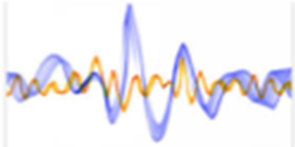
The work has been very wide ranging, allowing me to take on several interesting roles over the project lifecycle. Although this was a great challenge, I found it thoroughly enjoyable to be able to contribute to the project in so many different ways. I have conducted background research into existing IED detection systems; worked together with the rest of the team to analyse and make critical decisions about the direction of our project; assembled parts of the robot from electrical components and designed the layout of how all the parts fit together. The tasks at hand have required intense focus but have ultimately been very rewarding.

If I did the project again, I would make sure that the initial project planning is more detailed, so that everyone is allocated specific roles from the start and are all working towards a clear set goal. If this had happened I feel like I could have involved myself more with the project management process. I would also have proposed having agendas at project meetings to make sure they were as efficient and time-effective as possible.

Overall the project has allowed me to develop my technical knowledge as well as my teamwork skills. It also provided fantastic external opportunities; both the site visit to Leonardo and the workshop at the Department of Engineering at Cambridge University were brilliant ways to develop my passion for engineering, and have motivated and energised me to pursue engineering as a career in later life. I would strongly recommend Young Engineers to anyone with a passion for STEM subjects.

16 Appendix A: Project Brief

LEONARDO
AIRBORNE & SPACE SYSTEMS



4) Detection of a Signal

Much of the work done by Leonardo encompasses the detection of electromagnetic energy in the form of a signal. This is important and has many applications including defence, security and communications. In the modern world, phone signals and radio signals are just part of everyday life. The following briefs outline some current issues with detecting specific signals.

Mobile Detection

The latest threat to information dependent businesses is the mobile phone. They are the ideal device for stealing data, capable of storage and transmission of large quantities. Within secure organisations handling secure information, mobile phones can be a huge threat as they make it easy to leak or steal important data. For this reason, secure facilities often have rules on mobile phone usage and possession. Currently, there aren't many devices on the market capable of accurately and reliably detecting a mobile phone in the area. Many of them claim to be able to detect signals transmitted by phones but fail when tested by independent users. Your task is to create a working prototype which can detect the presence of a mobile phone in a building.

The prototype should be able to:

- ✓ Detect a mobile phone in the same room
- ✓ Inform the user if a phone is found
- ✓ Work for any mobile phone network
- ✓ Operate effectively even if the phone in on standby (not making a call)

Counter IED

On the front line, IEDs have become a serious threat to personnel and civilians alike. These devices can be detonated in a variety of ways, one popular method being by mobile phone. In suspicious situations where a bomb may be present, it would be helpful for personnel to be able to detect a mobile phone in the area and locate the potential threat. Currently, reliable technology does not exist for this.

You have been tasked with producing a working prototype.

The prototype should be able to:

- ✓ Search and detect for a mobile phone
- ✓ Locate the approximate location of the device
- ✓ Work for any mobile network

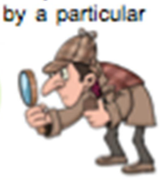
Ofcom Issues

Ofcom

Ofcom are the organisation which regulate communications in the UK. Recently, there have been claims that mobile network providers are operating outside their agreed frequencies – which is against Ofcom regulations. Each network has its own frequency band which it must operate within.
Ofcom wish to research these claims and check if all network providers are sticking to their allocated frequency ranges. Your team has been given the task of devising a way to detect the frequency at which network providers are operating at. This can be done by analysing frequencies transmitted by a particular device or a network's base station.

The system should be able to:

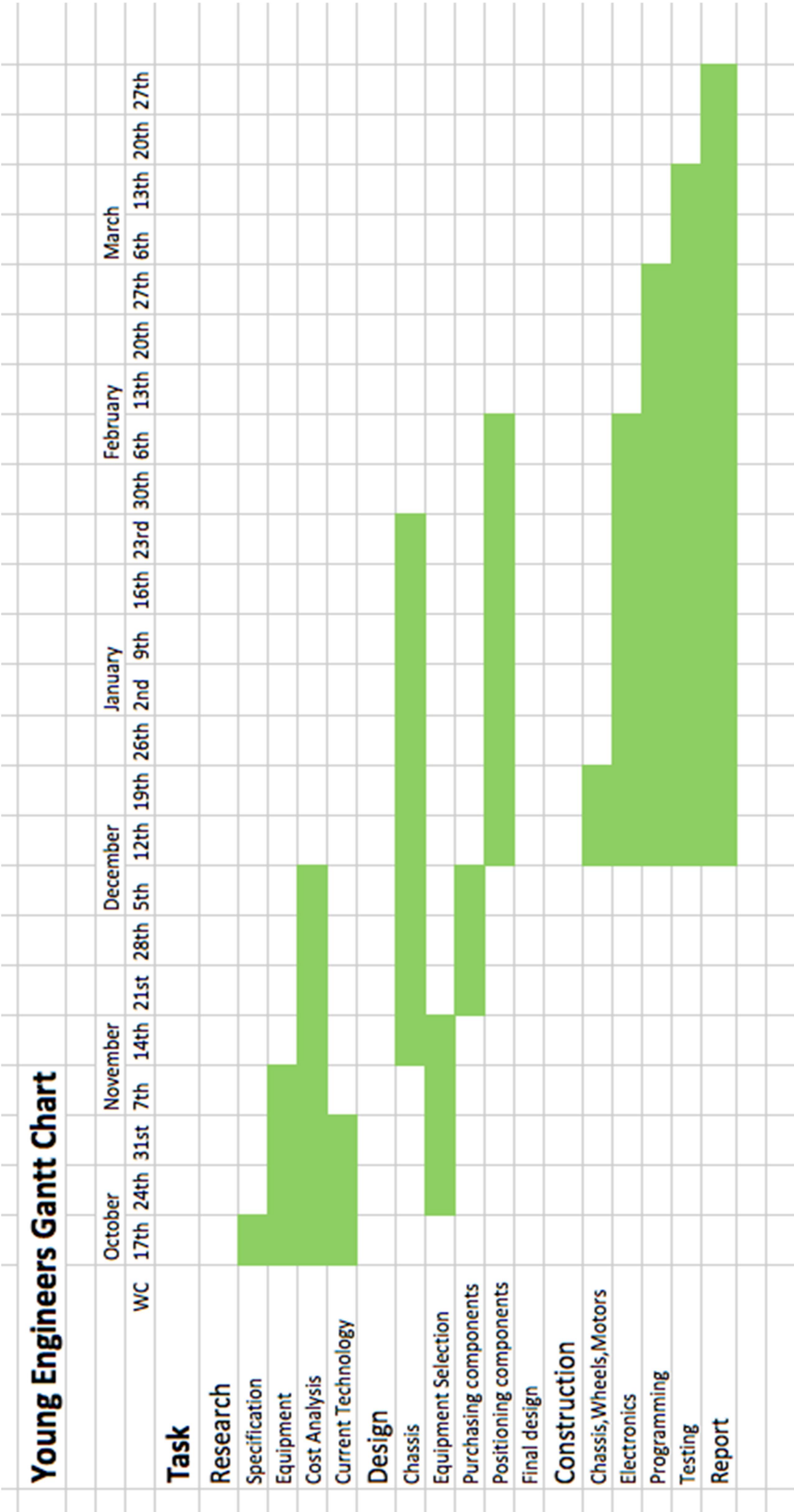
- ✓ Analyse the frequency of radio waves used by mobile networks
- ✓ Accurately give a value of the frequency band of particular waves
- ✓ Be easy to use for data analysis



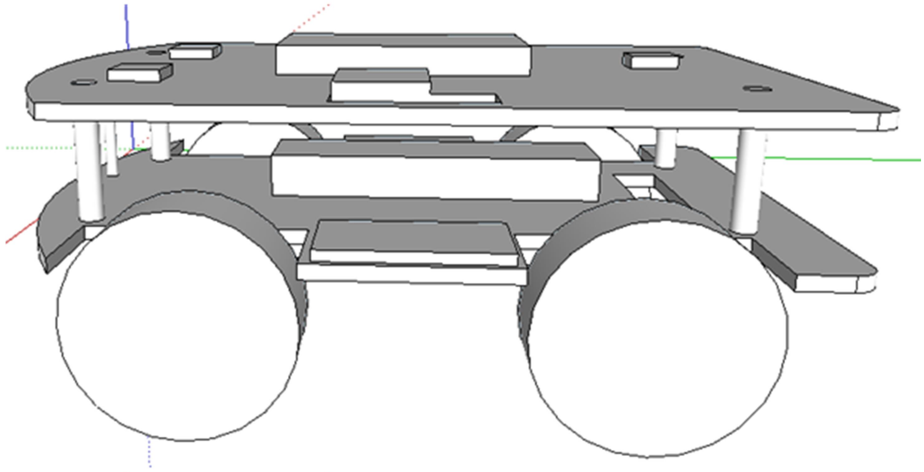
Issue 1

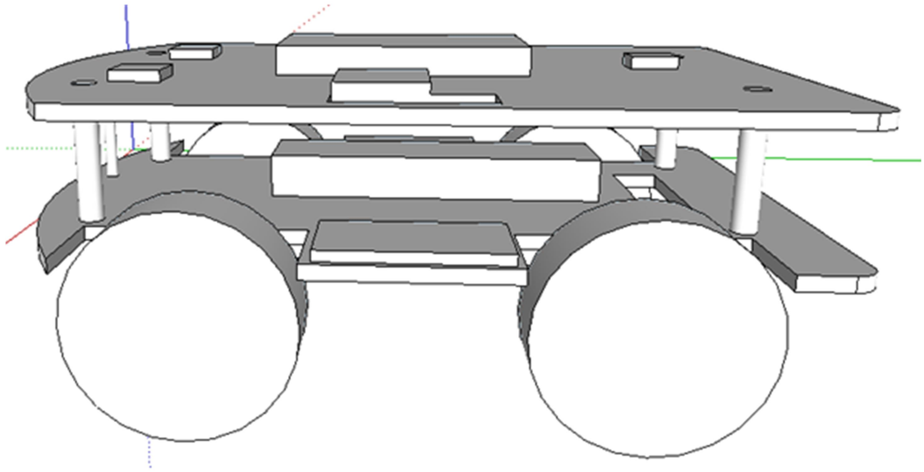
Page 5 of 5

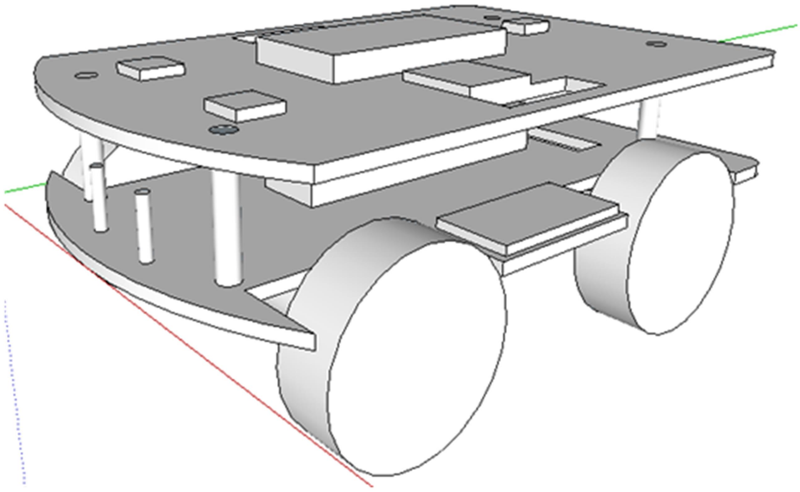
17 Appendix B: Gantt Chart



18 Appendix C: 3D CAD Model







19 Appendix D: Photos