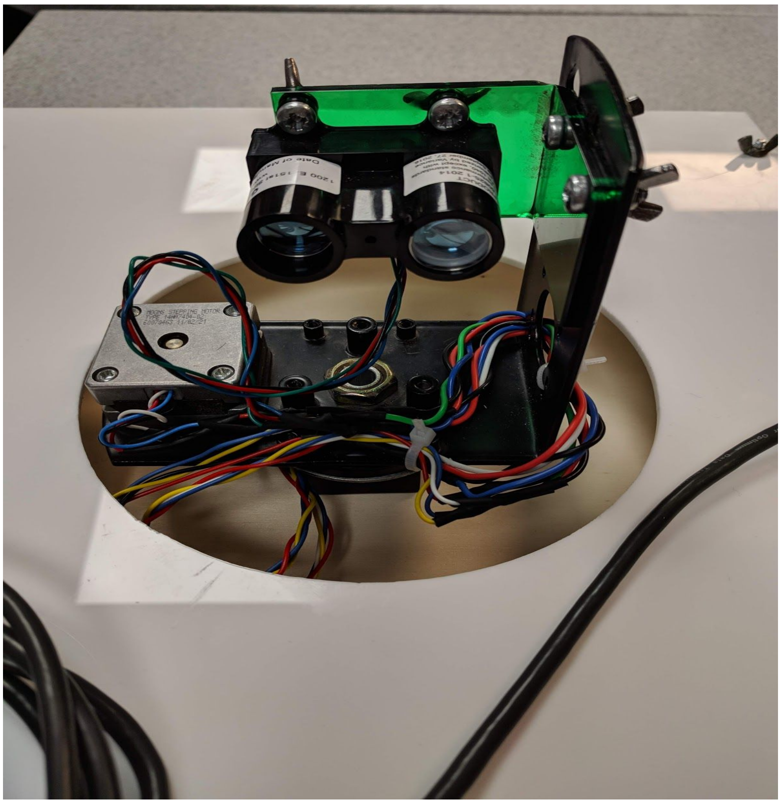
Design and Implementation of a RADAR like system



Merchiston Castle School Engineering Education Scheme March 2019

Contents

Acknowledgments **4**

Abstract **5**

Introduction **6**

Aim **6**

Team Members **7**
- Nikos 7
- Brodan 7
- Hector 7
- Han 7
- Robert 7

Leonardo **8**
- As a Company 8
- As Partners 8

Project Launch Day **9**

Principles of Radar **10**

Initial Ideas/Brainstorming **10**

Choosing our type of Radar **11**
- Option 1 - Repurposing of a "laser tape measure" 11
- Option 2 - LIDAR 11
- Option 3 - Sonar 12
- Our Decision 12

Stepper Motor versus Servo **13**
- Stepper Motor 13
- Servo 14
- Our Decision 14

Electronics **15**
- Lidar Lite V3 15
- Arduino and Motor Shield 16
- Prototyping 16
- PCB Progression 16
- Circuit Board 17
- Electronics Integration 18

Software **19**
- Code timeline: 19

Construction and Design **20**
Bracket 20
Casing Base 20
Side plates 21
Top plate 21
Assembly 22
Casing Material 22
Completed Case and Hardware 23
Testing **24**
Test 1 - Lidar 24
Test 2 - Lidar and Stepper Motor 24
Test 3 - Display 24
Test 4 - Scanning a Room 25
Test 5 - Outdoors 25
Glasgow Caledonian University Workshop Visit **26**
Workshop 1: Audio Engineering 26
Workshop 2: Digital and Robotics; Programming robots 26
Workshop 3: Mechanical Structural Integrity and Testing 26
Project Management **27**
Future Development Ideas **28**
Phased array system Radar 28
Synthetic Aperture Radar Systems 28
MIMO Radar 28
Conclusion **29**
Team Statements **30**
Nikos 30
Brodan 30
Han 30
Robert 31
Hector 31
Bibliography **32**
Glossary **33**
Appendices **34**
Appendix 1 - Leonardo project brief 34
Appendix 2 - Code 37
Appendix 3 - Minutes (Week 1) 58

Acknowledgments

First and foremost, we would like to thank Leonardo, and in particular, Sean Crawford and Kieran McGlynn, for their consistent support towards our project. The idea for this intriguing project came from them and Leonardo, so it is impossible to not think that without them, we would have never been able to try and complete this project.

In addition, we would like to thank Mr. Nicholls for all his continued support, and always making sure we were focused on our project.

Abstract

On the 16th November we visited the Glasgow Caledonian university for the EES launch day. We met with our sponsors, Leonardo who gave us our project aim. We discussed what type of radar would be most suitable for our use as well as other specifications of the radar we would build.

On the day we were given a problem solving task which was to suggest an “adaptation to a phone which would be beneficial to hillwalkers”. After a group discussion we came up with the adaptation that would program a route for hillwalkers before their expedition and then if they leave the route a warning system will buzz on their phone which only requires GPS and no mobile data. We were then required to speak in front of the other 9 schools doing the EES project. These tasks started to develop our teamwork and presentation skills

We quickly identified that the radar project came under four categories which were the construction of the housing and the radar, the electronics, the programming and finally writing the project report. Some categories overlap such as the programming and the electronics. Due to our team and each one of our individual talents we split the tasks between the group.

For the radar itself we built an infrared 360° Lidar with an acrylic mounting, using an Arduino to control the radar unit and interface to the laptop. We rotated the Lidar using a stepper motor. We displayed the data via USB to a laptop which showed a classic radar display. This required a lot of work to write the code from scratch, construct the electronics, design and implement the display and build the housing for the box. The biggest technical challenge was the programming so we could display the information given by the Lidar. The biggest team challenge was time management and ensuring that each member of the team was able to fit to the schedule.

The project outcome was a radar unit that worked as a USB device and interacted with a python program on a laptop. The radar unit was portable and could be quickly set up to scan a room or outdoor area. The display provided information about range to perimeter. The project fulfilled the criteria that we had been set by the engineers from Leonardo in all respects.

Introduction

We are 5 boys from Merchiston Castle School who are studying STEM subjects in the Lower Sixth, working in a partnership with Leonardo, a United Kingdom and Italy based engineering company. The Engineering Education Scheme provided us with an exciting and engaging opportunity to create something that we have never created before, but also to work as a team and simulate what working for a real engineering firm may be like.

The brief we received from our mentors was simple: build a radar. The purpose of it was to be of our own idea, thus giving us the opportunity to do what we would like with it. After intense deliberation, we decided our radar would be used for Building Surveying. We wanted it to be relatively small, simple to use and easy to understand, yet still creative. Helping us with our project, we had two engineers from Leonardo, Sean Crawford and Kieran McGlynn, along with Mr. Nicholls (Head of Science and Technology) from our School, Merchiston.

Robert (back), Brodan, Nikos, Hector, Han

Aim

Design and implement a radar like system which can scan the area around itself, identify nearby objects, determine their location relative to the radar and display the information in an appropriate way to the user.

Team Members

Nikos

Nikos is currently studying Maths, Physics, Chemistry and History. Nikos is our secretary and is responsible for setting the deadlines, writing the project report, taking minutes, and making sure everyone is on track.

Brodan

Brodan is studying Maths, Physics, Design Engineering and Computer Science. Brodan is the only one taking Computer Science at A-Level so he is the one who is in charge of any programming for the microcontroller used. This involved the manipulation of the data collected by the radar in order to to display what was desired. Brodan also has a further interest in electronics.

Hector

Hector is currently studying Maths, Further Maths, Physics and Chemistry. His research looked at the physics behind radar, and compared all the different options that we could use. He worked on the presentation as well as the project report and was involved with developing the display. Hector is planning to study natural sciences at university.

Han

Han studies Maths, Further Maths, Chemistry and Physics. As Han is a very good mathematician and he is responsible for many of the calculations to do with the data received by the radar. He hasn’t had lots of experience with engineering but he was interested in manufacturing so he also took part in the making of the casing for the Radar as well as coordinating the presentation. Han wants to study engineering in the future.

Robert

Robert is taking Maths, Physics, Design Engineering and Chemistry. Robert is very good with his hands, so he was delegated the job of actually making the radar, and thinking what design would work best. As Robert took Design Technology at GCSE level, he is used to using design software, so any work done in that domain was done by him.

Leonardo

As a Company

The company Leonardo is a United Kingdom and Italy based High-tech engineering company. It occupies a role as one of the biggest players in aerospace, defence and security. In January 2013, it was the biggest inward investor into the UK defence sector on top of producing exports that equated to about £1.3bn annually. The Edinburgh site develops and provides a new generation radar (CAPTOR-E radar, multi-mode Doppler pulse radar) for the RAF's Typhoon combat aircraft.

Leonardo Headquarters
Edinburgh

As Partners

Since Leonardo is primarily a defence company, specialising in early defense systems, giving us the task to build a radar only seems fitting. As the purpose of the radar was up to us, this allowed us to focus on a potentially wider array of uses than simply for example having it as an educational display.

CAPTOR-E Radar
made by Leonardo

Project Launch Day

On the 16 of October 2018 the team and Mr. Nicholls travelled to the Glasgow Caledonian University to attend the EES Project launch day. The day started off with the registration of the team and then a welcome address and an introduction to the EES by Dr. Tuleen Boutaleb. The school was joined by six other schools who had either one or two teams. After the introduction the team met our mentors from Leonardo.

The schools where then given a challenge to design “adaptation to a phone which would be beneficial to hillwalkers” the team and the mentors then went away and discussed an idea. After a group discussion the team came up with the adaptation that would program a route for hillwalkers before their expedition and then if they leave the route a warning system will cause their phone to buzz this idea would only requires GPS and no mobile data. After this design process we talked about our design in front of the other groups.

After lunch the team met with the mentors who gave the team the briefing on our project which was to “design and implement a radar system”. The mentors told the group about Leonardo and the history of the company and then with the group discussed initial ideas for the project.

We thank Mr Nicholls for accompanying us to glasgow and to the EES for such a Brilliant day. We also thank our mentors for making the the journey to Glasgow and for their warm introduction.

Principles of Radar

The word RADAR is an acronym for RAdio Detection And Ranging, and in its simplest form it consists of a transmitted radio signal aimed by an antenna in a particular direction, and a receiver that detects the “echoes” off any objects in the path of the signal.

The transmitter consists of an electronic circuit that oscillates at a specific frequency, usually much higher than those frequencies used for radio or TV broadcasts.

This signal is sent out in short bursts of electromagnetic energy, called pulses, through the antenna which produces a narrow beam like that of a torch.

An Antenna Dish

The distance to the target is determined from the time taken between transmitting the pulse and receiving the echo. This can be accurately determined because the radar signal travels at the speed of light, which is constant.

Initial Ideas/Brainstorming

Immediately after the project launch day, the group set out to discuss ideas for our radar. Ideas were passed around freely and no idea was dismissed. The ideas included using sound and ultrasound, attempting to build our own microwave radar system, building our own optical system and using radio. The group were keen to have a traditional style display with several members of the group excited at the prospect of learning how to use a Raspberry Pi. The ideas to be taken forward were assigned to team members to research.

Choosing our type of Radar

Early on in the project we decided to impose the following requirements on the radar:

1. It has to be able to take multiple readings per second
2. It must have a range greater than 20m

Although these specifications look simple they are in fact rather difficult to achieve as range finding apparatus is usually either expensive or in has only one or the other of the two desired attributes. As the purpose of our Radar was for Building Surveying, we decided that a decent range was important, but also one that would allow us to to make multiple readings in order to create a clearer image of a room.

Option 1 - Repurposing of a "laser tape measure"

This method initially looked rather appealing, considering the fact that some of these products have effective ranges up to 100m and are usually within the range of about 20-40 pounds. However, on further inspection these devices cannot take the required number of readings per second without sacrificing the minimum required range, as a high end variant (around £60) can only take around 3 readings per second. At first, this may sound fine but this would take a rather long time to make one rotation, and having a faster rotation is better. Unfortunately, there is going to be a trade off between resolution and rotational speed. Therefore the faster we can take measurements the less of a tradeoff we have to make.

Option 2 - LIDAR

Lidar is a surveying method that measures the distance to an object by illuminating the object with pulsed laser light. It then measures the reflected pulses with a sensor and therefore determines the distance to the object. The differences in the laser return times and the wavelengths can be used to make a digital representation of where the target is. Lidar stands for (light, detection and ranging)

In our opinion this is the far superior option as it allows for a roughly 40 metre range allowing for a large detection radius. This option also features easier integration with other systems such as the Arduino board. However, this option is on the more expensive side, yet still well within the budget, and as opposed to the laser tape measure option there is the ability to take multiple hundreds of readings per second allowing for a much greater resolution than the other option.

Option 3 - Sonar

Sonar works by emitting pulses of sounds and listening for echoes to recognize them as targets. Sonar is many used underwater for submarine navigation. Because sound travels at 340 m/s it is very easy to do a simple distance = speed * time calculation to work out how far away the target is.

From one perspective, Sonar seems like a good option since the equipment is relatively inexpensive. But the simple fact of the matter is that the range is miniscule compared to the others and the effects of background noise can not be understated as perhaps even being in a city could render the device useless. Setting this aside there is the fact that there is more that could be developed on this ourselves as the distances can be ascertained using speed distance and time and the time is relatively easy to measure.

Our Decision

In conclusion, we believe that the laser tape measure method although interesting doesn't work for our criteria. This is due to the small range and also since the budget we were given was sufficient to cover the more expensive options, the decreased range and resolution just simply wasn't worth it. Coupled with the fact that the configuration of the Lidar option will be a lot easier due to the fact that there is actually documentation on it and we are using the device for what it is made for allows for quicker manufacture and therefore more time to iterate the design. Essentially, we decided to use Lidar as firstly, what we wanted to look at (Building Surveying) had the perfect range for it with a radius of 43m. Using infra-red as the wavelength meant it was small enough to reflect of an object we were looking for such as a human. Essentially, the LIDAR method suited our aim the best.

Stepper Motor versus Servo

The members of the team responsible for the mechanism researched two different systems using either stepper motors or servos to provide a 360 degree rotation.

Stepper Motor

Stepper motors are DC motors that turn with discrete steps. The motor is made up of multiple coils which are grouped into what are called phases. By passing electricity into these different phases will cause the motor to turn. With a computer-controlled stepper motor you could achieve very precise position and speed control. The advantages of stepper motors are their precision at positioning as the stepper motor can move in repeatable steps this allows them to excel in applications requiring precise positioning this precision also allows for the precise control over speed. However, they have a low efficiency level as the drawing or current from power sources is independent of load so it will continue to be powered when it is not in use. Step counts can be in the region between 4 and 400 the most commonly available motor steps are 28 48 and 400. A way to achieve high positioning resolution is gearing adding gears can increase the steps of a motor.

One of the only inconveniences that goes along with this motor is the power consumption. It requires a constant DC 12v which in itself is not really a problem, it's the fact that the project needs to be portable and can carry 12v batteries. The batteries that have a long enough life and can be recharged are rather large meaning that a decision need to be made to either connect the sensor and motor through a swivel connector or have a scan pattern that goes back and forwards instead of a constant rotation. Our design’s favour the idea of the swivel connector but with the concern that we don't actually know how many steps are in a complete rotation the idea of going back and forward is quite nice.

Servo

Servos are used in many products as they are a small size but are powerful and are very energy efficient.

Servos are made up of a DC motor and a control circuit. The motor is attached by gears to the control wheel. It also contains a potentiometer, a three-terminal resistor that has a rotating contact that forms and adjustable voltage divider. As the motor rotates the potentiometer resistance changes so that the control circuit can regulate how much the motor has moved by. When the motor is near the intend position the power to the motor is stopped. The desired position of the motor is sent via electrical pulses through the signal wire. The motors speed is proportional to the actual position and the desired position, so if it is close it will turn slower and if it is far from the intended position it will turn faster. This means that the motor will run as hard as necessary increasing energy-efficiency.

Servos are controlled by electrical impulses with varying width through the control wire. There is a maximum and a minimum pulse and a repetition rate. A servo can normally only rotate 90 degrees in each direction amounting is a total of 180-degree movement. The neutral position is defined by the point where it has an equal distance from both full rotations. However, some can fully rotate these are called full rotation servos or 360 servos. They look like a normal servo, but its driver shaft allows for it to fully rotate however with the consequence of less precise control over precision. The full rotation servos have a driving shaft that can spin continuously with control over its speed and direction of spin.

Our Decision

We decided to use a stepper motor. We were able to acquired a stepper motor from the school workshop which appears to work nearly perfectly although there are slight problems with the calibration. However, it was satisfying to recycle a piece of apparatus. The motor itself has an interesting number of steps in a full rotation and as such we haven't been able to accurately work out how many steps there are. It seems to be a lot which is good as many stepper motors have only around 200 steps which lets them move in 1.8 degree steps, in comparison the one we have found seems to have no less than 400 steps meaning an individual step is not likely to be more than 0.6 degrees, this is very good as it gives us a higher resolution.

Electronics

Aim of the Electronics: To control the movement and operation of the radar (rotation of the sensor and distance measurements)

From the start we were not happy with the Arduino approach to the project, it simply didn't offer enough control over the Lidar, we wanted to be able to change the behaviour mid-run and this simply wasn't a possibility on the Arduino due to problems we were having in the serial communication between the Arduino and computer. We could get data perfectly well off the Arduino but just not into it.

Due to the problems with the Arduino we decided the best decision was to switch to a Raspberry Pi, we would still have the Lidar sensor controlled from the Arduino and have that set to be constantly sending data back to the Pi over USB. The motor would then be controlled with the GPIO pins of the Pi and a motor driver was built.

The problem was that although the Arduino was perfectly capable of sending back what step the motor would have been on (we were still using the old drivers) and the direction of the turn, the data concerning the distance measurement wasn't coming through, it would take one data point then fluctuate around it seemingly at random, and for this reason we decided to move the direct communication with the sensor to the Pi. This proved harder to do than initially expected, with the Raspberry Pi unable to detect the light sensor. However the problem may be down to the nature of the Pi we had been using, for the Pi that we were using had multiple dead pins, what this is down to we don't know, but it does seem logical that one or both of the I2C pins may be completely dead, and therefore the device undetectable by them. Thankfully the new Raspberry Pi appeared to be working completely therefore allowing us to continue working on putting the two systems into unison.

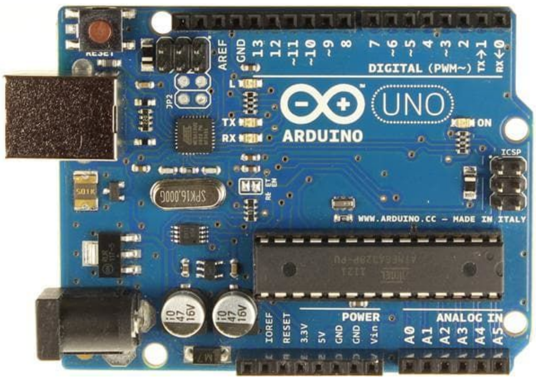
After a lot of work to develop first the Arduino and then the Raspberry Pi, the decision was taken to revert to the Arduino and make the whole radar unit into a USB device which could simply be plugged in to a laptop. This was because the drivers for the Lidar unit and the Raspberry Pi remained incompatible despite weeks of effort and research.

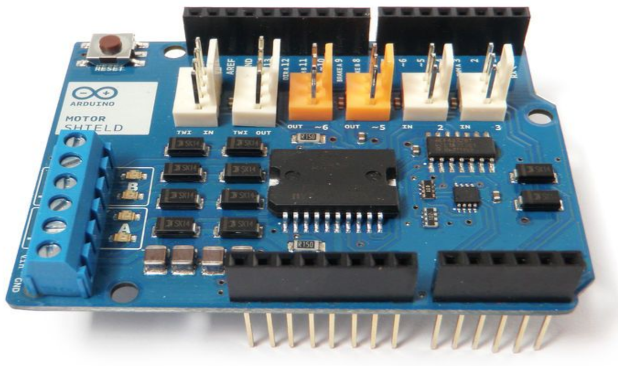
Lidar Lite V3

The Lidar Lite V3 is the model of sensor that we used for our radar. This device measures distance by calculating the time delay between the transmission of a near-infrared laser signal and its reception after reflecting off of a target. This translates into distance using the known speed of light. All of the timing is completed on board and the Lidar unit uses the I2C protocol for communication.

Arduino and Motor Shield

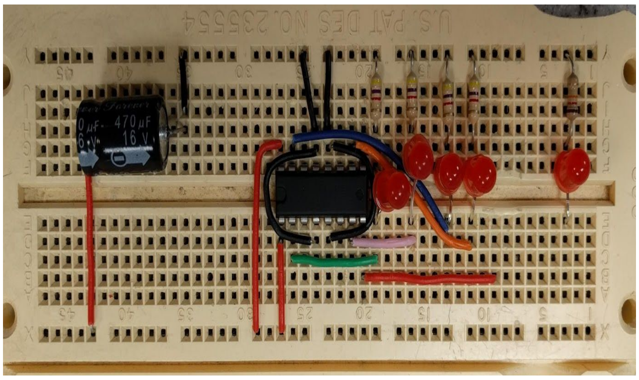
The Arduino chosen was the Arduino Uno as it was the cheapest and most readily available device. The Arduino motor shield was trialled and was used to test and understand the stepper motor. The motor shield was seen as excessive and a more basic motor driver was developed from scratch.





Prototyping

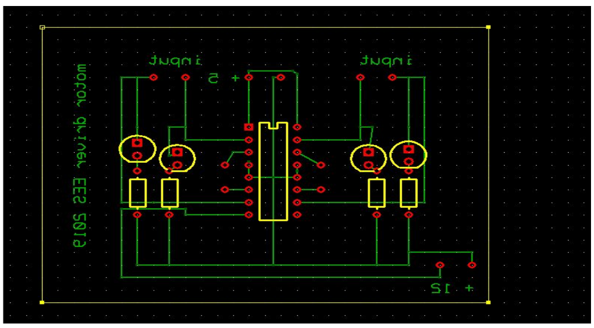
The initial prototyping stage was done on a breadboard. We decided to switch to using our own motor driver from the Arduino one as we found that the Arduino motor shield is not behaving in the manner that we wanted it to and thus switched to the custom built motor shield.



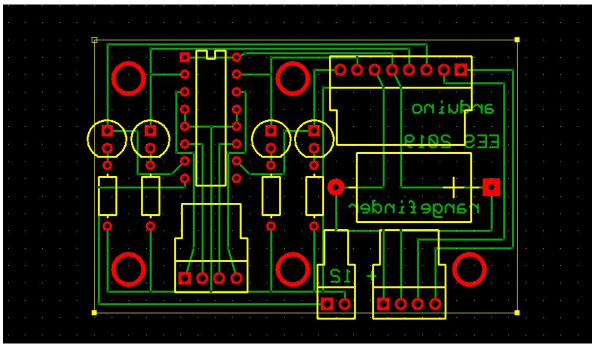
The motor driver was based on the readily available L293D IC with LEDs used to show the input data and a smoothing capacitor for the Lidar supply. Once the circuit was working, which was satisfyingly easy, the design was transferred to PCB.

PCB Progression

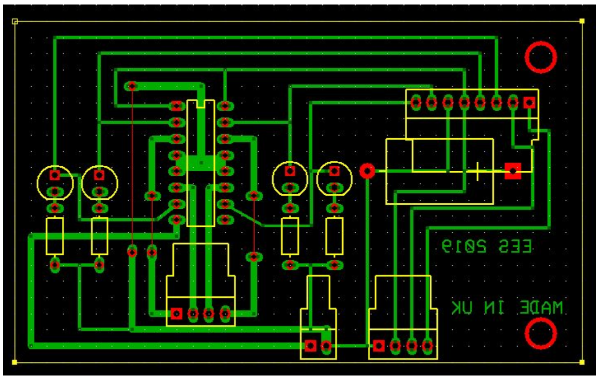
This initial iteration of the circuit board contained everything necessary for the motor driver with one important exception being a heatsink pad on the bottom for the ground pins. A trial etch also showed the tracks to be too narrow.



This next iteration of the PCB contained the circuitry required for the lidar integrated onto the PCB but still did not have the required traces for the 12v lines or the heat sink, it was also too small and the etching tank available didn't have the resolution necessary.



The final iteration of the design featured larger rails allowing for the resolution of the tank. It also has thicker rails for the 12v lines and a heatsink pad on the bottom for the driver.

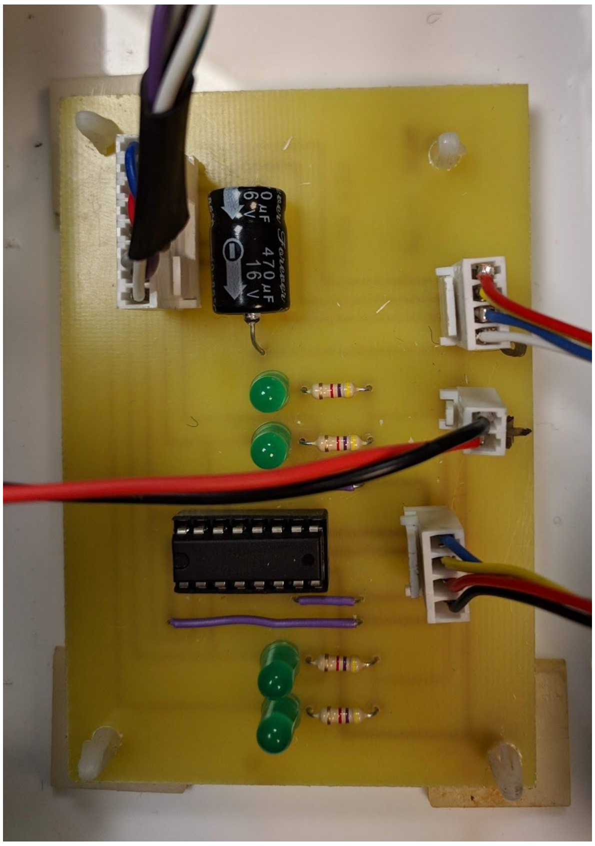


Circuit Board

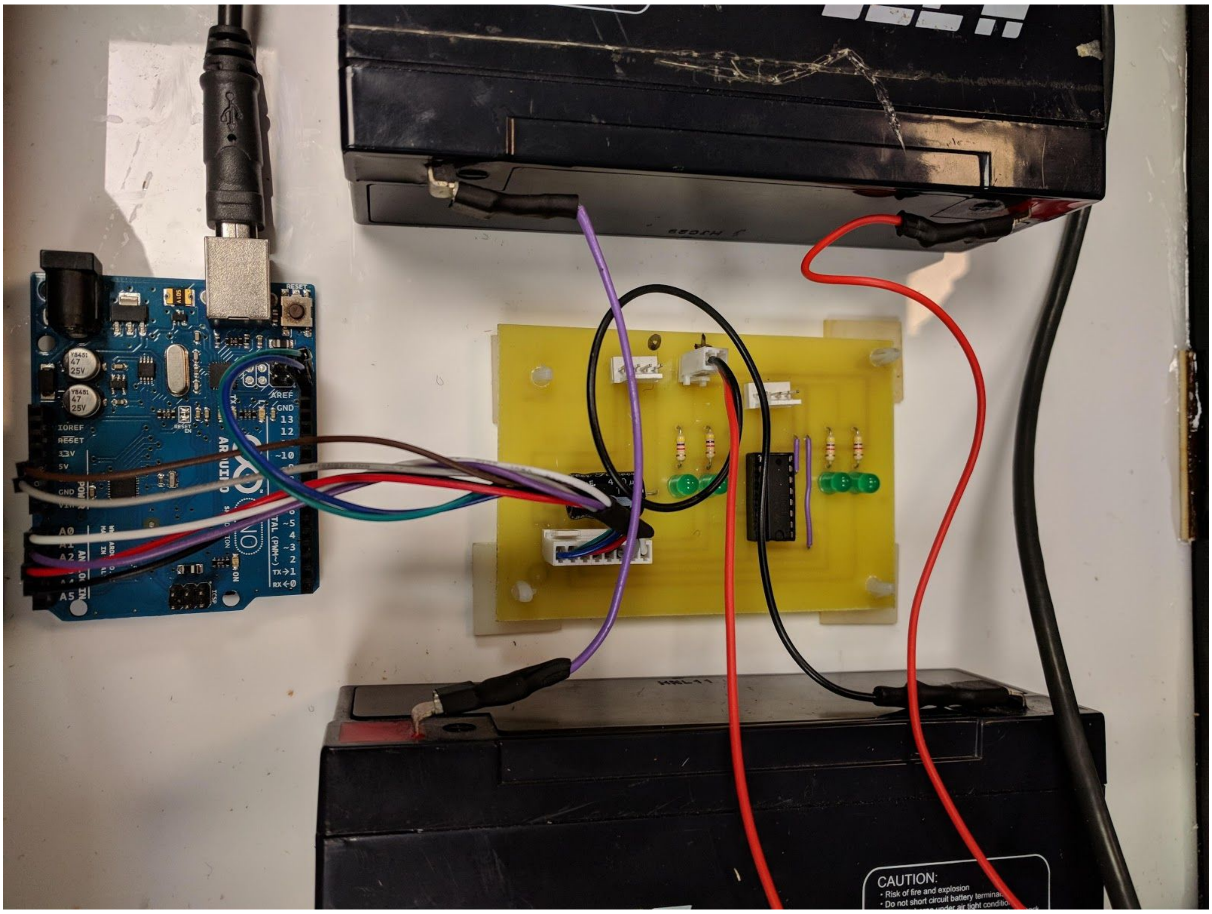
The finished circuit board was etched, drilled and tested. There was a break across four tracks due to an etching error but these were fixed with solder.

The assembled circuit board was tested and worked perfectly and as expected.

The wiring the the Lidar, motor, Arduino and 12V power supply were all completed using soldered MOLEX connectors for convenience and reliability.



Electronics Integration



The two 6V lead acid battery packs (recycled from a previous project) were mounted with velcro to the acrylic base.

The PCB was mounted using PCB mounts directly beneath the centre of the Lidar system so that the connections could be easily established.

The Arduino was mounted with velcro and the MOLEX wiring loom connected.

The USB cable was lead out of the box and secured against being pulled out.

Software

The development of the software took by far the longest of any of the aspects of the project. This is of course factoring in the various problems that occured with the use of a Raspberry Pi although the software that is now being used has gone through multiple iterations up to the point where it is. The first versions of the computer side code did not include any of the graphical capacity of the present iteration although the majority of the code was written before that point as the interface code for serial communication is rather long in comparison to the display code.

All of the code was written for this project and made use of the various libraries available. The code developed was, however, specific to this project.

Code timeline:

1. Arduino code written to control stepper motor. This was a relatively straightforward task and worked with few problems
2. Arduino code written for Lidar. The available libraries made this a relatively easy task
3. Python code developed to read data from Arduino
4. Raspberry Pi interfaced over USB to Arduino - worked as expected
5. Lidar accessed directly by Raspberry Pi. Version issues with the Rapberian operating system caused a lot of difficulties interfacing with the Lidar unit. This was abandoned eventually.
6. Arduino code written to run stepper motor and Lidar unit and return data to python console
7. Python code developed to use Lidar data to draw a green on black radar trace

The Arduino side of the code was a lot more simple than the computer side code as all it has to do is transfer the data from the light sensor to the computer and iterate the angle of the detector. Due to this it has remained much the same for the duration of the project, although the sequence for the stepper motor was changed at one point due to the inefficiency of a previous iteration.

The display code was written once the main interface code was reliably returning values over USB of position and distance.

The code is listed in Appendix 2

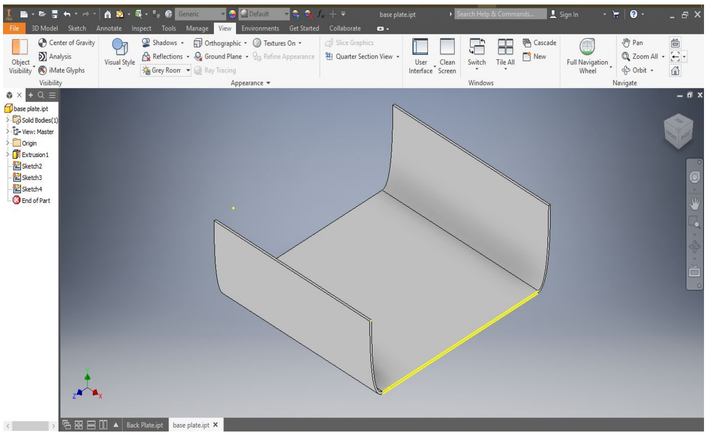
Construction and Design

Bracket


Production of Bracket

A laser cutter was used to cut out the template required for the bracket. We used a design that was created on a software called Pro Inventor, which is a type of CAD package. A strip heater was used to bend the bracket to the required shape. A wooden block cut by a band saw at a right angle was used as a forming block so that the bracket would be at a right angle. A marker pen and a steel rule where used to measure and mark where the bend was going to happen nuts and bolts were used to hold the bracket to the Stepper motor. Another set of nuts and bolts were used to hold the Lidar lite V3 to the bracket which connected it to the servo motor.

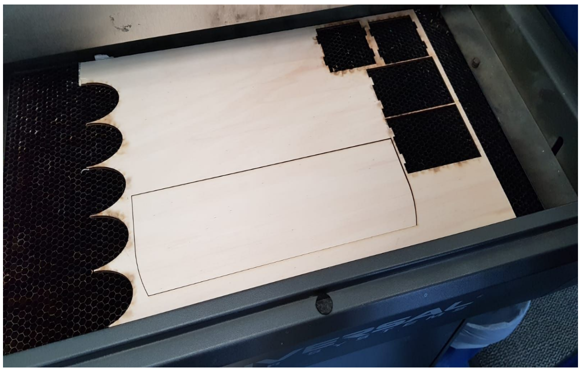
Casing Base


The image above is the original design of the base for our radar made using a CAD

The size of the plastic that was on offer needed to be cut to a smaller size to fit into the laser cutter. A bandsaw was used to cut the acrylic sheet into the required size which was 500x300x3mm The sheet was then placed into the laser cutter to cut out the holes required for the brackets to attach the top and the base plate to the casing. After the cutting was finished the sheet of acrylic was marked 100 mm away from the outward edge using a marker pen. A forming block was produced form polystyrene; it was cut out with a band saw and sanded down using a belt sander to flatten down the edges. The original idea was changed from a rounded edge to a more square edge, this was done as the rounded edge could not be produced to a high standard with the equipment that was available in our Design department. The acrylic was then placed into a strip heater where it was bent at the marker line that was 100mm away from the edge. It was bent using a forming block to an angle of 117 degrees. It was then marked again 50mm away for the edge and was bent again using the same forming block.

Side plates

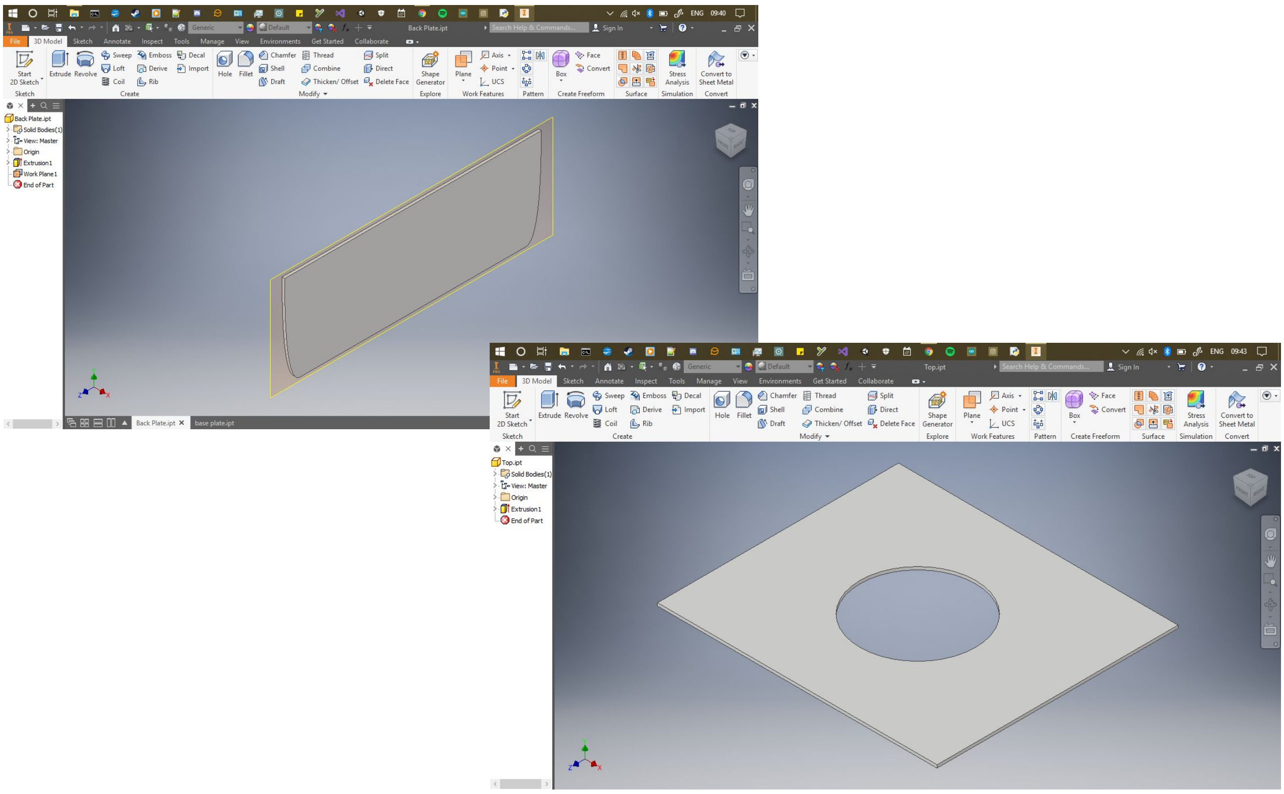
The template was produced in Inventor. The laser cutter was used to cut out the side plates which were designed with the specifications of the base. The manufacturing team had problems with producing the side plates with the right dimensions. The first prototype was too tall and not wide enough so the team redesigned it smaller and wider which lead to the final prototype which was used as the final side pieces for the project.



Production of the side plates for the casing

Top plate

The top piece was produced using the laser cutter to cut out the required piece. The original top was produced without the side plates but when the side plates where glued in the sides of the base where pushed in making the original base unable to fit. To fix this we redrilled the holes that were used to hold the top onto the base plate which then allowed for the top to fit on to the baseplates and the brackets used to assemble the whole of the base together.



Assembly

Brackets with welded bolt attached

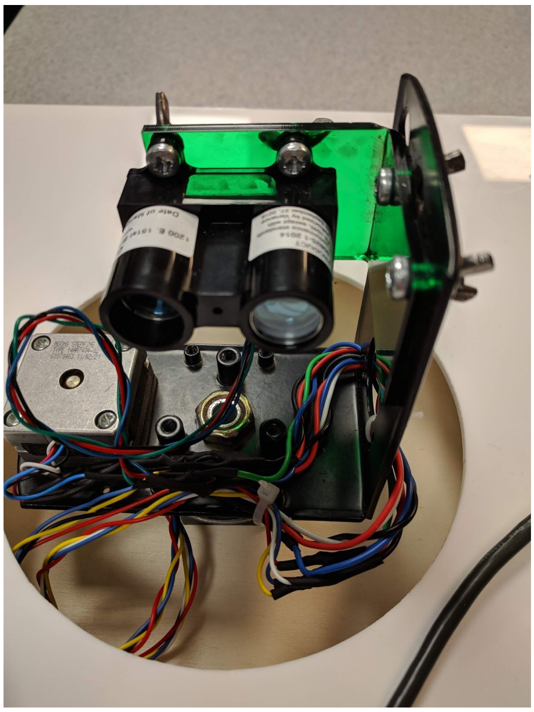
The final step in the manufacturing process was assembling the base together. The first step in assembling was producing the brackets to hold the baseplate to the top and also to hold the board used to mount the radar onto the case. The problem arose that the bolts would have to be held in place or attached to the brackets however with the sides being glued in their would be no access to the bottom of the bolts so they would just fall out when assembled. To solve this the team welded the bolts onto the brackets to hold them permanently in place and to stop the bolts from falling out when in use. The top and the base plate had holes of 4.5mm drilled into them using a pillar drill. The two parts where attached to the each other by using the brackets and bolts to hold the the two parts together. The side plates were glued on at each end of the baseplate using Araldite the side plates where held in by clamps and the forming blocks used to mold the base where used as wedges to hold the sides into the proper place.

Casing Material

The purpose of using our radar is for Building Surveying and acrylic is the most suitable plastic for the casing in terms of what we had available to us.

There are few reasons as to why acrylic is the most applicable:

1. It has excellent weatherability and resistance to sunlight (UV light). The radar will be used in outdoor so it will be much better if it is capable to withstand the weathering process and the sunlight.
2. It is rigid, with good impact strength, but also malleable. We had to bend the plastic on the sides so we found a malleable plastic. Acrylic can be easily cut and morphed to the shape that we need.
3. It is cheap compared to other plastic with similar properties.



Completed Case and Hardware

The physical side of the radar was completed on schedule

The image shows the Lidar unit and stepper motor mounted in the acrylic case

Testing

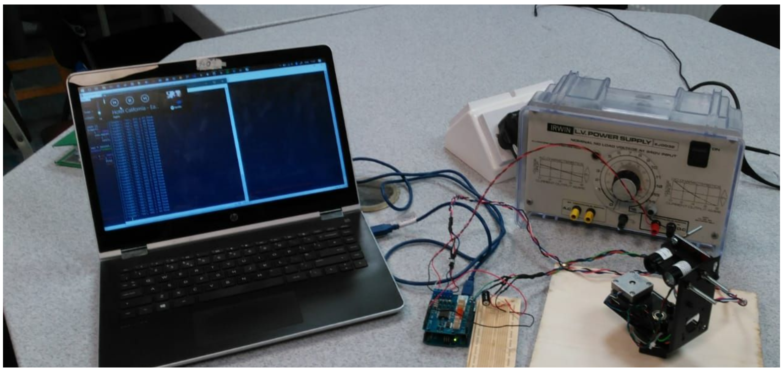
The testing process has been rather long in comparison to the time expected on the planning Gantt chart.

Various problems from tracks being in the wrong place on the PCB to the I2C bus being the wrong way round more than once has lead to the vast majority of the project simply being fixing problems that arose.

This was most prevalent in the transfer to the Raspberry Pi where at least a week was spent on the attempted transfer which caused such problems as the I2C bus not working to the serial communication via USB not returning a specific incredibly necessary value.

Test 1 - Lidar

The picture shows the Lidar system being tested via the serial monitor. The output shows a repeating pattern of distance values.



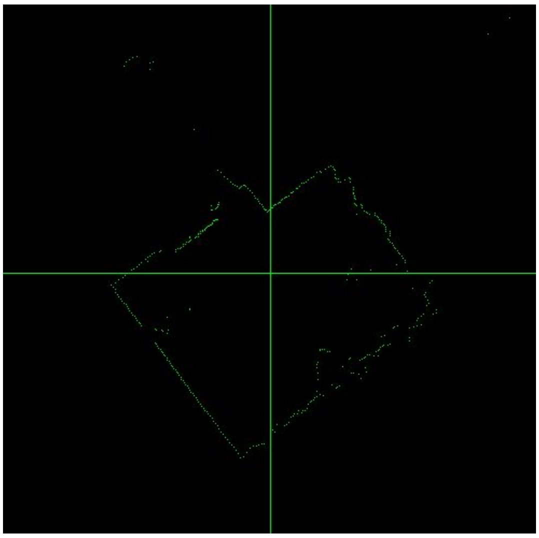
Test 2 - Lidar and Stepper Motor

The Lidar and stepper motor were tested together and provided data in a format necessary to draw the display.

Test 3 - Display

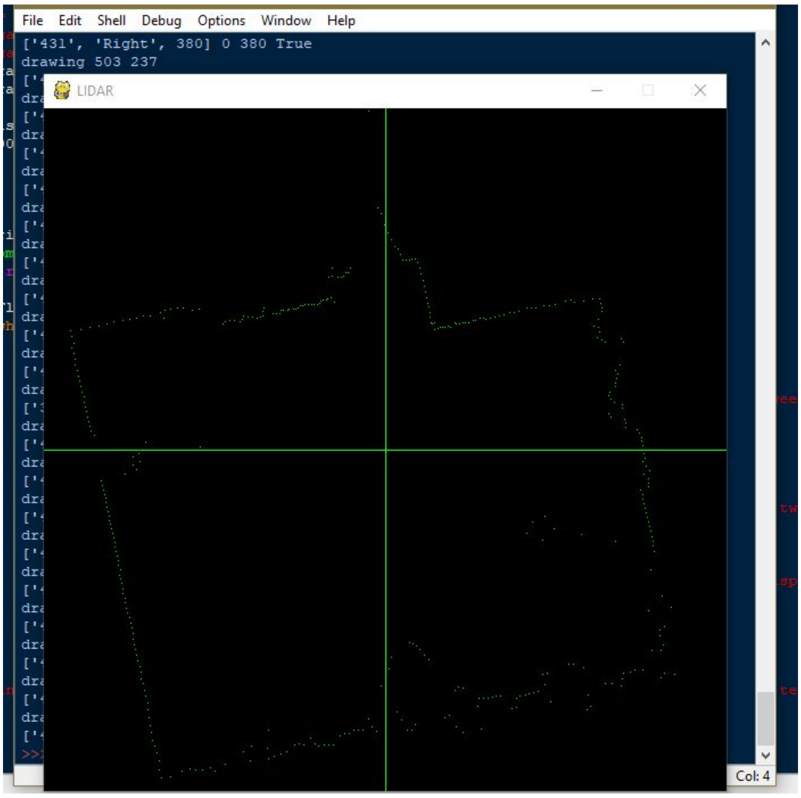
The output from the Lidar along with the data relating to the position of the stepper motor was rendered into an image by the python code

The image shows the boundaries of the Electronics lab where the initial testing was completed.



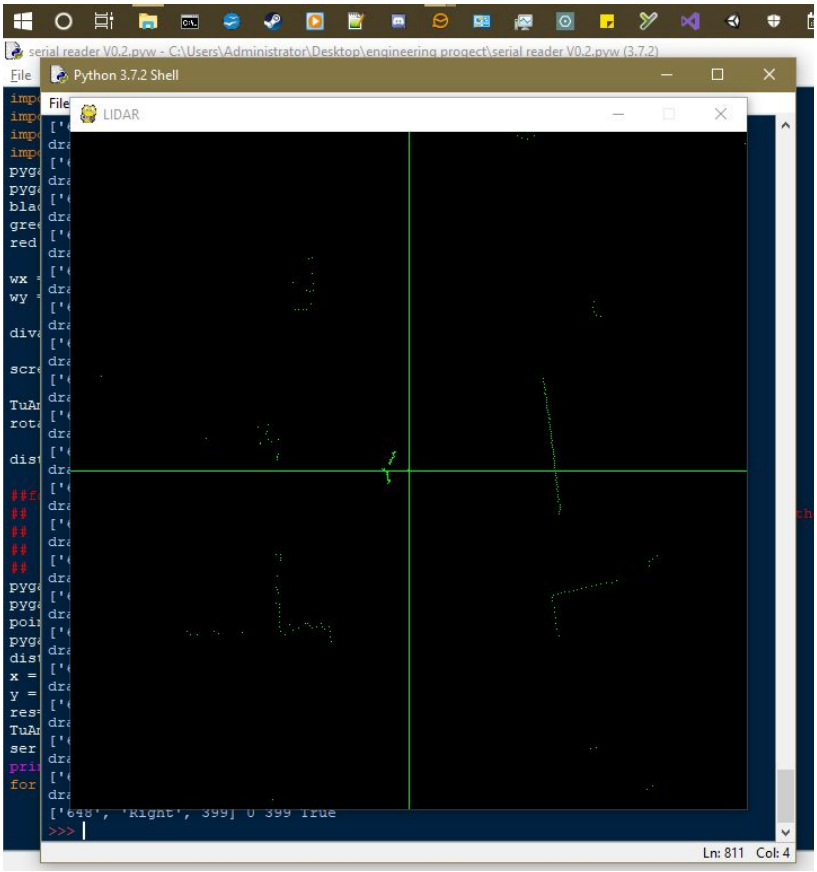
Test 4 - Scanning a Room

The Lidar system successfully scanned an entire lab with dimensions of around 10 m square.



Test 5 - Outdoors

The Lidar struggled to pick up the objects beyond 30 m but gave the outline of the buildings quite clearly.



Glasgow Caledonian University Workshop Visit

At the Glasgow Caledonian University Workshop, there were three different workshops that we took part in.

Workshop 1: Audio Engineering

In the first workshop, we went to the music room to find out how the soundboards work. Here we were able to learn about both analogue digital soundboards. It was quite interesting because our team never had an experience with controlling the soundboards before. The students at the Glasgow Caledonian University demonstrated by the soundboards by using speakers, mics and the guitars etc.

Workshop 2: Digital and Robotics; Programming robots

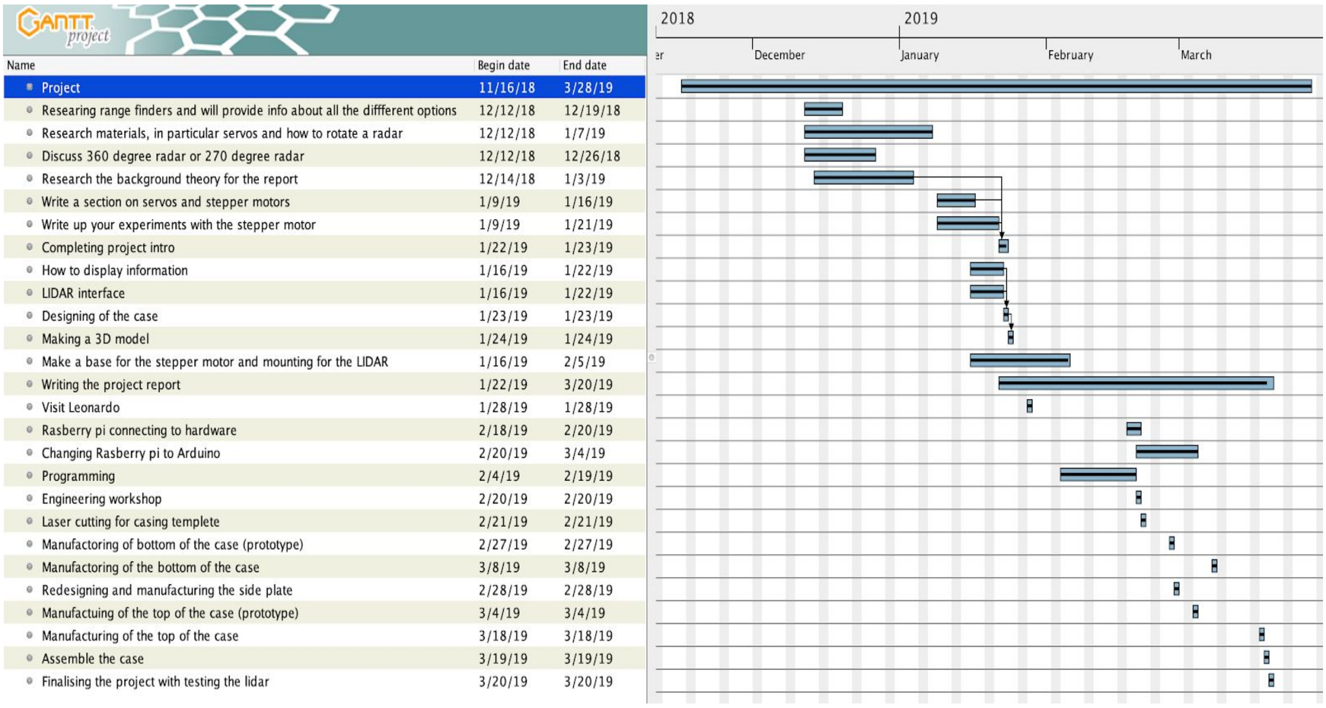
We also had an opportunity to try out programming and Lego mindstorms. In this workshop we made robots that can detect colour. We wrote a program so that the robot can travel forward while detecting the colour of the black line and the floor colour. It changed the direction as they detected the colour change. It was a good chance for us to try robotics and programming and we all found it very challenging.

Workshop 3: Mechanical Structural Integrity and Testing

In the third and final workshop, we went to a lab and did a practical. In this workshop, we learnt about the Young's modulus of a variety of materials. The method that we used to test these materials was the stretching method. We accomplished this by stretching three different metals including copper, mild steel and high percentage carbon steel. The stretching machine also drew a stress strain graph so that we could visualise the difference of stiffness for the three selected metals.

Project Management

The main way in which we managed our time, apart from Nikos keeping us on track, was through the use of a gantt chart.



Name	Begin date	End date
Project	11/16/18	3/28/19
Researing range finders and will provide info about all the diffferent options	12/12/18	12/19/18
Research materials, in particular servos and how to rotate a radar	12/12/18	1/7/19
Discuss 360 degree radar or 270 degree radar	12/12/18	12/26/18
Research the background theory for the report	12/14/18	1/3/19
Write a section on servos and stepper motors	1/9/19	1/16/19
Write up your experiments with the stepper motor	1/9/19	1/21/19
Completing project intro	1/22/19	1/23/19
How to display information	1/16/19	1/22/19
LIDAR interface	1/16/19	1/22/19
Designing of the case	1/23/19	1/23/19
Making a 3D model	1/24/19	1/24/19
Make a base for the stepper motor and mounting for the LIDAR	1/16/19	2/5/19
Writing the project report	1/22/19	3/20/19
Visit Leonardo	1/28/19	1/28/19
Rasberry pi connecting to hardware	2/18/19	2/20/19
Changing Rasberry pi to Arduino	2/20/19	3/4/19
Programming	2/4/19	2/19/19
Engineering workshop	2/20/19	2/20/19
Laser cutting for casing templete	2/21/19	2/21/19
Manufactoring of bottom of the case (prototype)	2/27/19	2/27/19
Manufactoring of the bottom of the case	3/8/19	3/8/19
Redesigning and manufacturing the side plate	2/28/19	2/28/19
Manufactuing of the top of the case (prototype)	3/4/19	3/4/19
Manufacturing of the top of the case	3/18/19	3/18/19
Assemble the case	3/19/19	3/19/19
Finalising the project with testing the lidar	3/20/19	3/20/19

The GANTT chart diagrammatically represents the time schedule we attempted to adhere to.

Our team held team meetings every wednesday at 7:00 p.m., which lasted from 30 minutes to 1 hour. The purpose of these meetings was purely to catch up on what progress we had made, and what progress we were going to make in the coming week. These meetings were very important in the production of our radar as it was here that we made our big decisions such as the type of radar we were using, initial designs etc.

On top of the GANTT chart, we set up a WhatsApp group chat to communicate anything that needed doing.

Future Development Ideas

Phased array system Radar

By relying on electronic signals instead of a rotating sweep, phased array radar is able to send out different sized beams to perform specific functions such as an individual search beam, and others for object tracking and altitude – all in one system. Therefore, it can track multiple targets and perform multiple tasks at the same time.

Synthetic Aperture Radar Systems

Synthetic Aperture Radar (SAR) uses a variety of complex maths to put together different scans together over an accurately known distance in order to generate a very accurate scan from many low resolution ones. Because it measures from different points it verifies whether an object exists and the shape of the object which is unachievable through a single scan.

MIMO Radar

MIMO stands for Multiple-Input-Multiple-Output, a technology that is coming up in communications in order to improve the coverage, data rate and/or signal quality. For the future radar the same improvements are needed. MIMO radars simultaneously radiate uncorrelated signals, for instance, in different directions, or in the same direction with orthogonal polarizations. This improves the coverage and the received information quality.

Conclusion

We were given an ambitious task to build a radar system and we managed to produce a Lidar based system that could:

- Rotate 360° by mounting on a stepper motor
- Use an Arduino to operate the radar's mechanisms and to interpret the information
- Display the information on a screen as a classic radar display
- Mark multiple points and outline the shape of a room, which can detect anything that is intruding the area
- Successfully work at ranges of up to 30 m

This radar we built had multiple uses. It can be used for indoor security of buildings such as a museum, but for us, the purpose of the radar was to survey rooms in buildings.

Our team kept on top of the work very well. We managed to stick to our agreed timescale and we didn't leave the project report too late but kept adding to the project which meant there was no panic at the end to cram our report in a few days. We trusted each other to get our jobs done as a massive project like this could not be done only by one person.

We all have learned from this experience and it has sparked further interest for some people in engineering.

We would advise any young scientists and engineers to give the EES a go as everyone will gain something out of it as well as gaining a stronger passion for engineering.

Team Statements

Nikos

Throughout this project, I enjoyed everything we did. As secretary of the project I was able to gain insight into different aspects of how an engineering firm works. I had a preconception that majority of the work would be spent in the workshop, building the radar, but this view was completely warped. In reality, a large amount of time was spent planning the project and actually writing the project report. I also partially acted as project manager which helped me develop my skills involving organisation, time management, planning and team communication. Overall, this experience has been great and has given me a better insight into what I may want to do in the the future.

Brodan

This project has brought with it many challenges. As the main contributor towards the electronics I have been left metaphorically banging my head against the table many, many, many times from things just not working, working how they are meant to and not how I want them to or just flat out being stubborn. Thankfully despite a few last minute setbacks I think it went rather OK. I have definitely learnt a lot from this project as it has forced me to use new things that I was definitely not familiar with. It has really helped with my personal development in electronics and I have learnt a lot from this and hope to go on learning in this field.

Han

Even though I am very interested in the physical aspect of sciences, I have not had many experiences with engineering. As soon as I heard that I could join to the engineering project, I was so excited due to the fact that I can actually try the engineering process including constructing and programming. However, it was much more challenging than I expected. From the planning and carrying out the project as a team was very tough. The physics and mathematics behind the radar was interesting but also hard to understand and apply to the project. Although the manufacturing and programming was very new to me, I learnt a lot of concepts how engineering works. I also learnt a lot about how to develop a long term project through good teamwork, time management and organisation. Overall, I am glad that I undertook the project as it helped me to improve my super-curricular understanding on engineering as I am very keen on taking engineering as a university course.

Robert

I had a fascination with engineering for much of my life however I never had a chance to know what engineering fully entailed. When I saw that there was a place open in the engineering team I jumped at the chance to join. Before the project I had experienced manufacturing with my subject choices but I also wanted to learn how to work with other materials. When we decided to use plastic I was interested in learning and working with a material that I had no prior knowledge and I was ready to learn about working with this new material. I also learned about other aspects of the project such as researching the about different types of motors and how they work and their level of accuracy. I felt the work that I did was challenging and taught me some lessons like starting earlier so that I didn't get to far behind on work. This experience has helped me understand engineering and has made me feel like taking the knowledge to the next level of education.

Hector

I have always enjoyed the theoretical aspect of Physics and Chemistry. However, I have never had the opportunity to go into engineering so with my skills in Maths and Physics I decided to give it a go. I originally researched how radars work and what type of radar would be most suited to us. This part of the project was very linked to my skills and the subjects I study. From then on I needed to learn a new skill, I had no experience with programming and electronics but it was a good experience learning how to program and how the program was used to operate the lidar. The project for me was also about how to work as a team. I had previously never worked on a project as a team, previously only by myself. It required people to stick to their roles and trust others to get the job done. It would not have worked if one person did all the programming, building the housing and writing the project report as there was not enough time. I felt our team used the whole project time effectively. We consistently worked with our time and did not leave the project too late.

Bibliography

https://www.scientificamerican.com/article/measuring-with-lasers/

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

https://sciencing.com/do-laser-distance-meters-work-6332366.html

https://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/

https://www.instructables.com/id/Raspberry-Pi-Arduino-Serial-Communication/

http://www.ti.com/product/MC34063A#

https://www.explainthatstuff.com/radar.html

http://www.abc.net.au/science/articles/2014/03/17/3964782.htm

https://www.researchgate.net/publication/272027572_RADAR_2020_THE_FUTURE_OF_RADAR_SYSTEMS

Glossary

Servo: Affordable mass-produced servomotors, often used in small scale robotics. Most, like ours, act as rotary actuators.

Stepper Motor: - A DC electric motor that divides a full rotation into a number of equal steps.

Microcontroller: A control device which incorporates a microprocessor.

CAD: (computer-aided design) software is used by architects, engineers, drafters, artists, and others to create precision drawings or technical illustrations.

I2C: A serial protocol for two-wire interface to connect low-speed devices like microcontrollers.

Driver: A program that controls the operation of a device such as a printer or scanner. (Computing) - a device or part of a circuit that provides power for output (Electronics)

Lidar: An optically based detection and ranging system

Appendices

Appendix 1 - Leonardo project brief

UNOFFICIAL DOCUMENTATION

LEONARDO
AIRBORNE & SPACE SYSTEMS

ENGINEERING EDUCATION SCHEME

DESIGN & IMPLEMENTATION OF A RADAR SYSTEM

PROJECT OUTLINE

Issue 1 November 2018

LEONARDO

UNOFFICIAL DOCUMENTATION

1. Leonardo MW

Leonardo is the largest inward investor in the UK defence sector and one of the biggest suppliers of defence equipment to the UK MoD.

Leonardo invests around £200m in the UK (around 10% of turnover) in Research and Development. Continued investment in innovation is enabling Leonardo to deliver some of the UK's most advanced technology programmes such as the new generation radar for the RAF's Typhoon combat aircraft.

The Edinburgh site is part of our Airborne and Space Systems Division which provides the sensors and electronic systems that sit under the skin of the world's most advanced aircraft, effectively acting as their brains, senses and nervous systems. Protecting pilots, alerting air crews to threats and providing crucial intelligence back to observers on the ground, the technology we offer can make the difference between life and death for our customers.

The Eurofighter Typhoon is the most advanced air multi-role fighter currently available on the world market.

1

LEONARDO

UNOFFICIAL DOCUMENTATION

2. Scope

This document describes the project challenge presented by Leonardo MW UK to the students of Merchiston Castle School, including a brief overview of the company, requirements and deliverables.

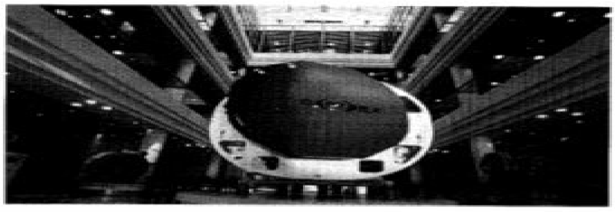
This document includes suggestion on possible implementations but all final decisions are ultimately up to the students undertaking this project.

UNOFFICIAL DOCUMENTATION

3. Introduction

Leonardo is a world leader in radar technology, operating a centre of excellence in Edinburgh. We design and manufacture radars for fire control and surveillance that have been bought by government and military customers around the world. Examples of these being CAPTOR radar for the Eurofighter Typhoon, the Seaspray Radar for the US Coast Guard, US Customs and Border Protection and the Raven for Saab's Gripen Fighter aircraft.

You will be designing a simple low cost radar system that should be able to detect objects around it. Applications for this might include obstacle finding for an autonomous car or robot, or object detection for a security system.



The CAPTOR radar designed for the Typhoon

3

UNOFFICIAL DOCUMENTATION

4. Mentors

You have been assigned two industrial mentors from Leonardo to provide you with support during this project. We will do our best to help with any questions (both technical and non-technical) you have during the project. Also feel free to ask us any questions you have about STEM, university, and working in the engineering industry. Our contact details can be found at the end of this document.

4.1. Kieran McGlynn

I am a Graduate Test Systems Engineer at Leonardo, currently working on CIRCM which is an infrared countermeasure. I focus on hardware at Leonardo but I do have some software experience from University. I went to Robert Gordon University before I started, where I studied Electronic and Electrical Engineering.

4.2. Sean Crawford

I am a Graduate Digital Hardware Engineer at Leonardo and I work on the processor for a Radar. During university I also did placements at Leonardo in RF (Radio Frequency) Engineering and Firmware Engineering (which basically means digital logic design). Before starting at Leonardo I studied Electronic and Electrical Engineering at the University of Edinburgh.

LEONARDO UNOFFICIAL DOCUMENTATION

5. Project Brief

5.1. Project Aims

Design and implement a radar like system which can scan the area around itself identify nearby objects, determine their location relative to the radar and display the information in an appropriate way to the user.

5.2. System Design

Over the course of this project you will be making engineering decisions which will have implications in the design and implementation of the radar system. Any decisions should be carefully thought out and any impact they might have going forward considered.

One very significant decision you will have to make early on is the medium that the radar will use to sense objects. A classic radar uses pulses of RF (radio frequency) radiation to measure distance by measuring the time between the pulse being send out and being reflected back. However, RF components are very expensive, very fast processing is required (before radio waves travel at the speed of light) and high power RF radiation can be dangerous (the front of a radar is kind of like a big high powered microwave). So working with RF may not be easy or cheap. Some alternatives that you could consider are using an ultrasonic or infrared rangefinder which use ultrasonic sound and infrared light respectively to measure the reflections from an object. Another alternative might be a Doppler sensor which uses RF waves but instead of measuring the time taken for a pulse to come back it sends out a continuous wave and measures the Doppler shift of the reflected wave; this allows the speed of an object to be calculated (but not the distance).

Some of the other decisions you may have to make are mentioned below along with some suggestions of choices you might make and the questions you should possibly ask yourself when making these decisions. However as mentioned in the introduction everything here is simply a suggestion and all final decisions are left up to you. Note that this is also a reasonably open-ended project and you are welcome to extend the functionality of the basic system if you have any interesting ideas.

5.3. Hardware

- The system will need a sensor element to perform the sensing function of the radar
  - What medium should the radar use for sensing? (e.g. light, laser, pulse RF, Doppler RF, Ultrasound)
  - Is your chosen sensor cheap and readily available?
  - Does the sensor require any additional electronics to drive it?
  - How will the radar control this sensing element?

5

LEONARDO UNOFFICIAL DOCUMENTATION

  - Is there someone at your school that you can ask for help?

- What development tools will you use?
  - How do your hardware decisions affect this?
  - Are there learning resources available?

- How will you implement the software?
  - What do you need to develop software for? Just a microcontroller or do you need separate software on a computer or a phone to display the data?
  - Are there software libraries to help interface with the hardware that you are using?
  - Are you going to split the software into different modules? (This can allow the development to be divided between people but can also cause problems when the modules are brought together.)

6. Deliverables

The first and most important deliverable is that of the **Project Report**. It is acceptable that only the final report is delivered at the end of the project period.

The other deliverable is a working radar system which detects objects and displays the information back to the user.

It is better to have a finished report and an unfinished prototype than a working prototype but an unfinished report.

7. Referencing

External software libraries may be used as needed, however any libraries used in the final project **MUST** be referenced in the final report.

Source code from external sources may be used if necessary but it students are encouraged to wherever possible think of their own solutions to software problems and write their own code. All external sources **MUST** be referenced in the final report.

Diagrams and images may be used where necessary, although it would be encouraged that all diagrams are created by the students. Any diagrams/images not created **MUST** be referenced in the final report.

Appendix 2 - Code

Version 1 - Computer side

```
import serial
import time
import pygame
black = (0,0,0)
white = (255,255,255)
TuAm = 2            #
                    # set these to the same values as there
counterparts on the Lidar
rotate = 100        #

#-------dont touch-------#
res= []
TuAm+=1
for i in range (0,TuAm):
    res.append([])
    for j in range (0,rotate):
        res[i].append([])

ser = serial.Serial('COM10', 9600)
print("comunicating")

for i in range (0,TuAm):
    for j in range (0,rotate):
        Flag = False
        while Flag == False:
            if(ser.in_waiting >0):
                Flag = True
                line = ser.readline()
                line = str(line)
                line = line.split(" ")
                line.pop(3)
                line.pop(0)
                line.append(j)
                print(line,i,j,Flag)
                res[i-1][j-1].append(line)

#------------dont touch anything above this line--------------#

#output in the form ("distance", "direction", "step") also outputs more
unintelligible data to text document in the same directory as the
program
```

Version 2 - Computer side

```
import serial
import time
import pygame
import math
pygame.init()
pygame.display.init()
black = (0,0,0)
green = (0,255,0)
red = (255,0,0)

wx = int(input("res x: "))
wy = int(input("res y: "))

divam = float(input("scaling: "))

screen = pygame.display.set_mode([wx, wy])

TuAm = 0          # set these to the same values as their counterparts
on the Lidar
rotate = 400
dist = 100

##for j in range(0,100):
##    x = 250+(int(dist)*(math.cos(math.radians(0.9*j))))   # these two
lines work out the coordonates of the point for the display
##    y = 250+(int(dist)*(math.sin(math.radians(0.9*j))))
##    pygame.draw.line(screen,green,[x,y],[x,y],1)
##    pygame.display.flip()
pygame.draw.line(screen,green,[(wx/2),0],[(wx/2),wy],1)  # these make
the crosshair
pygame.draw.line(screen,green,[0,(wy/2)],[wx,(wy/2)],1)  #
points = []
pygame.display.set_caption('LIDAR')
dist = 200
x = 0
y = 0
res= []
TuAm+=1
ser = serial.Serial('COM10', 9600)
print("comunicating")
for i in range (0,TuAm):
    for j in range (0,rotate):
        Flag = False
        while Flag == False:
            if(ser.in_waiting >0):
                Flag = True
```

```
                line = ser.readline()               #
                line = str(line)                    #   controls the
serial comunication between the lidar and the computer
                line = line.split(" ")              #
                if len(line)>3:
                    line.pop(3)
                    line.pop(0)
                    line.append(j)
                    x =
(wx/2)+(divam*(int(line[0])*(math.cos(math.radians(0.9*j))))) # these
two lines work out the coordonates of the ponit for the display
                    y =
(wy/2)+(divam*(int(line[0])*(math.sin(math.radians(0.9*j))))) #
                    x = int(x//1)
                    y = int(y//1)
                    pygame.draw.line(screen,green,[x,y],[x,y],1)   #
draws the point on the display where the point is
                    print("drawing",x,y)
                    pygame.display.flip()
                    print(line,i,j,Flag)
                    points.append([x,y])
##for i in range(0,len(points)):
##
pygame.draw.line(screen,green,[points[i-1][0],points[i-1][1]],[points[i
][0],points[i][1]],1)
##    pygame.display.flip()

#output in the form ("distance", "direction", "step") also outputs more
uninteligeble data to text document in the same directory as the
program
```

Version 3 - Computer side

```
import serial
import time
import pygame
import math
pygame.init()
pygame.display.init()
black = (0,0,0)
green = (0,255,0)
red = (255,0,0)

wx = int(input("res x: "))
wy = int(input("res y: "))

divam = float(input("scaling: "))

screen = pygame.display.set_mode([wx, wy])

TuAm = 0         # set these to the same values as their counterparts
on the Lidar
rotate = 400

dist = 100

##for j in range(0,100):
##    x = 250+(int(dist)*(math.cos(math.radians(0.9*j))))   #   this is
a pice of code for troubleshooting the display with a fixed distance
##    y = 250+(int(dist)*(math.sin(math.radians(0.9*j))))   #   set
within the program
##    pygame.draw.line(screen,green,[x,y],[x,y],1)
##    pygame.display.flip()
pygame.draw.line(screen,green,[(wx/2),0],[(wx/2),wy],1)  # these make
the crosshair
pygame.draw.line(screen,green,[0,(wy/2)],[wx,(wy/2)],1)  #
points = []
pygame.display.set_caption('LIDAR')
dist = 200
x = 0
y = 0
res= []
TuAm+=1
ser = serial.Serial('COM10', 9600)
print("communicating")
for i in range (0,TuAm):
    for j in range (0,rotate):
        Flag = False
        while Flag == False:
```

```
            if(ser.in_waiting >0):
                Flag = True
                line = ser.readline()               #
                line = str(line)                    #   controls the
serial communication between the lidar and the computer
                line = line.split(" ")              #
                if len(line)>3:
                    if (line[0]>0):
                        line.pop(3)
                        line.pop(0)
                        line.append(j)
                        x =
(wx/2)+(divam*(int(line[0])*(math.cos(math.radians(0.9*j))))) # these
two lines work out the coordonates of the point for the display
                        y =
(wy/2)+(divam*(int(line[0])*(math.sin(math.radians(0.9*j))))) #
                        x = int(x//1)
                        y = int(y//1)
                        pygame.draw.line(screen,green,[x,y],[x,y],1)
# draws the point on the display where the point is
                        print("drawing",x,y)
                        pygame.display.flip()
                        print(line,i,j,Flag)
                        points.append([x,y])
for i in range(0,len(points)):

pygame.draw.line(screen,green,[points[i-1][0],points[i-1][1]],[points[i
][0],points[i][1]],1)
    pygame.display.flip()



#output in the form ("distance", "direction", "step") also outputs more
unintelligible data to text document in the same directory as the
program
```

version 1 - Arduino side

```
int delaylegnth = 0;

// function for right turn
--------------------------------------------------------------------------
--------------- //
void TurnRight(){
  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, HIGH);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, HIGH);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);

  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, LOW);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, LOW);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);
}


// Function For Left Turn
--------------------------------------------------------------------------
-------------//

void TurnLeft(){
  digitalWrite(9, LOW);  //ENABLE CH A
```

```
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, HIGH);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, LOW);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);

  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, LOW);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, HIGH);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);
}

void setup() {

  //establish motor direction toggle pins
  pinMode(12, OUTPUT); //CH A -- HIGH = forwards and LOW = backwards???
  pinMode(13, OUTPUT); //CH B -- HIGH = forwards and LOW = backwards???

  //establish motor brake pins
  pinMode(9, OUTPUT); //brake (disable) CH A
  pinMode(8, OUTPUT); //brake (disable) CH B

//
------------------------------------------------------------------------
-- Main Program
------------------------------------------------------------------------
--- //
```

```
Serial.begin(9600);           // this begins serial comunication //
Serial.print("--------------------- connection established
---------------------- \n");

delaylegnth = 1;     // efectively sets the speed of the rotation //

int TuAm = 30;        // sets how many times the platform goes through
the left right cycle //

int rotate = 400;   // sets how far the platform rotates //


// bear in mind the platform will rotate left first //


for (int k = 0 ; k < TuAm ; k++ ) {

  Serial.print("left \n");
  for (int i = 0 ; i < rotate ; i++){
    TurnLeft();
    Serial.print(i);          // the serial function sends data back
throught the serial port back to the   //
    Serial.print("\n");       // computer so that the data being sent
back will be outputed in the terminal //
  }
  Serial.print("right \n");
  for (int j = 0 ; j < rotate ; j++){
    TurnRight();
    Serial.print(j);
    Serial.print("\n");

  }
}
// -------------------------------------------- End of Main Program
-------------------------------------------- //
}

void loop(){}




#######################################################################
#######################################################################
#######################################################################
#####
```

Version 2 - Arduino side

```
#include <Wire.h>
#include <LIDARLite.h>
LIDARLite lidarLite;
int delaylegnth = 0;


// function for right turn
------------------------------------------------------------------------
--------------- //
void TurnRight(){
  digitalWrite(9, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(12, HIGH);

  analogWrite(3, 255);

  delay(delaylegnth);

  digitalWrite(9, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(13, HIGH);

  analogWrite(11, 255);

  delay(delaylegnth);

  digitalWrite(9, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(12, LOW);

  analogWrite(3, 255);

  delay(delaylegnth);

  digitalWrite(9, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(13, LOW);

  analogWrite(11, 255);

  delay(delaylegnth);
}


// Function For Left Turn
```

```
------------------------------------------------------------------------
-------------//

void TurnLeft(){
  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, HIGH);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, LOW);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);

  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, LOW);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, HIGH);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);
}

void setup() {
  //establish motor direction toggle pins
  pinMode(12, OUTPUT); //CH A -- HIGH = forwards and LOW = backwards???
  pinMode(13, OUTPUT); //CH B -- HIGH = forwards and LOW = backwards???

  //establish motor brake pins
  pinMode(9, OUTPUT); //brake (disable) CH A
  pinMode(8, OUTPUT); //brake (disable) CH B

//
------------------------------------------------------------------------
```

```
-- Main Program
------------------------------------------------------------------------
--- //

Serial.begin(9600);           // this begins serial comunication //

bool flag  = false;

int delaylength = 0;

int rotate = 0;

int TuAm = 0;

delay(100);
while (flag == false){
  if (Serial.available()>0){
    flag = true;
    delaylength = delaylength * (Serial.read() - '0');
    Serial.print(delaylength);
    Serial.print("aFLag1 \n");
  }}

delay(100);
flag  = false;
while (flag == false){
  if (Serial.available()>0){
    flag = true;
    rotate = rotate * (Serial.read() - '0');
    Serial.print(rotate);
    Serial.print("aFLag2 \n");
  }}

delay(100);
flag  = false;
while (flag == false){
  if (Serial.available()>0){
    flag = true;
    TuAm = TuAm * (Serial.read() - '0');
    Serial.print(TuAm);
    Serial.print("aFLag3 \n");

  }}

delay(10);

Serial.print("--------------------- connection established
---------------------- \n");
```

```
lidarLite.begin(0, true); // Set configuration to default and I2C to
400 kHz //

lidarLite.configure(); // Change this number to try out alternate
configurations //

for (int k = 0 ; k < TuAm ; k++ ) {
  int dist = 0;


  for (int i = 0 ; i < rotate ; i++){
    TurnLeft();
    Serial.print(i);          // the serial function sends data back
throught the serial port back to the   //
    Serial.print(" left ");       // computer so that the data being
sent back will be outputed in the terminal //
    dist = lidarLite.distance();
    Serial.print(dist);
    Serial.println(" cm");
  }

  for (int j = 0 ; j < rotate; j++){
    TurnRight();
    Serial.print(j);
    Serial.print(" right ");
    dist = lidarLite.distance();
    Serial.print(dist);
    Serial.println(" cm");
  }
}
// ----------------------------------------------- End of Main Program
-------------------------------------------- //
}

void loop(){}


########################################################################
########################################################################
########################################################################
#####
```

version 3 - Arduino Side

```
#include <Wire.h>
#include <LIDARLite.h>
LIDARLite lidarLite;
int delaylegnth = 0;

// function for right turn ------------------------------------------------------------------------------------ //
void TurnRight(){
  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, HIGH);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, HIGH);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);

  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, LOW);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, LOW);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);
}


// Function For Left Turn ----------------------------------------------------------------------------------//

void TurnLeft(){
```

```
  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, HIGH);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, LOW);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);

  digitalWrite(9, LOW);  //ENABLE CH A
  digitalWrite(8, HIGH); //DISABLE CH B

  digitalWrite(12, LOW);   //Sets direction of CH A
  analogWrite(3, 255);   //Moves CH A

  delay(delaylegnth);

  digitalWrite(9, HIGH);  //DISABLE CH A
  digitalWrite(8, LOW); //ENABLE CH B

  digitalWrite(13, HIGH);   //Sets direction of CH B
  analogWrite(11, 255);   //Moves CH B

  delay(delaylegnth);
}

void setup() {

  //establish motor direction toggle pins
  pinMode(12, OUTPUT); //CH A -- HIGH = forwards and LOW = backwards???
  pinMode(13, OUTPUT); //CH B -- HIGH = forwards and LOW = backwards???

  //establish motor brake pins
  pinMode(9, OUTPUT); //brake (disable) CH A
  pinMode(8, OUTPUT); //brake (disable) CH B

// ------------------------------------------------------------------------ Main Program
------------------------------------------------------------------------ //
```

```
Serial.begin(9600);          // this begins serial comunication //

lidarLite.begin(0, true); // Set configuration to default and I2C to 400 kHz //

lidarLite.configure(); // Change this number to try out alternate configurations //

delaylegnth = 5;     // efectively sets the speed of the rotation //

int TuAm = 1;        // sets how many times the platform goes through the left right cycle //

int rotate = 100;   // sets how far the platform rotates //


// bear in mind the platform will rotate left first //


for (int k = 0 ; k < TuAm ; k++ ) {
  int dist = 0;


  for (int i = 0 ; i < rotate ; i++){
    TurnLeft();
    Serial.print(i);          // the serial function sends data back throught the serial port back to
the   //
    Serial.print(" left ");       // computer so that the data being sent back will be outputed in the
terminal //
    dist = lidarLite.distance();
    Serial.print(dist);
    Serial.println(" cm");
  }

  for (int j = 0 ; j < rotate; j++){
    TurnRight();
    Serial.print(j);
    Serial.print(" right ");
    dist = lidarLite.distance();
    Serial.print(dist);
    Serial.println(" cm");
  }
}
// ------------------------------------------- End of Main Program ------------------------------------------- //
}

void loop(){}
```

##

version 4

```
#include <Wire.h>
#include <LIDARLite.h>
LIDARLite lidarLite;
void setup() {
// ----------------------------------------------------------------------- Main Program
------------------------------------------------------------------------ //

Serial.begin(9600);          // this begins serial comunication //

lidarLite.begin(0, true); // Set configuration to default and I2C to 400 kHz //

lidarLite.configure(); // Change this number to try out alternate configurations //
}
int dist = 0;
void loop(){
dist = lidarLite.distance();
Serial.print(dist);
Serial.println(" cm");
}
```

##

version 5

```
int Delay = 0;
#include <Wire.h>
#include <LIDARLite.h>
LIDARLite lidarLite;
void right(){
  digitalWrite(A0,LOW);  // step 1
  digitalWrite(A1,HIGH);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 2
  digitalWrite(A1,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 3
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 4
  digitalWrite(A1,LOW);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
}
void left(){
  digitalWrite(A0,HIGH);  // step 1
  digitalWrite(A1,LOW);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 2
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 3
  digitalWrite(A1,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 4
  digitalWrite(A1,HIGH);
```

```
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
}
void Reset(){
  digitalWrite(A0,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,LOW);
}

void setup(){
  Serial.begin(9600);
  lidarLite.begin(0, true);
  lidarLite.configure();

  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A2,OUTPUT);
  pinMode(A3,OUTPUT);
  Delay = (5);
  int TurAng = 400;
  int TurAmm = 1;
  for (int j = 0; j<TurAmm  ; j++){
  for (int i = 0; i<TurAng; i++){
    Serial.print(i);
    Serial.print(" ");
    Serial.print(lidarLite.distance());  // this outputs the data to the laptop or other device over
USB
    Serial.print(" ");
    Serial.print("Right \n");
    right();
  }
  for (int k = 0; k<TurAng; k++){
    Serial.print(k);
    Serial.print(" ");
    Serial.print(lidarLite.distance());
    Serial.print(" ");
    Serial.print("Left \n");
    left();
  }
  Reset();

}
}
```

```
void loop() {

  // put your main code here, to run repeatedly:

}
```

##
##
######################

version 6

```
int Delay = 0;
//#include <Wire.h>
//#include <LIDARLite.h>
//LIDARLite lidarLite;
void right(){
  digitalWrite(A0,LOW);  // step 1
  digitalWrite(A1,HIGH);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 2
  digitalWrite(A1,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 3
  digitalWrite(A1,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 4
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 5
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
```

```
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 6
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 7
  digitalWrite(A1,LOW);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 8
  digitalWrite(A1,HIGH);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
}
void left(){
  digitalWrite(A0,LOW);  // step 1
  digitalWrite(A1,LOW);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
  digitalWrite(A0,HIGH);  // step 2
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 3
  digitalWrite(A1,HIGH);
  digitalWrite(A2,LOW);
  digitalWrite(A3,HIGH);
  delay(Delay);
  digitalWrite(A0,LOW);  // step 4
  digitalWrite(A1,HIGH);
  digitalWrite(A2,HIGH);
  digitalWrite(A3,LOW);
  delay(Delay);
}
void Reset(){
  digitalWrite(A0,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);
  digitalWrite(A3,LOW);
}
```

```
void setup(){
  Serial.begin(9600);
  //lidarLite.begin(0, true);
  //lidarLite.configure();

  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A2,OUTPUT);
  pinMode(A3,OUTPUT);
  Delay = (10);
  int TurAng = 400;
  int TurAmm = 1;
  for (int j = 0; j<TurAmm  ; j++){
  for (int i = 0; i<TurAng; i++){
    Serial.print(i);
    Serial.print(" ");
    //Serial.print(lidarLite.distance());
    Serial.print(" ");
    Serial.print("Right \n");
    right();
  }
  for (int k = 0; k<TurAng; k++){
    Serial.print(k);
    Serial.print(" ");
    //Serial.print(lidarLite.distance());
    Serial.print(" ");
    Serial.print("Left \n");
    left();
  }
  Reset();

}
}


void loop() {

  // put your main code here, to run repeatedly:

}
```

Appendix 3 - Minutes (Week 1)

Below is a snapshot of our minutes from week 1.

Meeting 28/10/2019

Meeting time is Wednesday 6:30 to 7:30 and Monday Engineering Activity 28/11/18-(18:30-19:20)

1. Given an Introduction on Radar from Mr Nicholls
2. Project outline: To create a radar. To map out contours and map on the scale of a room? A field? What's the purpose? Surveying? Security? Perimeter security? Need to develop an aim of the radar.
3. Discussed the Challenges: Technicalities … programming, electronics. Physical construction. Time management.
4. We gave out roles to each person in the group so we work more efficiently:

- Nikos: Team Manager/ Secretary
- Brodan: Team Manager/ Coder
- Han: Mathematician/ Builder
- Bobby: Builder
- Hector: Mathematician

5. To do:

- Figure out which type of Radar we want to use for our project.

“Science can amuse and fascinate us all, but it is engineering that changes the world.”

Isaac Asimov