UNIVERSITY OF SOUTHAMPTON

Electronics and Computer Science

A Group Design Project Report Submitted for the Award of:

MEng Electronic Engineering, MEng Electromechanical Engineering, and MEng Computer Science

By:

Damon Roberts

Matthew Brooks

Arjun Patel

Man-Leong Chan

Project Supervisor: Professor Kirk Martinez

Second Examiner: Dr Christopher Freeman

GDP 4

NXP Cup Autonomous Racing Car

31/01/2018

Abstract

The NXP Cup is a race designed to encourage an interest in autonomous vehicles (AVs) and embedded programming, where entrants have to set the best lap of an unknown test track using a standardised car kit. This report details the development of a car to compete in the 2018 round. A PID based controller was implemented, leveraging data from a dual linescan camera array. The final product was able to complete the most complex test track that we could construct reliably with consistent lap times. The project gave an insight into the challenges faced by full-size autonomous vehicles, despite its relatively controlled conditions.

Contents

List of Figures

List of Tables

1 Introduction

Autonomous vehicles (AVs) are on the cusp of becoming a part of everyday life, with many companies conducting real world tests. Waymo [1] has been developing autonomous vehicles since 2006, and in 2017 began operating an autonomous taxi service in Phoenix, Arizona. Formula E is busy organising the world's first full scale autonomous racing league, Roborace [2], as a supporting event to the main championship, and as a test bed for real time computing and machine vision technologies. The sensing systems and the sheer volume of data that needs to be processed are key aspects holding autonomous cars back, alongside the social and political challenges involved in bringing such a disruptive technology to market.

The NXP Cup, which will be introduced in more detail in the next section, hopes to introduce students to some of these concepts within a more controlled environment using an accessible platform. This report details the development of Southampton's entry into the 2018 round of the NXP Cup, beginning with an overview of the competition, and current autonomous vehicle technology. This is followed by a summary of our own project goals and stretch goals, followed by details of the development of the vehicle. The development was split into four sections, which provided each team member with a well contained section of the project, allowing much of the work to be carried out in parallel. These four sections are: physical design; machine vision; development of the control systems and finally, wireless communications for telemetry and debugging.

2 The NXP Cup

The NXP Cup is a series of national and international races, based around an NXP designed autonomous car. Teams from schools and universities compete against each other in a series of local heats, before progressing onto an international final round. The competition is designed to be easily accessible to competitors with a large range of skills and experiences. This is achieved by providing a large amount of working example code and implementing rule rules prohibiting the use of expensive high-powered processors and sensors or upgrading the car itself, with better motors or large physical changes. This means the teams who succeed will be those whom created the best algorithms rather than who spent the most money.

The car itself is based around a complex remote-control car style chassis, featuring adjustable suspension, adjustable camber and toe angles for the front wheels, adjustable servo driven steering geometry and two 7.2V motors driving the rear wheels independently. This model, known as the 'Model-C' is in the process of being updated with a new, simplified design called the 'Alamak'. We received two of the new cars, however the various boards included with them were equipped with the wrong connectors for the cameras and other peripherals, and libraries for them were not yet available. Due to these issues, only the older 'Model C' was used for this project. The new 'Alamak' cars will be made to work in time for the completion later this year, as Southampton has entered one of each model.

The rules for the competition this year [3] are mostly unchanged from the previous years, with the key points listed below:

- The original and unaltered equipment must be used as the entry.
- No part of the car shall exceed dimensions of 250mm (W) x 400mm (L) x 305mm (H).
- You may change the orientation of the servo motor and related linkages.
- You may add a "skin" to the car, but it must be removable during inspection.
- You may adjust or remove springs, linkages, and other non-essential pieces.
- One processor - No auxiliary processor or other programmable device allowed.
- The car must use an optical sensor to navigate, no other navigation technique is to be used.

Aside from these rules, competitors must submit a technical report to NXP, and upload all their code to an open source repository. All cars are inspected before the race commences, and no changes can take place after this inspection.

The car has to follow a 60cm wide track. The track itself is made from various sections of matte white ABS with 25mm wide black stripes along each edge. The track pieces are standardised, however the track layout the car will be competing on is unknown. The track can have a variety of obstacles, including inclines and declines, dark tunnels, a series of small 25mm tall rumble strips, and crossroads. The car must also be able to detect a finish line on one of the straight sections of track, or a penalty will be applied to the final time. Rules during the race include the following:

- The car fails to leave the starting area within 30 seconds after beginning of the race [+1 second].
- The car fails to stop within 2 meters of the finish line or leaves the track after crossing the finish line [+1 second].

- Three or more wheels leave the racetrack [Disqualified from that attempt].
- The team fails to get prepared for the attempt within the two minutes [Disqualified from that attempt].

3 Review of Current Autonomous Car Vision Systems

Autonomous vehicles for use in a real-world environment make use of multiple, far more complicated sensor systems than those used in the NXP Cup. Most AVs use a combination of LIDAR, radar, and cameras to provide 360-degree vision around the car. The challenge when using so many sensors is utilising them all together to gain a coherent image of the cars surroundings [4]. LIDAR is a key technology used in most AVs, as it can generate a 3D view of the surrounding area far more efficiently than a camera can, whilst being less vulnerable to lighting conditions [5]. LIDAR works by rapidly pulsing a spinning laser. Some of these pulses reflect off the surroundings and are then detected by the LIDAR unit. Measuring the 'time-of-flight' of each pulse can be used to build a 3D map of a vehicles surroundings, based on more accurate distance data than can be estimated using a camera, and with fewer interference issues than radar.

Following the build-up of a detailed image, deep learning tends to be used to make decisions [6]. This means machine learning algorithms can be trained to detect certain road markings, signs, road edges, and objects on the road and respond accordingly. The challenge in this approach is the sheer range of possible data and therefore the training and processing power required to make a truly robust autonomous car is vast, especially if a car is expected to work independently on all types of road surface and in all conditions. Thankfully, in the NXP Cup there is a standardised track, and no obstacles to avoid, so there is a much smaller possible range of input data. This means, for the purposes of the cup, a machine learning approach would be excessive, and very difficult to implement on the KL25Z. Nor would LIDAR or radar sensors be of much use.

Recently, AVs have begun being equipped with infra-red sensors alongside LIDAR and conventional cameras. This is because LIDAR struggles in adverse weather conditions such as rain or fog [7]. IR sensors allow the car to detect objects on the road in conditions which may scatter the laser pulses, such as heavy rain or fog, or in extremely dark environments where cameras would struggle. Work is also being carried out on designing a solid-state LIDAR. Currently, the spinning mechanism is a key point of failure in LIDAR sensors, something which isn't ideal on a safety critical system. It also adds greatly to the complexity and cost of these sensors, whilst a solid-state solution is likely to be far more reliable, and cheaper to mass produce [8].

4 Review of the Social and Economic Aspects of Autonomous Cars

There are three commonly cited benefits of AVs, all of which will be discussed in the following section. Firstly, a fully autonomous fleet is likely to greatly reduce deaths and damage caused by vehicle accidents. Secondly, there is a potential for a massive reduction in the number of cars owned worldwide through car sharing and autonomous taxi schemes [9]. Thirdly, they may enable new levels of mobility to those who can't afford a vehicle or are unable to drive. All these, coupled with the increased proliferation of battery electric vehicles (BEVs), plug-in hybrid electric vehicles (PHEVs), and other low emission vehicles acting to reduce the carbon footprint of the transport industry paints an inspiring image for the future of the transport industry.

4.1 Safety Benefits

It is likely that greatest advantage of autonomous vehicles is the potential to reduce the number of incidents on the roads. This will be achieved by vastly reducing accident numbers by eliminating human error from the system. The statistical likelihood of being involved in a car accident varies widely dependent on experience, age, location, and other metrics, however most sources agree that the average driver will be involved in at least one accident, and probably two, in their lifetime. Comparatively, Waymo's vehicles have covered over 3.5 million miles of real world driving [10], and over this time, they were involved in only three accidents. Two of these were caused by human drivers rear-ending a Waymo vehicle whilst it was stopped at a traffic light, and the other occurred whilst a human driver had taken over control of the vehicle. Most people are unlikely to drive anywhere near that distance in their lifetime, and as such already show a better safety record than the average person, however, there are obvious limitations in this data. The vehicles have mainly been running in 'ideal' conditions in suburban America, with simple, wide roads and far less traffic than a complex scenario, such as rush hour in London.

Despite this, conclusions can still be drawn from these results. The main one is that until all cars, or at least a large majority are autonomous, accident rates are unlikely to decrease dramatically. This is due to the unpredictability of driving, currently, humans are still far better than computers at reacting to complex situations, and at predicting what other people will do. Autonomous cars may be extremely good at following the rules of the road, and driving 'by-the-book', however still struggle to react as a human would. Recently, a self-driving shuttle bus in Las Vegas was slowly reversed into by a truck during its first two hours of operation [11]. The vehicle had stopped behind the truck when it detected the chance of a collision, just as it was programmed to, however it did not ever attempt to move out of the way as any human driver would have. This may come across as an obvious oversight in testing or programming, however it highlights just how many situations there may be where current autonomous technology is limited.

4.2 Economic and Environmental Benefits

Worldwide, costs and damages incurred due to road accidents amount to 2-3% of many developed countries GDPs [12]. In the UK, the government calculates the annual costs of accidents per year. The annual report considers physical damage, costs of healthcare

and repairs, and working hours lost due to death or injury. In 2016, this totalled nearly £36 billion, or 1.3% of the UK's GDP [13].

Clearly, fewer cars on the road would reduce the environmental impact of transport due to lower cumulative emissions, alongside lower manufacturing volumes and raw material usage. This would be possible through the usage of on-demand AVs, which can simply be summoned like a taxi whenever you need them. In cities where space is at a premium, reduced car ownership will free up space on roads for pedestrians and bicycles and have knock-on benefits to public health by improving air quality. Many commercial logistics companies, and truck manufacturers are exploring the potential for autonomous technologies in that sector. Volvo has been testing the idea of autonomous truck 'platoons'. These are tightly packed convoys of trucks that drive far more closely together than a human could, whilst following with a 'lead truck'. This technique can reduce fuel use by the convoy by more than 10% [14]. In 2016 they completed a route across Europe, and have recently been given the green light to begin testing in the UK [15].

In areas with high population densities with enough demand, autonomous vehicles may be able to calculate optimal 'ride-share' routes, ensuring they're always carrying a few people at a time. A highly connected car may also be able to calculate optimum routes to save power by avoiding traffic, or plan to recharge itself during times where excess renewable power is available at a charging location. Obviously, such a dramatic change is reliant on many technologies maturing, and a shift away from the mind-set that cars are 'status symbols' and more of a service to be utilised.

4.3 Unsolved Issues

There are however, still several key issues with the adoption of AVs. Aside from the unproven and immature nature of the sensors and control systems, there is a strong social stigma against the concept of self-driving cars, and the potential reduction in car ownership. Highly publicised vulnerabilities affecting millions of computer processors, such as Spectre [16] and Meltdown [17] do not help to build faith in a system that would have people's lives in its hands, no matter how small the real life risks are. There are also legal matters, who is liable for a crash [18]? The car manufacturer, the company that supplied the sensors, or the city council that didn't repaint their road markings? There are issues relating to privacy and data collection, let alone the potential requirements for widespread policy changes [19]. If a car can drive itself, would you need a licence to 'drive' it?

There is far more to bringing a new technology to market than just the engineering, particularly for one like AVs; which have such far reaching effects. It is likely that these factors will end up being the rate limiting step in the adoption of self-driving vehicles, rather than the technology behind them. As shown in this report, even simple hardware can be used to control an 'autonomous' vehicle but convincing someone to use our car for their daily commute would prove nigh on impossible.

5 Review of Previous NXP Cup Work

Southampton competed in the NXP Cup last year, and therefore we were given the choice of continuing on using their work as a base, *or starting from the beginning.

Their final design is detailed in their report [20], and featured a single camera, hall effect sensors to measure rear wheel speed and a PID controller to set the steering angles and maintain drive wheel speed. Their design was successful as they achieved good times during the actual NXP cup and met most of their objectives. They were far below budget as well. They stated that potential improvements to their design include some form of look ahead corner detection, a more complex PID controller, implementation of reactive control, more reliable finish line detection, and improved headlights. The 'look ahead' data would be used to brake before the corners and fed into a PID controller with variable gain that could utilise different values dependant on track layout. Potentially, this system then could be used to develop a more advanced predictive controller rather than a reactive one, however the processing power and cameras available may be the limiting factor in this approach.

We decided to start fresh for several reasons. The main reasoning behind this decision was that if we had decided to continue using their code and designs from last year, we would risk taking a blinkered approach to finding solutions, by just using their assumptions and potentially missing better ideas. Secondly, understanding other people's code can sometimes be more difficult and time consuming than writing your own. Finally, attempting to improve on a very successful project would have risked our group not producing enough unique work. On the hardware side, many of the physical components they used/designed, such as the camera mounts and board holders were missing, damaged or could easily be improved, so most of that side of the project would have ended up being changed anyway.

6 Project Goals and Stretch Goals

With the aim of the project to compete in the NXP Cup, there is a list of essential goals which were needed to be achieved. The rover must obviously conform to all the rules laid out by NXP in order to qualify for the competition, as well as being capable of successfully racing around an unseen racetrack in the fastest possible time.

6.1 Main Goals

Hence, the main goal of the entire project was to build a successful car that could compete in the competition. As the event is only held in March, it was decided that the measure of success for the rover would be a comparison to last year's car. If our design could reach within 0.5s of their times, using their test track layout, this goal would be met. As their group had one more member than we did, this is a reasonable goal to try and achieve.

Throughout the project, particular deliverables were set on the way to the final goal. These are detailed the Gantt charts in Appendix C and D, and are summarised below.

- Establish basic car movement.
- Attain vision of the track.
- Navigating straight track sections.
- Navigating single corners.
- Navigating multiple corners.
- Negotiating the unique track pieces (chicane, hill, tunnel, and crossroad)
- Stopping at the finish line.

6.2 Stretch Goals

The stretch goals were to be implemented once we felt we had reached the limit of a particular solution or discovered an issue that wasn't solvable using the existing hardware. These included:

- The addition of torque steering/traction control/an electronic differential.
- Self-optimisation using the telemetry to determine the most efficient set of coefficients for the control algorithms.
- Addition of another camera, either for increased resolution, or as a 'look ahead' camera to identify upcoming obstacles.

7 Division of Labour

The work on the car itself was divided up based on the individual group members' expertise, and what parts they wanted to work on. Within this section, each group member is referred to by their initials.

DR = Damon Roberts, MB = Matthew Brooks, AP = Arjun Patel and MLC = Man-Leong Chan.

7.1 Division of Project Work

The rough sections of the work carried out throughout the project are listed below. When members of the team had spare time, they assisted in the building and testing of the other project portions.

- Physical Components – DR
- Camera and Machine Vision – MB
- Control Algorithm – AP
- Building, Testing and Tuning – MB, DR & AP
- Communications and Hosted Software – MLC

7.2 Division of Report Writing

The report was written collectively with the author of each section shown below, before being merged and formatted by DR and proofed by the whole of the group.

- Abstract – DR
- Introduction – DR
- The NXP Cup – DR
- Review of Previous Work – DR & MB
- Project Goals and Stretch Goals – AP
- Division of Labour – AP
- System Overview – AP
- Physical Design and Hardware – DR
- Machine Vision, Servo Centering and Overall Embedded Program – MB
- Control Theory – AP
- Telemetry – MLC
- System Testing – MB
- Limitations – MB
- Lessons Learnt – AP
- Project Management – AP
- Conclusions – DR

8 System Overview

Below is a flowchart (Figure 1) showing the systems and subsystems that make the car work, and the various power and data flows between them. Overleaf is a labelled photo of the car, to illustrate each component more clearly, followed by a description of each main component.

Figure 1: System Flowchart

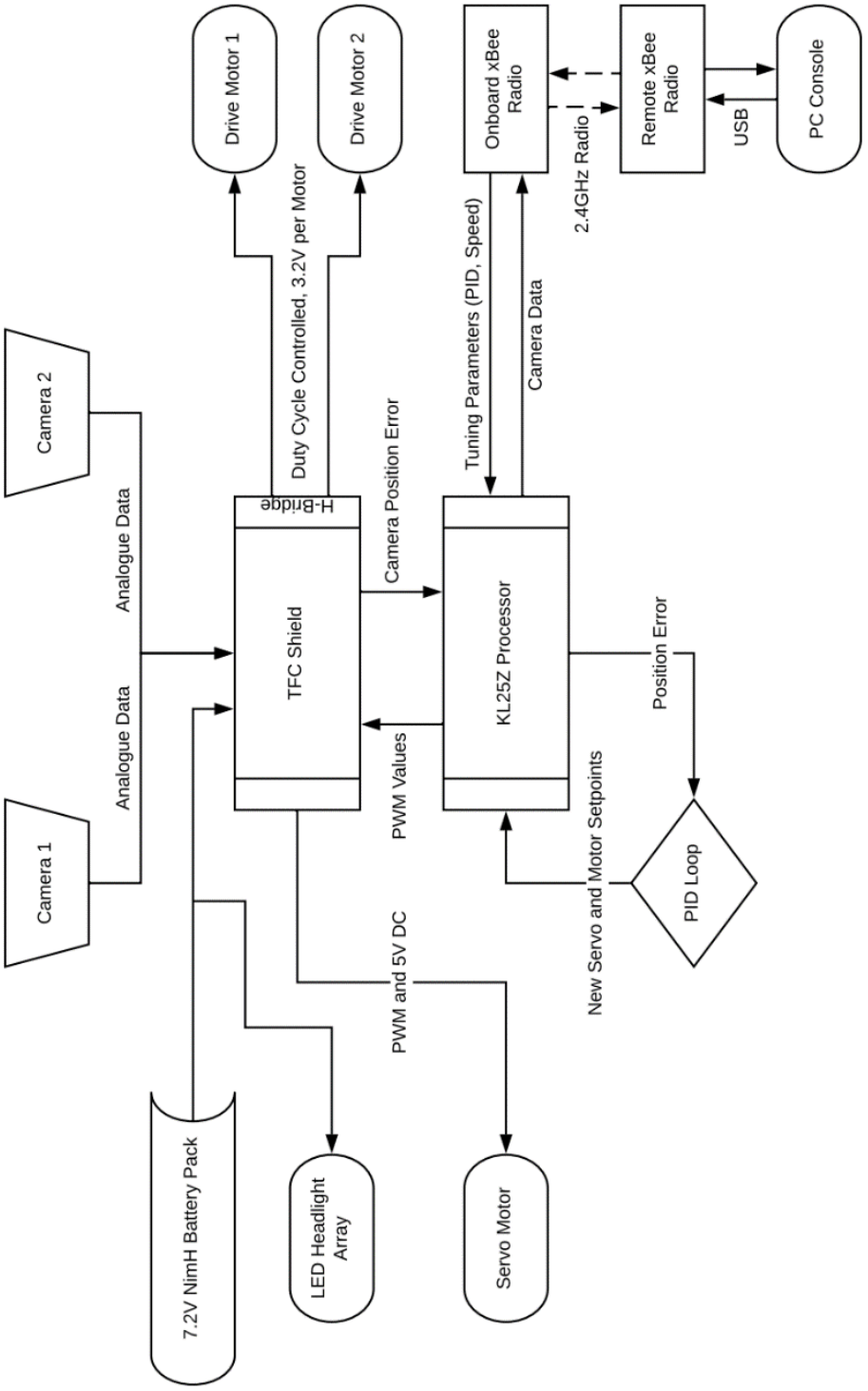
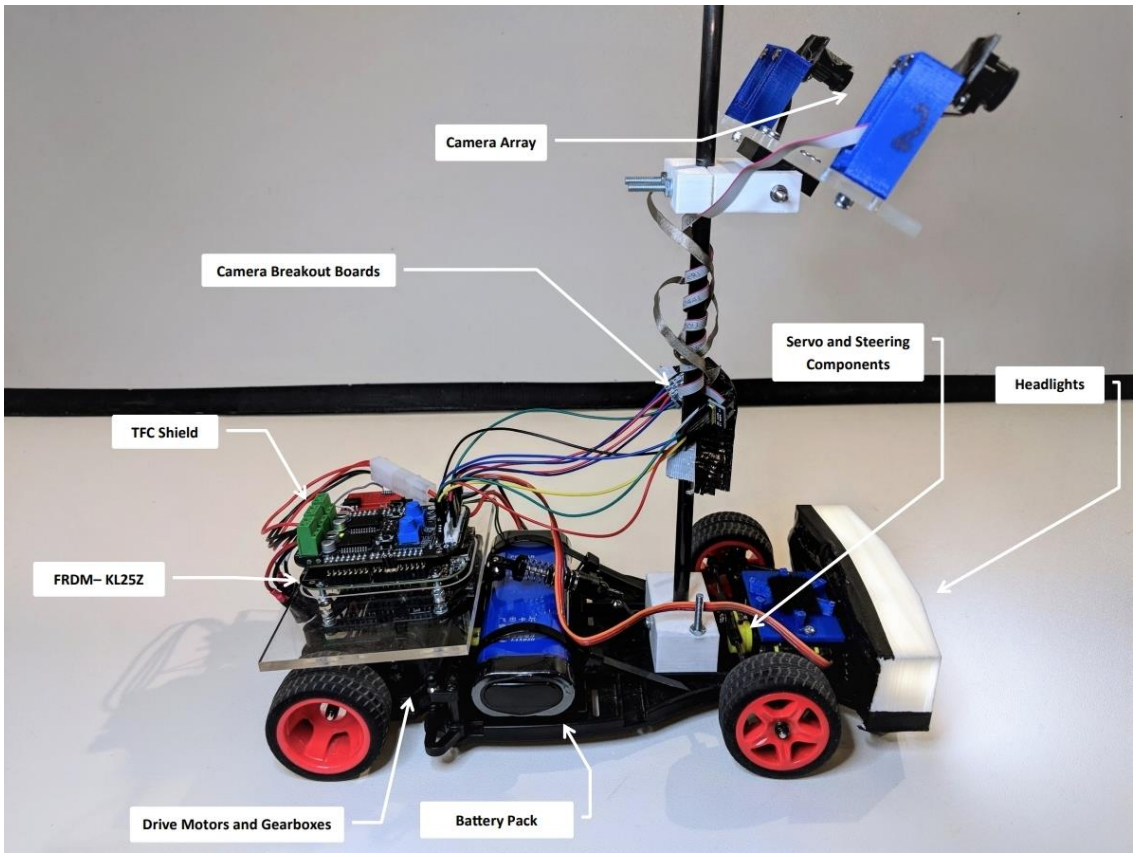


Figure 2: Labelled Diagram of the car.



Key Components:

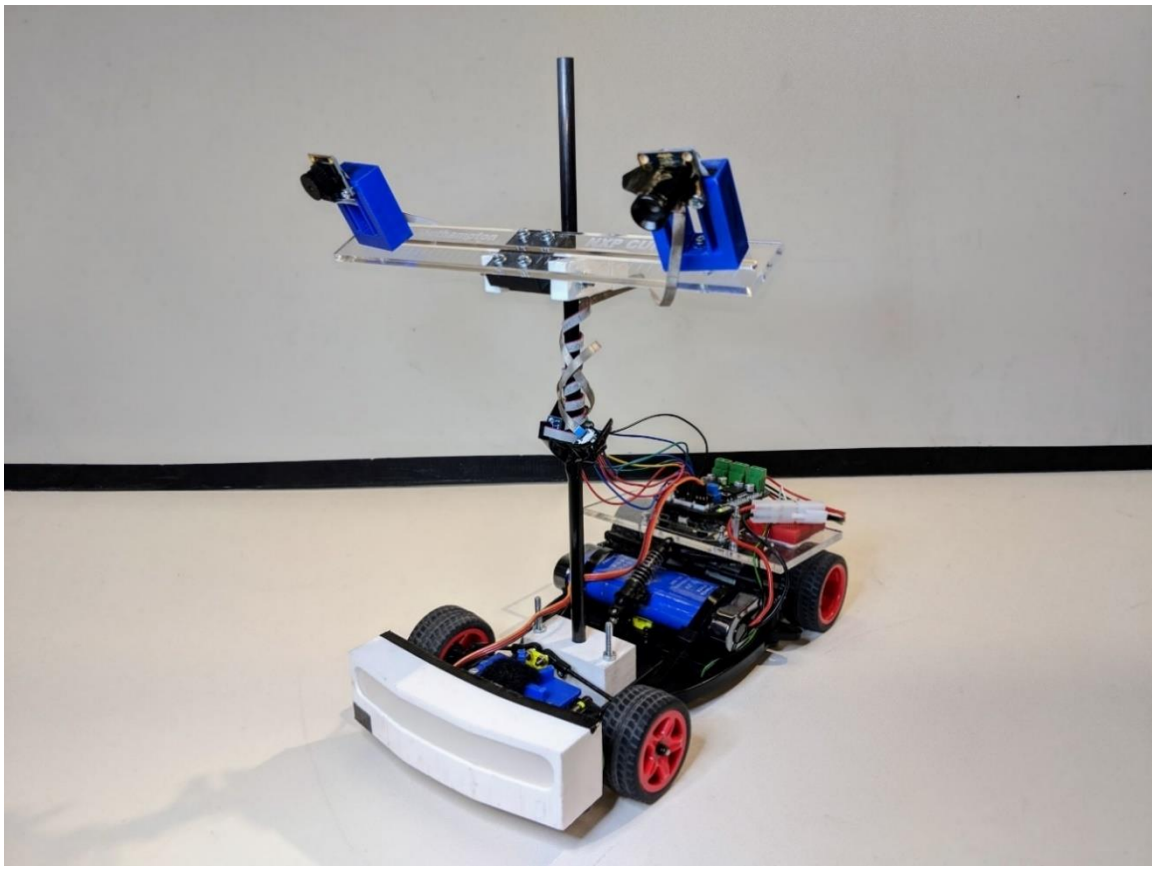
- FRDM-KL25Z [21]: The processor board used to control all the functions on the car. It features a 48 MHz ARM processor, with 128KB of flash memory for storing programs, and 16KB of SRAM. Programming of the board takes place via an online SDK, known as the Mbed compiler[22]. Mbed supports programming in C and C++, it is managed by ARM with the goal of making embedded programming more accessible. It does this mainly through simplifying the setup of registers and IO through built in functions.
- TFC Shield [23]: This is a breakout board designed specifically for the NXP cup. It consists of two H-bridges to drive the motors, a pair of 5V servo outputs, a pair of camera inputs, two ADC inputs, and a variety of buttons, switches and potentiometers which are programmable using a supplied Mbed library.
- Cameras – TSL1401 [24]: The cameras are simple linescan cameras, which produce a 128*1 image. Each ‘pixel’ is a single analogue value, which is equivalent to the brightness incident on that part of the sensor.
- Motors: The drive motors are 7.2V, 5A DC motors. The car is fitted with two, allowing each rear wheel to be driven independently. The servo is a simple S1213 model, commonly used for a range of hobby projects.
- Battery Packs: The batteries are 7.2v units, using a NiMH (nickel–metal hydride) chemistry. They are rechargeable, and less volatile than an equivalent lithium cell.
- Chassis: The camera mounts, with the exception of the carbon fibre rod, headlight array, and board mount were all manufactured by us, the rest of the parts visible are standard chassis parts that came with the car kit.

9 Physical Design and Hardware

The first step in getting the car going was simply assembling a working car from the boxes of parts left over from the last few years of competition. We received one working car that required few changes, and a second backup chassis that appear new, however was missing some essential drivetrain parts, a few chassis pieces, and a rear wheel. Similarly, various stand-offs and screws had been stripped or damaged, and had to be replaced. Therefore, the project began by simply putting together a working car from the mixture of parts. Following this, new camera mounts, headlights, board mounts, and breakout boards were designed. As most of the manufacturing could be done rapidly via 3D printing and laser cutting, iterative design methods were used for many of the parts. Laser cutting tended to produce far more accurate results than 3D printing in a fraction of the time, and therefore this became the preferred method of construction.

9.1 Final Chassis and Car Overview

Figure 3: The car in its final state.



The final car design is pictured above in Figure 3, showing the camera mount, board mount, and headlights. The design process and work behind each part will be explained in the following chapter. Overall, the physical design was very successful, with no parts holding the performance back and no large breakages due to crashes. Throughout probably hundreds of crashes during testing, the only damage sustained was a small crack in the diffuser in the lighting array, which did not affect its performance. As it tended to sustain the full force of most impacts, this is impressive.

9.2 Manufacturing Methods

9.2.1 Methodology Notes

It was decided to use mainly M3 machine screws as they matched the standard screws used throughout the car, and therefore all the pre-drilled holes, aiding attachment of components. Using a standard size of screw also meant any re-assembly following travel to the competition or component changes wouldn't require excessive amounts of parts to be carried. Various M3 screw lengths were readily available from the ECS stores, whilst RS Components was used to acquire a few sizes not available in labs, alongside locknuts and wide radius washers.

9.2.2 3D Printing

The 3D printed parts were all produced using the 'Up! Plus 2' [25] 3D printers available in labs, printed in PLA. The printer can print parts up to 140 x 140 x 135 mm, however only the headlight unit came close to using up this volume as it was 140mm wide. The flexibility provided by 3D printing allows for complex shapes to be produced. This allowed for several neat design touches, such as making captive slots for nuts, allowing pieces to be attached easily despite the tight spaces and improving the overall finish of the vehicle. Despite this, the 3D printers had a fair share of issues, the specifics of which will be discussed in the next few sections. Prints often failed partway through for no discernible reason, or the spacing between bolt holes or the diameters of the holes would be slightly off, resulting in some parts not fitting together at all or requiring manual modification. As many of the parts took several hours to print, these issues resulted in lots of time and material being wasted.

9.2.3 Laser Cutting

The laser cut parts were cut using the Epilog Laser Mini 24 [26] laser cutter. The laser cutter can cut wood, plastic, and thin sheets of metal. It has a cutting area of 600mm x 300mm, which means far larger components can be produced than with the 3D printers, however only 2D designs can be realised easily. 3D parts can be made by using multiple layers of material stacked together, however this causes a complex design process and potential joining issues. The advantages of laser cutting lie in the accuracy of the machine. The laser head is belt driven via stepper motors, which allows for extremely accurate cuts, and smooth curves to be produced. Cut times are also on the order of minutes, rather than hours, leading to far quicker iterations of parts if necessary. 5mm Perspex (acrylic) was used for most of our parts, as it provided enough strength for our needs, without being too flexible. The acrylic is very brittle, so care must be taken when screwing parts to it. This issue was countered by using large radius metal washers between fixings and the acrylic to spread the force across a larger area.

9.3 Camera Mounts

One of the most important parts of the car, aside from the electronics, is the camera mounting. Various designs have been used previously in the NXP cup, with most teams opting for a single camera mounted high up in the centre of the car. Other options have included low mounts very near the front, and even a '3rd person' camera mounted above and behind the vehicle. The most successful cars from past years tended to use the single, centre mount approach.

9.3.1 Requirements

To begin the design process for the camera mount, a list of requirements was established (Table 1), in conjunction with the data we had gained from initial handheld camera tests. These tests showed that the camera tended to be most reliable when looking straight downwards toward the track and had a maximum range of 40-50cm. At shorter ranges, the edges could still be detected successfully, however the narrow field of view (~60°) meant that the edges of the track were out of view when the camera was positioned in the middle. This would mean the car would struggle to centre itself. Based on this, the ideal location for the camera was surmised to be as high as possible, as far forward as possible, and mounted looking downwards at an angle greater than 20° from horizontal.

Table 1: Camera Mount Design Criteria.

Requirement	**Importance and Justification**
Height Adjustment – Must be stable and allow repeatable positioning.	Essential – A range of camera heights will need to be tested to find the optimum point.
Angle Adjustment – Must be stable and allow repeatable positioning.	Essential – A range of camera angles will need to be tested to find the optimum point. Must be stable and allow repeatable positioning.
Strong – Should not break under reasonable impacts.	Essential – The car is likely to be crashed multiple times during testing, and the final mount will need to survive being transported to qualifying rounds.
Remains in Set Position – Will not move under reasonable impacts and during use.	High – The camera should not move easily during handling, operation, and most crashes that occur, otherwise time will be wasted regularly re-aligning the camera
Left/Right angle Adjustment – Must be stable and allow repeatable positioning.	Medium – The camera should always be pointing forwards and centred on the car; however, a small amount of adjustment may be useful if we discover the camera sensor isn't aligned on the PCB, or the mounting holes on the car chassis are slightly misaligned.
Lightweight – Does not add significant weight to the car.	Medium – The car itself is reasonably lightweight, and as such sometimes lacks grip when cornering. As the camera mount is likely to be tall, weight should be concentrated as low as possibly or stability may be impacted, due to the raised centre of mass.
Reusable Parts – Mount should be modular is possible.	Low – As the camera mount is likely to go through several iterations, the ability to reuse certain parts if needed will save time (3D printing) and materials. Repairs will also be quicker.

9.3.2 Single Camera Mounts

9.3.2.1 Fixed MK1 and MK2 (3D Printed and Laser Cut)

A very simple mount was designed first, mainly to assess the capabilities of the 3D printers, consisting of a single fixed piece, inclined at a 20-degree angle. This mount would be attached to the very front of the car, and perhaps unsurprisingly was not very good. The 3D model of this design is shown below in Figure 4. The subsequent Figure 5 shows the printed model on the left.

Figure 4: The Model for the MK1 Mount

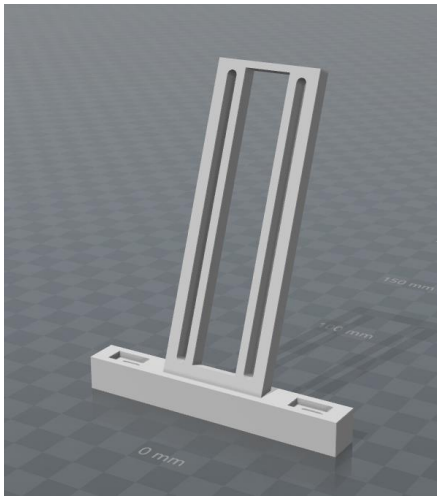
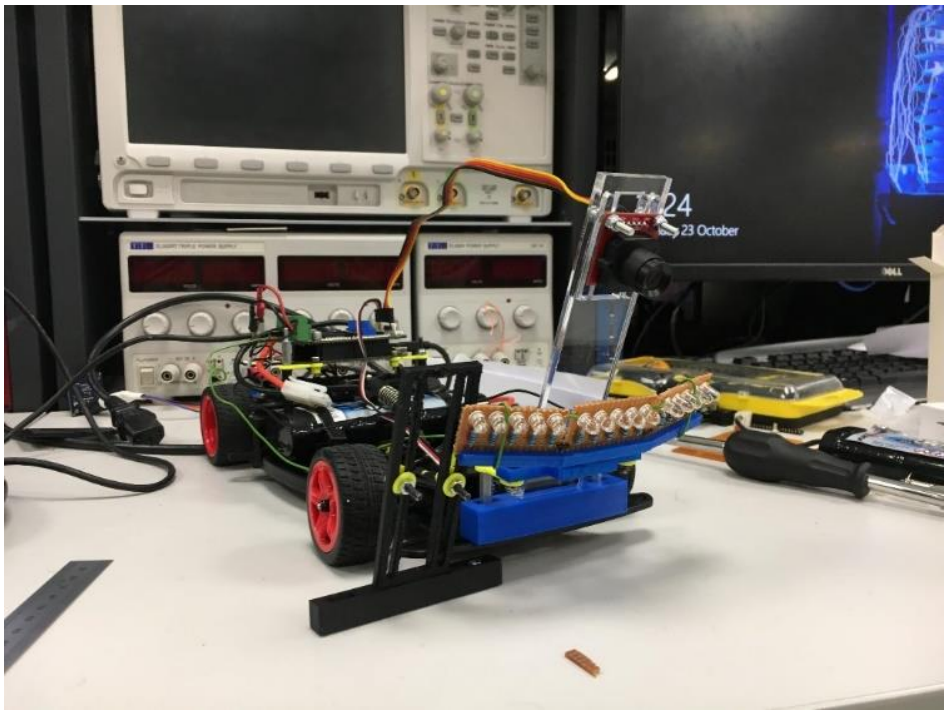


Figure 5: The MK1 Mount (Left) and the MK2 Mount (Right).



The mount was successful in that it held the camera in place and allowed for the first iterations of the control code to be tested but had many limitation and issues. The three long slots provide space for mounting bolts, and the camera connector, however the long thin sections were very weak, which meant large washers were required (Shown in Figure 5 on the left mount) to screw the camera on without them bending and potentially

snapping. On the upside, it provided a solid proof of concept for several ideas that were to be used throughout the manufacture of other parts. As shown in, the shallow slots on the left and right sides are slots for nuts, which are sized so that the nuts are held in place, allowing the mount to be tightened in place easily. These mounting points are also slotted, which allows for slight left/right adjustment, and compensates for any measurement/printing inaccuracies. The later MK2 mount, shown on the right in, improved upon the first design by using a laser cut acrylic section to mount the camera on. Acrylic is far more mechanically stable than PLA and had no flexibility issues like the first design. This acrylic piece simply slotted into a 3D printed base block which was then attached to the chassis. This design was far more successful and allowed a large amount of testing to be carried out. Due to improvements in the control algorithms, the limited height, and therefore view of the track became a limiting factor. A longer piece of acrylic could be installed; however, this style of mount did not allow for any adjustments other than height, which did not meet the design specification and limited the variables we could change during testing. Furthermore, this design caused the camera to be above and nearly in front of the headlight array (this is pictured in Figure 3 and detailed in the Lighting section) this caused issues with glare 'whiting' out the camera image. Due to the numerous limitations with this design, an entirely new mount was designed.

9.3.2.2 Tall Central Rod Design MK1 and MK2 (3D Printed)

This design consisted of a base block mounted to the chassis, and a hinged camera mount, joined using a rod. This design provided adjustable height and camera angle and was attached centrally in the car. Pictured below in Figure 6 are the three components that went into the initial design. From front to back, the camera mount, upper section, and base are shown. The camera mount fits into the upper section and is attached using a thin threaded rod to create a hinged joint, tool free adjustment is possible as it is fixed using a lock-nut on one end and a wing-nut on the other. A 250mm long M10 threaded rod was sourced to form the upright rod. The upper sections hole was designed to be slightly smaller than the outside thread diameter of the rod, which meant the PLA could be tapped by hand and threaded onto the rod. A lock-nut above and below secured the piece in place securely. Finally, a similar tapping method was used in the lower base block to attach the rod to the base. In the diagram below, a captive hexagonal hole can be seen in the rearmost part. This doesn't go the entire way through the part and allowed a lock nut to be inserted within the block. When installed, the nut would be hidden entirely, resulting in a neat finish and a strong, stable connection. Finally, the entire mount could then be attached directly to the chassis using long M3 bolts. The drawback to this design was the overall weight, the threaded rod was extremely heavy, nearly doubling the weight of the car. The weight itself was not a large issue as the drive motors are very powerful, however the centre of mass of the car ended up being raised significantly causing understeer and body roll through corners.

Figure 6: Third Camera Mount Components.

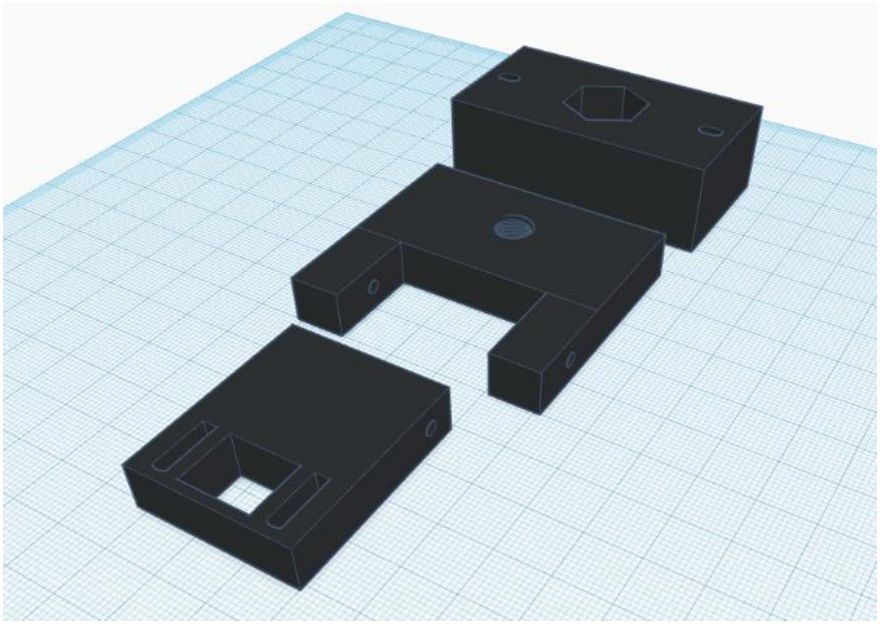
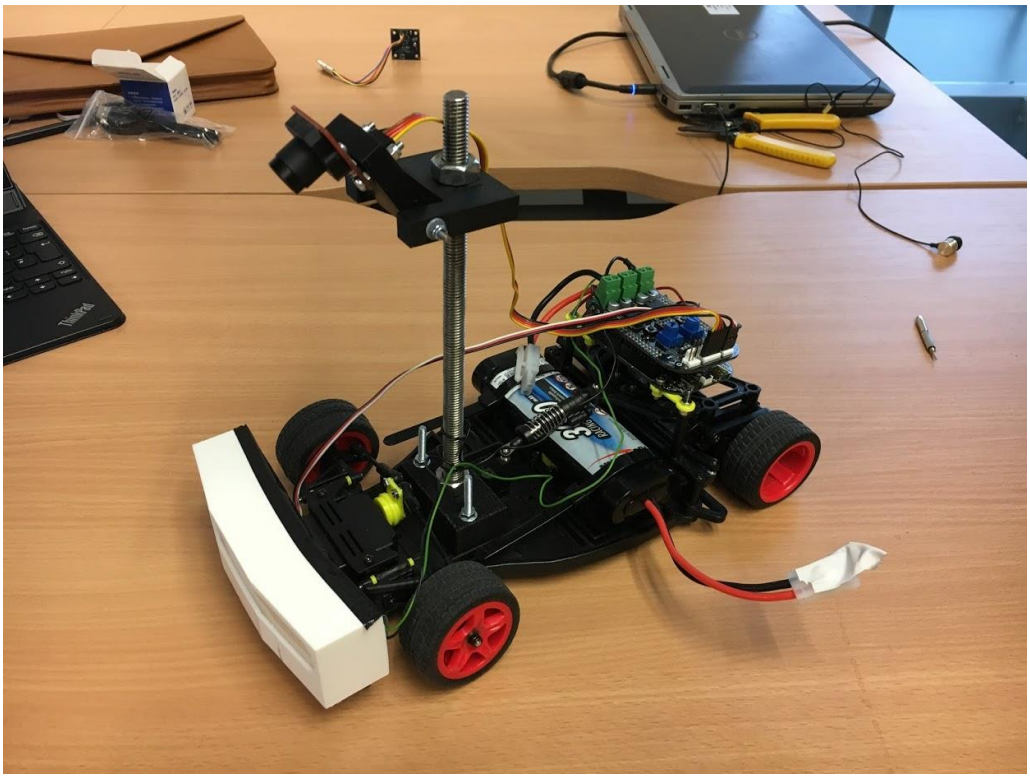


Figure 7: Constructed Mount, Fitted to the Car.



Following this, the threaded rod was replaced with a lightweight carbon fibre rod, meaning the upper and lower sections had to be redesigned. The upper section was split into two parts, to form a clamp (Figure 8) that could be tightened using bolts, whilst the base required a more complex redesign. A similar clamp could not be used here, as there are limited fixing points on the car chassis and getting all these points aligned with a moveable clamp would be extremely difficult. Therefore, a design that used bolts to clamp the rod into the base was created.

This is detailed in Figure 9 as it is best explained with a diagram. Vertical slots sized for an M3 nut were placed on either side of the carbon fibre rods, with holes left for bolts to

be inserted from both sides. These would then screw through the nuts, and press against the rod, clamping it in place. This is not an ideal way to secure carbon fibre, as it tends to crack under direct, high pressure load, however not much force is needed to hold the rod adequately, so as long as care is taken the risk of damage is low.

This design proved highly successful and is shown in full in Figure 11. The carbon rod had to be shortened for the competition to ~320mm as otherwise the car would exceed the height limit. This design met all the design requirements given. The majority of these parts are still in use on the final car, having not suffered any breakages. The hinged design allows just the camera mount part to be replaced when needed if cameras are changed. The various methods used to attach the rod allow for movement if they are involved in a crash. This reduces the force on the cameras, various connectors, and mount itself if the car crashes.

Figure 8: Upper Clamp Drawing (Dimensions in mm).

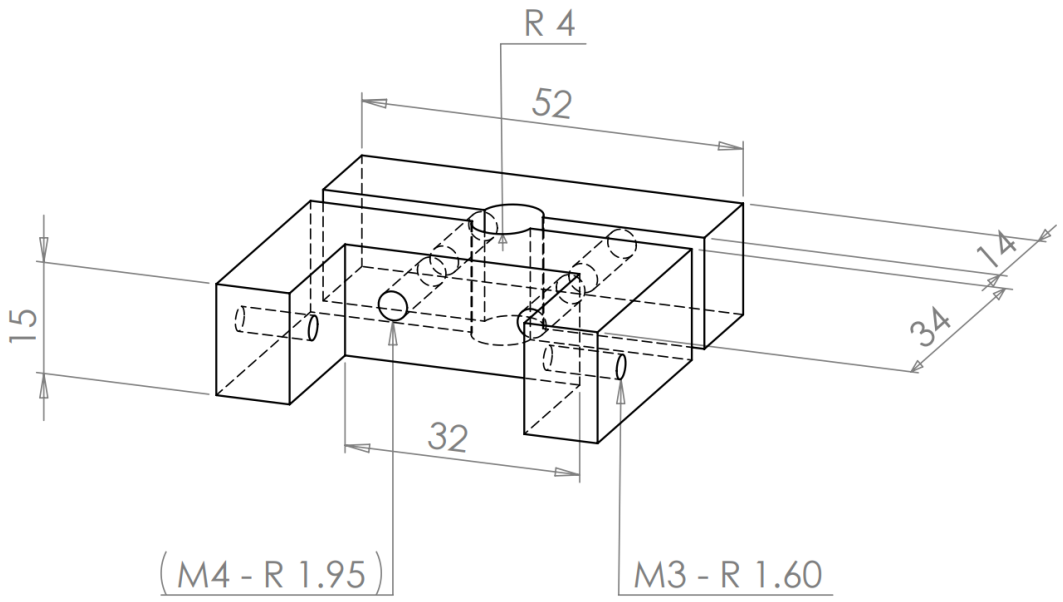


Figure 9: Base Block Drawing (Dimensions in mm).

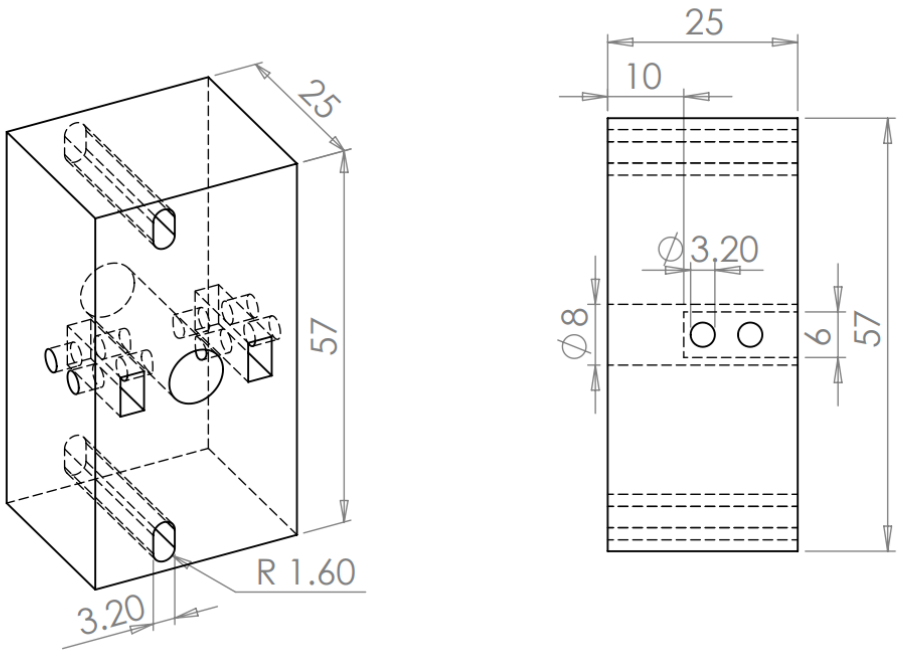


Figure 10: Various Components Being 3D Printed.

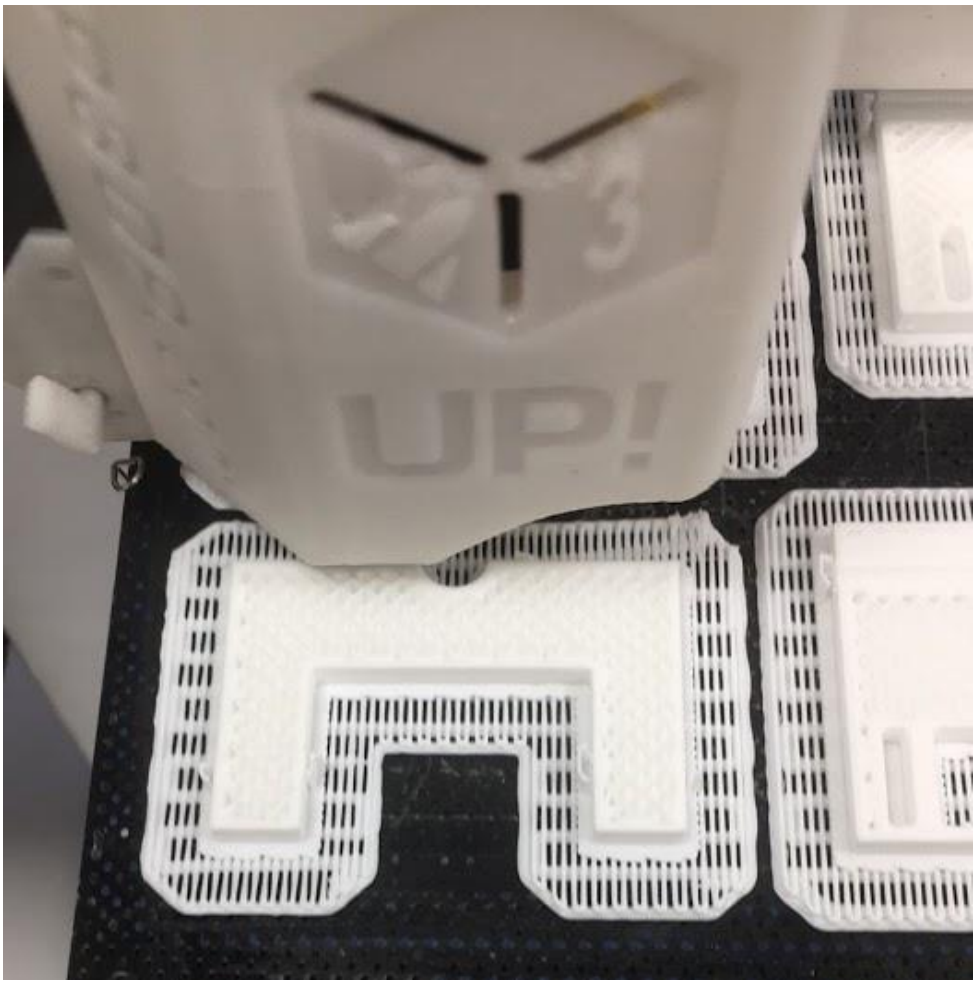
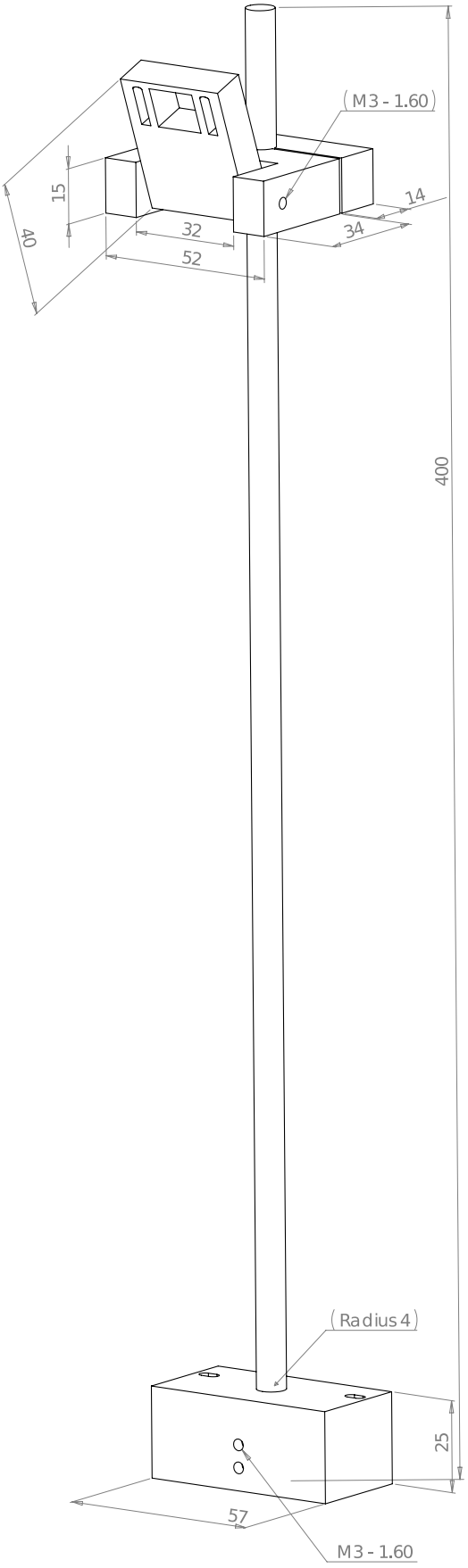


Figure 11: Drawing of the Assembled Mount.



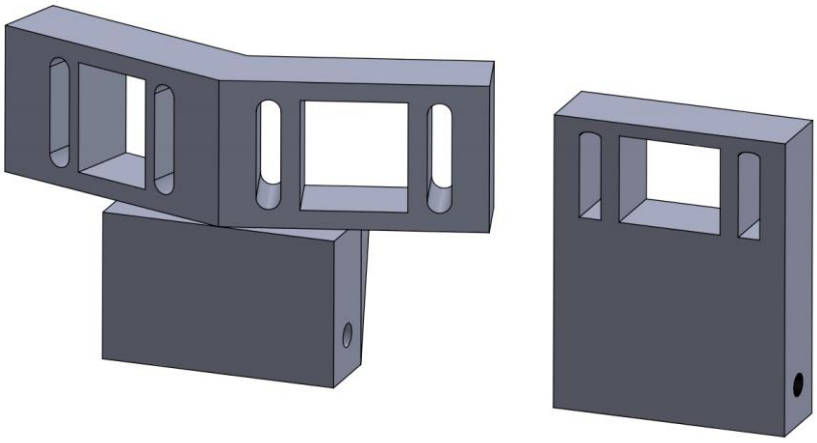
9.3.3 Twin Camera Mounts

During our tests, we established that the cameras did not have a wide enough field of view to see the entire track in certain situations. During straights, this caused small oscillations, which sometimes caused the car to enter a corner at a non-optimal angle. For complex sections of track, like S bends, the car would end up too far from the centre of the track and risked crashing. As we could not raise the camera any farther, or look further down the track reliably, it was decided that we would attempt to use two cameras that looked outwards slightly. These two images could then be stitched together in the software, and if aligned correctly should give a much wider field of view.

9.3.3.1 Initial Twin Camera Design (3D Printed)

The first design, shown in Figure 12 built upon the initial single camera mount. It was compatible with the previous design, meaning it could be tested quickly, with only one 3D print required. The angle was chosen based on the previously assumed field of view. This fixed angle proved to be one of the downsides to this design. For the cameras to see the track, it had to be tilted downwards like all the other camera mounts, however, this meant that the two cameras ended up looking at the track at a diagonal. The fact that we did not have two same identical cameras intensified this problem. As one camera suffered from vignetting, one had a narrow field of view and one had a strange, curved 'fish eye' view, the cameras required alignment or for lots of the central data to be ignored. As we could not find a source for a duplicate of any of the camera models we did have, the purchase of two cameras would have been required. Despite all these issues, this design showed promise as we could see each edge of the track, however alignment would need to be possible if it were to be useable.

Figure 12: Left: Twin Camera Design with different slot widths. Right: Single Camera Mount showing the similarities.



9.3.3.2 Final Twin Camera Design (3D Printed and Laser Cut)

Taking that knowledge into account, a different twin camera mount was designed, shown in Figure 13 and Figure 14. The laser cutter was used to cut a wide, narrow rectangle with a central slot. On this, two small camera mounts were fitted. These mounts once again used a similar design to all the others, but this time with a small hole in the bottom aligned vertically rather than horizontally. This allowed for independent angle adjustment for each camera to compensate for their slightly different views and to minimize overlap in the centre. The laser cut section was then mounted to a small T shaped section, which was attached to the previous mount to form another hinged joint.

This mount was very successful and is the one used in our final product. It is still slightly hampered by the two different cameras; however, the range of available adjustment makes this a solvable problem. Throughout testing it has proved strong enough to survive repeated crashes, and whilst driving stays in place. Crashes tend to make the cameras move slightly, resulting in a loss of alignment which can have dramatic effects on the operation of the car, and therefore a fixed position mount will be made for the competition.

Figure 13: Final Twin Camera Design.

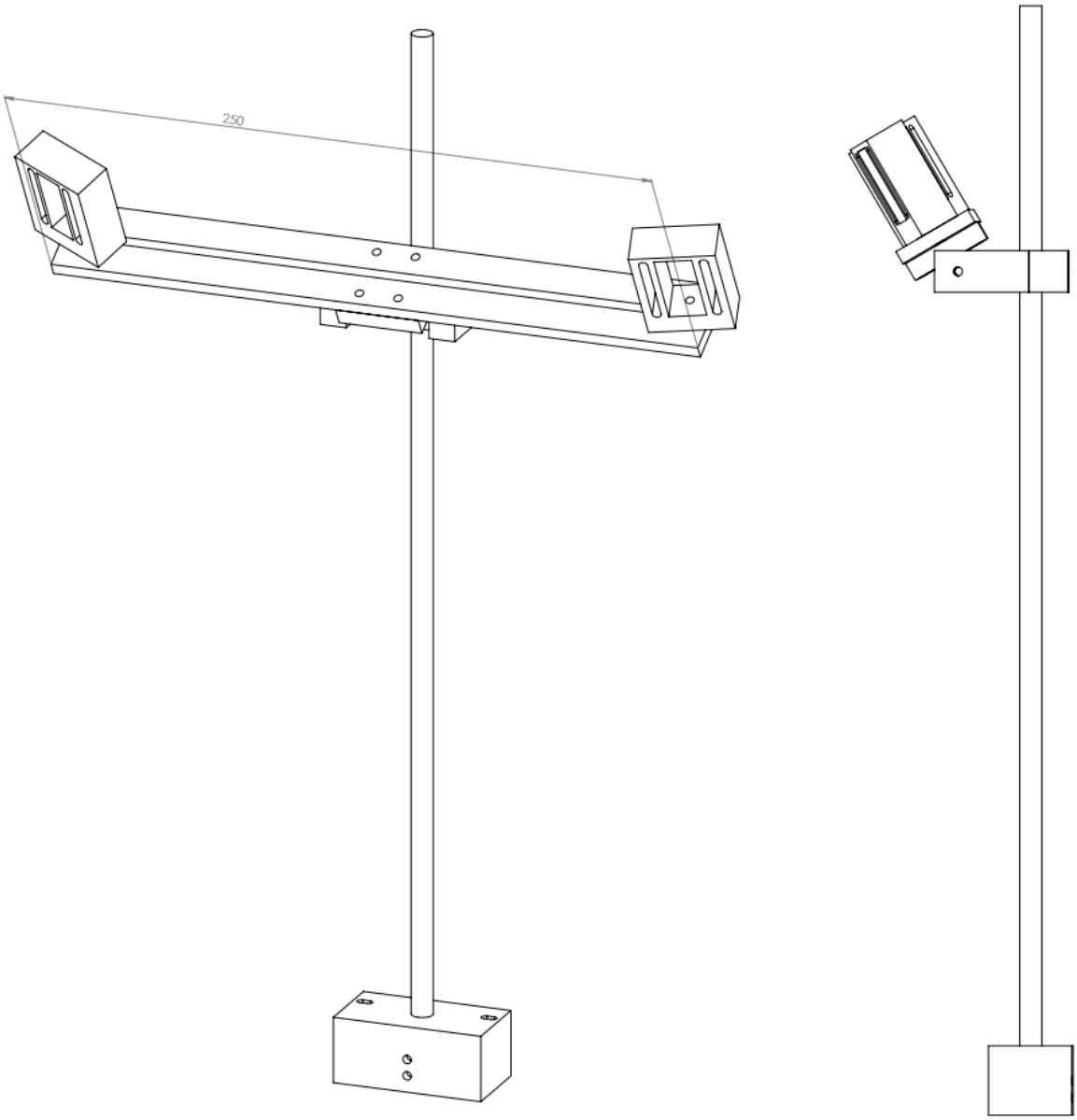


Figure 14: The Laser Cut Section. The graduated markings every 5mm assist in positioning the cameras.

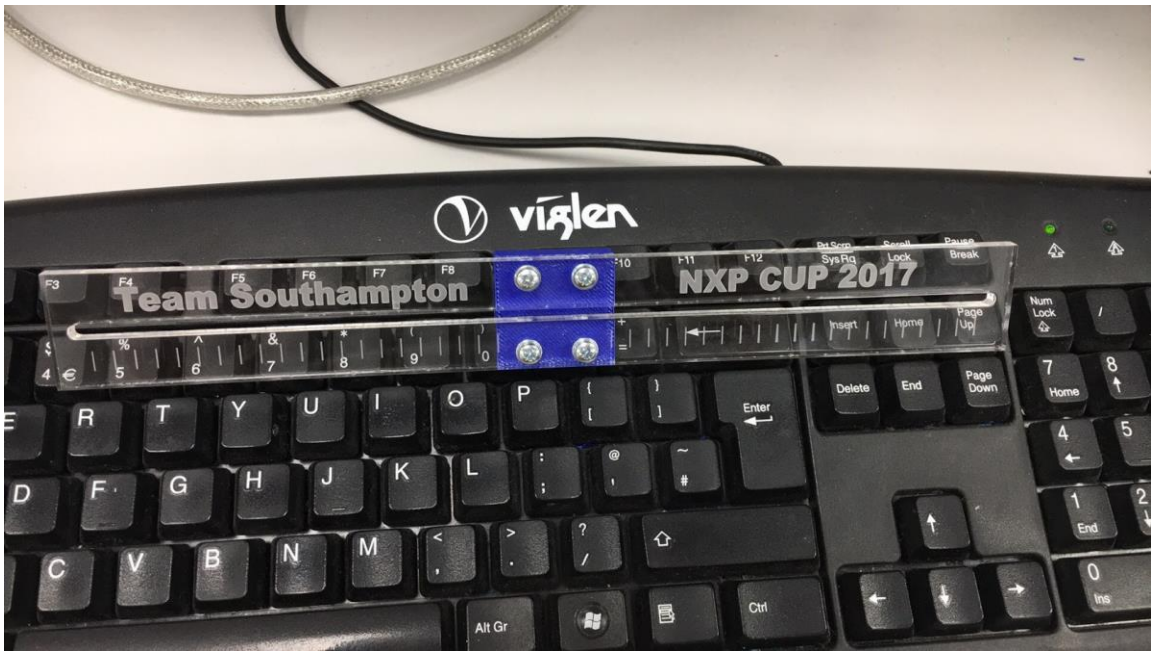
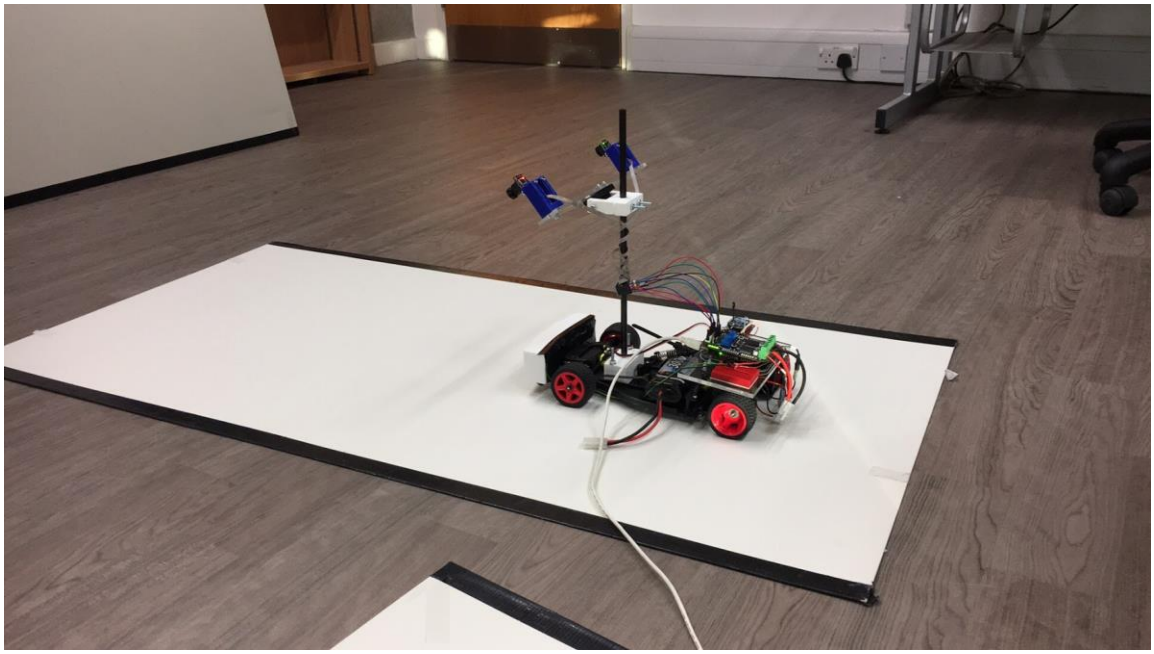


Figure 15: Initial Testing of the Mount.



9.4 Lighting

The reasoning behind adding headlights to the car are twofold, firstly, there is the potential for the track to include a tunnel, which at worst could cause the car to mistakenly identify a track edge and crash. Secondly, the lighting array would increase the contrast between the black track edges and the white track, which increases the likelihood of the edges being identified correctly and reduces the effect of variable lighting conditions are present whatever tests are taking place.

Initial tests were carried out using a simple LED array constructed by the group from last year, and they showed a general improvement in following the track, however, the array was not very stable, not mechanically strong and was then unfortunately damaged in a crash. There were also the glare issues mentioned in the previous section. Their design used 3 sections of stripboard, each with 5 LEDS. This was mounted on a 3D printed part, which was composed of 3 differently angled straight sections to provide a wider beam, however, the lighting was not uniform, and on occasion the car seemed to detect the dark patches as track edges and would crash due to this. With knowledge of those limitations in mind, a set of design criteria was established.

Table 2: Lighting Design Criteria.

Requirement	**Importance and Justification**
Even Lighting – No distinct change in beam intensity.	Essential – The dark patches may be detected as track edges.
Strong – Should not break under any impacts.	Essential – The car is likely to be crashed multiple times during testing, and the light array will be mounted on the front of the car, and therefore is likely to bear the brunt of most crashes.
Bright – Illuminates enough track to make a difference.	High – The lighting would just be a waste of power if it didn't have a far enough throw, however this depends on ambient light levels as well.
On/Off Switch – Easy way to disconnect the lights from power.	Medium – A large array of LED's is likely to draw a large amount of power and may not be needed in all situations.
Adjustment – Variable direction or intensity.	Low – Angle adjustment provide a benefit in the throw distance of the light; however, this provides a point of failure for unknown benefit.

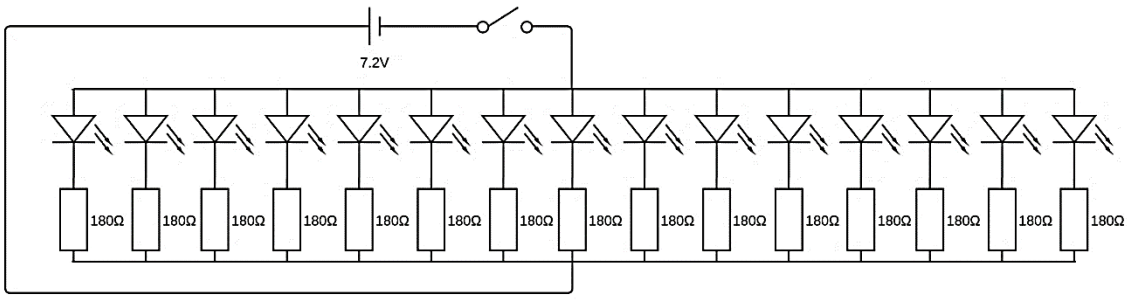
9.4.1 Circuit

As the array from last year had been damaged during testing, a new one had to be built. Once again, 15 LEDs were used, in a parallel configuration. High output Cree C512A-WNN 9000K White LEDs [27] were used, and in particular were chosen due to their cool white output. This ensures the track is brightened up as much as possible compared to a warmer tone, resulting in greater contrast. The parallel configuration was chosen to ensure even if one fails, the majority of the array will continue working. The circuit diagram is shown below in Figure 16.

The circuit itself was constructed on three small sections of stripboard. Wire was used in connecting each section, providing a flexible join. A curved or flexible PCB was considered; however, it was quickly deemed to be too complex and expensive to justify, especially when stripboard gave acceptable performance. A simple inline switch was used, which was mounted to a breadboard next to the KL25Z on the board mount. The

power connections were soldered directly to the battery inputs on the TFC, as there are no 7.2V outputs available. This also means the array is powered in parallel to the board itself, meaning the car can be run with the lights off. Ideally, the lights would be powered separately, but only one battery is allowed. The circuit draws ~300mA from the battery. Therefore, during operation the diodes dissipate 1080mW, whilst the resistors dissipate a further 1110mW, totalling just over 2W. This results in ~50% efficiency, which isn't ideal, however a buck converter could not be used to step down the voltage as there were no available PWM outputs on the TFC. Compared to the power used by the motors, 2W is not very significant. Furthermore, the battery packs being used are rated to at least 2500mAh, and therefore the reduction in run time due to the array is not significant.

Figure 16: LED Array Circuit Diagram.



9.4.2 Housing

The headlight array was to serve a dual purpose, also acting as a bumper for the car, and as such has to be very robust. To eliminate any glare in the cameras, the entire array was enclosed, whilst to remove the patchiness of the light output, a diffuser was designed as well.

9.4.2.1 Body

The main body of the light array proved to be unexpectedly complex to design. A variety of lofted cuts and complex curves were needed to aim the light beam. Adding to this complexity, a narrow, curved slot had to be created to hold the diffuser in place. The housing was made to be as wide as the car, this would protect the fragile wishbones holding the front wheels in place from any fontal collisions. The car came with a small bumper as standard, and those mounting holes were utilized to attach the light to the car.

, the housing is composed of three flat sections, which form a convex curve when viewed from the front. The gap for the LED's to shine through is tapered from ~8mm at the back, to 18mm at the front, which should help direct the light down onto the track. There are slots on either side of the rear section to help hold the stripboard in place. The semi-circular slots at the base provide space to insert nuts, with the bolts being inserted from below the chassis into the housing. Narrower, captive slots could have been used, however these sometimes 3D print at slightly incorrect sizes, due to the 10-hour print time, these larger slots provided a safer, simpler design choice. They are tall enough for a standard spanner or needle nose pliers to be inserted to hold the nuts in place whilst tightening. The diffuser and LED array can both be removed without removing the housing itself, so it is also unlikely that the nuts will need to be used very often.

Finally, there is a slot that spans the entire front of the housing, shown in Figure 17. This slot is used to hold the diffuser in place in the body in a neat manner, whilst also recessing it slightly, providing protection. The diffuser is a small, 140mm x 35mm section of 1mm

thick acrylic, which etched using the laser cutter. The diffuser is used to scatter the relatively directional light from the LED's, removing the light and dark patches from the beam. A high speed, high resolution random noise pattern was etched across the entire piece. This created a 'frosted' texture on the front face of the acrylic. Testing the array with and without the diffuser showed good results (Figure 18), the array illuminated nearly the whole width of the track in quite bright light conditions, whilst eliminating the dark patches. Tests with an Android light meter app [28] showed a drop in light levels of approximately 45%, which is quite high, however the headlight still provides adequate brightness, and this would be hard to improve upon with the available equipment.

White PLA was used to print the array, as this will increase the light emission from the headlight due to its higher reflective properties than other colours. Light output could be increased by coating the inside of the light with a proper reflective material, however performance was adequate without, and this would have also added thermal insulation which may have caused issues if the light was run for a long time.

Figure 17: Base of the Housing

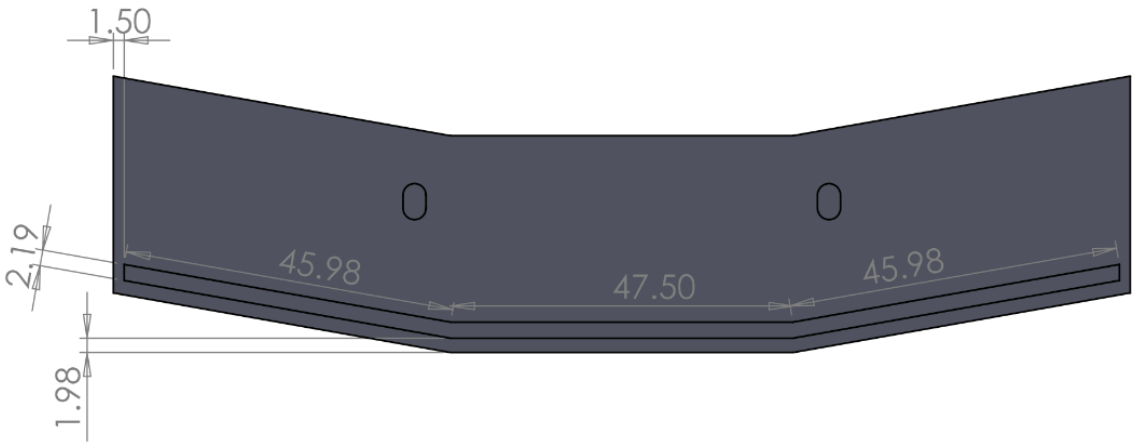


Figure 18: Beam Pattern Comparison. The left image is with diffuser, the right without.

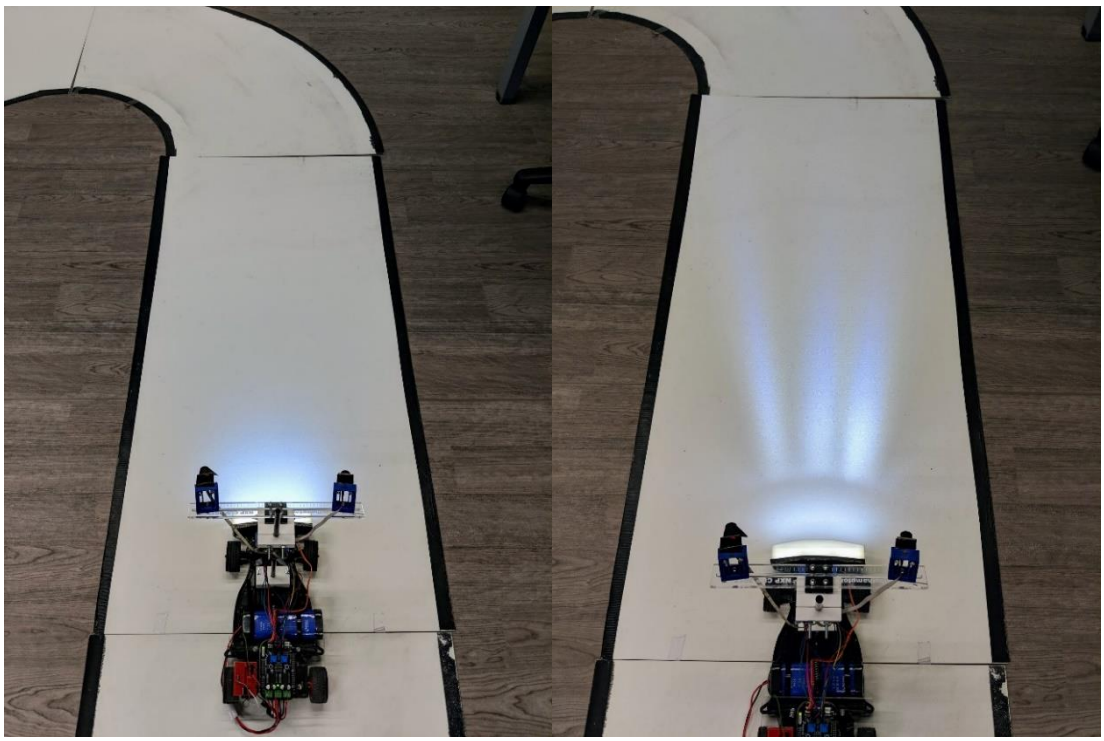
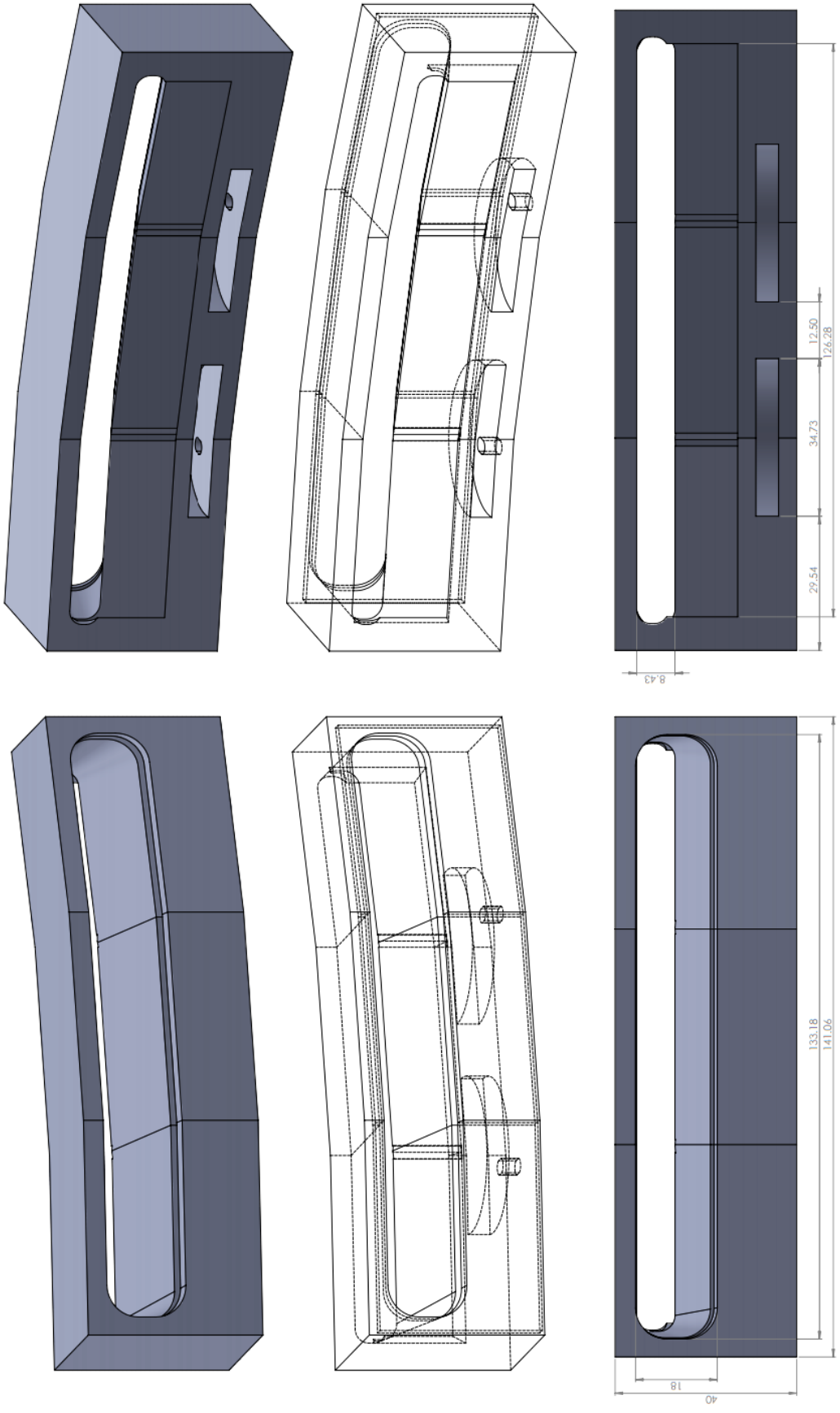


Figure 19: Headlight Housing, the front is the bottom/left set of images, and the back is the top/right set.



9.5 Board Mount

The car came with a selection of plastic stand-offs and mounting brackets, however none of these were able to hold the board securely. On occasion, it would wobble enough that some of the leads would disconnect, either causing a motor to lose power or the board to reset. It also meant using any of the on-board buttons cause the board to flex. As there were no large, flat surfaces on the car, one had to be added. The headlight switch and xBee also needed space to be attached, and therefore a large plate that covered the entire the back section of the car, whilst protecting the back wheels, was designed, shown below in Figure 21.

The plate is made from 7mm thick acrylic, and includes mounting holes for the KL25Z, and another set of holes to attach it to the chassis. When looking down at the car, the rear section of the chassis is an H shaped section of plastic, with an I-beam cross section. The narrow, middle section of the I-beam is almost the same width as the M3 bolts used throughout the car, and therefore to attach the plate, 4 holes were drilled in this section. Wide washers were used to spread the load on the plastic, and to space the plate out so it cleared the top of the wheels. The plate has been entirely successful, with plenty of space to mount a small breadboard, and the xBee, with room to spare to add other components in the future if needed.

Figure 20: Schematic of the Board Mount.

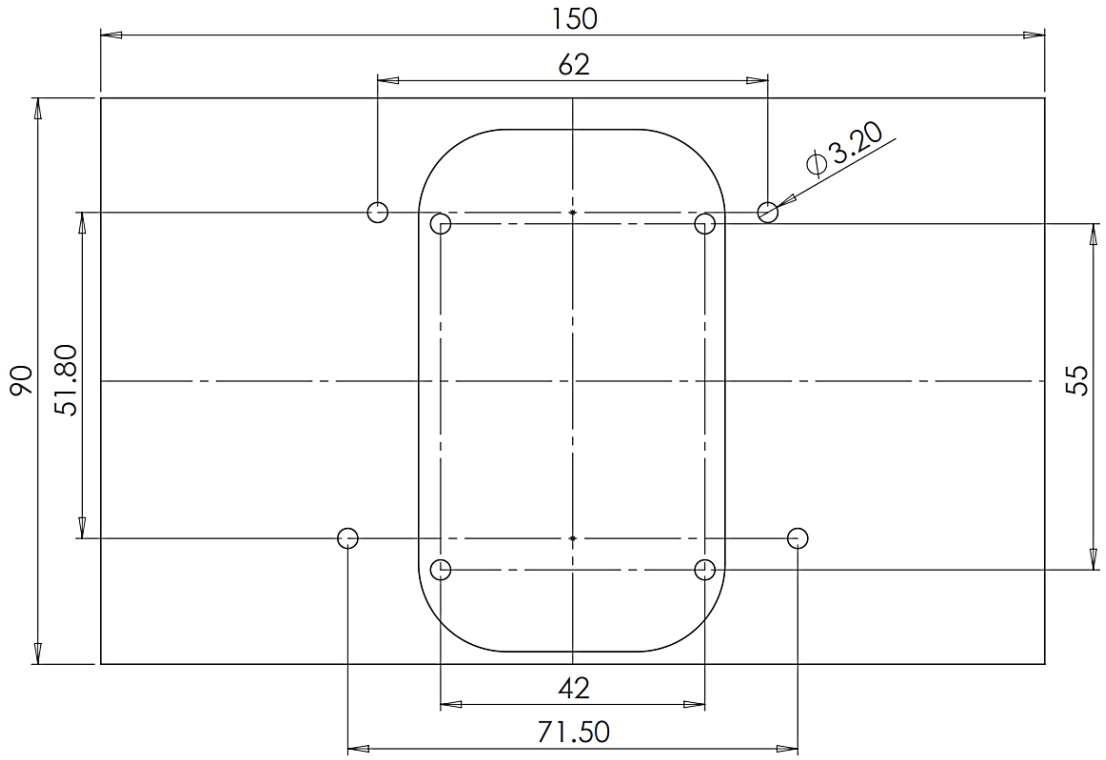


Figure 21: The Board Mount.

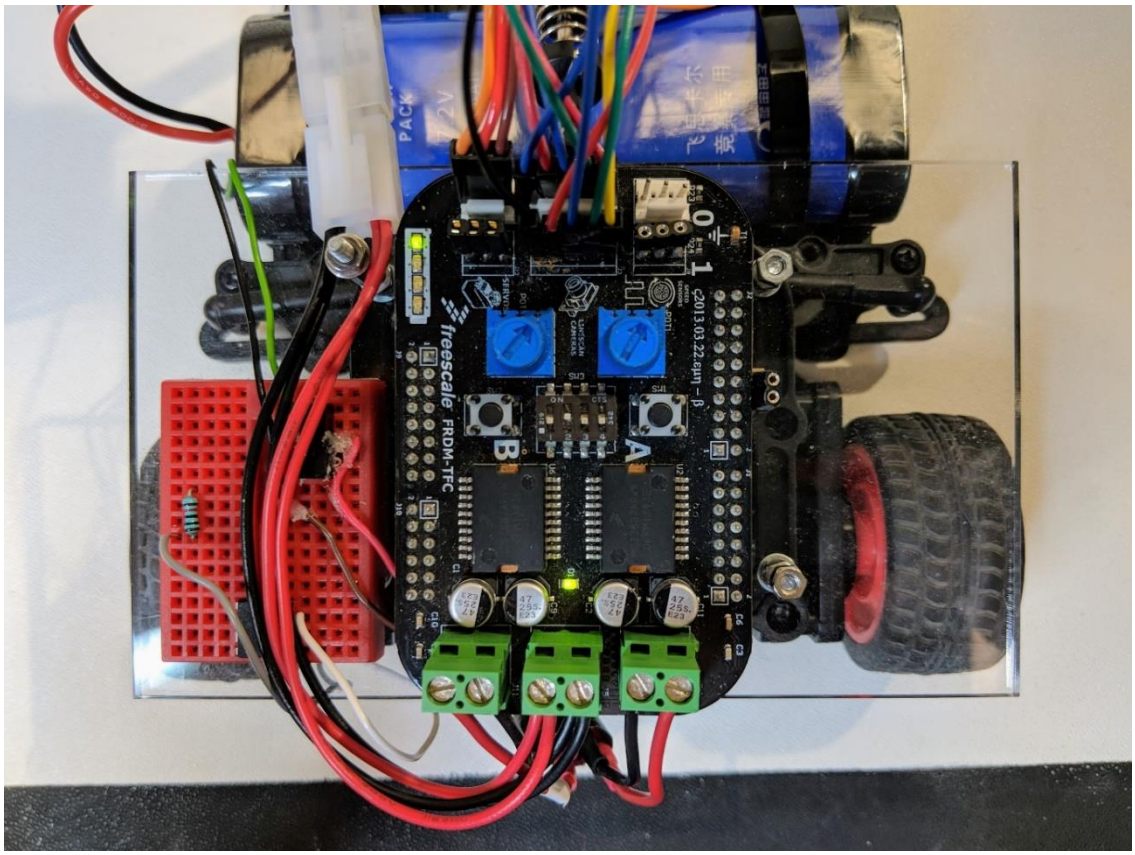
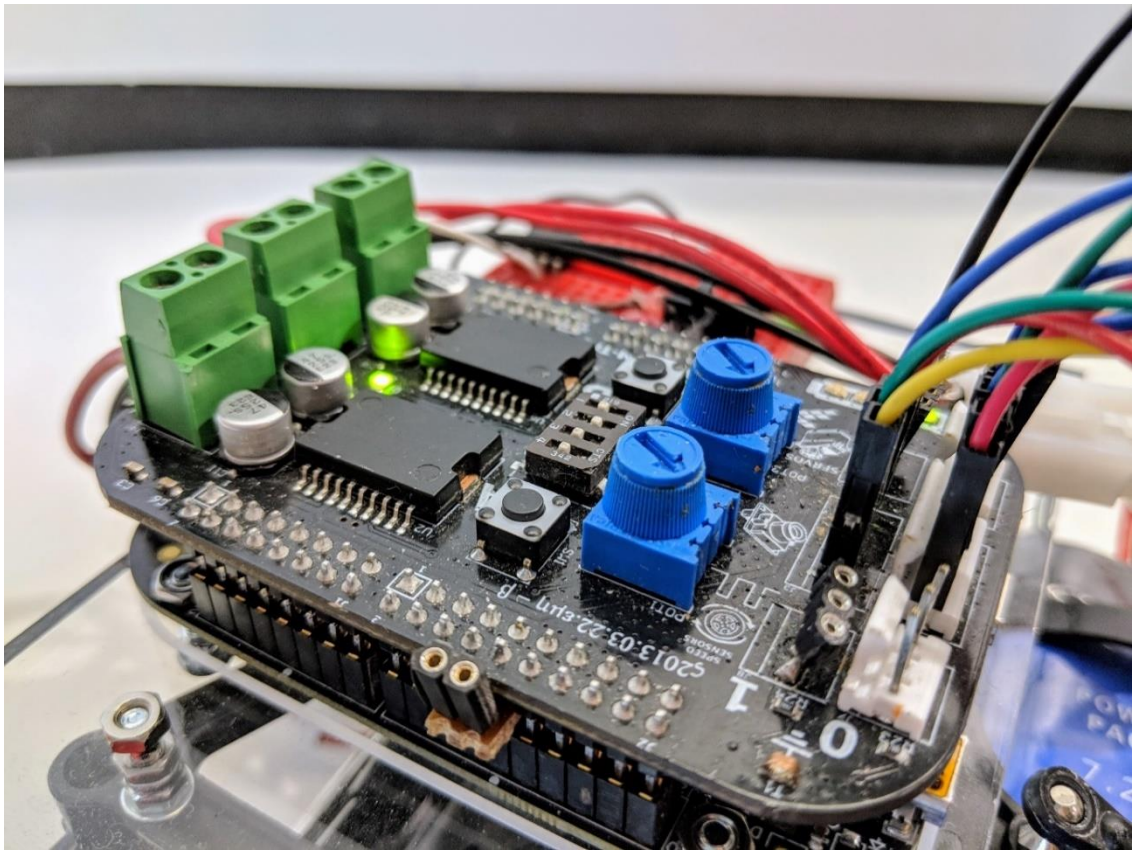


Figure 22: Detail of the attachment method (bottom left of image).

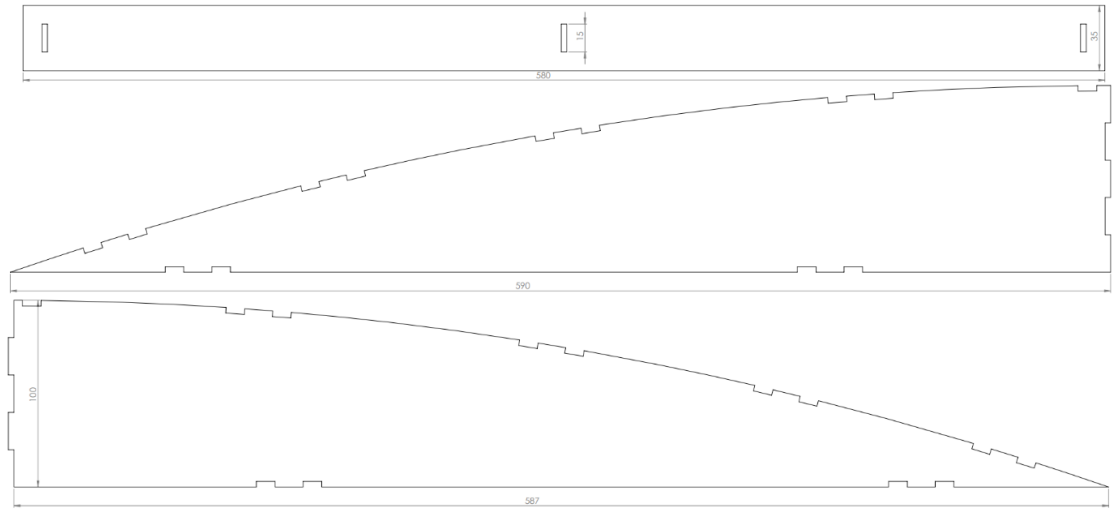


9.6 Ramp Track Section

One of the more worrying obstacles that may come up in the competition is a ramp, as this will change the shape of track the cameras see, and the speed of the car. There is a chance that the car could stall on the upslope, speed up too much on the downslope, or even fly off the side, risking damage and disqualification. Prices for a ramp supplied by NXP ranged from £100-£300, which made building our own the only real option to test this.

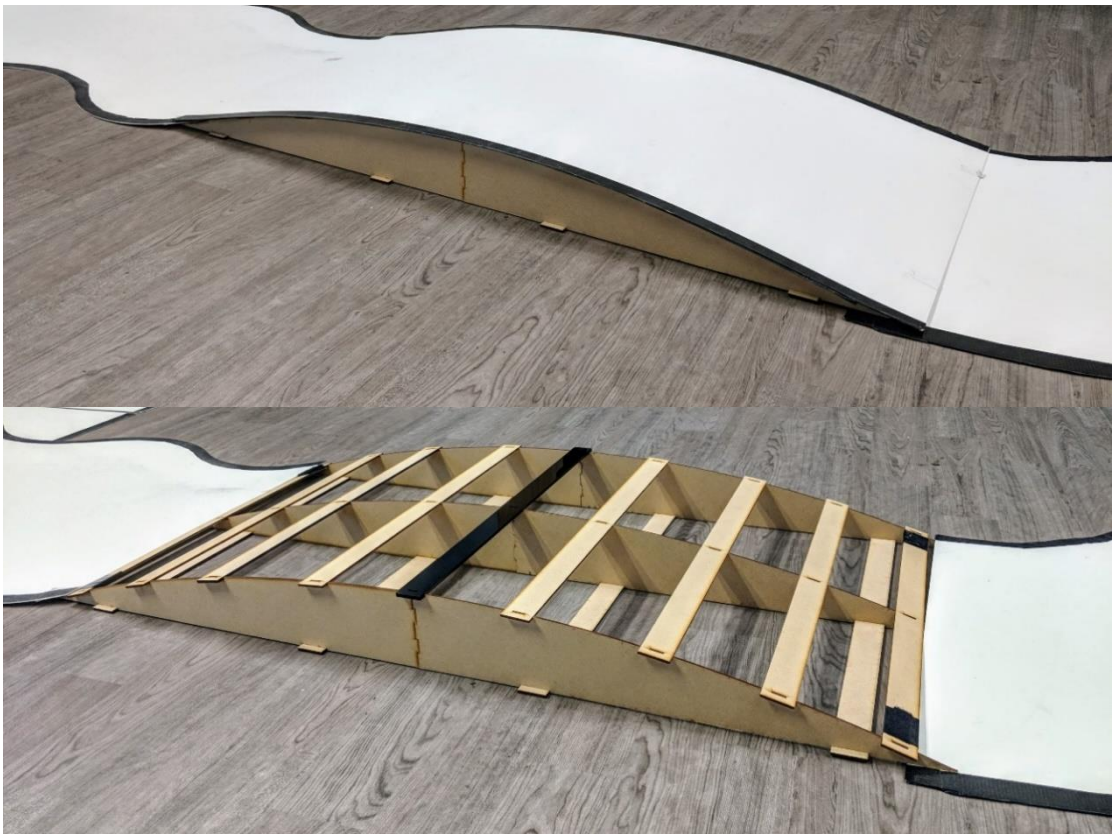
The spec of the hill used in the competition is available online. It follows a simple curve, with a maximum height of 100mm at the crest, and a length of 1.2m, the same as a normal, straight track section. As the university laser cutter uses 600mm x 300mm sheets of material, the frame was designed in two sections. MDF was chosen as it is cheaper and less likely to crack than acrylic. The three different parts used are shown below in Figure 23.

Figure 23: Ramp Components.



The finished product used 13 slats, and 6 ramp sections. The middle and bottom parts shown above fit together ‘back-to-back’. This made 3 full curved sections, two sides and a central one supporting the middle of the slats. The slats simple press fit onto the tabs and hold each section together. It is shown constructed below in Figure 24.

Figure 24: The constructed ramp, shown with and without track attached.



Having cut out all the sections, several minor issues were discovered. An error had been made in the calculation in the tab width at the crest of the hill, so a narrower slat had to be cut to hold the two halves together. Secondly, a tab should have been included at the bottom as well, which would make the overall structure far more rigid. The tabs on the vertical ends of each ramp section end also could have been made wider at the ends, so they fitted together mechanically like a piece of a jigsaw puzzle, rather than just slotting into each other. Despite these niggles, the ramp performs well in practice. A section of straight track was fitted with small squares of Velcro, as were the slats on either end of the track. This holds the track section in place securely, and the ramp together. Testing using the ramp allowed a hill detection function to be developed using the accelerometer on the car.

10 Machine Vision

The NXP cup requires that a linescan (produces an image 1 pixel tall) camera is used to navigate the track and there is no processing or alteration of the camera data on anything other than the KL25Z, this means things such as auto-focus or contrast enhancement are forbidden. Throughout the project multiple cameras were used. Whilst working with the single camera we used a Freescale line scan camera based on the TSL1401CL sensor[29]. Later in the project we implemented a solution using multiple cameras, for this we used 2 newer cameras NXP sent us as part of the new Alamak car kits.

10.1 Camera Interface

The cameras detect light using a line of 128 photodetectors, the more photons incident on a photodetector, the more photocurrent is generated. Therefore, a large photocurrent in a photodetector denotes a brighter area. The light is focussed on the line of photodetectors via a lens, the focus of this lens can be adjusted depending on how far ahead the camera should be looking.

Interfacing with the linescan camera requires 3 signals: analogue out, serial clock and the serial input to the sensor. The serial input to sensor defines the start of the data output sequence, this goes high when data is being read from the photodetectors. The clock signal then determines the data rate. Therefore, for each subsequent clock edge, a reading from 1 of the photodetectors is pushed to the analogue output. [24]

To interface with the linescan cameras we used functions in the open source TFC library provided for the board, this library sets up all the required IO allowing us to interact with the camera data, and provides two arrays to store camera data in, and can be updated at up to 1000Hz. The camera data is stored in arrays of 128 unsigned integers. The typical values for the cameras were 4000 for a track reading (white) and 1500 for an edge reading (black) though the readings for the black track section varied depending on the camera and the lighting conditions, reaching a maximum 3000 in some cases.

10.2 Camera Aims

The aim of the camera section of the project was to determine the position of the car on the track. This position was then transmitted to the control algorithm. To that end the camera needs to be able to detect the dark edges of the track, from these edges the position of the car relative to the centre of the track could be calculated.

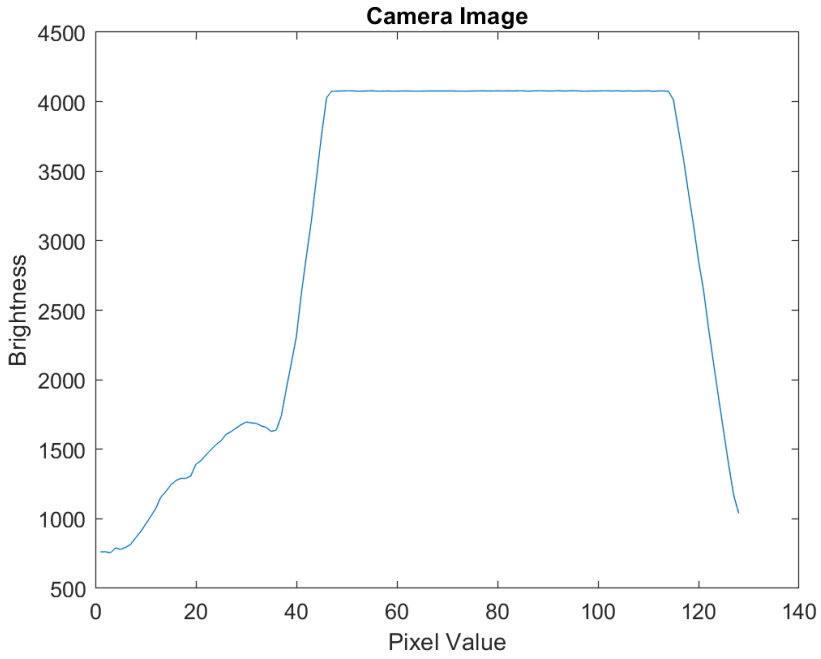
This means the cameras must consistently be able to see and detect at least one edge of the track. The processing of the camera data must not generate false positives from the noise around the track. In addition, the algorithm must be quick, if too much time is taken to process the camera data the control will not be able to react in time to changes in the centre position.

Ideally the camera should also be able to detect the stop marker on the track which takes the form of 2 black rectangles in the centre of the track. However, this was not implemented during the project due to time constraints.

10.3 Single Camera

For the initial stages of the project one camera was used. The issue with the single camera was the heavy vignetting image, meaning the values on either end of the image were always seen as black. This is shown below in Figure 25. The data in Figure 25 was generated by placing the car such that only the left-hand side of the camera should only see track. Yet a false edge was still detected as the last 10 pixels rapidly drop in value. This vignetting, alongside the narrow field of view of the camera contributed to the move to a dual camera setup.

Figure 25: Typical Camera Data.



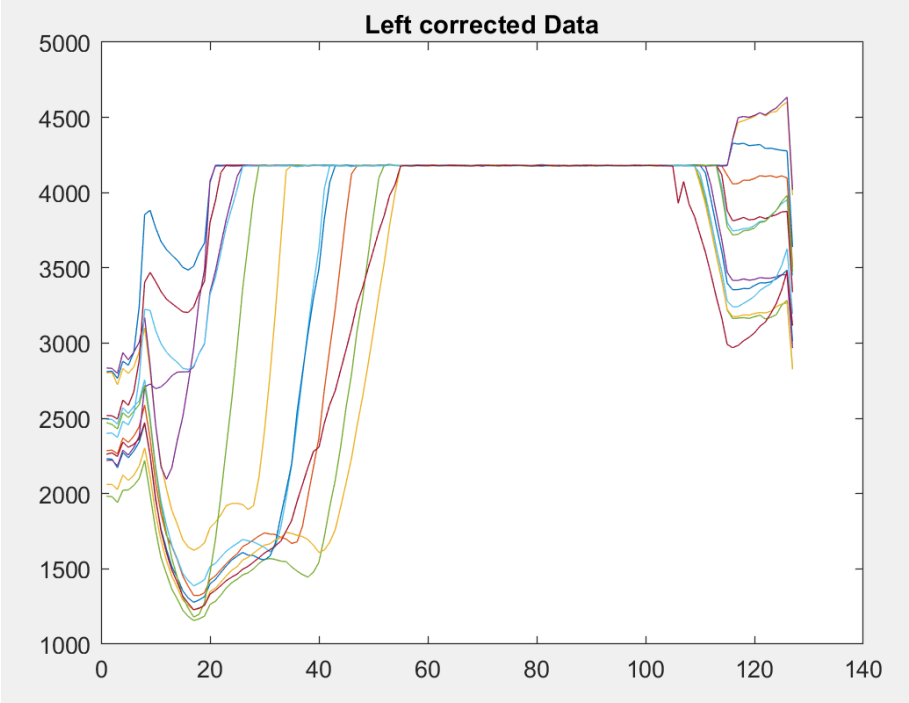
The vignetting effect rendered edge pixels on each side of the camera useless, affecting the first and last 20 pixels of the image. A vignetting filter was implemented to reduce the effect, despite this, 5 to 10 pixels of the camera image had to be discarded as they contained no useful data. This removal of the edge pixels reduced the field of view meaning there were times when the camera could not see any edges of the track.

10.3.1 Left to Right Scan

The first algorithm implemented decided on the location of the track edges by testing when the pixel value rose and fell below a threshold value. The linescan image was read from the leftmost pixel to the rightmost pixel. The first value that was above the threshold value was determined to be the left edge. Following this, the first value to fall back below the threshold was the right edge of the track.

Considering the simplicity of this algorithm, it was reasonably effective and was used to implement early bang-bang control which could complete laps of a simple track reliably. The main issue with this algorithm was that the algorithm often detected both edges on the far left of the track, particularly in areas with lightly coloured floors. This was compounded by the vignetting filter amplifying the edge values.

Figure 26: Effect of Vignetting.



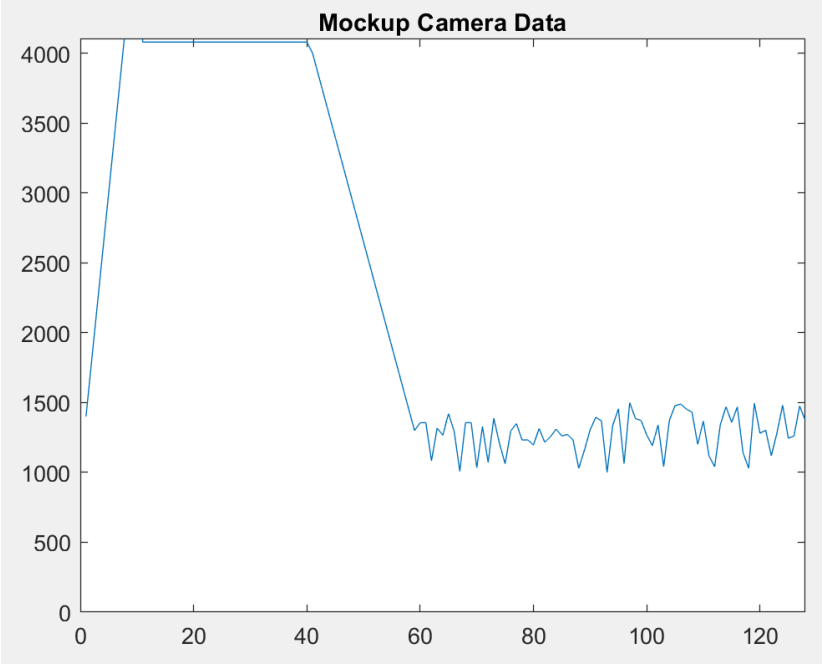
Shown above in Figure 26 is a selection of data where the car is positioned on the track to the left of the centre. With the threshold at 3500, the peaks around pixel 10, which are caused light patches on the floor next to the track on the edge may trigger an incorrect edge detection. The data on the right-hand side are the effects of the vignetting and the filters attempt to remove it.

10.3.2 Centre Outwards Scan

To prevent noise on the edge of the images triggering the threshold detection the algorithm moved from scanning left to right, to scanning from the centre of the image outwards. When the pixel value on either side dropped below the threshold value this was identified as the edge of the track.

This approach was successful when the centre of the camera image was on the track, a case which is true in most situations, however, when the centre of the camera view was the floor or a track edge then the centre pixel value would be below the threshold and the algorithm would detect this as both the left and right edge.

Figure 27: Mock Camera Data.



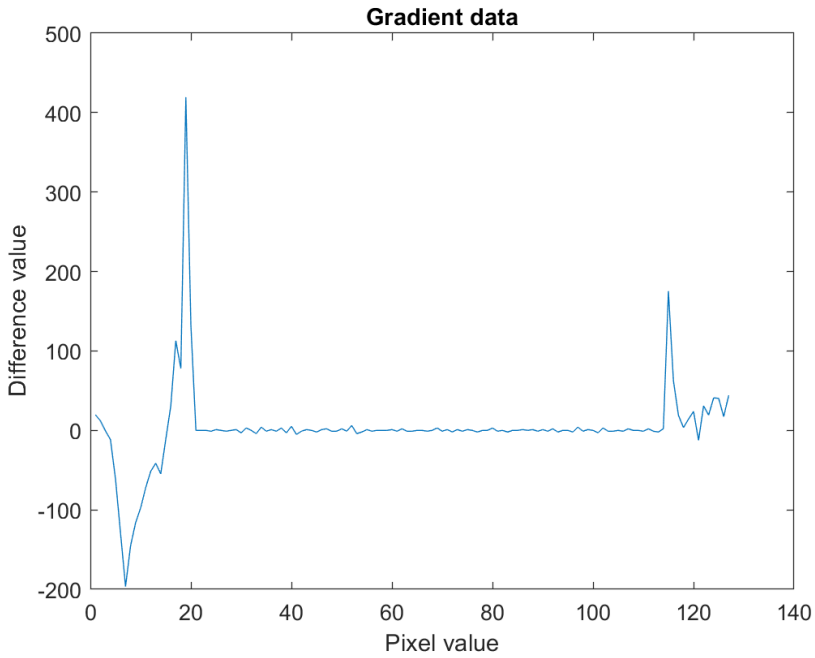
For example, in Figure 27 the algorithm would scan right from 64 and find the first pixel is below the threshold putting the right-hand edge at 64. Then the left scan would take place, once again beginning at 64, and detecting another edge. This sort of camera input occurs occasionally when the car is turning a corner, and when it did was guaranteed to cause a crash.

Despite this issue this camera algorithm worked for most of the time, however it eliminated any chances of the car recovering from an overshot corner, or a bad entry to the crossroads. This became the final algorithm used for the single camera approach.

10.3.3 Gradient Scan

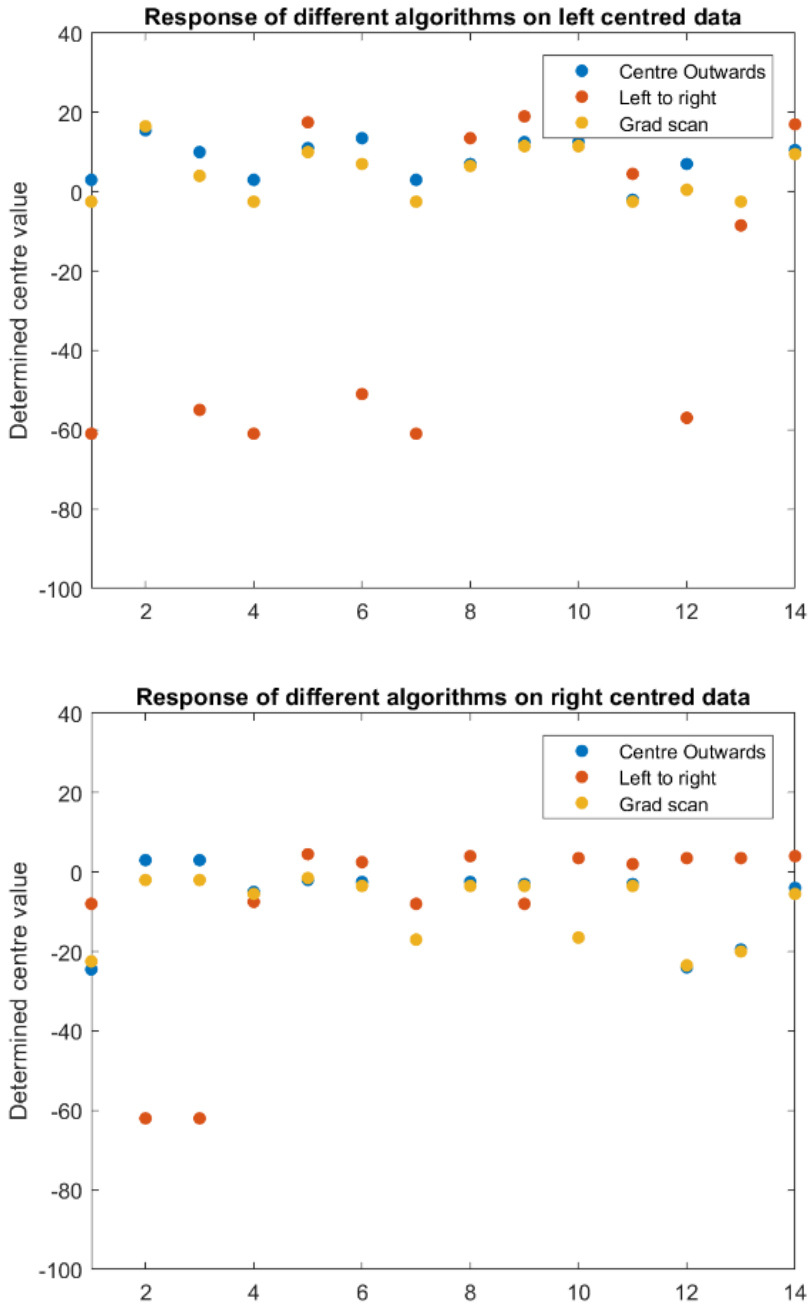
Instead of working off detecting the edges using a simple threshold another approach attempted was using the change in subsequent pixel values to detect the edges. In theory this should be successful as going from the black of the edges to the bright white of the track should be a far greater change in value than any changes occurring on the floor the track was laid on.

Figure 28: Typical Gradient Scan Data.



Shown above in Figure 28 is the gradient data, the large positive spikes denote the edges of the track. This represented a distinct improvement in accuracy compared to the previous methods, however its real-life performance was much the same, and still suffered from the same issues as in the last section, as well as also being affected by vignetting, but to a smaller extent.

Figure 29: Comparison of Centre Point Values from Each Algorithm. The upper graph shows data from the left side of the track, the lower graph shows data from the right side of the track.



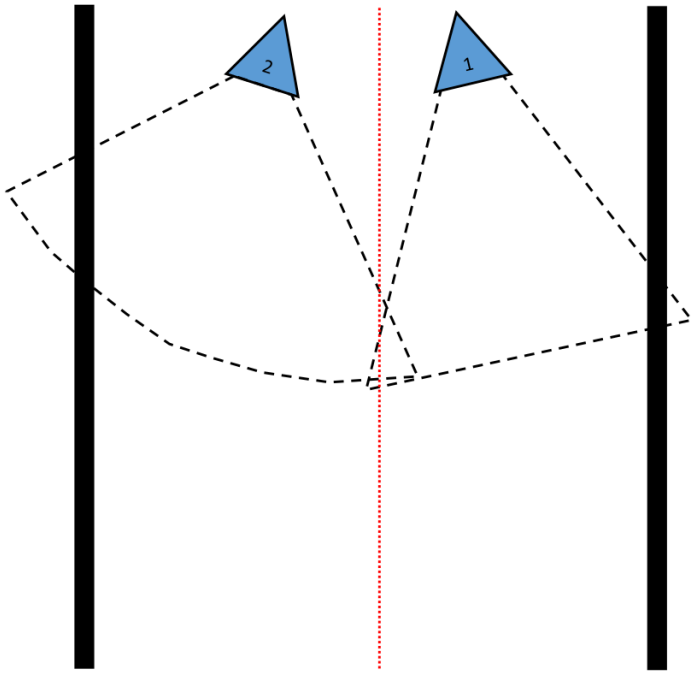
As both the threshold and gradient scan methods detected the centre of the track in roughly the same location the threshold method was used chosen for its quicker processing times. This meant the full control loop could run at a faster rate which meant even if a few centre position values were less correct, the overall reaction times of the car were reduced.

10.4 Dual Camera Algorithms

Having run out of ways to improve the previous camera in software, we began tests using two new cameras received from NXP.

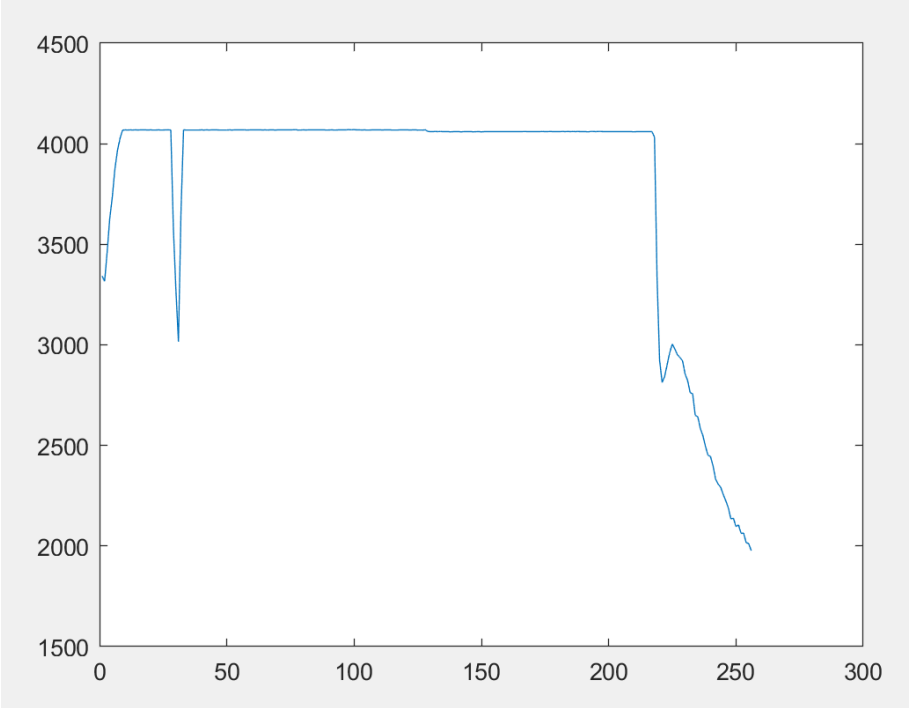
Testing of the new cameras showed they were both a distinct improvement over the older model, as they did not suffer from anywhere near as much vignetting, and one was equipped with a ‘fish-eye’ lens, giving a very wide field of view. However, this wide-angle camera didn’t quite ‘see’ in a straight line, rather, it saw an arc centred around the camera due to the sensor being slightly misaligned with the lens. The other camera had a narrower field of view, but it was far more linear. To make use of both these cameras, a dual camera mount was designed. The logic behind using both cameras was that we would end up with a higher resolution image, which could more easily cover the entire width of the track, whilst giving leeway to ignore the edges of each camera image which were vignetted or strangely positioned. To make this clearer, a diagram (Figure 30) is included below. The red line shows the calculated centre point, based on the views from each camera, the black dashed areas illustrate the approximate view each camera has.

Figure 30: Diagram illustrating the new setup. Camera 2 (left) is the fisheye one.



The disadvantage of using two cameras is the increase in processing time, as twice as much data must be physically read, and then run through the position finding code, however this did not cause a noticeable delay in the control loop. Secondly, the cameras are very sensitive to alignment issues, particularly if their viewpoints do not overlap, which can end up resulting in false edge detections. As camera mounts were quite stable, calibration was only required after serious collisions.

Figure 31: Typical Data from the new setup.



A typical view of the output of the dual cameras is shown above in Figure 31. The dip at ~40 on the left side of the image is a problem with the left camera and was accounted for in the camera algorithm. Also, it was noted that the new cameras gave the black track edges far higher values, often between 2000 & 3000 so for the following algorithms the threshold was set higher at around 3900.

10.4.1 Centre Outwards Threshold Scan

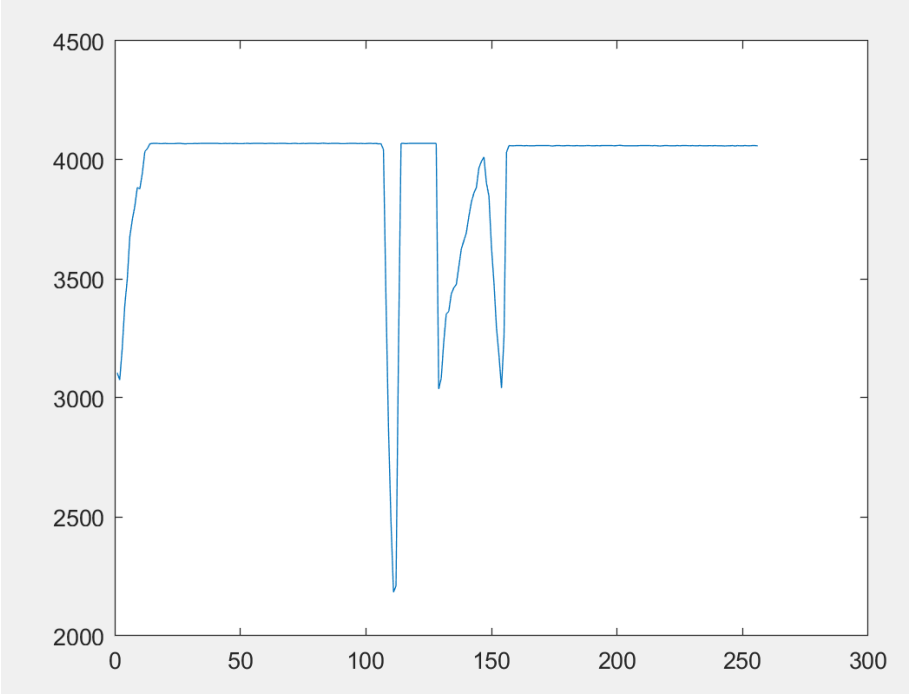
When the dual cameras were first mounted on the car the centre-outward scan was re-implemented, with corrections to account for the situation shown in Figure 31, where the track is entirely on the left or right-hand side of the image, meaning one side of the view contains no edges.

This was fixed by adding a flag in the algorithm that was set if the centre value was below the threshold value. If the flag was set, the algorithm would first scan left from centre to find the first value above the threshold, this was determined to be right edge. The next value to fall below the threshold was then assumed to be the left edge. If no edges were found the algorithm would scan right from centre and the first value above the threshold would set as left edge and then the next pixel with a value below the threshold was the marked as the right edge.

For the data shown in Figure 31, the algorithm would scan from the centre of the image, pixel value 128. First it would scan left and find the dip at a value of ~35. This would be recorded as the left edge. Then it would scan right from 128 and find the dip at ~210 and this would be recorded as the right. The centre value is then calculated to be $Centre = \frac{35+210}{2} = 122.5$. An offset of -128 is applied to make the values range from -128 to +128 instead of 0 to 256, giving a centre value of -5.5.

This method worked on most datasets, but it still failed in some places due to the problem with left most camera, or where the floor was particularly light.

Figure 32: Data causing a failure of the centre outwards scan.

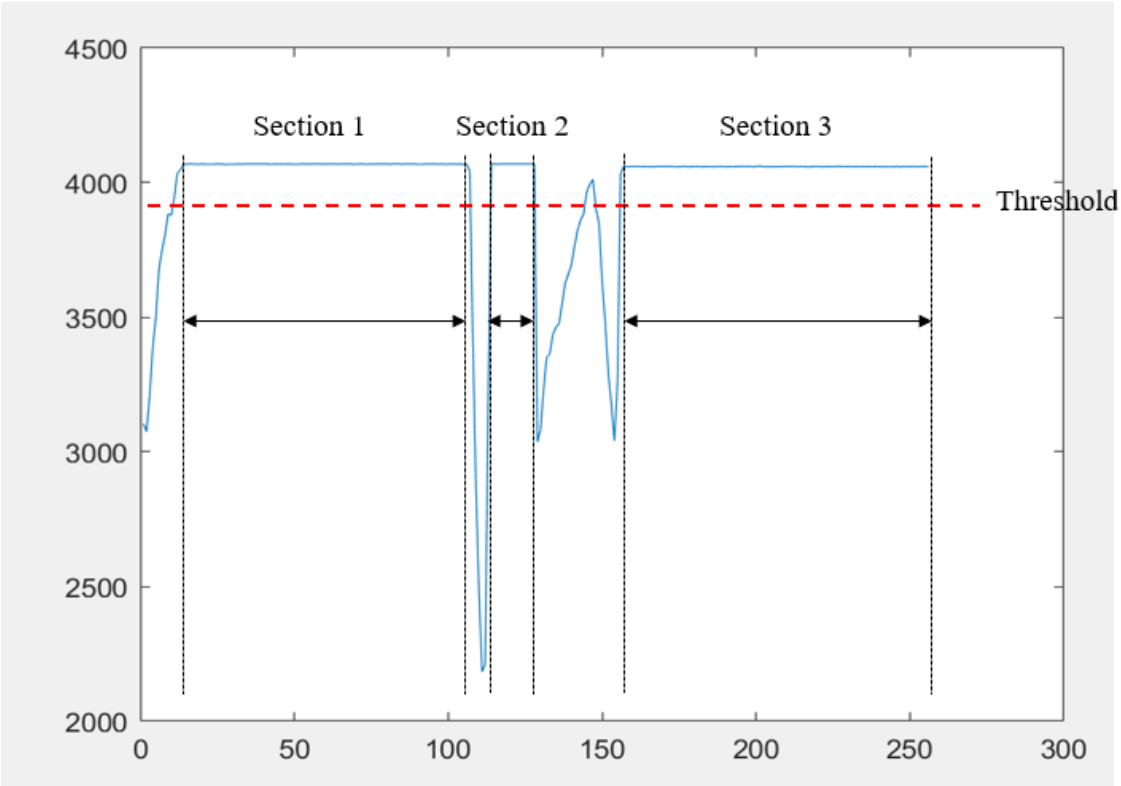


For example, if the centre algorithm was run on the data shown above in Figure 32, the algorithm would find the nearest section of track to the centre of the image, which is the high section between 110 and 130 and use the dips around there as the left and right edges. For this image the car was placed on the far left of the track and anything to the left of the dip at ~155 is noise from the floor/camera.

10.4.2 Track Width Method

To fix this problem, a different type of algorithm was developed, and is shown in Figure 8. This algorithm would scan left to right through the camera data. Once a value went above the threshold, its position was stored and then the width of the following segment of track was recorded until the pixel value then fell below the threshold, then the right edge was then recorded. This was repeated until the entire image had been analysed. The greatest width was assumed to be correct, and the corresponding left and right position values were used to calculate the position of the car.

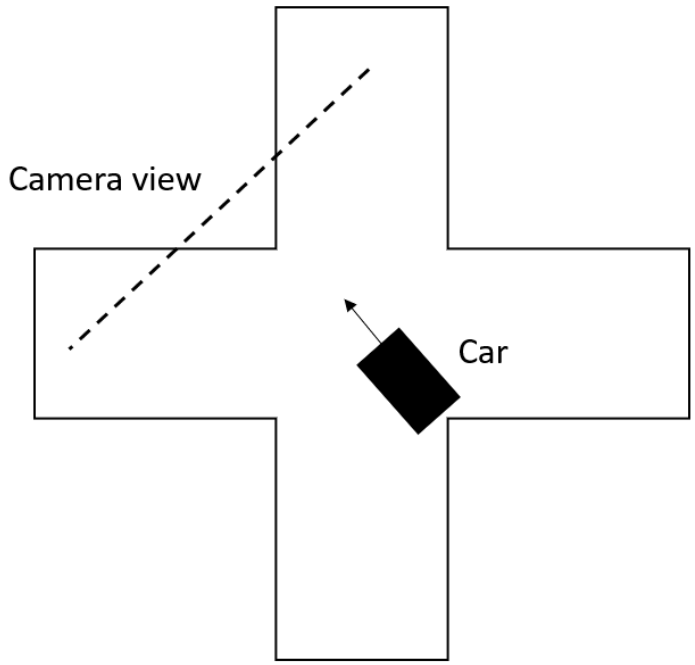
Figure 33: Pictorial Example of the Track Width Algorithm.

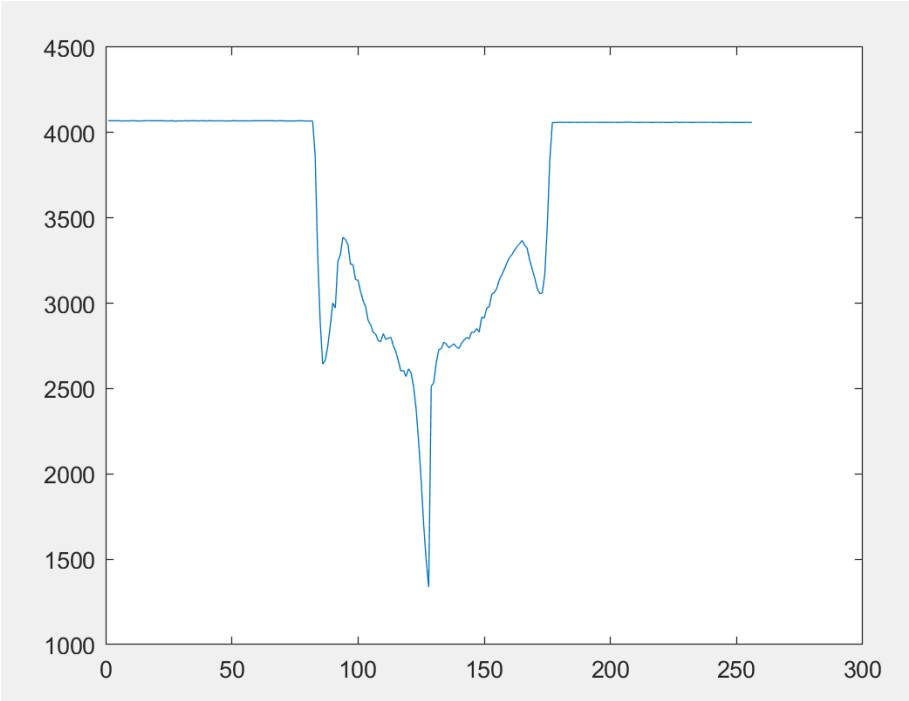


This solves the problem described previously. In this case, the new algorithm would find three track sections, the one on the left, the one on the right and the centre section. The largest of these three sections is the rightmost section. Therefore, the left edge is taken to be ~155, however, the right edge was not found. When an edge is not detected, it is assumed to be the edge-most value on the side where the edge was not detected. The performance of this algorithm is demonstrated in Figure 33 and gives a more accurate result than the previous algorithm.

Only two test cases caused this algorithm to fail. Firstly, if the camera cannot see the track at all due to the control responding slowly, or overshooting a corner dramatically, it sometimes detects the floor as track, or behaves erratically. The other is if the car comes into the cross road at a 45-degree angle. This produces the data shown below in Figure 34

Figure 34: Top: The situation that causes the algorithm to fail. Bottom: The data from this situation.



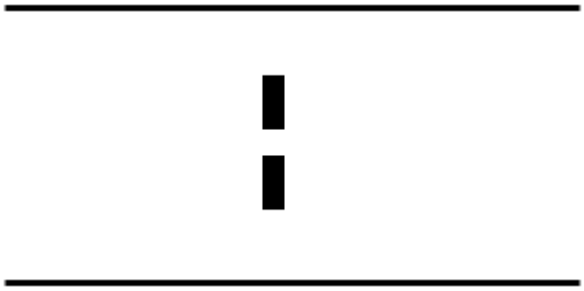


As shown above, the centre of the image is composed of the black edges of the crossroad, and noise from the floor. From this data the camera cannot determine whether the car should go down the left or right (upwards in the diagram) side of the crossroads. However, this scenario should only occur if the car is entering the crossroads at a very extreme angle. Therefore, with a reliable control system the car should never reach this state.

10.5 Cameras – Further Work

Given more time on the project there are two main things that could have been implemented. Firstly, detection of the finish marker is required for the competition. This is will be implemented before the actual NXP Cup Qualifiers, however the main goal for this project was to get the car running reliably round the track, and achieving good lap times, with finish line detection as a stretch goal, given the timeframes for the project. This would be implemented as a hard-coded solution since the finish marker will produce a unique camera image. The finish marker is shown in Figure 35.

Figure 35: Example of the finish marker.



For example, under the track width detection algorithm detailed above, an image containing a view of the finish marker should be split into a minimum of 5 sections. During testing of this algorithm, it has been established that this is a very rare occurrence in other situations, so a potential method would be to count the number of detected track segments. If there were 5 or above the edge detection code would call a stop function that would bring the car to a halt.

The second feature that could have been added is yet another camera, this time a look ahead camera with a view much further forward than the 2 mounted on the car. The idea of the look ahead camera is that it could calculate which section of the track was coming up. This information could be used to adjust the speed of the car, either allowing the car to ‘brake’ for corners, or even to calculate a proper racing line through them. This is possible, as each corner is standardised. The challenge in implementing this on the car would be finding a camera that could see far enough ahead, in enough detail, to be able to determine what type of track section was in front of the car. The current camera being used on the rover could only see noise if mounted at a shallow enough angle to see adequately far ahead. To implement this a different camera, which may not even be allowed for the competition, would need to be procured, and a detailed model of the car would need to be generated.

11 Control Theory

The control system utilises the output from the machine vision to direct the behaviour of the rover, shown in Figure 36. Using a control loop, it manages the speed of each of the back driven wheels plus the steering angle of the front wheels. The camera works as a feedback mechanism, providing the controller with a value for the position of the car on the track so that the error from the set point can be calculated. A perfect controller would always keep the car in the centre of the track with no oscillations.

Figure 36: Control Flow Diagram

There are multiple different control strategies which could be used to control the rover, including model-based control, proactive control, and reactive control. Although a model-based controller would provide an extremely quick and reliable response [30], it requires a mathematical model of the system. The rover would need to be modelled using complex data processing to derive an approximate system equation to fit [31]. Within the short time frame of this project, the rover needed to be built alongside the development of the controller. Therefore, after each revision of the rover design where the machine vision setup is changed or moved, the system would need to be re-modelled. The controller would need to be synthesized based on the characteristics of the new data, which would be very time consuming. Furthermore, problems could arise at the NXP Cup itself if any last-minute changes are made on the day.

A proactive controller which utilised pre-set movements could have been implemented. This is possible because even though the track layout is not known, as the course is made up of specific predefined sections. For example, all corners of the track are made up of the same 90 degree turns, so theoretically a routine could be implemented that takes the car through the perfect racing line every time. However, problems would be encountered when trying to recognise the difference between the types of obstacles, as the chicane track section could easily be mistaken for a corner. For each obstacle, unique routines would require. Additionally, the limitations of the line scan cameras available, coupled with the speed of the rover, means that the obstacles may not be sensed early enough to identify them, and then and perform the routines.

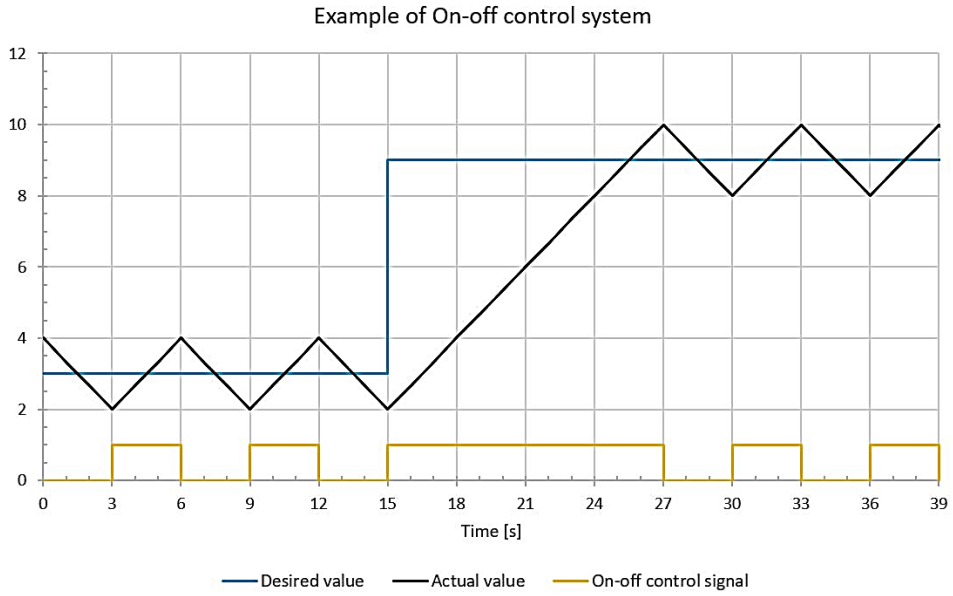
As the system is relatively simple, the rover can be kept in the centre of the track using a reactive controller utilising the feedback from the camera. Various algorithms could be used, each with variables which can be tuned to try and find an optimal control algorithm. The goal is for the output to be stable and achieve smooth, reliable, and repeatable behaviour, where a balance needs to be found between disturbance rejection and reference tracking. Disturbance rejection is the ability of the system to maintain the output at a given set point whenever a disturbance causes a deviation from it. Whereas reference tracking is the response of the system to implementing changes in the desired set point [32]. When tuning the variables, the stability of the system, meaning there are no unbounded oscillations, is essential. Furthermore, for the rover, reference tracking is much more important than disturbance regulation.

11.1 Control Algorithms

11.1.1 On-off Control

Also known as bang-bang control, this method produces a fast response and easy to implement control methods [33]. The algorithm can eliminate large errors, is simple and effective, but due to the features of on-off control it easily causes system instability and can be inadequate in ensuring system accuracy [34]. On-off control works by setting the output to one of two values depending on the input. Figure 37 shows the typical output of the control algorithm where the effects of the abrupt switching between the two states can be seen. This means the controller is most effective in cases where the output is binary, or very discrete. Brisk transitions between continuous states tend to oscillate more, which may cause wear on components. Another complication which can occur depending on the inertia of the process being controlled, an oscillating error signal may arise about the desired point in a saw tooth fashion. Alternately, if controller has a narrow hysteresis between its switching values this may induce rapid on-off switching.

Figure 37: Example response of on-off control with an update time of 3 seconds. It shows the oscillations about the set point, and linear response to the change.



In the case of the rover, the camera algorithms determine the distance (error value) of the car from the centre of the track. This is then used to determine if the car is to the left or right of the centre. Based upon this, the servo position and hence the steering angle is set to the maximum left or right value to reduce the error and re-centre the car. The speed is kept constant and not changed with this controller. The code is provided in the design archive. As this controller is quick and simple to produce, it allowed for the testing of initial camera and car designs at the start of the project.

11.1.2 Discretised Control

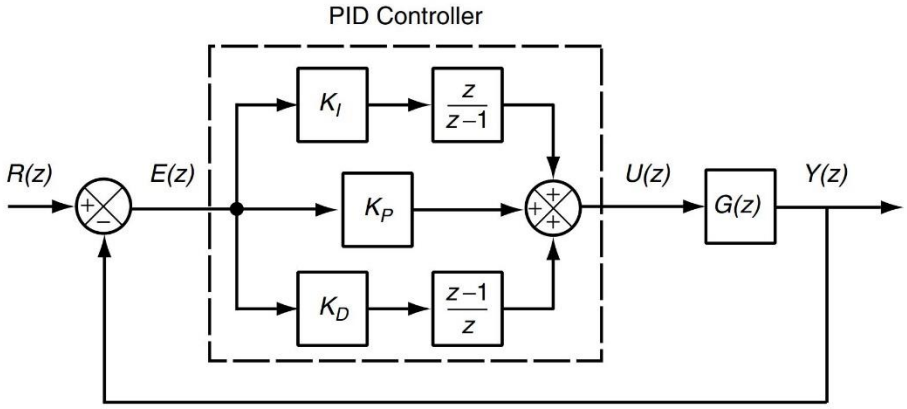
This algorithm builds on the on-off control by providing a few more output states by converting the system from on-off to discrete. This control algorithm does this by setting certain values for the discrete error from the track centre for ranges of the normalised error. The granulated value is the final error value that is used to set the motor speed and steering angle. The range in which the continuous data can be discretised can be tuned to

provide better results. This controller is very simple to produce but provides a smoother effect than on-off control. The code is provided in the design archive.

11.1.3 Proportional – Integral – Derivative (PID) Control

A PID controller uses the three terms to affect the output to apply optimal control. From the error, the controller calculates the correction depending on the proportional (P), differential (D) and integral (I) terms. The general system of the controller is shown Figure 38 where in the case of the rover is the plant 'G(z)' and the correction variable 'U(z)' is the values passed to the servo and motors.

Figure 38: Feedback loop with PID controller [35].



The 'P' term provides a correction proportional to the error, such that a larger error produces a larger correction. As an error is required to produce any response from this term, there may be an error between the reference 'R(z)' and the actual 'Y(z)' values which is the residual error. The value of the proportional gain constant 'K_P' can be tuned for each case there this control algorithm is used. The greater the gain the greater the response to a change in the output but if it too high the system can become unstable. In the control algorithm the P term usually contributes the largest part of the correction value [36].

The 'I' term uses both the magnitude of the error and the duration as it is the sum of the error over time which is an accumulated error. This removes any residual error which may have been caused by the 'P' term by adding a control effect. The gain constant 'K_I' is another variable which needs to be tunes and will case instability if set too high. Another problem with the 'I' term is integral windup [35] which occurs when a large change in the error occurs thus the 'I' term accumulates a large error. As the output of physical systems are limited and can saturate the windup which will lead to overshooting the set point. The overshoot will persist until the accumulated value is offset by errors in the opposite direction. The problem can be addressed multiple ways, such as only enabling the integral function below certain error values or initialising the integral to the desired value.

The 'D' term is the anticipatory control using the current rate of change to predict the future trend of the error [37]. This mean the term produces a larger correction value when the rate of change of the error is large. The derivative gain 'K_D' control the magnitude of the control produced and can be tuned accordingly. This term improves the settling time of the system but if the gain is set too high it can produce an excessive response and overshoot.

A selective use of the PID terms can be used to control a system with varying degrees of effectiveness. Firstly, for the rover only proportional control algorithm was used, then a PD controller and finally full PID was implemented. This allowed to test each part of the controller with the rover. The code for each of the algorithms is provided in the design archive.

11.2 Establishing Values and Results

11.2.1 On-off Control

This controller simply tried to set reach the set point by setting the servos to either fully left or fully right. This controller was good for testing initial data values produced from camera system and allowed us to test if the rover could follow a simple track. The algorithm produced a lot of oscillations at all points of the track as the car was almost always steering. This sometimes caused the rover to drive very close to the edges of the track whilst winding its way along straights, which meant it would meet corners at random angles, some of which were so extreme that the rover could not turn fast enough to make it round reliably.

11.2.2 Discretised Control

The conversion shown in Table 3 shows how the rover uses a discrete range of steering angles based on the error values set from the vision system. With this controller the oscillations, particularly on the straights were greatly reduced, which meant the rover could drive at faster speeds to further test the camera. Plus, the smoother controller removed the issue of meeting corners at sharp angles.

Table 3: Discrete Control

Continuous	< -0.75	-0.75 to -0.15	-0.15 to 0.15	0.15 to 0.75	> 0.75
Discrete	-1	-0.3	0	0.3	1

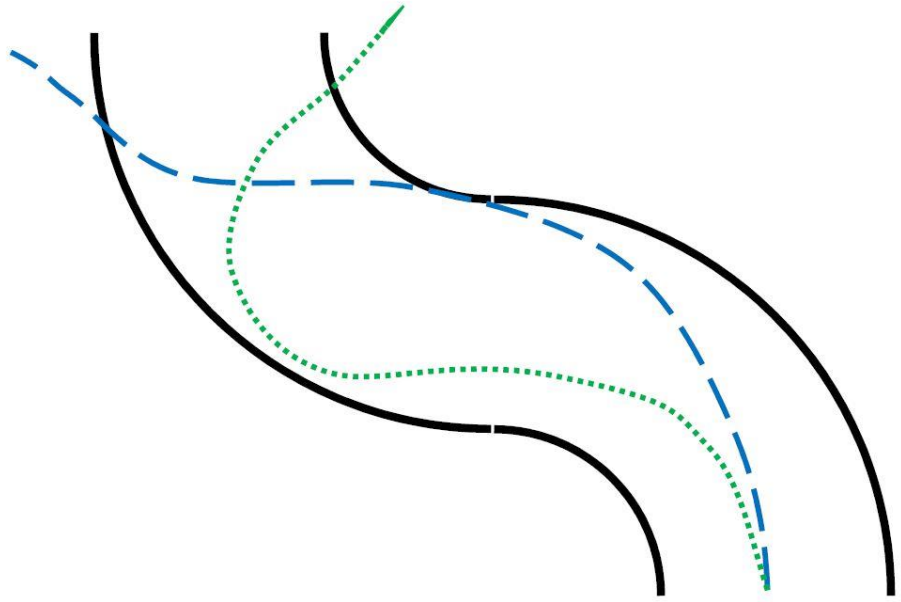
11.2.3 Proportional Control

The controller used a normalised position error from the centre of the track multiplied by the gain value to generate the value used for the steering angle. Through some testing it was found that one way to increase the speed at which the car completed the track is the proportional control was only used to control the servo angle. The speed at which the motors ran was able to be set to a constant unless the error from the machine vision reached a threshold value. Once this occurred the speed would immediately drop and then ramp back up. This meant if the value was set correctly, the rover would maintain a high speed throughout most of the course and slow down for corners, accelerating out of them.

The value of the gain was set by trial and error to a value which allowed the rover to complete a simple square track. The algorithm provided an improvement in lap times compared to the previous controllers and now the car could comfortably negotiate the chicane track piece. However, with proportional controller it was unable to complete complex sets of corners, such as a left turn followed immediately by a right turn. The paths of the rover are shown in the Figure 39, trying the complete this track section, the rover was not able to turn fast enough to the first corner (blue line) when the gain value was set to a low number. When the gain value is increased it reacts too sharply and

overshoots the turn (green line). With the additional elements of the PID controller could be used to remove the overshoot and keep the car in the centre of the track.

Figure 39: Drawing showing the different paths the rover takes when using proportional control at different gain values



11.2.4 PD Control

The addition of the 'D' control element increased the time it took to tune the gain values of the rover but greatly improved the performance removing a lot of the oscillatory nature which existed before. This allowed for the car to remain on the track when the speed was further increased and enabled it to negotiate multiple complex turns. The 'D' term was generated by assuming each loop of the control loop takes the same amount of time, so the value is equal to the difference between the current error and the previous error divided by a constant time period. The 'P' and 'D' gains needed to be tuned again after each change to the car and when the machine vision was updated through the project.

11.2.5 PID Control

A problem arose where the servo steering was not always centred when the rover microcontroller thought it was. This caused a residual error where the car would not stay perfectly in the centre of the track, with the addition of the 'I' term this could be compensated for. The controller generates the 'I' term by adding the errors in each control loop if the value is within a certain bound, to prevent integral windup. The tuning was all undertaken manually using trial and error methods. First 'K_P' is determined by keeping the other gains set to 0. Secondly, 'K_I' is increased until residual errors are removed in a sufficient time frame and finally 'K_D' is increased until the response time is as fast as possible, without becoming too 'twitchy'. The final rover controller is tuned so that it slightly overshoots initially, which helps it to negotiate the turns faster and reducing the effects of understeer.

11.3 Unique Track Feature Detection

11.3.1 Hill Detection

One of the more unusual track sections that could be faced in the NXP cup is a hill. From the study of competitions from previous years, many cars have struggled against this obstacle. The issue is twofold as the rover may require extra power to be sent to the motors to make it up the hill and maintain speed, but needs to brake on its way down, particularly if there is a corner soon after the hill. Furthermore, when driving over the hill all four wheels need to be on the track otherwise the rover will fall off.

Testing using our PID control and speed settings showed the rover could easily make it over the hill, however it would pick up a large amount of speed, and without wheel speed sensors this had to be dealt with in another way. To combat this issue, we utilised the inbuilt three axis accelerometer. The three values vary from -2 to 2. By driving over the hill multiple times and outputting the data via a serial link, it was found that the value of the X-axis accelerometer reached -1.5 every time. Testing of other obstacles very rarely caused this value to be reached. Therefore, a function was created to add a braking effect to the car by running the motors in reverse for a short period of time. To add to the reliability, to reduce the likelihood of the car braking at the wrong time, once the -1.5 threshold is reached, a moving average is taken to ensure the car is actually on a hill. Only once this is confirmed will the car brake.

12 Telemetry

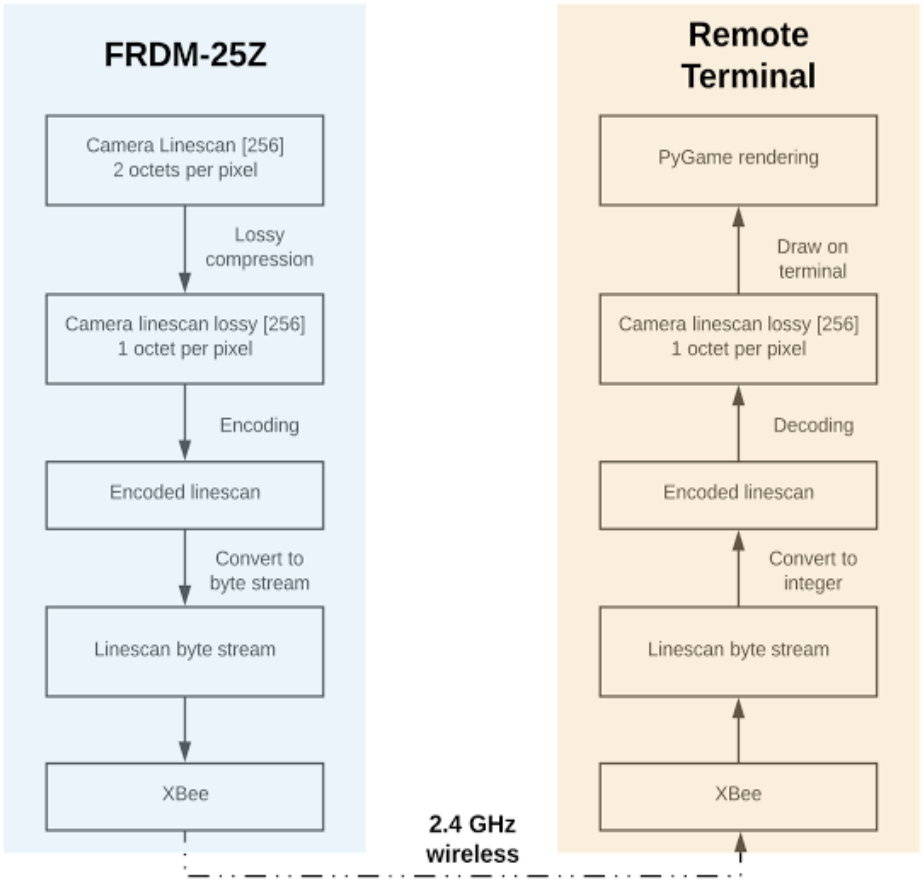
Autonomous vehicles are designed to be independent, free from human interaction and wireless controllers. Radio communication is therefore not allowed during the competition. However, a set of radio modules are able to be used on the vehicle primarily for debugging purposes, to aid in understanding the real-world environment that the car faces throughout its development. Ultimately, we would like to monitor all metrics from every parameter and sensor on the car from a remote terminal. The goal is to receive a live feedback as the vehicle drives itself around the track, in order to verify its decision-making process and collect data if the system misbehaves.

The alternatives to a wireless communication system is either connecting the car to a portable terminal through a USB cable, or by installing an LCD screen which display the status of the car. A USB cable however has a finite length and as the vehicle travels along the track, the person monitoring with a portable terminal will have to travel at roughly the same pace as the vehicle. This solution would create a substantial distraction to the person monitoring the system as it requires a high level of multi-tasking, never mind the potential health and safety issues. One solution would be use a much longer cable, such that it can reach all points on the track, but for the same reason the cable itself would get in the way of the car as it would eventually get tangled. At some point, the USB connectors would also suffer damage due to a crash, or simply overuse. Similarly, the LCD screen also causes a lot of trouble for the person monitoring the system. Given the limited amount of space on the KL25Z board and car itself, the LCD screen would be either be extremely small, or potentially unfeasible to install. This means reading the on-screen data would be next to impossible, and someone would still need to follow the car around to see the screen the entire time.

12.1 Overview

The development of our telemetry system involves three main components - Hardware, Communications protocol, and software. Hardware provides the underlying ability to communicate between the vehicle and our remote terminal. The protocol to deliver messages correctly from one end to another. Finally, the software visualises the information from the vehicle such that we can understand the content easily.

Figure 40: Overview of the workflow from the camera to terminal rendering



12.2 Understanding Domain

12.2.1 Data Rate

As far as our telemetry system is concerned, the data rate is the largest issue to overcome. Each of the on-board line-scan cameras generate a 128x1 pixel image at 30 fps (frames per second), with a bit depth of 16 (2 octets) inherited from MBed [21]. The means our dual-camera system produces 122 KB (122,880 bytes) per second of raw data. This becomes an immediate concern to us, as a number of radio modules have a similar maximum data throughput as our camera read-in rate.

$$Datarate = 128 \times 30 \times 16 \times 2$$

On top of the camera data which is fed multiple times per second, other on-board parameters like PID coefficients would also need to be transmitted every few seconds.

12.2.2 Memory

Since the FRDM-KL25Z only has 128KB of flash memory [21], space for a message queue is extremely limited. During normal operation, the flash memory can hold less than 1 second worth of encoded messages. Therefore, it demands a radio module which can consume and deliver messages in real-time. Messages that don't get delivered in time will most likely to be dumped and the data will be forever lost. The same would apply if there is a sudden loss of connection.

12.3 Hardware

The choice of hardware is limited by our domain which immediately restricts our range of selection. As well as the domains mentioned in the previous section, it has to be low cost due to our limited budget and it has to be low power which is limited by the power output from the board. Most importantly it was to be compatible with our KL25Z, such that it has the relevant IO to communicate with the board.

12.3.1 nRF24L01

The nRF24L01 [38] is a radio from a very popular range of proprietary transceivers by Nordic Radio. They include many features such as low power mode, acknowledgement, and channel management. It has a default non-ACK data rate of 900 kbps but configurable to a maximum of 2000 kbps. Although there are limited resources available in the industry to indicate that they're compatible, a handful of online resources were available, hinting that it may be possible. Attracted by its transmission rate, we first experimented with an nRF24L01.

The nRF24L01 provides a SPI connection in which it can theoretically communicate to our FRDM-KL25Z board. On the other end, an identical nRF24L01 is connect to a **USB to NRF24L01 Module CH340T SPI** [39], in which it is connected to a remote terminal. The remote terminal is running a python library **pynrf24** [40] in order to talk to the **CH340T SPI**.

However due the clash in wiring nRF24L01 SPI to FRDM-KL25Z and the TCF shield, as well as some irritating compatibility issues with CH340T, we as a team decided to move on to other simpler radio modules.

12.3.2 XBee S2

The XBee is an equally popular transceiver with features like low power and small footprints [41]. They are commonly used in mesh type sensor networks. However, compared to the nRF24L01, it has a significantly lower data rate - 240kbps max. The primary reason for choosing the XBee S2 is that it is proven to be compatible with our FRDM-KL25Z board and provides great reliability. Documentations for configuring XBee in different scenarios are easily accessible as well [42][43][44].

12.4 Data Compression

12.4.1 Bit Depth Compression

MBed stores line-scan camera data as an unsigned short integer type, which has a size of 16 bits (2 octets) ranging from 0-65535, (Appendix E). As the pixel becomes brighter, the camera would output a higher value and vice-versa. However, after a closer look at the camera data, it was noticed the sensor could saturate, and would rarely output a value of over 5000. This was confirmed by shining a torch directly towards the camera. This meant that the values ranging between 5000 and 65535 were wasted. Therefore, we could boldly cut-off any value higher than a certain threshold, which was set at 6000. With the new upper threshold, we can comfortably cover the entire range of value with 13 bits instead of 16, reducing the required data rate by 20% [45][46] Note, although this methodology might look similar to bit packing [47], when data is compressed to a lower bit representation, we do not unpacked data back to a higher bit representation at end point.

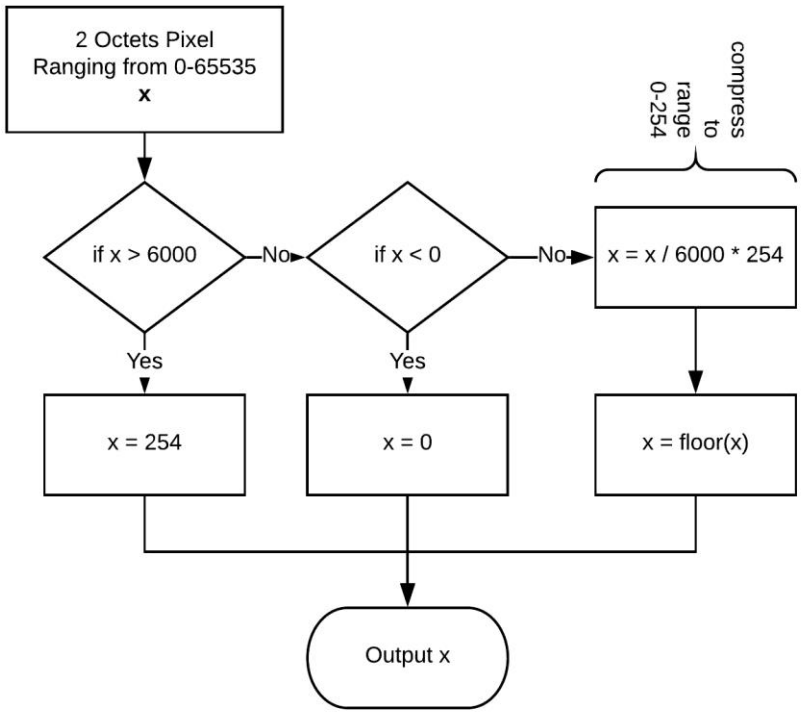
12.4.2 Lossy Compression

Under closer inspection, we discovered that the camera produces a pretty good exposure with wide ranging white and black values, meaning that white is truly represented with a large number (> 4000) and black with a low value (<470). This allows lossy compression to be implemented, with minimal information loss [48]. In fact, if I pack the awkward 13 bits representation into a nice octet (8 bits), my information remains largely intact.

Therefore, we have successfully compressed a 16-bit signal down to an 8-bit signal, a reduction of 50%. Data values now span from 0 to 254 with 255 as a reserved value for special signal which we will mention under the chapter 'Protocol'.

The workflow for the data compression is shown overleaf in Figure 41.

Figure 41: Bit depth compression workflow



12.4.3 Performance Vs Compression

Based on the knowledge from lossy compression, we understand that the light and dark values are distinct enough to allow more aggressive data compression. One example would be delta coding where instead of storing the exact value itself, difference between integers are stored. In principle this is totally a valid decision. However, the XBee transmits 8-bit values, meaning that however you wish to compress your data, you have you submit a list of 8 bits representation at the end of the day. This means further compression from 8-bit values can be done, but at the expense of CPU power and time [49][50]. In other words, any compression performed from this point would require repacking of bits before the data could be send. With CPU resources at high demand, further compression from 8-bit representation is strongly discouraged. The problem intensifies if the CPU struggles to compress and pack data in real time, information may need to be stored and risk memory overflow.

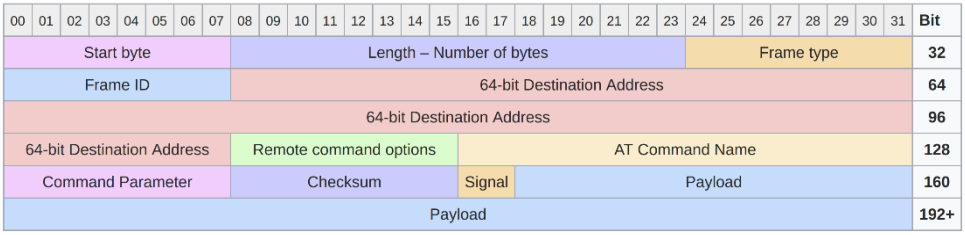
12.4.4 Frame Rate Reduction

Frame rate reduction is the least preferred methodology of all but is very effective. The telemetry system is designed to be independent of time, in other words messages can arrive at any moment in time. When the bandwidth reaches its maximum capacity, line-scan data which is stuck at the bottleneck will be dumped and lost. This encourages the radio to work at its maximum capacity under pressure but allows the frame rate to drop when necessary.

12.5 Protocol

Our telemetry data protocol is built on top of the XBee AT protocol [51]. Responsibility for establishing a connection as well as addressing and ordering is done natively by the XBee S2. Its protocol guarantees the order of packet arrival and provides a basic checksum for data integrity; however, its lack of acknowledgement means that the arrival of packets cannot be verified.

Figure 42: A Typical xBee Packet



The payload is designed to carry 3 types of data - line-scan camera data, PID coefficients & system status. The data is initiated and finished by a set of special characters. This helps the receiver to identify the nature of the incoming data, and process accordingly.

12.5.1 Special Signal

Special Signal is formed by a special signal prefix followed by special signal content. From the previous chapter ‘Data Compression’, we have mentioned that value 255 is reserved for a special signal. In fact, value 255 is a special signal prefix; when the receiver listener program detects value 255, it knows that a special signal is to come indicating that the system will change state. Therefore, a valid special signal begins with a 0xFF and is followed by the special signal content (E.g. 0x00 for line-scan data). This signal will repeat at the end of the message to indicate the change of state.

Table 4: Special Signals

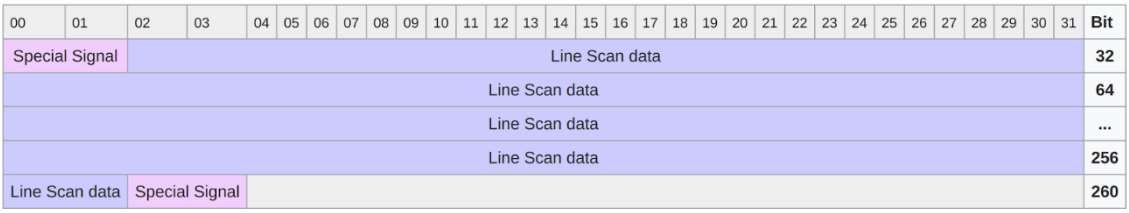
	Meaning	**Decimal**	**Hexadecimal**	**Bit**
Prefix	Special signal prefix	255	0xFF	1111 1111
Content	Line Scan	000	0x00	0000 0000
Content	PID	102	0x44	0110 0110
Content	System state	153	0x99	1001 1001

For example, a line-scan message will look like this:

0xFF 0x00 [Line scan data] 0xFF 0xFF

At this point, the terminal would understand that the line-scan is fully delivered and it's waiting for the next line-scan, PID coefficient or a system state message.

Figure 43: Example of a line-scan payload.



12.5.2 Design of Special Signal

Special signal accounts for less than 1% of the total data transmitted, but their appearance play an important role to ensure the messages are delivered correctly. Although these signals are formed by regular 8-bit message, their binary values are carefully chosen such that a corrupted special message can easily be identified and not be misinterpreted. Any faulty special messages will be discarded, and the system will continue to wait for a correct signal.

Gray code inspires the formation of these special signals, it inherits the property where they maximized their hamming distance from each other [52][53]. Therefore, it is able to detect up to 3 noisy bits and ultimately safeguard the integrity of the special signal.

12.6 Requests

When talking from the remote terminal to the car, the protocol is much simpler. It begins with a signal header, then followed by the payload if applicable.

Figure 44: Table of request identifiers

Signal	**Signal header**	**Payload (Octet)**
Update PID	0x41	6
Update bearing	0x42	2
Power off	0x5A	-

12.7 Error Handling

Our protocol was designed to be fast and lightweight, with minimal memory usage in mind. For that reason, acknowledgement (or negative-acknowledgement) is purposefully left unimplemented. This meant that the sender would not have to make a copy of recent messages in case a message was lost. Important information is instead protected by forward error correction.

12.7.1 Nature of Common Errors

1. Most error exist in the form of data loss. Each xBee has a given amount time between their last communication and when they sleep. Once they enter the sleep mode, a new message is required to wake the XBee up. During this period of time, the XBee will not be able to listen to any messages. Data that are sent between the first packet and the time when XBee formally wakes up will be forever lost.
2. Data corruption is almost non-existent, after logging 5 sets of one-minute transmission, no data corruption was detected. No further experimentation was taken as we are not concerned about the potential of minimal noise in the data.

12.7.2 Frequency of Errors

The frequency of data loss depends on the rate of system idleness. If the communication has been idle for a period of time, we can safely expect the first 300ms from the first message will not be received.

12.7.3 Forward Error Correction

Forward error correction protects special signals. Since the data corruption rate is so minimal, special signals are protected by repetition codes.

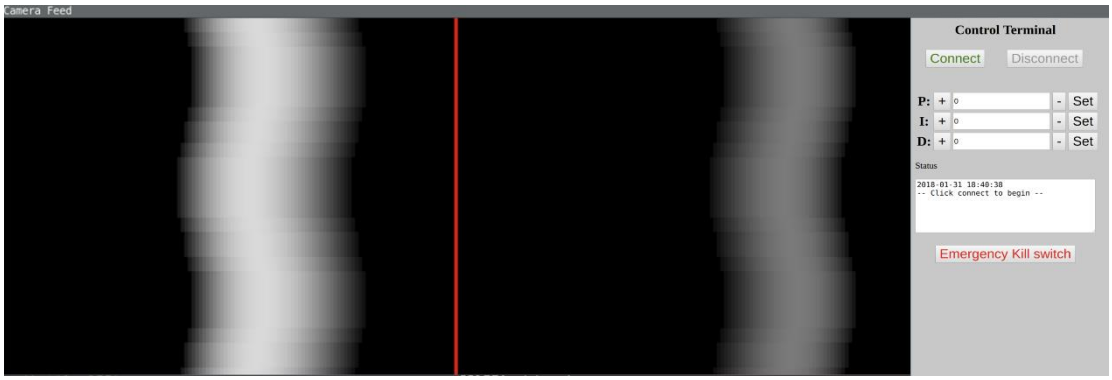
12.7.3.1 Error-Correction-Bit to Data Ratio:

Assuming camera feed is running at 30 fps and PID coefficient is sent one per second:

$$(4 + 4) \,/\, (2048 + 6) \;=\; 0.3\,\%$$

12.8 Software

Figure 45: GUI Software on Remote Terminal

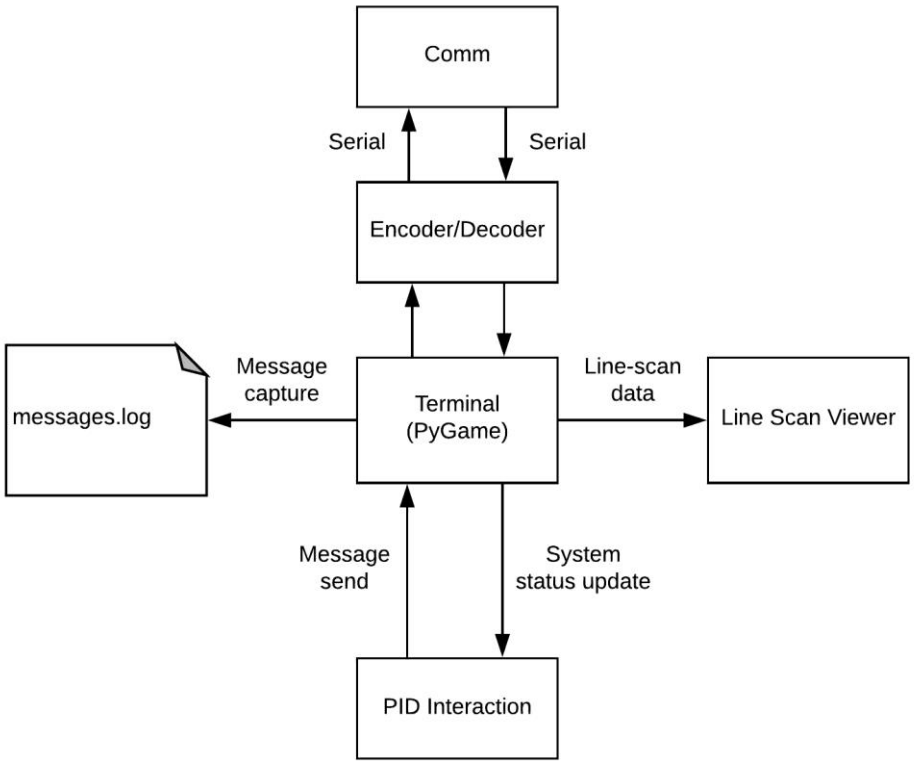


Remote terminal software is responsible for displaying the camera sensor data as well as the PID coefficients. The software is developed in Python due to its cross-platform capability as opposed to C. This allow us to be terminal-hardware independent in case if we decide to run the terminal software on other computers. During the first development phase, the camera screen draw was done using python.graphics. However, its redrawing capability on the same canvas was very limited and does not offer quick enough frame rates. This meant most of the received data would need to be stored in a queue or risk being dumped as the queue reaches its maximum size. For the same reason, the software became unable to display live data as the runtime increased.

In the second version of this software development, PyGame was used instead of python.graphics and the results were immediately noticeable [54]. As well as the increase in performance, PyGame was able to reduce the load in a single processor as it utilizes the graphics card available on the remote terminal (tested on a Dell XPS 15 9550) [55].

Each pixel-value is visualized by a monochromatic pixel representing the pixel's brightness. As the line-scan is printed across the screen, track boundaries should light-up as bright white and other area should be patches of black. A screenshot is shown in Figure 45.

Figure 46: Class structure of terminal software



12.8.1 Features

12.8.1.1 Line Scan Stacking

Each line scan received is stored in a history array, this enables us to view the past 100 lines worth of line-scans. As the software receives a line-scan provided by the receiver, the scan is added to the top of the line-scan stack, followed by the past 99 line-scans. This is especially useful as the vehicle changes its bearing. As the vehicle corrects itself whilst moving around the track, the track boundaries would shift towards either side of the line-scan. This stack of line-scans would provide a good indication if the car is staying well within the track's boundary.

12.8.1.2 Lost-Packet Safe

In an ideal world, each line-scan would provide exactly 256 pixels, allowing the data to be drawn from one side to another with no gaps. However as this is not always the case, the software will have to adapt to loss of packets. As it receives a line-scan, it zero-pads the array of length 256. Any 'gaps' within this array would be set to 0 and would be drawn as a dark patch. Assuming that the lost rate is low, the noise should not affect the visualization of the overall line-scan stack.

12.8.1.3 GUI for PID Interaction

Graphical user interface is created with user-friendly buttons for adjusting PID coefficient while the vehicle is on the track. This aid our team to perform fine-turning during our development and provide assistance when the vehicle is adapting to a new environment.

12.8.1.4 Logging

On top of storing the past 100 line-scans in the history array, all messages received from the receiver is saved in a log file on the remote terminal for further debugging purposes. These generated logs are large in size and it is advised to be cleared from the remote terminal regularly.

12.8.2 Testing

The following equations are used to simulate the camera feed of a moving vehicle as it travels along the track. It generates two white regions on either side of the line-scan data simulating the edge of the track. These two regions will shift according to a randomly scaled sin wave to simulate the changing of bearing as the vehicle travels along the track.

This equation generates the brightness of a certain pixel according to their position along the x-axis and a constant value - shift.

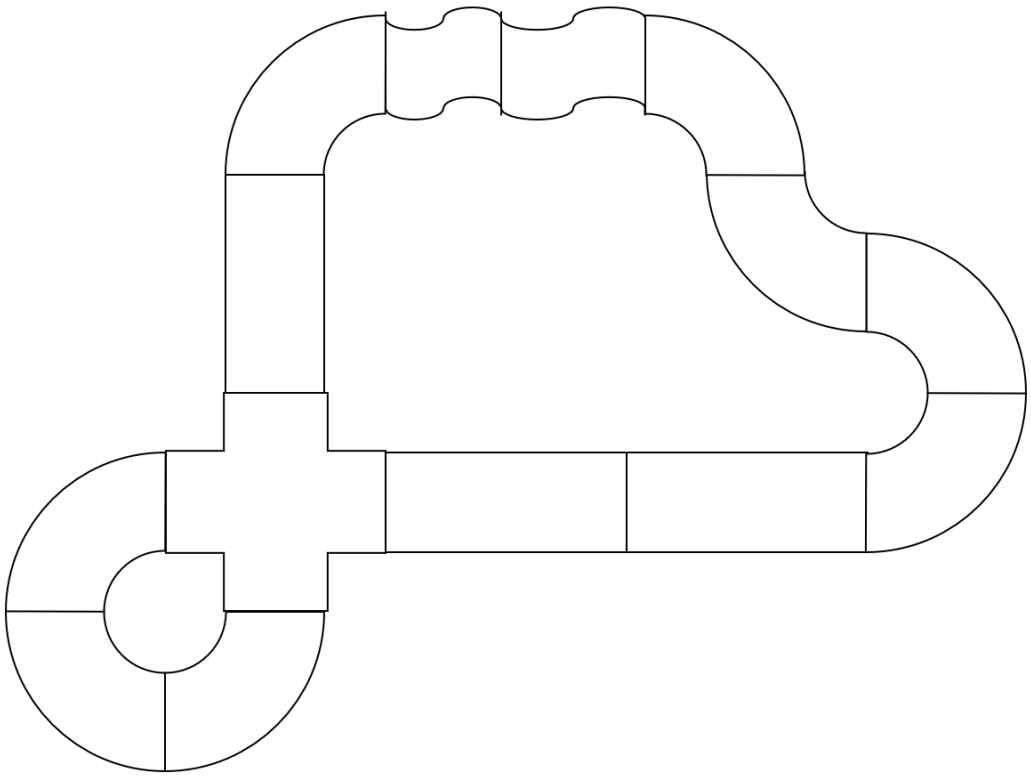
$$brightness(x, shift) = -\frac{(x - shift - 50) \cdot (x - shift - 100) \cdot (x - shift - 200) \cdot (x - shift - 238)}{60000}$$

Where:

$$shift = int(3 \cdot sin(\frac{y}{50}) \cdot Random(0, 2))$$

13 System Testing

Figure 47: The track layout used for the final system test.



The track shown above was chosen as our testing track as it is the most difficult track we could construct using the pieces available.

The 270° bend into the crossroads is challenging because the car may come out of the turn into the crossroads at an angle. This creates a risk that it will take a wrong turn at the crossroad. Understeer would cause the car to end up travelling around the track the wrong way, whilst oversteer could cause it to end up driving around the corner indefinitely.

The hairpin can cause issues if the car oversteers as the cameras may lose sight of whole track if it does not remain centred on the track through the hairpin.

The long straight is included to test the top speed of the car and ensure it doesn't end up going into corners too quickly, whilst the chicanes test that it doesn't slow down too much during the corners.

When testing the car some conditions were kept as constant as possible. These were the lighting of the track; the front lights of the car were always on and putting the blinds up in the lab gave constant lighting conditions. The other was the charge of the battery, these tests were carried out with a freshly charged battery, as running the car for extended periods of time drained the battery, reducing its voltage output and therefore the speed for the same drive motor PWM values. For the results shown below constant PID values were used, though given more time it would be beneficial to gather results at a range of PID values to show the effect of PID on lap time. The PID values used were P = 0.88 I = 0.002 and D = 0.2

The times shown below were recorded using a stopwatch from a constant start point, this start point was to the the right of the crossroads, and travelling clockwise, however both directions produced similar times.

Table 5: Results at Most Consistent and Reliable Speed

Motor PWM on Corners	Motor PWM on Straights	Lap Time
0.4	0.63	10.95
0.4	0.63	FAIL
0.4	0.63	10.53
0.4	0.63	10.68
0.4	0.63	11.50

Table 6: Results at Faster Speed

Motor PWM on Corners	Motor PWM on Straights	Lap Time
0.45	0.68	FAIL
0.45	0.68	FAIL
0.45	0.68	10.21
0.45	0.68	FAIL
0.45	0.68	FAIL

The results above were generated without changing the PID values, and as shown above there is an ideal speed for this set of values which is reasonably consistent at getting the car around the track. There was a lot of time spent finding the ideal PWM values shown in Table 5, these results are not listed as the runs were mostly failures or much slower lap times. The second table shows that if we increase the motor PWM we can probably generate a faster lap, but we were not able to gather enough data at that speed to prove that the result is not an anomaly. This is because of the inconsistency of the car getting around the test track at higher speeds. Unfortunately, testing various PID values was an extremely time-consuming process, and further optimisation was not possible due to time constraints.

13.1 Testing against Last Year's Results

With regards to the goal of matching last years' time, our testing took place on a more complex track, and our best times were between 0.1s and 1s slower than last year's times, dependant whether they used their electronic differential. Their track and results are shown below for reference. In order to measure our car against our goal of being competitive with their design, testing was carried out using their track as well.

Figure 48: Last Year's Test Track [20].

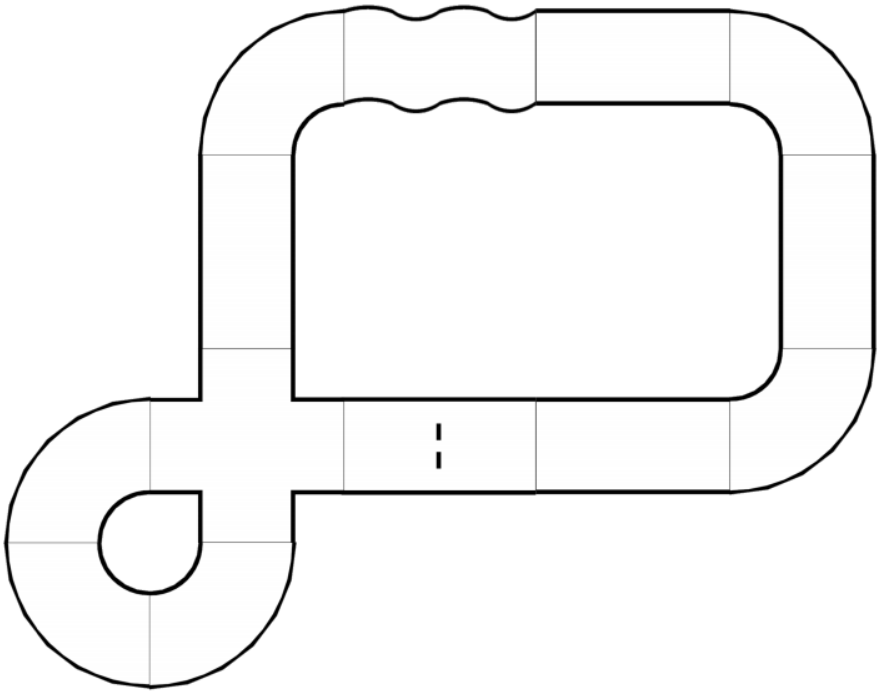


Table 7: Previous Years' Results Without Differential [20].

Test	1	2	3	4	5	6
Desired speed (rad/s)	120	120	125	125	130	130
Camera focal length (m)	0.15	0.15	0.15	0.15	0.15	0.15
P	2.20	2.00	2.80	3.00	3.20	3.00
I	0.60	0.60	0.50	0.50	0.50	0.40
D	0.25	0.25	0.30	0.30	0.30	0.40
time 1 (s)	10.37	10.12	9.97	10.53	10.00	9.72
time 2 (s)	10.15	10.45	10.58	10.00	10.14	9.81
time 3 (s)	10.36	10.69	10.49	10.22	10.03	9.81
time 4 (s)	10.42					9.86
mean average time (s)	10.33	10.42	10.35	10.25	10.06	9.80

Table 8: Previous Years' Results With Differential [20].

Desired speed (rad/s)	130	135	140
camera focal length (m)	0.15	0.15	0.15
P	2.20	2.20	2.20
I	0.57	0.55	0.55
D	0.44	0.50	0.50
differential value	0.60	0.60	0.60
time 1 (s)	9.98	9.10	9.37
time 2 (s)	9.91	9.39	9.01
time 3 (s)	9.84	9.48	9.96
time 4 (s)			9.21
mean average time (s)	9.91	9.32	9.39

Several laps of their test track were completed, using the PID and PWM values from our more consistent set of results, and are tabulated below in Table 9. These times average to 9.98s, which is within 0.6s of the best time they managed to achieve. This just misses our goal of being within 0.5s of their times, however it is extremely close. It is worth noting that their car ended up placed very highly in the overall NXP Cup rankings, and by that metric our car is still a success.

Table 9: Results on Last Year's Track

Motor PWM on Corners	**Motor PWM on Straights**	**Lap Time**
0.4	0.63	10.02
0.4	0.63	9.98
0.4	0.63	10.13
0.4	0.63	9.86
0.4	0.63	9.91

14 Limitations

The main limitations come from the camera communications and the hardware of the car. Some of these limitations were reduced as best we could via the design choices we made, however some were created by our own decisions.

The limitations from the camera are the lighting conditions and the resistance to crashes. For example, if the track is too dark then there is not a large enough contrast between the light and dark sections of the track meaning the camera algorithm cannot accurately determine the centre position, this lead to the development of the headlight array to guarantee consistent lighting. However, if there is too much light i.e. direct sunlight the camera cannot determine any difference between the light and dark sections of the track the camera just sees all white.

The camera mounts also make the car less resistant to crashes and collisions. A crash of the can easily misaligns the cameras, and dramatically affect the control. The camera also places a constraint on the speed as it only looks ahead a certain distance, if the car is moving fast enough the camera cannot process the image fast enough to find the centre point and deliver it to the control algorithm in time for it to respond to any changes upcoming corners/corrections that are needed. Another issue we discovered late in the testing is that speed sensing should have been implemented, as it would have allowed the motor speeds to be kept constant, and not affected by battery drain. This meant a lot of testing time was spend adjusting speed values, when more important work could have been carried out.

A decision not to implement wheel speed sensors early on in the project ended up having a large effect on the reliability of the car, however this was only discovered later on during testing. As the battery voltage drops, the speed of the car drops. We assumed that this drop would be negligible, however having conducted several long testing sessions, it became apparent that at the same motor PWM values, the actual speed of the car decreased relatively linearly throughout testing. As the optimal PID values are governed by the speed the car is travelling at, this meant a significant amount of time during testing was spend just changing the speed values up or finding working PID values that would later stop working as the battery charge levels changed. Wheel speed sensors would have alleviated this issue, by ensuring the wheels remained at a given speed, rather than a specific PWM value, and a traction control system to be implemented.

Finally, issues with the communications system, partially caused by unavoidable interference from the TFC board delayed its implementation dramatically. This in turn slowed down the tuning process as the car had to be reprogrammed each time a variable had to be changed, or unexpected behaviour needed to be debugged. If the communications had been implemented sooner, this tuning could have been done remotely, in real-time and allowed much more optimisation of values.

The main unchangeable limitations are the grip of the tyres, as they are not allowed to be changed under NXP Cup rules. At higher speeds, the car tends to either oversteer or understeer due to a lack of grip, and too much speed. Short of adding mass, or some kind of aerodynamics to the car, this must be worked around by ensuring the car is running at an optimal speed for the track section it is on/approaching, by being able to slow down for corners and speed up for straights.

15 Lessons Learnt and Future Improvements

If this project were to be undertaken again with the benefit of hindsight, the main change made to our methodology would be to make use of some sections of the code created by the code created by the previous team. Choosing to start the code from scratch removed the risk of carrying over any errors they may have made and allowed us to develop a novel solution. However, with respect to the communications and telemetry issues faced, the previous code could be used as a baseline to get that up and running so that it could have been finished much earlier and used for the tuning of the control.

More time spent optimising the control would also have been beneficial, as more values for the variables could have been tested to find the fastest speed that the rover can be controlled. Lastly, we should have allocated more time to writing the report as with exam season so soon before the deadline, the report was written in a very short time frame.

Finally, more time should have been devoted to initially planning the car and thinking through the possible repercussions of design decisions. This may have highlighted the use of speed sensing far earlier, and potentially had positive repercussions on the final outcome of the project.

16 Project Management

To manage the group throughout the project, we coordinated the team using the three services; email, Slack, and Facebook Messenger. Email allowed us to co-ordinate regular group meeting with our supervisor and was used to provide us with feedback from the two project reviews. Slack was used to discuss different sections of the rover due to the way different sub-groups can be created plus was used to share datasheets and links to other useful documents. For more informal conversation and discussing when to meet in the lab space, Facebook Messenger was used.

In order to share documents with one another for the project reviews and the writing of the final report, we utilised Google Drive, with a collaborative Word document being used for the final report. The Google Drive allowed us to share large files, particularly 3D models and videos for the presentations. The embedded programming for the rover took place using the Arm Mbed OS developer site, which includes an online SDK. A shared account was created, allowing us to collaborate on the code throughout the project, and keep previous versions easily accessible.

As with all Part 4 projects we were given a section of the dedicated laboratory space in Building 16. This allowed us to design and build various components of the rover. The Maker Space facilities were also locating in Building 16, which is where all the fabrication of custom parts took place. Due to the large size of the track which needs to be set up to test the rover, we obtained access to a larger space known as the Active Lab on level 3 of Building 32, where the track could remain setup, and more complex circuits could be designed.

DR managed the project budget, and all expenditure is shown in Appendix A. Out of the allocated £400, £297.16 was used. The largest purchase was two of the new Amalak car kits from NXP for $100 each. These were not used in the project as they were supplied with missing parts and no supporting materials for their custom controller boards, however, will be made to work in time for the actual competition round entered in March 2018. The rest of the budget was spent on various spare boards to allow simultaneous testing of code (one person could test a camera, whilst another board is mounted on the car), communications components, and various mechanical fixings that were not available through the university.

The original Gantt chart, charting the progress we expected to make, alongside an up to date one are included in the appendix. Generally, the timings were reasonably accurate. Getting the car moving initially happened far quicker than expected, however other complexities delayed the implementation of several stretch goals.

17 Conclusions

The goal of the project was to build an autonomous racing car that was able to compete in the NXP Cup and navigate an unknown track at a high speed. To that end, the project was a success. It successfully completed hundreds of laps of various test tracks throughout the project, including all the obstacles we had access to. The final test track used for testing was the most complex we could build with the track section and floor space available. Reliable operation and timings were achieved on this track. The ability to achieve faster times was limited mainly by the mechanical grip of the front tyres, however the addition of speed sensing, traction control, and a 'look ahead' ability, as mentioned in the limitations section could result in faster times.

The main goal of being within 0.5s of last years' best time was not quite achieved, with our times being around 0.6s slower than their bests, however they had an extra team member, and produced one of the fastest cars in last year's competition, and as such even this result is still a success.

Despite not meeting some of the goals, and only discovering several limitations in the car late in testing, we have shown the potential in our current hardware and software design. Therefore, we should be able to dramatically improve the performance of the car in time for the competition.

References

[1] "Waymo." [Online]. Available: https://waymo.com/. [Accessed: 29-Dec-2017].

[2] "Roborace." [Online]. Available: https://roborace.com/. [Accessed: 29-Dec-2017].

[3] NXP, "Rules 2017/2018 | NXP Community." [Online]. Available: https://community.nxp.com/servlet/JiveServlet/download/335083-3-411052/NXP Cup 2017_18 EMEA Rules_NEW.pdf. [Accessed: 30-Dec-2017].

[4] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[5] B. Schwarz, "LIDAR: Mapping the world in 3D," *Nat. Photonics 2010 47*, Jul. 2010.

[6] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving."

[7] M. Harris, "Night Vision for Self-Driving Cars - IEEE Spectrum," *IEEE Spectrum*, 2017. [Online]. Available: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/do-selfdriving-cars-need-night-vision. [Accessed: 29-Jan-2018].

[8] B. Luey *et al.*, "Luey 1 A LIGHTWEIGHT, COST-EFFICIENT, SOLID-STATE LIDAR SYSTEM UTILIZING LIQUID CRYSTAL TECHNOLOGY FOR LASER BEAM STEERING FOR ADVANCED DRIVER ASSISTANCE."

[9] T. Higgins, "The End of Car Ownership - WSJ," *Wall Street Journal*, 2017.

[10] "On the Road to Fully Self-Driving."

[11] S. Gibbs, "Self-driving bus involved in crash less than two hours after Las Vegas launch | Technology | The Guardian," *The Guardian*, 09-Nov-2017.

[12] "WHO | Road traffic injuries," *WHO*, 2018. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs358/en/. [Accessed: 29-Jan-2018].

[13] "Accident and casualty costs (RAS60) - GOV.UK." Department for Transport, 2016.

[14] G. Janssen, H. Zwijnenberg, I. Blankers, and J. de Kruijff, "Future of Transportation Truck Platooning," *Tno*, no. February, pp. 1–36, 2015.

[15] G. Topham, "Semi-automated truck convoys get green light for UK trials | Politics | The Guardian," *The Guardian*, 25-Aug-2017.

[16] P. Kocher *et al.*, "Spectre Attacks: Exploiting Speculative Execution *," 2017. [Online]. Available: https://spectreattack.com/spectre.pdf. [Accessed: 29-Jan-2018].

[17] M. Lipp *et al.*, "Meltdown," 2017. [Online]. Available: https://meltdownattack.com/meltdown.pdf. [Accessed: 29-Jan-2018].

[18] S. H. Duffy and J. P. Hopkins, "Sit, Stay, Drive: The Future of Autonomous Car Liability," *SMU Sci. Technol. Law Rev.*, vol. 16, 2013.

[19] J. M. Anderson *et al.*, *Autonomous vehicle technology : a guide for policymakers*. .

[20] L. Harlow, O. Johnson, A. Piekarski, M. Stenham, and R. Taylor, "UNIVERSITY OF SOUTHAMPTON Faculty of Engineering, Science and Mathematics A group design project report submitted for the award of Electronic Engineering, Electrical & Electronic Engineering, Computer Science," 2017.

[21] "FRDM-KL25Z|Freedom Development Platform|Kinetis MCU|NXP." [Online].

Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/l-seriesultra-low-powerm0-plus/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z. [Accessed: 30-Jan-2018].

[22] "Development Platform for Devices | Mbed." [Online]. Available: https://os.mbed.com/. [Accessed: 30-Jan-2018].

[23] "Freescale Cup Shield for the Freedom KL25Z | NXP Community." [Online]. Available: https://community.nxp.com/docs/DOC-93914. [Accessed: 30-Jan-2018].

[24] "TSL1401CL Datasheet," 2016. [Online]. Available: http://ams.com/chi/content/download/250163/975677/file/TSL1401CL_DS000136_3-00.pdf. [Accessed: 29-Jan-2018].

[25] Up!, "UP Plus 2," *https://www.up3d.com/*. [Online]. Available: https://www.up3d.com/up-plus-2/. [Accessed: 01-Jan-2018].

[26] Epilog, "Epilog Legend Series: Epilog Mini 18, 24 and Helix." [Online]. Available: https://www.epiloglaser.com/products/legend-laser-series.htm. [Accessed: 01-Jan-2018].

[27] CREE, "Cree® 5-mm Round LED C512A-WNS/WNN." [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/1524/0900766b815247e8.pdf. [Accessed: 28-Jan-2018].

[28] "Lux Meter (Light Meter) – Android Apps on Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=com.tsang.alan.lightmeter&hl=en_GB. [Accessed: 28-Jan-2018].

[29] NXP/Freescale, "Freescale Cup line scan camera - Part I | NXP Community," 2013. [Online]. Available: https://community.nxp.com/videos/1469. [Accessed: 29-Jan-2018].

[30] C. Brosilow and B. Joseph, *Techniques of model-based control*. Prentice Hall, 2002.

[31] J. A. Rossiter and J. A. Rossiter, *Model-Based Predictive Control : A Practical Approach*. .

[32] E. Zafiriou and M. Morari, "Setpoint Tracking vs. Disturbance Rejection for Stable and Unstable Processes - IEEE Conference Publication," *Am. Control Conf.*, 1987.

[33] H. Kim, J. Kim, M. Shim, J. Jung, S. Park, and C. Kim, "A digitally controlled DC-DC buck converter with bang-bang control," in *2014 International Conference on Electronics, Information and Communications (ICEIC)*, 2014, pp. 1–2.

[34] Q. Meng, Y. Wang, F. Xu, and X. Shi, "Control strategy of cement mill based on Bang-Bang and fuzzy PID self-tuning," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2015, pp. 1977–1981.

[35] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, "PID Controllers," in *Feedback Control of Computing Systems*, Hoboken, NJ, USA: John Wiley & Sons, Inc., pp. 293–335.

[36] Kiam Heong Ang, G. Chong, and Yun Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[37] H. Unbehauen, “CONTROL SYSTEMS, ROBOTICS, AND AUTOMATION,” in *CONTROL SYSTEMS, ROBOTICS, AND AUTOMATION*, vol. 1, EOLSS, 2009.

[38] “nRF24L01 / 2.4GHz RF / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01. [Accessed: 31-Jan-2018].

[39] Bastille Research, “nRF24LU1+ breakout board.” [Online]. Available: https://github.com/BastilleResearch/mousejack. [Accessed: 30-Jan-2018].

[40] Jpbarraca, “Pynrf24.” [Online]. Available: https://github.com/jpbarraca/pynrf24. [Accessed: 30-Jan-2018].

[41] Sparkfun, “XBee 2mW Wire Antenna - Series 2 (ZigBee Mesh),” *Sparkfun Electronics*, 2013. [Online]. Available: https://www.sparkfun.com/products/10414. [Accessed: 30-Jan-2018].

[42] T. JIMB0, “Exploring XBees and XCTU.” .

[43] S. Schmit, “XBee AT Mode - Transmit and Command Mode Example.” 2014.

[44] M. Bowen, “XBee Configuration: Coordinator and Router.” 2013.

[45] Jan R. Magnus and Heinz Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. 1999.

[46] P. G. Howard, “The design and analysis of efficient lossless data compression systems,” 1993.

[47] D. Lemire, “Fast integer compression: decoding billions of integers per second.” 2012.

[48] “Usability of irreversible image compression in radiological imaging. A position paper by the European Society of Radiology (ESR),” *Insights Imaging*, vol. 2, no. 2, pp. 103–115, Apr. 2011.

[49] K. J. Anil, “Image data compression: A review,” *Proc. IEEE*, vol. 69, 1981.

[50] B. K. C. Krste and Asanovi$\$’c;, “Energy-aware lossless data compression,” *ACM Trans. Comput. Syst.*, vol. 24, pp. 250–291, 2006.

[51] Digi International, “XBee Datasheet,” 2009.

[52] B. P. Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, “Numerical Recipes: The Art of Scientific Computing,” *Cambridge Univ. Press*, 2007.

[53] M. Maxfield, “Gray Code Fundamentals.” 2011.

[54] P. Community, “PyGame Documentation.” [Online]. Available: http://www.pygame.org/docs/. [Accessed: 30-Jan-2018].

[55] W. McGugan, *Beginning game development with Python and Pygame: from novice to professional*. 2007.

Appendices

Appendix A Project Budget

GDP Budget		Grand Total of Expenditure	£297.16
Order Details	**Items**	**Quantity**	**Cost**
Date: 2017-12-08	Xbee USB Adapter	2	£30.00
Supplier: HobbyTronics	DELIVERY CHARGE	1	£2.88
		Total	**£32.88**
Date: 2017-11-07	Adafruit 1404 Servo	2	£27.46
Supplier: Mouser Electronics	M3 Spanner	1	£12.80
		Total	**£40.26**
Date: 2017-10-30	M4 x 50mm	1	£4.16
Supplier: RS Components Ltd		**Total**	**£4.16**
Date: 2017-10-30	KL46Z	1	£30.43
Supplier: Mouser Electronics	C-GRID III 16 CKT PC	2	£5.28
	C-Grid 20ckt	1	£3.16
	C-Grid 12 CKT	1	£2.00
		Total	**£40.87**
Date: 2017-10-30	M3 x 10mm Pan Head	1	£2.65
Supplier: RS Components Ltd	Pan Head Machine Screws, M3 x 20mm	1	£3.28
	M3 Lock Nut	1	£3.47
	M3 Hex Nut	1	£2.38
	Nylon Washer M3	1	£1.79
	Steel Rule	1	£9.30
	3m USB Mini Type A Cable	2	£5.28
		Total	**£28.14**
Date: 2017-10-17	CreeC503D-WAN 9000K White LED, 5mm	50	£9.42
Supplier: RS Components Ltd		**Total**	**£9.42**
Date: 2017-10-17	NXP CUP Car Kit - Amalak	2	£141.42
Supplier: NXP USA		**Total**	**£141.42**

Appendix B Original Project Specification

School of Electronics and Computer Science
ELEC6200 MEng Group Design Project

Project Specification and Plan

Title:
Autonomous Electric Car Racing

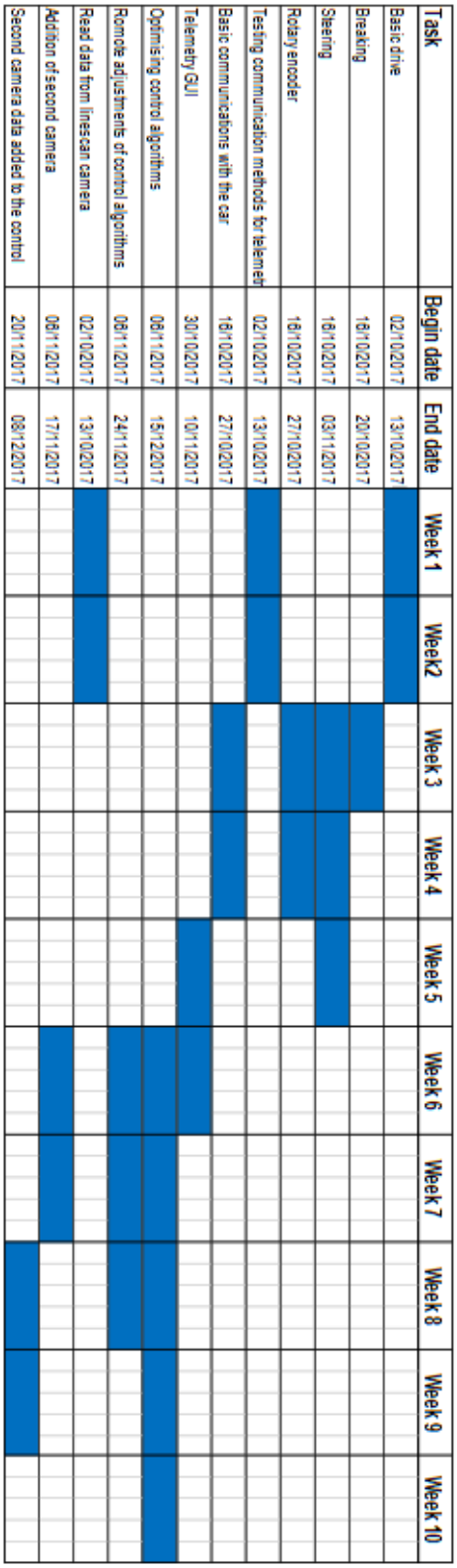
Supervisor:
Kirk Martinez

Team Members:
Matthew Brooks
Arjun Patel
Damon Roberts
Man-Leong Chan

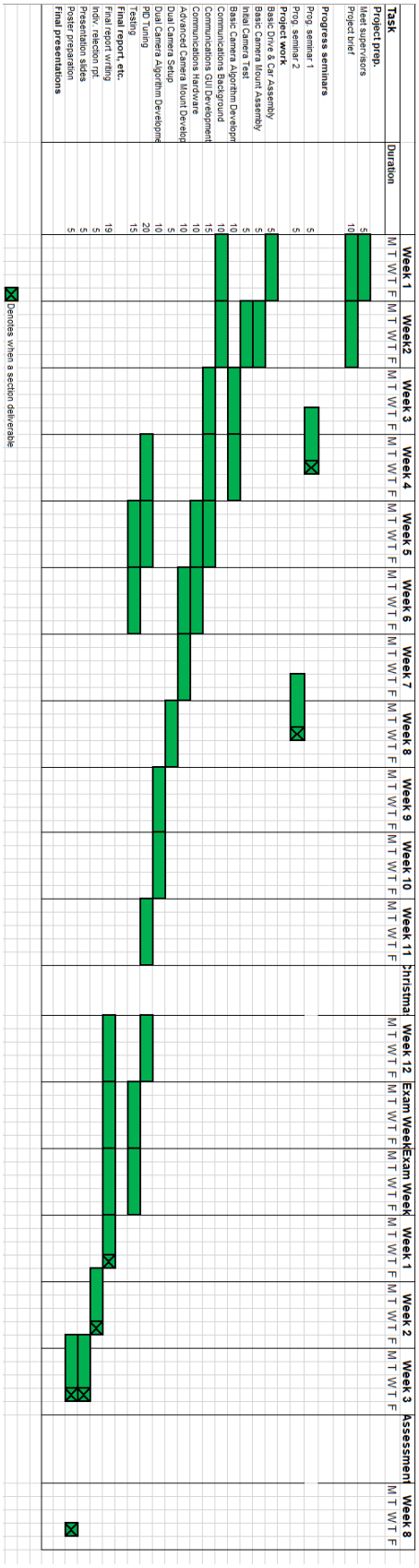
Customer:
NXP Cup

Project Specification:
The aim of the project is to enter the NXP cup where the challenge is to drive around an unseen racetrack as fast as possible with a small autonomous car. Therefore, this project involves the design and implementation of: motor control, steering control and image processing of the linescan camera. In addition to these features the project will also require a telemetry/testing suite to improve the development of the control algorithms used.
Stretch goals for this project are: implementation of an additional linescan camera for faster cornering, the addition of torque steering and some self-optimisation using the telemetry to determine the most efficient set of coefficients for the control algorithms.

Appendix C Original Gantt Chart



Appendix D New Gantt Chart



Appendix E MBed Data Type Specification

Integer Data Types

C type	stdint.h type	Bits	Sign	Range
char	uint8_t	8	Unsigned	0 .. 255
signed char	int8_t	8	Signed	-128 .. 127
unsigned short	uint16_t	16	Unsigned	0 .. 65,535
short	int16_t	16	Signed	-32,768 .. 32,767
unsigned int	uint32_t	32	Unsigned	0 .. 4,294,967,295
int	int32_t	32	Signed	-2,147,483,648 .. 2,147,483,647
unsigned long long	uint64_t	64	Unsigned	0 .. 18,446,744,073,709,551,615
long long	int64_t	64	Signed	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807

Appendix F Project Archive Data Listing

Contained in the project archive is the final MBed programming running on the car, as part of this program there are three libraries: cameraLib, controlLib and motorLib. The open source libraries for the TFC board and the accelerometer are not included.

Also included are the CAD files for the board mount, final camera mount, headlight array and ramp. As well as a variety of drawings and photographs of those parts.

The telemetry software and all dependencies are also included.

A readme is provided within the file itself.