Good Referencing & Weekly Blog as appe...

Do Not Remove from CO145

AVC Report

ENGR 101 2015 T1

Name: [redacted]
Student Number: [redacted]
Team/Lab Session: Team 5, Thursday 2pm – 4pm
Team Members: [redacted]

Abstract

In industry, it is incredibly rare for a programmer to work alone on a specific task. Nowadays teams work in groups to achieve a certain goal. The majority of the time, developers are also given time and monetary constraints to which they must keep to. As a result, the autonomous vehicle challenge will give us our first glimpse into how projects are handled in industry but within the confines of the ENGR101 course. We will also gather the basics concepts of mechatronics and robotics which will be useful in future engineering courses and also in industry.

Introduction

Scope

The scope of this challenge is to allow us to understand fundamental concepts of electronic engineering and the basics behind robots and mechatronics. We will have learnt skills in lectures and in previous laboratories that will enable us to create an autonomous vehicle to navigate a maze in a group of five to six people. We will also be introduced to modelling for creation of parts needed for our robot. As this is our first real "group" task, we will be introduced to group programming and working as a team as well as meeting a deadline and keeping the project under budget.

Motivation

The motivation for this challenge is that robotics is a major part of an electronic engineering degree, students must also have the basics understood for other engineering majors such as software engineering and network engineering. As a result, on completion of this challenge we will have gained vital skills and techniques that will allow us to continue in the engineering specialisation and possibly even industry. These skills and techniques may be required in future engineering assignments and laboratories. Investigating tools such as 3D printing and creating models will be useful for future engineering papers where modelling is a key component. As well as this, working as a team to fit a brief under budget and in time is commonplace amongst industry and as a result this challenge will prepare us for what is to come in the future.

Aim

The aim for this challenge is to work as a team to successfully build, construct and program an autonomous vehicle that is able to navigate a four quadrant "course". The first quadrant is a single line in which the robot must follow, the second is a line maze with intersections and dead-ends, the third is a maze with no lines but walls and the fourth is a maze with walls and obstacles. The robot should wirelessly transmit its location back to a host computer and should not require any human interference. This will need to be achieved under budget, within our time constraints and successfully by working as a team with equal contribution from all.

Objectives

The objectives for this challenge are to:

- Learn the basics of a PID system and how it operates.
- Be able to understand and use modelling programs to create parts for our robot.
- Work as a team to create an aesthetically pleasing, recyclable autonomous vehicle that is able to navigate a maze using analogue and digital distance sensors, a colour sensor and motors.
- Develop the automated vehicle so it is made under budget and within our time constraints.

Anticipated Benefits

As this lab is one of the first real "group" tasks we will encounter at university we will learn the basics of functioning as a team and how different roles should be distributed. We will also learn what not to do when functioning in a group and what works well for future team activities. We will also have an idea of how to plan a project so that it comes in under budget and on time.

Background

An autonomous vehicle is defined as a device "which is able to perceive its environment, decided what route to take to its destination and drive it.... Without the need for human supervision or operation." (Yeomans, 2014). In order to fit this definition, our "vehicle" must be able to detect its surroundings and respond appropriately. No "human supervision or operation" means that we are not to guide, touch or help our device through the quadrants, doing so would not result in our robot being "autonomous".

The challenge for this task to be able to complete a four quadrant maze, the first quadrant consists of a "wiggly white line on black background", the second quadrant consists of a white line arranged in a maze pattern, the third quadrant is made out of a 100mm high wall and the fourth quadrant is the same configuration as quadrant three but has additional obstacles (in our case two automatic doors). Our robot must also communicate to a host computer and send its location (in terms of quadrant), look aesthetically decent (i.e. not a mess of wires and parts) and be recyclable (i.e. the hardware must be able to be reused). When it comes to challenge day, we will be given 15 minutes to prove that our vehicle can meet these specifications to our customers – the Engineering department staff and ENGR101 tutors. We are allowed to restart our robot from the start as many times as we liked however we cannot touch the robot in any way once it starts going around the course (Victoria University of Wellington: School of Engineering and Computer Science, 2015). An image of some of the track can be seen in Figure 1 above.

FIGURE 1: PART OF THE COURSE OUR ROBOT HAD TO NAVIGATE

Our robot will be constructed from several different sensors and motors all connected to an Arduino. An Arduino is in essence a microcontroller developed for students and other robotics amateurs that want an introduction into the mechatronics world, it is open source and cheap to run (Cuartielles, 2013). The Arduino can be configured using a special integrated development environment. The language that the Arduino is coded in is based upon C/C++ (Arduino, 2015) and in our case the microcontroller we will be using is the Arduino Uno. In this report, the terms short and medium range IR sensors, QTR line follower sensor, Vicmoto motor driver board, DC motors, LiFe batteries and Xbee radio transceiver will be used. Below is a summary of what each component does.

An IR sensor, short for infrared sensor, measures infrared light in order to detect the "distances" between it and objects in front. It works by detecting energy given off by other objects (Venice Locksmith, 2014). For our AVC we will be given both digital (short IR) and analogue (medium IR) sensors so our robot can detect objects around itself. Our digital sensor returns either a 0 (nothing in range) or 1 (something in range), for our model the range is between 0.5cm and 5cm (Pololu Corporation, 2001-2015). Our medium analogue distance sensor is capable of measuring distances between 4cm and 30cm (Pololu Corporation, 2001-2015). Determining the actual distance with sensor is difficult however as it returns a numerical voltage which needs an equation to be converted into a distance – more on this in the "method" section. A QTR line sensor (in our case the Pololu QTR-8RC) is a colour sensor which reads the reflective values given off by the colours underneath it (Pololu Corporation, 2001 - 2015). We are able to interpret these values through a library and use them to follow a line. For our motors, we were required to use a VicMoto motor shield which gives us the required extra pins to connect the DC motors as well as other sensors onto our Arduino. The DC motors are small electrical machines that convert direct current into mechanical power (Daware, n.d.). They were one of the first type of motors widely used (Herman, 2009). The battery we will be using to power our device is a LiFe (Lithium-Iron-Phosphate battery) which according to the supplier "offers you more power and capacity in a smaller and lighter package" (Hobbico Inc., 2015). The use of a battery means USB power is not necessary allowing for a lot more flexibility and mobility. The transceiver we will be using to transmit the location of our device is an Xbee controller. An Xbee controller is similar to a Bluetooth one in that it is a type of personal area network (PAN) and can only transmit/receive within a small range (Digi International Inc., 1996 - 2015). Xbee transceivers allow us to easily "broadcast" messages to other Xbees in the area making it easy and efficient for our autonomous vehicle challenge.

Lastly, in order to control our robot, we will be using a range of different control systems. One of these systems is the PID controller (stands for the proportional, integral, derivative controller). In essence, a PID controller calculates an "error" value based on where the line is underneath the QTR sensor. Following this an algorithm calculates proportional, integral and derivative values. These values are calculated based off the calculated past, present and future values based on its current rate of change (Araki, 2003). The proportional value allows for the current error to be corrected, the integral allows for future errors to be corrected (i.e. moving in a squiggly line when it's supposed to be moving straight) and the derivative allows for the error to be corrected slowly instead of rapidly to avoid "jerky-ness". These values can then be used to adjust the motors so that the system (in our case an autonomous vehicle) can stay on track. A diagram of how the PID controller operates can be seen in Figure 2.

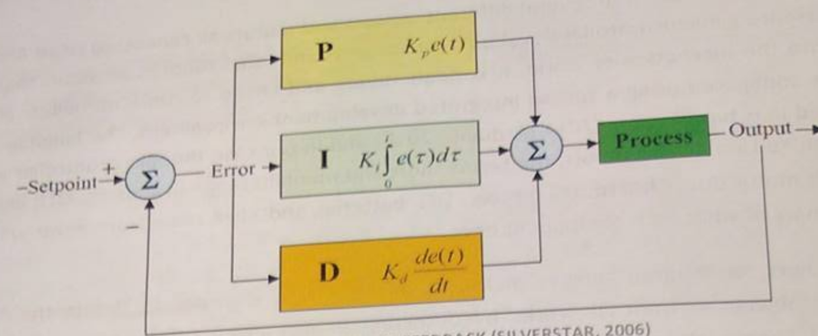

FIGURE 2: PID CONTROLLER FEEDBACK (SILVERSTAR, 2006)

We used several different tools to develop parts for our robot. When we needed 3D models for parts that needed to be laser cut/3D printed we used the open source 3D computer-aided design program FreeCAD (available from http://www.freecadweb.org/). FreeCAD allows for beginners to develop small/simple models that can be sent to be 3D printed or laser cut. This can be done using a drag and drop user interface with different types of shapes. Another group member opted to use Sketchup when modelling up their ideas. This is similar to FreeCAD however this team member was more experienced with SketchUp so chose to use it instead. When we needed to get our models 3D printed, we opted to use one of two 3D printers the ECS department had on offer. We simply saved the model as a .stl file and uploaded to the printer's computer (using a USB stick) and pressed print. For objects that needed to be laser cut, we emailed the electronics department (electronics@ecs.vuw.ac.nz) with the model saved and attached as a .dxf file, our group name and our lab time specifying the plastic's thickness and colour. It would then be ready for collection within the next few days.

Method

Note: A more accurate version of events can be seen in our group's "weekly blog" which can be found at the end of this report.

During the first week back from the mid-trimester break, we were introduced to our team of five, from there we developed rough sketches on concepts for our vehicle. We argued and reasoned why we thought each design was better than the other but eventually came to a unanimous decision on what design we would go with. Upon coming up with this decision we developed a relatively loose schedule of events and milestones and when we wanted things completed. This was documented in a Gantt chart which can be seen in Figure 3 below. It was created using GanttProject, available for free from http://www.ganttproject.biz/.

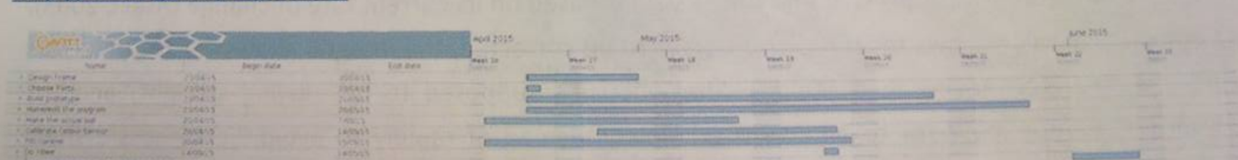

FIGURE 3: THE INITIAL GANTT CHART WE CREATED SHOWING OUR TIMELINE OF EVENTS

After completing our projected timeline, we divided up the tasks that needed to be done into two separate parts; one being software, the other being the "design" aspect (i.e. development of the body, placement of the mounts, aesthetics etc.).

For the design we started off by attaching our IR sensors and DC motors by using electrical tape and attaching it to a modified CD, this can be seen in Figure 4 with one of the earliest versions of our design. By performing basics tests such as manually turning our robot and pushing the robot through the maze we determined our final specifications for our prototype and where the wheel bases, IR and QTR sensors, Arduino and it's shields could go. Following this, we mocked up designs for the motor mounts and base on FreeCAD, an open-source modelling tool that allows the schematics to be exported in order to be 3D printed or laser cut (available at http://sourceforge.net/projects/free-cad/files/FreeCAD%20Windows/). To be 3D printed, the schematics needed to be saved as a .stl format, for laser-cutting the file needed to be saved as a .dxf file. These were designed both during the ENGR101 laboratory in CO145 and also during spare time at one of our team member's houses. Following the initial construction of our robot, we reconstructed some of the mounts to fit the specifications that we wanted and fine-tuned (i.e. sanded/filed down mounts, added in more supports) and changed where the LiFe battery and Arduino Uno were mounted in order to maximise weight distribution, an example of us fine-tuning our design can be seen in Figure 5. Following this we joined all the parts together and created our final prototype. The parts were either joined together with screws and nuts, hot glue and in some cases blue tack. To top it all off, we added pipe-cleaners to give our robot a nicer look and also for cable management. An image of our final design can be seen in Figure 6.

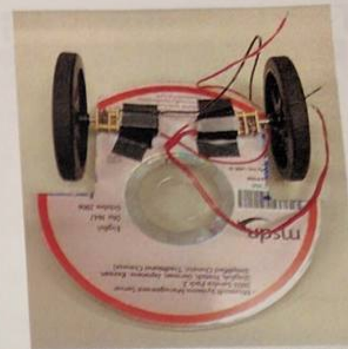

FIGURE 4: WEEK 1 DESIGN OF OUR VEHICLE

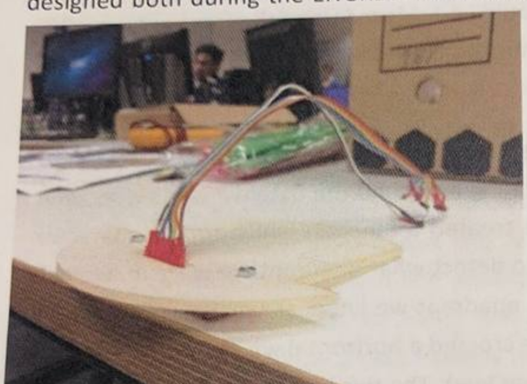

FIGURE 5: LASER CUT BASE OF OUR PROTOTYPE WITH COLOUR SENSOR

A small "lip" on the bottom of the base can be seen in figure 6, this was placed to allow the robot to move on the course/reduce friction and not get stuck on the course.

The other important aspect of our vehicle was the software. The software was mostly developed in CO145 (and in some cases on team members' Windows laptops) running a distribution of Linux. It was done through the Arduino 1.6.1 integrated development environment (IDE) which is coded in C/C++. We first started by creating a basic outline of our code using comments and began by developing our code quadrant by quadrant. Developing quadrant one and two were our main priorities. We began by coding up tests for our QTR and IR sensors so we could see how they worked (see Figure 7) – this was so we could check the tolerance value for the white strip and also detecting the values that the IR and QTR sensors returned. Following this, we investigated different libraries that would enable us to create the features we needed. Upon further

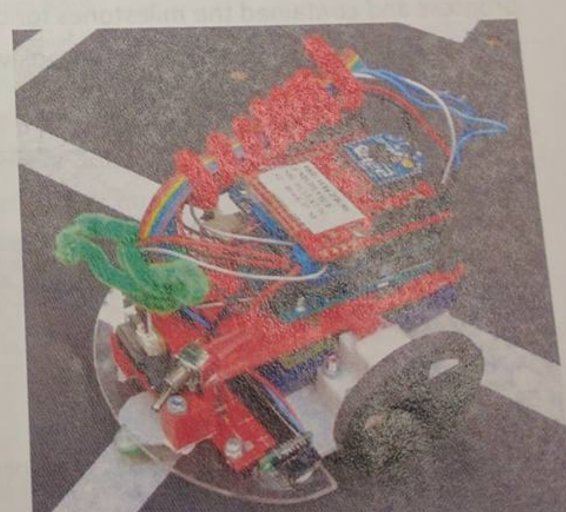

FIGURE 6: OUR FINAL DESIGN

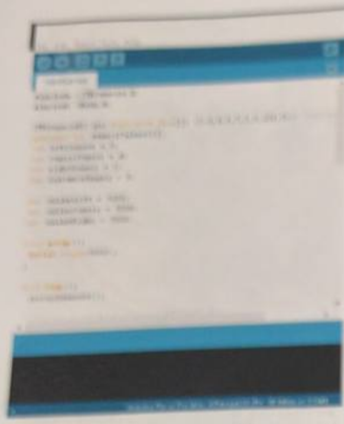

FIGURE 7: AN EXAMPLE OF OUR QTR SENSOR CODE

PID library and used the error function and derivatives/integrals to PID control (Roberts, 2015). Upon developing the first two quadrants, we then began to code up quadrant three. This was done by reading the two digital sensors we had placed on the left and right of the vehicle and a front analogue sensor. Using a range of logic and testing by printing to the serial monitor, we determined which way the vehicle had to turn depending on what sensors were on or off. We decided that if the analogue sensor was under a certain tolerance (i.e. if it was heading straight into a wall) that it needed to turn.

We then began to code quadrant four however ran out of time. If we had the time, we would have made the robot check for the line (before the door), it would know it was in quadrant four and wait for the door to open. This would be achieved using the front analogue IR sensor. Following this, the robot would use quadrant three code to make its way through the rest of the track.

One of the key aspects of the design was networking. Coding up the Xbee was relatively straight forward as it involved using the code provided by Arthur and manipulating it to send the data we wanted to send (e.g. QUADRNT 1, QUADRNT 2 etc.). In order to do this, we created a function which took a character array and broadcast it to all nearby Xbee receivers. In order to detect what quadrant we were in, we had separate functions in each of our "quadrant" code. The first quadrant we knew as we always started in quadrant one, the second quadrant was determined when we crossed a horizontal white line on the track (i.e. the values provided by the QTR sensor would have all been low). The third quadrant was determined when the digital sensors on the sides of the robot detected something on both sides and the fourth quadrant was determined when it detected a horizontal line while inside the third quadrant.

Upon finishing the project, we redesigned our Gantt chart so that it more accurately displayed our progress and contained the milestones for our AVC. This can be seen in figure 8 below. As we progressed through our project, we also wrote a weekly blog. This can be seen at the end of the report.

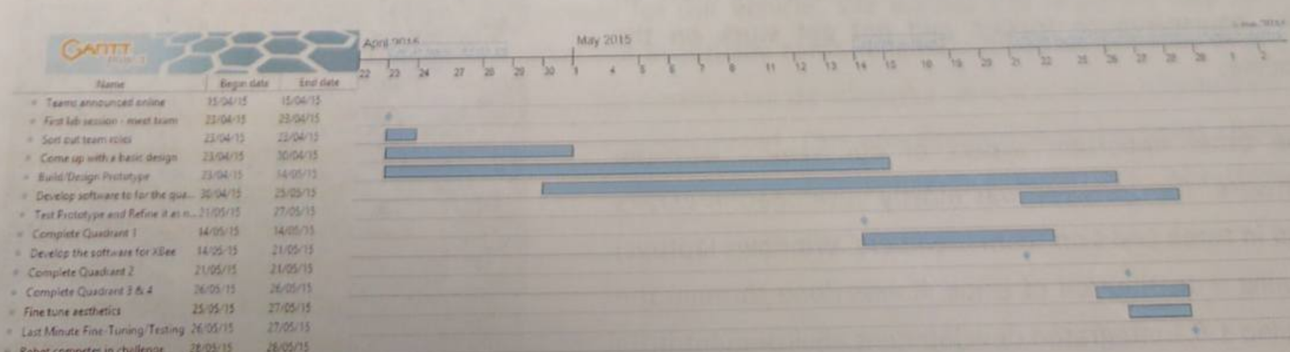

FIGURE 8: OUR REDESIGNED, MORE ACCURATE GANTT CHART

Results

During our ENGR101 lab in CO145 on Thursday 28th May, we were given fifteen minutes in order to prove our robot could navigate the maze to our "customers" – the ENGR101 tutors and other members of the Engineering faculty. During the fifteen minutes, we were allowed to pick up our robot (if it were to get stuck or break down) and restart it from the start of the track. During the demonstration, we were not to touch the robot in any way, shape or form.

FIGURE 9: OUR ROBOT TURNING AROUND AT THE START OF Q4

Our first attempt was not entirely successful, we managed to get to approximately the 75% mark (part way through quadrant three) where upon turning around a left hand corner, our device got stuck. For our second attempt, our robot made it to the start of quadrant four however as we hadn't configured it properly, it thought that it had reached a dead end and turned around. One of these attempts can be seen in the video at this link: https://goo.gl/WafYhq (a video still can be seen in Figure 9). This continued for the next couple of attempts until we managed to figure out when to release the robot at the correct time. Once we had figured this out, we managed to get it past the first door but it was a matter of luck to get it past the second door. Eventually after about five or six tries we managed to get it through both doors and progressed to 100%. Throughout the attempts, the robot performed flawlessly through quadrants one and two (see figure 10 and the video link above) and only got stuck in quadrant three on the first attempt. The other times, the robot simply failed because we ran out of time to code quadrant four and instead had to rely on timing instead. Although the robot did get dangerously close to the walls in quadrant three and four, our code managed to allow our robot to pass through the course relatively well.

FIGURE 10: OUR ROBOT PASSING THROUGH Q1

Having completed 100% of the course, our robot managed to get into the finals to be held on Tuesday 2nd June. As a result, we will have more time to create quadrant four code and fine-tune code for quadrant three. Come Tuesday, the robot should perform substantially better than it did on test day.

Discussion

Overall, I believe that the project went well in terms of keeping to budget constraints however I did have a couple of issues regarding time management. We could have better organised ourselves in terms of time as we spent quite a lot of time towards the end of the project getting everything together. I think the time period of six weeks was generous however the time spent outside of the laboratories would need to be worked on. For example, our group (for the first couple of weeks) only worked on the robot during our weekly two hour laboratories, something we regretted as the deadline loomed closer. As a result, we spent the last two days leading up to the challenge (and the morning of the challenge) working on our robot non-stop. This meant we did not successfully manage to code quadrant four at all and we ended up fine-tuning and testing quadrant three up until the last half an hour before the challenge were to take place. Because we had to refine our quadrant three code so late in the process, we did not manage to fully test it and although it made it through five out of six times, it did get dangerously close to the walls when taking left turns. As the robot managed to get to 100% albeit with a lot of trial and error, we could have properly got to 100% with quadrant four code had we managed our time a little better. As a result of getting 100% we did however make it to the finals giving us another week to program our quadrant four code and actually refine the code we had for quadrant three. Consequently, as how we reach quadrant four does not contribute to the actual grade (as long as we get there) it shouldn't matter what approach we took (Victoria University of Wellington: School of Engineering and Computer Science, 2015). Because relying on trial and error is incredibly risky it would have been better to actually code quadrant four instead of relying on timing.

Another aspect that is closely related to our time constraints problem was access to the CO145 engineering lab. Although in the weeks leading up to the challenge we had mostly unrestricted access (except for when another ENGR101 lab was running) I found when it came to the week leading up to challenge the laboratory was always packed and it was impossible to get a seat. If somehow we did manage to get a seat, we were only allowed one group member per team in the laboratory at the same time. This did not allow us to collaborate properly on the challenge and we had to rely on one person in the group to do the work at that point. Next year, I would recommend having a backup lab with another practice track so that groups can work together and aren't turned away from the lab. During the week of challenges, we weren't allowed to use the track during other groups' lab times. As a result of this, our group was forced to make a practice test course upstairs in the hallways of the Cotton building (see figure 11). We could not accurately test our robot on this course either as the reflectivity of the mats were slightly different to that of the proper course.

FIGURE 11: OUR PRACTICE COURSE WITH Q1, Q2 AND Q3

Access to resources was also a problem we had with 3D printers. As CO145 only had two in operation we often had to wait for other groups to finish their prints before we were allowed to start. As some groups would print out large cuts (which arguably would have been better laser cut) we would have to wait for a long period of time. Other groups would also cut in front of our turns stealing the printer and forcing us to wait even longer for a printer to become available. We ended up resorting to coming in outside of laboratory times and early in the morning to get our models printed so we could use them during our allocated lab slot.

In terms of team work, I believed the majority of the team pulled their weight and contributed well to our autonomous vehicle. Although some members were more efficient than others most of us managed to contribute to either the body or the software or in some cases both. One criticism I would have would be that five in a group may be a bit too many. During the allocated lab slots, we often had a couple of team members floating around simply because there was no room for them to do any work. Although they would do work in other ways, during some of the lab sessions nothing much could be done because we only had two Arduinos to test with (we also had a team member's personal Arduino to use). I found working as a group on this project contributed significantly however and gave me a great insight into team work and sharing the load of a project.

Hardware problems were another massive issue we had to deal with in particular a lot of the connections between the hardware connection (i.e. the wire) and the actual hardware itself. Our Pololu QTR Line Follow sensor was not in the best shape possible when we received it, resulting in us having to solder almost all the pins back together over the course of about two weeks. Another issue with our QTR sensor that we didn't pick up on for a while was that our ground pin needed to be soldered back together, this hindered our ability to complete quadrant one code in the first two or three weeks of prototyping. Instead of soldering our QTR sensor back together, we did ask for another one however there were none available. In my opinion, the department should have some spare parts in the situation that a QTR sensor was not immediately repairable otherwise other teams could be severely disadvantaged. In terms of other hardware not available, when we asked for a switch one had to be removed from a robot that had already competed in its challenge and again, there was not a switch that was readily available. We also had a problem with our VicMoto motor driver board connecting to our Arduino Uno. Although the majority of the time the two connected with no significant problems, sometimes, some of the pins would not connect properly resulting in our left motor spinning backwards at full power. This was an issue during early development of our prototype as we could not figure out why our DC motors weren't functioning as they were supposed to.

Another hardware issue I had was the lack of knowledge on how the Xbee works. Although the Xbee was covered briefly in the lectures I still didn't have much of an idea on how it actually worked. We were given a script to copy and paste into our code (with some adjustments) however I didn't like this as I didn't properly understand how it operated. I did however look up some documentation on the internet about Xbee which clarified things slightly but in my opinion networking, in particular Xbee, should be covered more in lectures – similar to how the IR, QTR and control systems were covered by Arthur.

Lastly, I have a few concerns that our robot managed to make it to the finals. As mentioned in the results section, we only got to 100% by luck and a lot of trial and error. In my opinion, only the best, well-programmed robots should make it to the finals and the only criteria that the robot make it to 100% should not theoretically apply. In the AVC rules it states that "the top six (or seven) robots that earned the most points will go to a Grand Challenge Finals Night during week 12". I believe that our robot is probably not in the top six (or seven) however I am not complaining that we made it to the finals. I think that more criteria should be added or robots should be entered into the finals at the judges' discretion.

In terms of aspects of the challenge that went well, I believe that using wheels over tracks was a great advantage. Speaking to other groups who used tracks, they experienced far more difficulty than us and one group eventually ended up switching to using wheels instead. The fact that we had two Arduinos was incredibly useful too. One of our team members lent us their personal Arduino Uno allowing two members of the group to code up the software at the same time. This was also handy when it came to designing the body as it meant we didn't need to stop coding for measurements to be taken or parts to be assembled etc. Lastly, having the tutors around greatly increased my understanding of the topic and I believe they were fundamental in allowing us to complete the challenge. Having them around for advice or guidance allowed us to solve problems that would have taken us a lot longer to solve by ourselves. I also believe the effort put in by Arthur was fantastic, he was always available from early in the morning to late at night when we needed something re-soldered, required a battery or we needed to ask him a question.

Conclusion

To conclude, the autonomous vehicle challenge allowed us to think critically and constructively in a group environment under cost and time constraints. It introduced us to what is now commonplace amongst industry and how most tasks these days are not done independently. It also gave us an insight into the basics of mechatronics and robotics which will be covered in more depth later on in the Engineering degree. During this challenge, our robot managed to complete the aim which was to successfully navigate the entire four quadrants of the AVC maze constructed by Arthur and as a result, made it to the finals. Throughout this challenge, we have been encouraged to work in a cooperative team environment and put in extra hours outside of designated laboratories to complete this task in time and given a certain amount of "money" to spend. Because of this challenge, we will have the fundamental knowledge of completing a task under budget, in time and successfully as a group, a skill vital amongst industry and throughout our time at University.

References

Araki, M. (2003). *Control Systems, Robotics, and Automation - Vol. II - PID Control.* Kyoto, Japan: Encyclopedia of Life Support Systems (EOLSS).

Arduino. (2015, March). *Frequently Asked Questions*. Retrieved May 31, 2015, from Arduino: http://www.arduino.cc/en/Main/FAQ

Cuartielles, D. (2013). *Arduino FAQ - with David Cuartielles.* Malmö University: Prototyping Futures.

Daware, K. (n.d.). *Universal Motor - Construction, Working And Characteristics*. Retrieved June 4, 2015, from ElectricalEasy.com: http://www.electricaleasy.com/2014/02/universal-motor-construction-working.html?m=1

Digi International Inc. (1996 - 2015). *XBee*. Retrieved June 5, 2015, from Digi: http://www.digi.com/lp/xbee/

Herman, S. (2009). *Industrial Motor Control 6th Edition.* Boston: Delmar Cengage Learning.

Hobbico Inc. (2015). *LiFeSource Lithium-Iron-Phosphate Batteries*. Retrieved May 31, 2015, from LiFeSource: http://www.lifesourcebatteries.com/

Pololu Corporation. (2001 - 2015). *QTR-8RC Reflectance Sensor Array*. Retrieved June 5, 2015, from Pololu: https://www.pololu.com/product/961

Pololu Corporation. (2001-2015). *Pololu Carrier with Sharp GP2Y0D805Z0F Digital Distance Sensor 5cm.* Retrieved from Pololu: https://www.pololu.com/product/1132

Pololu Corporation. (2001-2015). *Sharp GP2D120XJ00F Analog Distance Sensor 4-30cm*. Retrieved from Pololu: https://www.pololu.com/product/1136

Roberts, A. (2015, May 5). ENGR101: Lecture 26. *Control systems, mechatronics*. Wellington, New Zealand.

SilverStar. (2006, October 26). *File:Pid-feedback-nct-int-correct.png*. Retrieved from WikimediaCommons: http://upload.wikimedia.org/wikipedia/commons/4/40/Pid-feedback-nct-int-correct.png

Venice Locksmith. (2014, June 24). *Home Alarm PIR Tech*. Retrieved from Home Security Technician's Notebook: http://steveslockandsafe.com/venice-locksmith-home-security-technicians-notebook/

Victoria University of Wellington: School of Engineering and Computer Science. (2015, April 21). *AVC Rules*. Retrieved from Victoria University of Wellington: School of Engineering and Computer Science: http://ecs.victoria.ac.nz/foswiki/pub/Courses/ENGR101_2015T1/LectureSchedule/AVCrules_2015.pdf

Yeomans, G. (2014). *Autonomous Vehicles - Handing Over Control: Opportunities and Risks for Insurance.* London, UK: Lloyd's Exposure Management. Retrieved from https://www.lloyds.com/~/media/lloyds/reports/emerging%20risk%20reports/autonomous%20vehicles%20final.pdf

Weekly Blog

Week 6 – 23rd April 2015

Matthew Lee

This is the first week of our AVC challenge! So far we have introduced ourselves to our team members and allocated brief "jobs" for each person in the group. We have come up with a basic design for the prototype and an idea of what the code should look like/do. So far in this lab session we have read through the rules and what equipment we are allowed to use to construct our robot and what the robot will be required to do.

Week 6 – 23rd April 2015

Connor Moody

This week was the beginning of the AVC. We began with a session that involved working as a team and we found that we worked well when we each specialised in a certain area. Before launching into the actual designing of the robot we sat down and discussed the hardware we were planning to use to complete the course. We chose to use wheels over tracks because one of the team members explained that tracks are difficult to turn on the spot with. We chose medium sized wheels to get a balance between speed and the fine accuracy of small wheels.

The overall shape of the body was going to be round. We believed that this would reduce the likelihood of getting the robot caught on any walls. For quadrant three we also wanted the digital sensors for detecting the walls to be in line with the axles of the wheels for better turning radius's and for more accurate detecting of intersections.

We allocated jobs within our team so we could specialise. I volunteered to do the programming for the robot because it is one of my strengths.

Week 7 – 30th April 2015

Matthew Lee

This week we started putting some of our plans into action. We started developing parts for our prototype in FreeCAD and should be ready to 3D print/laser cut them by the end of the week. We have started to construct the prototype as well. This was done using an old CD and spare plastic off-cuts from other groups' laser cuttings. In terms of software, we have started to create code that allows us to calibrate the sensors we are allowed to use. The digital distance sensor code has been written and we have nearly finished developing the code for the analogue sensor so that it returns an actual distance (in cm) instead of some arbitrary number. The aim for this week is to have a substantial chunk of the prototype's design complete and the majority of the sensor code done.

Week 7 – 30th April 2015

Connor Moody

This week we started building our prototype by using a CD as the base. Our designers begun to design and build the parts that would be used for our prototype, the first of which were motor mounts. The motor mounts allow us to fix the motors onto the base. We had also found a piece of acrylic that could be used as a temporary mount for the Arduino board. This was planned to sit above the base so the battery could fit underneath.

On the software side of the project we had begun to calibrate our distance sensors in particular the analogue distance sensor so we could extract a real length in centimetres out of the unitless number the sensor measured.

I spent extra time on the Friday of this week to getting the motors working with the code. I did this my simply making the robot loop through some code that would make the motors spin one way and then reverse the direction. About halfway through testing it appeared to stop behaving as expected and started spinning on the spot regardless of the direction I sent to the wheels.

Week 7 – 30th April 2015

Jonathan Carr

During this week we have our first priority has been planning our project; discussing as a team the sensors we will need to use in order to complete the challenge, how the code will need to react to the sensor values in order to make it through each quadrant, and how the physical robot itself will need to be designed in order to pass the course. We have also started prototyping using a CD as a base and some simple 3d printed parts, as well as writing code to interpret input from the sensors on the robot.

Week 7 – 30th April 2015

Michael Vega

This we made the first priority of our project to plan out what we should do for this project, we split everyone into two groups in which they felt they would contribute strongly to the team last week. We began the start of our prototype designs by getting our pieces from Arthur and seeing what we were able to create from these objects which consists of 3 CD's, 2 motors (which had to be fixed), the three types of sensors. We used a CD to start things off and cut some of the disc off and found a place for the motor to be. In terms of the code we had matt and Connor to work on the sensors. We also started to create our motor brackets on FreeCAD which took a long time to do since FreeCAD continuously erased our progress.

Week 8 – 7th May 2015

Matthew Lee

In terms of design and the build of our robot we have completed the basics of our prototype. We have the final shape sorted and printed and our body is ready to be sent off to be laser cut. A lot of the parts that needed to be printed have been printed and are ready to be attached to the body once it has been printed. For the software of our robot we have successfully developed code to read from the colour sensor and return whether or not it is on the line and returns an "error" distance. This number will determine how much power each motor needs in order to correct itself and to stay on the line for quadrant one. We have had quite a bit of trouble with our colour sensor, in particular the connections between the wires and the sensor and in some cases the pins. We have had to solder the sensor back together again multiple times taking up crucial time. We have also been having some trouble with FreeCAD crashing and losing our designs. The aim for this week is to have all the software for the sensors complete and the prototype built.

Week 8 – 7th May 2015

Connor Moody

At this point we have a working prototype with motors and Arduino mounted. We attached the QTR sensors and to the laser cut base. Currently, the prototype is our functioning robot and is limited but hopefully by next week we should be able to start constructing and programming the final robot.

In the software department we have been finalising the calibration of the sensors, in particular the QTR sensor. The sensor behaves exactly as expected and we can calculate the distance from the centre of the sensor to the centre of the line in the form of an 'error' value. We implemented PID controller that would take this error value and return a value that can be used to offset the speeds on the robot. This addition of the PID controller has not been tested yet.

Like last week the wheels are still acting strangely at times with no apparent pattern. We tried changing the polarity of the motors without any success.

Week 8 – 7th May 2015

Jonathan Carr

This week we continued designing and creating the parts required to create a functional prototype, so that we can begin creating code that can be tested on the actual course. We've made some progress in having our robot following the white line quadrant 1, but there is still much calibration and tuning to do before our robot performs reliably and efficiently in this area.

Week 8 – 7th May 2015

Michael Vega

This week we continued doing our design and programming for our prototype/project as that is really our main objective. We made the final design for our base for the prototype which we will probably use for our final design. We also had to 3d print a lot of our parts out which took a long time. We also began to work on our code for the 1st quadrant. But we ran into a lot of problems as we had faulty wires so we had to re-solder it and the code seem to work fine.

Week 9 – 14th May 2015

Matthew Lee

We have nearly completed the body of our prototype, we are currently waiting to print out the brackets for all of the sensors. In terms of software we have developed code that determines how much power to give each wheel based on the "error" number it receives from the colour sensor. This will allow us to complete quadrants one and two. It also has methods for correcting itself when it is not going straight meaning it should drive in a relatively straight line. Again, we have been having trouble with our colour sensor not performing well. We have had to solder some of the connections again costing us valuable time for building the robot. We have also had problems with the connections to our motors as the wires keep coming undone meaning they need to be soldered as well. For the networking aspect of this project, we have developed the code for the XBee sensor which is required to send back our location to a computer near the track. It will take an input from the robot and output it to another XBee receiver telling it what quadrant the robot is currently in.

Week 9 – 14th May 2015

Connor Moody

This week our final robot is starting to take form. The QTR sensors, Arduino and motors are all connected and are working well. We are getting new motor mounts printed soon that will allow digital sensors to be mounted above the wheels as we originally planned.

Multiple issues have arisen with the wires connected to the QTR sensor have been torn off resulting in extra time being put in that could've been used more productively elsewhere. Progress has slowed.

When the robot moves off of the line in quadrant 1, both wheels go to full power and doesn't move back onto the track as it should. We tried fine tuning the PID controller to prevent that robot from moving off of the line. Other than that the PID controller is working wonders.

Week 9 – 14th May 2015

Michael Vega

This week our prototype was almost complete. We just need to finish printing out the brackets that we designed so we can have the sensors on top of the brackets. We've had trouble getting our colour sensor not working properly as it did not transmit the numbers after having the white line under it. Our robot also had problems as one wheel would always rotate the other way. This happened after the robot fell. We stood there in anger as we tried to find out by its not working properly, the problem was just that the motor board was not connected properly as that caused the sensors and motors to act weirdly. In terms of the software we started to develop the methods in which the robot can tell that's in a certain quadrant. We also had written the base layout for the XBee.

Week 10 – 21st May 2015

Matthew Lee

This week we managed to successfully make it through quadrant one and the majority of quadrant two. We are currently working on a new body and designing the parts for it in FreeCAD. This should hopefully be ready and 3D printed by the end of the week. For the software, we are developing code to read from the sensors and allowing it to control the motors in order to make it through quadrants three and four and hopefully to the end. We are also developing methods that will allow the robot to know what quadrant the robot is in. So far we have determined the robot will know it is in quadrant one (as it is the first quadrant), it will know It's in quadrant two when it crosses the first white line (like an intersection), it will know it's in quadrant three when it detects the walls with its distance sensors and lastly it will know it's in quadrant four when it hits the door. We are also developing an LED detection system, this will allow us to know what quadrant the robot thinks it's in while XBee sensor is not active. Our aim for this week is to get through the bulk of quadrant three and develop code that allows the robot to know what quadrant it's in.

Week 10 – 21st May 2015

Connor Moody

We have managed to get the robot to complete quadrant 1 and quadrant 2 with only the occassional mishap. We discovered that the issue with the wheels spinning the wrong direction back in Week 7 was the result of the Vicmoto shield not making good connections with the pins of the Arduino. I believe that the pin responsibly for dictating the direction that the motors spin wasn't connecting with the Vicmoto shield. From now own we need to ensure that the shields are connected as best as possible before testing.

Spending extra time earlier in the week I managed to identify that when the robot move off of the line that the error calculated by the PID system was causing an overflow in one of the values sent to the motors. This meant that the wheel would be set at a very high velocity resulting in it moving of the line and never correcting its path.

We currently don't have any quadrant 2 code yet the robot seems to make it through quadrant 2. It even follows the shortest route without any unnecessary dead ends. I tested this numerous times an I still haven't been able to figure out exactly why it takes the best route.

The final robot has been built this week. We have also started to test the robot in quadrant 3 with the distance sensors. We currently only have the sensors blu-tacked on so we can play around with the positioning to get it just right before fixing them in place.

Week 10 – 21st May 2015

Jonathan Carr

Our robot is now built, created from 3d printed parts, and parts cut from acrylic, although there have been discussions about possible changed to our design, so changed may be made before our final build is complete. Also, some sensors and parts are currently attached with temporary means such as blu-tak, so we will likely use more permanent means to mount these sensors once we have determined the optimal position for each sensor. We have completed quadrants 1 and 2 (with a little bit of luck), but still have a lot of work to do to complete quadrants 3 and 4 of the track.

Week 10 – 21st May 2015

Michael Vega

This week we had finished our robot. Now we just had to test/edit/adjust our robot in the maze which we continuously had to make edits to the program. We also made changes to where the sensor would be so we got rid of the extra part that was part of the motor bracket, and we placed the sensor at the front just before the wheel. We also used blu-tak as a temporary placeholder for our little 'thumb' to lift the robot up and to hold the sensors that was placed in front of the wheels in place. We also edited our program for the XBee to send information to the main computer. We managed to complete quadrant 1 and 2.

Week 11 – 28th May 2015

Matthew Lee

This week is the final week of development. This week we have developed the majority of the code so that our robot successfully makes it through quadrant one, quadrant two (with actual quadrant two code not with quadrant one code like before), and quadrant three. We have set up our XBee sensor so that it periodically broadcasts what quadrant the robot is in to all other XBee receivers in the vicinity. Our prototype is almost complete and we are fine-tuning and testing our creation for the test day on Thursday.

Week 11 – 28th May 2015

Connor Moody

Over the last weekend I took the robot home to work on quadrant 2 code. I mocked up a course using black paper and white plumbing tape. I developed some code that would allow the robot to turn on the spot left, right, or to do a full 180. This code can be used in quadrant 2 and in quadrant 3 when accurate turns need to be made.

I also developed that algorithm that would be used to navigate quadrant 2. The idea was to take any right turns if it was possible. It will then prioritise going straight second followed by the robot turning left. If the line is lost then the robot does a 180 degree turn.

When testing the code I had developed in the weekend on the course I found that there were false positives of a dead end in the second quadrant. This would make the robot move away from the line and sometimes picking up on a line from a completely different part of quadrant 2. I realised that the dead end case made the robot turn left which would generally take it away from the desired path. To fix this I made the robot turn clockwise on the spot. This solved the problem and now the robot takes quadrant 2 with no problems.

We merged the Xbee code written by a teammate with the code on the robot and we tested this with the computer setup next to the course. We decided that we would transmit the changes in quadrants rather than transmitting the current quadrant every loop of the Arduino. It seemed redundant to transmit the current quadrant every loop when it is possible to infer that.

There were countless hours spent on the Wednesday evening and Thursday morning, the day of our assessment. During this crunch time we managed to code quadrant 3 to work about 80% of the time the robot goes through. Unfortunately, we ran out of time and were unable to implement code for quadrant 4.

During the 15 minute assessment time we managed to get up to 80% of the course done within the first three runs. We had luckily been able to make quadrant 2 and 3 run quite fast. This meant we each run was quick and we could effectively run it more times in the 15 minute time limit. We ran it again and again. We managed to get to the first door just in time for it to open before the robot detected a dead end and turn around. The robot made it to the second door, a completion of %90 of the course. In the 13th minute we gave our robot its last chance to get further. By sheer luck we managed to time the launch so the robot arrived at the first door and second door just in time. We had been lucky in getting the robot to the end. With an aim of 80% for the assessment run I couldn't believe that we had gotten our robot to the end in the eleventh hour.

Week 11 – 28th May 2015

Jonathan Carr

Our robot is now complete, although slightly differing from original designs and ideas as we have made improvements to some of our initial thoughts over the past few weeks (such as the position of sensors and motors). Our robot is currently making significant progress through quadrant 3, and sometimes quadrant 4, and we are now making fine adjustments to the code before the final test in order to be more reliable within the final two quadrants.

Week 11 – 28th May 2015

Michael Vega

This was the final week for making our adjustments for our robot. We had the design very similar to what we had to the start but the main change was the sensor being moved to the front before the wheels. we had our robot go through quadrant 3 with it doing well, but we don't have a code for quadrant 4 so we had to let the robot run off quadrant 3 code, this turned out to be great but bad at the same time as we can get to the door but then have it turn away but we can get through the maze with the right timing of the doors. We have to fine tune it before the actual test on Thursday.

During the test – 28th May 2015

Michael Vega

During the test we had very small problems with the robot, one being that it got stuck against the wall and a corner. Second being that it reached the door and turned away thinking that it is a dead end. We took several attempts with a lot of time to spare. while our robot was going through the maze I was timing the door and how fast the door would close in a certain time, I counted that the 1st door take 10 seconds to close and 10 to open and the second door took 5 seconds to close and 5 seconds to open. I also counted how long the robot takes to reach the door, I timed it and got above 1 minute, the robot would reach the door at about 1 minute 10 on the clock so we had to release the robot when the door had a just open and it successfully did it.