MANCHESTER 1824
The University of Manchester

School of Electrical and Electronic Engineering

Embedded Systems Project

DESIGN REPORT #4

Title: Final Report

Group Number: 37

Group members name:	ID Number	I confirm that this is the group's own work.
Federica Donald	10095518	☒
Alkinoos Sarioglou	10136315	☒
Elvis Taimal Gongora	10146823	☒
Kai Guo	10201372	☒
Ruoning Mi	10431554	☒

Tutor: Dr. Emad Alsusa
Date: 08/05/2019

Contents

1. Executive Summary – Introduction

The Embedded Systems Project is a useful learning experience for the members of the team, as they are faced with the challenge to design and construct an autonomous robot, while being an active part of a team.
All of the above is achieved by keeping in mind the main aim of the project: the buggy needs to follow a white line around a track autonomously, while overcoming obstacles. During the full duration of the project, a very well organised procedure is followed by the entire team to ensure objectives are achieved on time and the development has the anticipated progress. In the first semester, the required knowledge is acquired by the group members on how the buggy navigates around the white line. Also, time is spent on planning and designing what components will be used meanwhile a CAD model is created for the buggy chassis. The second semester includes assembly of all the components and programming the microcontroller for the buggy to run efficiently around the track. This report concludes the Embedded Systems Project and summarises the work undertaken on planning, designing, building and operating the buggy the group created. There are 4 main parts: the final system's components summary, the team organization, the budget and the analysis of the heats.
Firstly, the components used are the most crucial aspect of the system as they ensure its successful operation. These include mechanical, electronic, control and software components. The main mechanical components used are: DC Motors, Gearboxes, Encoders, Wheels and Tyres and the Chassis. In terms of electronic components, the reflective sensors on the stripboard, the motor drive board, the microcontroller board and the Bluetooth module are most significant. Furthermore, the sensor, speed PID controllers and the Bluetooth guarantee a successful control of the movement. Finally, as far as the software is concerned, libraries, classes and objects with different functions of the algorithm are implemented to complete the correct functionality. Overall, the buggy successfully integrated the sensor circuit, the motors on the chassis and demonstrated great communication between components.
The following CAD design (Figure 1) and the actual picture of the buggy (Figure 2) show the configuration of components on the chassis. The chassis is shaped as a semicircle at the front and rectangle at the back which makes it good-looking and getting cut on a sharp edge is avoided. It can also be seen that the motor drive and the microcontroller board are on top while the battery pack and the stripboard with the sensors underneath.

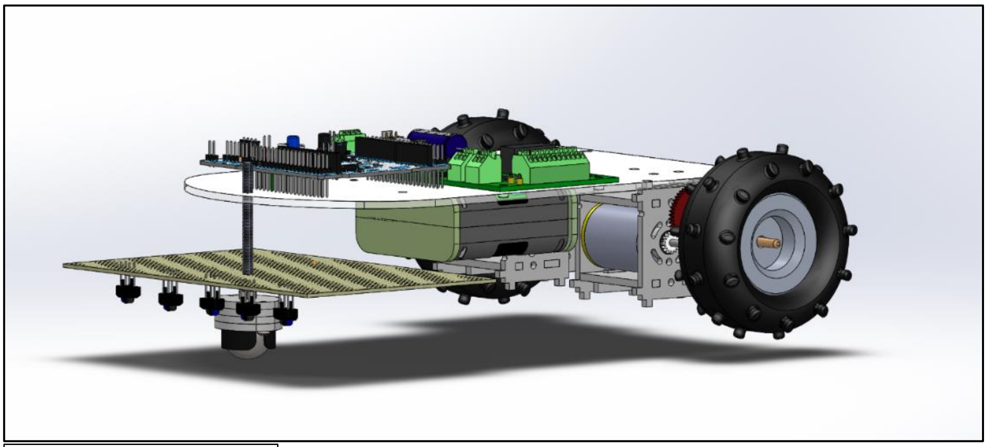

Figure 1: CAD Design

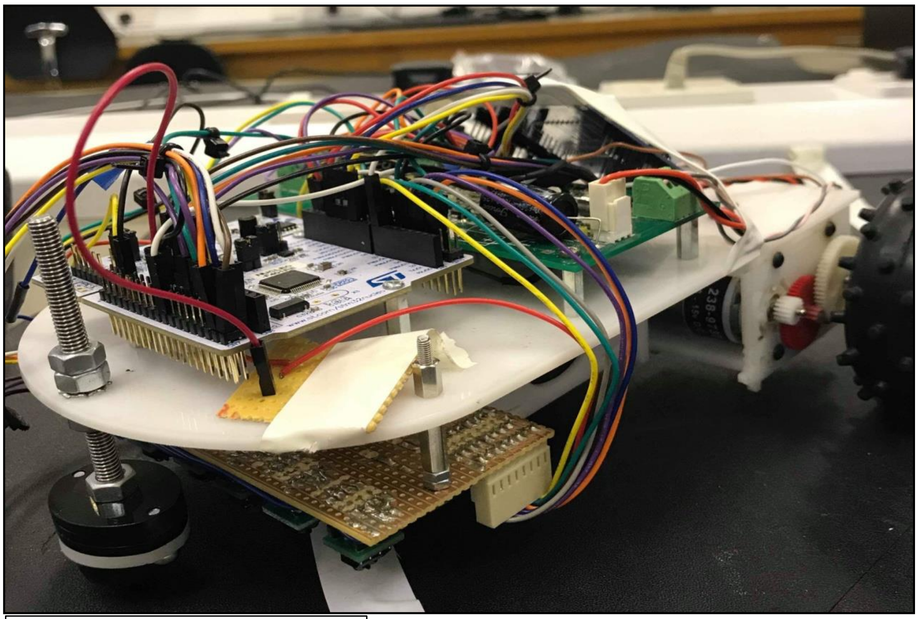

Figure 2: Picture of the Buggy

In addition to that, the code carried out the objective of the project successfully by implementing the control algorithm for the correct adjustment of power for the motors depending on the readings from the sensors. However, some changes were required like the connections on the microcontroller board to be able to accommodate the LCD screen.

For all the above to materialise, good organisation of the team and a suitable plan was required. The plan included a time schedule for all the main objectives to be achieved on time. These include report writing, CAD and PCB Design, Interface of Motors and Sensors, Assembly of the System, Successful Movement and Steering and the Track Length Estimation Algorithm. Some changes did occur during the implementation of the plan, e.g. no PCB fabrication and no tilt sensors implementation, which were related to timing issues. Furthermore, the 5 members of the group were divided into 2 teams to achieve efficient division of workload. One team worked on the hardware and the other on the software part of the buggy. A Gantt Chart was also created to keep track of the project's progress and ensure all deliverables are achieved on time. A critical analysis of the team's performance reveals that this was accomplished due to the strong organisation skills of the team, which resulted in high scoring on the technical demonstrations and the reports with few points missing. For that reason, changes in a future project would include more productive breaks and more productive use of the budget. Also, difficulties that arose did not affect the normal operation of the team.

When it comes to budget, apart from the freely-provided components, the spend was 18.92£ in total. That results in an overall cost of £252.51 for the whole system. However, towards the actual value of the buggy the hard work of five people for 1 year should be counted. For this reason, the price that the buggy would be sold for is much higher. The high price is supported by the good performance in the Heats and the Final Race.

Before the heats, the team performed many tests to check the full operation of the buggy and completed minute corrections. These involved improvements of the turnaround function and increase of the speed. The evidence of testing proves the development of these parts. The first attempt in the Heats was unsuccessful, because of a problem in the turnaround function. However, the buggy completed the track in 20 seconds in the second attempt and qualified in the Final Race. In the Final Race, the successful features of the buggy like the fast and smooth movement boosted its performance and achieved the Best Track Length Estimation Award, while the problematic features (big size, messy wiring) did not disrupt the correct operation of the system. The components summary gives a better insight into these features.

2. Final System Components Summary

2.1. Mechanical components

The mechanical components of the system can be summarised as:

- Motors
- Encoders
- Gearbox
- Wheels/Tyres
- Chassis
- Screws, nuts, spacers
- Ball castor
- Battery case

It is obvious that the most crucial part is the motor module, which is composed of a DC motor, a quadrature encoder and a gearbox. This was chosen during the development of the buggy among three options. Gearbox 3 was selected as this produced the torque needed for the buggy to climb on a slope of 18°. The motor module is attached to a rubber tyre through a gearbox. The chassis was developed by measuring the components placed on it and calculating their weight distribution so that no significant bending occurs. During the assembly procedure the electrical components, the ball castor #955, the motors with the rims, screws, nuts and spacers were placed on the chassis. Finally, a battery case was added since there was no original way of fixing the batteries safely on the design.

2.2. Electronic components

The electronic components of the system can be summarised as:

- Motor drive board
- Switch
- Sensors
- Resistors
- Darlington Array
- Stripboard
- Bluetooth module
- Microcontroller board
- LCD screen

The motor drive board was powered by the rechargeable batteries which together generated a maximum of 10 V for the buggy to work properly. This board controls the current and voltage output to the motors depending on the speed set by the microcontroller board. On the NUCLEO microcontroller board the code was transferred for processing in order for it to send the suitable electrical signals to the electrical components of the buggy for efficient operation of the system. Furthermore, a switch was used to control the ON and OFF condition of the system. Additionally, the contribution of the electronic components used on the stripboard was integral to the success of the sensor implementation. More specifically, the 6 resistors of 68 Ω, the 6 resistors of 10 kΩ, the ULN2003A Darlington Buffer that was used to draw less current from the microcontroller board to turn on the LEDs and most importantly the 6 TCRT5000 line sensors. This type of sensor was chosen among several options since it was the most accurate and sensitive when detecting the white part and contains a daylight blocking filter. More complex components include the Bluetooth module which was used to receive a signal and force the turnaround of the robot. Finally, an LCD screen was added to display measurements that the buggy computed, like the track length estimation.

2.3. Control components

The control components of the system can be summarised as:

- Sensor and speed PID controllers
- Proportional, Derivative, Integral Action
- K_p , K_d , K_I , Error
- Bluetooth

The control algorithm of the system guarantees a correct adjustment of the motor speeds depending on the readings from the sensors, so that the buggy navigates correctly above the line. For the algorithm, two PID controllers were used. First of all, a sensor PID controller was used to monitor the error of the position of the buggy when compared to the centre of the white line. The proportional, derivative and integral action depending on their constants provided corrections to the position of the buggy by converting this error into a speed action for the motors. Therefore, a smooth movement above the line was ensured. Furthermore, a speed PID controller read measurements from the encoders in order to fix the speed of the motors with the same value on the straight section and adjust the speed when the buggy deals with an uphill or a downhill. The Bluetooth module applies a different kind of control by controlling the turnaround. The signal was sent from a mobile device and determined when the turn will occur. After that, the normal line-following algorithm resumed its operation.

2.4. Software components

The software components of the system can be summarised as:

- Libraries
- Classes
- Member Functions
- Data Members
- Objects
- Main Code

The software part of the system is of great importance as it controls the operation of all the components and determines whether the communication is effective between them. It uses different parts to achieve this. First of all, the code uses functions or parts of code from pre-written libraries. These provided useful functionality for different components like the encoders and the LCD screen. Then, classes were constructed which helped with the modular design of the code making it easier to debug. Each class represents a different part of the system, e.g. motor, Bluetooth etc. It includes all the important information for this particular part (data members) and many member functions that it can be used with. One example is the class of Motor which contains data members like bipolar, PWM etc. and member functions like constant speed, forward rotation, backward rotation etc. Then, in the main code, objects of the classes were created to represent the objects of the actual system, e.g. MotorLeft and MotorRight etc. All the functions performed by the objects were connected in the main-code and interacted with each other giving rise to the operation of the whole system as an entity.

2.5. Parts that worked well and Changes

The parts of the system that were fully-functioning can be summarised as:

- Stripboard with reflective sensors
- Motors
- Communication of components with microcontroller board
- Chassis design
- Software Design

First of all, the stripboard including the sensor circuit performed exceptionally in the sensor demonstration and therefore it was mounted on the chassis without any further adjustments. Also, the performance of the motors was satisfactory as they responded to the full range of speeds and reacted quickly to the commands of the microcontroller. This suggests that the communication of the components and the software design was accomplished, as the sensors would send the readings to the microcontroller, which would process the outputs and adjust the speed of the two motors to control the direction of the buggy. Finally, the chassis design was successful as it managed to accommodate all the components required and without causing significant problems to the movement.

However, there were things that required change during the project.
These can be summarised as:

- No tilt sensor used
- Connections on the microcontroller
- Addition of LCD screen to display measurement of distance

To begin with, tilt sensors were not used on the system in the end due to timing issues. These were planned to detect incline or decline and help the speed control algorithm adjust the speed of the motors. Changes were also made in the connections of the microcontroller board since some pins which were initially used were needed for the LCD screen. The screen proved to be crucial for the system as the results of track length measurements would be displayed on it. As a result, a change in the connections had to be completed and wiring diagrams had to be redrawn.

2.6. Summary

In conclusion, the organisation of the components reviewed in this section was done in steps during the progress of the project development in order to observe changes and the interaction between the parts of the system. The improvements and next steps would then be decided. This would not be possible without a very strong team organisation and planning that took place within the team.

3. Team Organisation and Planning

3.1. Summary of Project Objectives

The main aim of the project was to create an autonomous line-following buggy that will follow a white line placed in the middle of a track. The objectives to complete along the way include:

- Reports including the Motor and Sensor Characterisation, the Proposal and the Final Report
- CAD Design and Chassis Fabrication
- PCB Design and Fabrication
- Interface of Motors
- Interface of Sensors including the tilt sensors
- Buggy Assembly
- Successful movement / Steering
- Code Development including the track length estimation code

The milestones of the project were the following:

- Report deadlines throughout the entire year
- Technical Demonstrations including motor control, sensor interface, steering demonstration and the heats
- PCB Fabrication Submission
- Final Report
- Race Day

Objectives changed mainly throughout the second semester as the project became more practical driven. The first change of objective was not creating the PCB for the sensors. This was decided not to be completed as the group considered it an unnecessary use of resources as the stripboard had been completed to a high standard and it worked. Additionally, the time constraint the group had placed on the integration of both hardware and software to ensure the buggy was ready for each technical demonstration requirement became an issue for completing the PCB. Secondly, the tilt sensors were not implemented into the buggy system. These were interfaced and tested however the group decided that the possibility of improving speed control would be sacrificed for the added complexity and time cost this would place.

3.2. Organisation of the Team

Organisation included Alkinoos, Elvis and Kai on the hardware team whilst Federica and Ruoning were on the software team. Towards the end of the project, once hardware was completed, Alkinoos joined the software team to help with the last technical demonstrations and final adjustments and refinements.
The main changes to the Gantt Chart included the timescale of the objectives set. Apart from the main compulsory deliverables, most of the objectives took up more time than anticipated. However, this was not a huge issue as the team had split into two groups for this reason. Everyone could be participating and working on other parts of the project in order for the group to act in a parallel fashion and keep the flow of work going. This improved the organisation, but it led into a few communication issues between departments as comes with a divisional type structure of working.

3.3. Critical Analysis

From a critical analysis point of view, it should be pointed out that all deliverables were achieved on time. This is mainly due to the fact that objectives were set for deliverables before the deadlines to ensure that they were completed early. The group completed the technical demonstrations to the best of its ability given the time available, with only a few points missing or not being completed fully. However, the majority were completed successfully resulting in high scoring. The reports were also completed on time due to high organisation skills with the group having a strict policy to deliver reports: Compromising of bullet points to get the members thinking about the content of the sections usually required to be complete 2 weeks before deadline; then, an initial flowing and fully written draft usually required to be complete a week before the deadline and a final draft to be complete 2 or 3 days before the report deadline allowing for any errors to be negated. This procedure resulted in improved performance in the reports from one time to another and high scoring in the last reports.

3.4. Difficulties

The only difficulties present within the group have been illnesses, interview and exam absences. These problems were resolved and overcome by the rest of the group continuing as planned as the person absent was only missing for 1 or 2 days.

3.5. Summary

The group worked very well as a team, with clear leaders set out from the beginning, a grounded communication network and clarified deadlines so no tensions arose. If team work and productivity were to be improved, it would be suggested to include incentives such as breaks and/or praise. The Draugiem Group found that the "top ten percent most productive employees didn't actually work any more hours than anybody else, they took more breaks" [1] stated on Forbes about how stress reduces productivity. Therefore, if as a group there were clear breaks used for creative purposes (allowed through the timetabling of the semester) the group could perhaps have been more productive in the long run throughout the project. Another improvement would be the more efficient use of the available budget with more additional components purchased for added features in the buggy. It is understandable that the budget and the overall cost of the robot have been important aspects of the project.

4. Budget vs. Outturn

4.1. Spend Costs

In the building stage of the project some free parts were provided. However, additional components were required for the system to be complete. For that reason, 6 Tilt sensors, 5 mini PCBs and 3 TCRT sensors were ordered. The cost is 18.92£ in total. The prices and quantities of the ordered components are listed in Table 1. However, this price is above what was actually needed. This was because the white box of the team containing all the important components was taken away mistakenly for a month and the group did not have the required components to complete the system. Therefore, some parts needed to be reordered resulting in more budget usage. Also, due to time limits and technical reasons, tilt sensors were not used in the end. Nevertheless, the cost was well within the £40 limit for the project.

Item	Quantity	Price Per Item (£)	Cost (£)
Tilt Sensors	6	2	12
Mini PCBs	5	1	5
TCRT Sensors	3	0.64	1.92
		Total:	**£18.92**

Table 1. Price for Ordered Parts

4.2. Buggy Price Proposition

The total cost for the buggy, including the freely-provided parts and the ordered parts, is shown in Table 2.

Item	Quantity	Cost (£)
Resistors	12	1.2
TCRT5000	6	3.84
HM-10 Bluetooth Module	1	16.36
Chassis Material	1	42
Front Wheel	1	2.5
Rubber Tyre	2	2.9
Motors	2	14.22
Gearbox	2	14
NUCLEO-F401RE	1	15
Application Shield	1	33.09
Motor Drive Board	1	30
Sensor Mini PCBs	12	12
Battery Holder	1	3.5
Batteries	8	16
Insulation Tape	1	2
Cable Ties	3	0.6
Stripboard	1	3.5
Tilt Sensors	6	12
Chassis Fabrication	1	27.8
		Total: £252.51

Table 2. Total Cost of the Buggy

As Table 2 shows, the total cost for the buggy is £252.51. If this was connected to a real business, more factors than the total cost of the components would be considered. These include the time and effort spent on designing, programming and constructing the system. According to the National Minimum Wage Standard, minimum wage for people aged from 21 to 25 is £7.38 per hour [2]. Assuming total time spent on the project to be 200 hours, the minimum wage for each person is £1476. The total wage for the 5-people-team is £7380. To make a profit, the selling price is increased to £8000, so the profit of each buggy is £367.49. The high selling price is also justified by the buggy's accuracy on estimating the length of a track since it received the prize for Best Track Length Estimation on the Race Day and its result was within ± 0.1 cm error. The buggy can be sold for a higher price if more functions are added to it based on customers' needs and preferences.

5. Analysis of Heats

5.1. Preparation for the Heats

The full construction of the system was implemented when all the extra components were ordered and received. The technical demonstrations showed that the functionality of all the components was competent. Following the last technical demonstration, which included following a small track of the line, either straight or curved, the team had to perform minute corrections to the system in the hardware or the software part to ensure an efficient and quick performance around the track for the heats.

In the steering demonstration, the buggy moved successfully in the straight and curved parts of the line and it was able to use the encoders to implement the speed control algorithm. The controlled stop worked perfectly too. However, there was a problem with the turnaround, as the buggy would receive the Bluetooth signal but after reversing direction it would miss the line. Therefore, it was obvious that improvements had to be made on this part when preparing for the heats.

The weeks before the heats all the work done was based on improving the code regarding the turnaround. It would be crucial to have a reliable algorithm running after the $180°$ turn. The software team worked in trying different versions of the turnaround function and results were improved significantly using a section of the track provided in the labs. Evidence of results of improvement can be seen in Table 3 in Section 5.2. At the same time, the efforts of the team were focused on increasing the speed of the buggy. The PWM of the motors in the straight section was increased from 70% to 85% of the period in the bipolar mode. Also, the speed in the turns was boosted from 65% PWM to 80%. These changes caused some problems especially in the sharp turns where the buggy would miss the line many times. The solution was to re-tune the sensor PID controller by increasing the values of K_p and K_d. A crucial part to mind was to make sure that it can complete the sharp curves. These turns contributed significantly when deciding the values of the constants. The change of the parameters led to important observations on the movement of the buggy during testing which are mentioned in Table 4 (Section 5.2). The ultimate goal was to make the buggy run with the mentioned values of PWM smoothly above the line.

A few days before the heats, a problem came up with the battery charger. The batteries could not be charged, resulting in the team swapping them every time they were running low. However, this problem did not affect the time schedule of the team. On the last day before the heats, a test of the whole system was performed. The tests on the provided tracks proved that the buggy could follow the line and therefore all the components were fully-functioning.

5.2. Testing and Evidence of Results

- **Turnaround Function Improvement**

The improvement of the turnaround involved developing and testing the function which would be called at the turnaround point by Bluetooth. Table 3 includes different versions of the function which were tested (from oldest to newest) but also observations on the performance.

Turnaround Function	Observations and Improvements

`void turnaround (void) {`
`Motor2.write(0.5f);`
`Motor1.write(0.5f);`
`wait(1);`
`Motor2.write(0.8f);`
`Motor2.write(0.4f);}`
	The buggy stops and turns smoothly. Need to add the Bluetooth

<table>
<tr><td></td><td>functionality and to detect the line after the reverse turn.</td></tr>
<tr><td><pre>else if (hm10.readable()) {
                d = hm10.getc();
                if(d == 'A'){
                    MotorOne.write(0.8f);
                    MotorTwo.write(0.4f);
                    wait(3); } }</pre></td><td>The buggy now performs a turnaround when the Bluetooth signal is sent. It turns around but the wait function is not useful to detect the line afterwards.</td></tr>
<tr><td><pre>void turnaround () {
        MotorOne.write(0.5f);
        MotorTwo.write(0.5f);
        wait(1);
        while (angle<180) {
            MotorOne.write(0.7f);
            MotorTwo.write(0.3f);
            leftwheel.showspeed1();
            rightwheel.showspeed2();
            angle1=leftwheel.angle1();
            angle2=rightwheel.angle2();
            angle=angle1-angle2;
            pc.printf("%.3f",angle);
        }
        angle1=0;
        angle2=0;
        angle=0;
        setspeed();}</pre></td><td>The buggy is not able to detect when the angle becomes 180° due to problem with the encoders. The microcontroller cannot perform the expected functionality correctly with this function. Another way of detecting the line needs to be developed.</td></tr>
<tr><td><pre>void turnaround () {
        MotorOne.write(0.5f);
        MotorTwo.write(0.5f);
        wait(1);
        MotorOne.write(0.7f);
        MotorTwo.write(0.3f);
        wait(2);
        while ((out1.read()<0.36) && (out2.read()<0.36)
&& (out3.read()<0.36) && (out4.read()<0.36) &&
(out5.read()<0.36) && (out6.read()<0.36)) {
            MotorOne.write(0.7f);
            MotorTwo.write(0.3f);
        }
            setspeed();}</pre></td><td>The buggy performs the required functionality and the code seems to be suitable. However, the turning is fast and when trying to detect the line again sometimes it is not successful and turning continues. Some factors like the sensor values and the “wait” functions need to be improved.</td></tr>
<tr><td><pre>void turnaround () {
        MotorOne.write(0.5f);
        MotorTwo.write(0.5f);
        wait(1);
        MotorOne.write(0.7f);
        MotorTwo.write(0.3f);
        wait(1);
        while ((out3.read()<0.8)) {
            MotorOne.write(0.6f);
            MotorTwo.write(0.3f);
        }
            MotorOne.write(0.5f);
            MotorTwo.write(0.5f);
            wait(0.7);
            setspeed();}</pre></td><td>The condition for re-finding the line is improved as the central sensor is exactly on the line when the buggy stops after the turnaround. However, when detecting the line instead of following the line, the buggy turns abruptly, misses the line and stops. The microcontroller finds difficulty in transitioning to the original function. The transition needs to be improved.</td></tr>
<tr><td><pre>void turnaround () {
        MotorOne.write(0.5f);
        MotorTwo.write(0.5f);
        wait(1);
        MotorOne.write(0.7f);
        MotorTwo.write(0.2f);
        wait(0.5);
        while ((out3.read()<0.45)) {
            MotorOne.write(0.7f);
            MotorTwo.write(0.15f);
        }
            MotorOne.write(0.5f);
            MotorTwo.write(0.5f);
            wait(1);
 while (!((out1.read()<0.36) && (out2.read()<0.36) &&
(out3.read()<0.36) && (out4.read()<0.36) &&
(out5.read()<0.36) && (out6.read()<0.36))) {

values=(3)*out1.read()+(2)*out2.read()+1*out3.read()+(-
1)*out4.read()+(-2)*out5.read()+(-3)*out6.read();
            error=ref-values;
            integral = integral + error;
            derivative = error - lasterror;
            lasterror=error;
if ((error>0.75)|| (error<-0.75)) {</pre></td><td>This solution was found after the first heat and successfully completed the turnaround at the second heat. The problem is solved by copying the setspeed() function itself after the condition for re-finding the line instead of forcing the microprocessor to transition to the function. The Bluetooth signal is received, the turnaround is performed accurately, the line is detected correctly and the buggy continues following the line. The turnaround function is now successful.</td></tr>
</table>

`controlaction=(kp*error)+(ki*integral)+(kd*derivative);`
`MotorOne.write(0.75f - controlaction);`
`MotorTwo.write(0.75f + controlaction);`
`}`
`else {`
`constantspeed(); }}}`	

Table 3. Different Turnaround Functions and Results

- **Improving the Speed – Adjustment of Constants**

A lot of testing time was spent on increasing the speed of the buggy and ensuring a smooth and quick navigation. The team tested different values of constants for the sensor PID controller and the observations were recorded to help with the correct tuning.

Values of Constants in Sensor PID Controller	Observations and Improvements
KP = 2 KD = 0 KI = 0	The movement is not smooth, the buggy oscillates above the line as the correction is very big. K_p needs to be decreased.
KP = 1 KD = 0 KI = 0	The movement has significantly improved but still it is not perfectly smooth on the straight part. K_p needs to be decreased.
KP = 0.12 KD = 0 KI = 0	The buggy moves very smoothly on the straight part of the track. Next improvements relate to the turns so K_d needs to be tuned.
KP = 0.12 KD = 0.2 KI = 0	The buggy deals with the turns better than before, but the correction from K_d is abrupt and not accurate. K_d needs to be decreased.
KP = 0.12 KD = 0.15 KI = 0	The buggy successfully completes all kinds of turns, even sharp ones but it moves slightly off the line. K_I needs to be tuned.
KP = 0.12 KD = 0.15 KI = 0.5	The offset from the line has not improved. Other values of K_I need to be tested.
KP = 0.12 KD = 0.15 KI = 0.2	The movement is not better as well, and it decreases the accuracy of the navigation. As the integral part does not help significantly, it is better if it is not used at all.
KP = 0.12 KD = 0.15 KI = 0	The buggy navigates efficiently on the straight part and deals with the turns. The constants are the appropriate for the Heats.

Table 4. Different Constants and Results

All the above improvements and tests were focused on achieving a successful performance in the Heats and qualification for the Final Race.

5.3. Performance in the Heats

In the first attempt of the Heats, the buggy did not complete the track. The start was good and quick with the buggy making the first left turn. The second right turn leading to the slope was successfully completed as well. With the buggy approaching the turnaround point, the Bluetooth signal was sent. The signal resulted in the buggy turning by $180°$ quite accurately. Following this, the robot tried to continue following the line but instead of that it turned abruptly to the right, which resulted in hitting the side wall, missing the line and failing to complete the rest of the track. This result suggested that, before the second attempt, the team had to brainstorm solutions for the problem.

The analysis of the problem suggested that the microprocessor could not transition efficiently from the turnaround function back to the original function. The solution was found just before the second attempt. Now, instead of transitioning to the main function after the turnaround, the code of the main function was copied after re-finding the line in the turnaround function. That way the code would stay in the turnaround function without having to transition between two functions which caused obvious problems.

The second attempt in the Heats was successful. The buggy overcame the first part in the same way as before, completed the turnaround and continued following the line until the end without facing a problem with the other turns. It also completed the run with a controlled stop at the end of the white line. The time was 20 seconds, which resulted in the qualification to the Final Race.

5.4. Successful and Problematic Buggy Features

The buggy made it successfully to the Final Race, which suggests a good design and efficient communication between components, e.g. sensor measurements to microcontroller and microcontroller to motor control.

The most successful features in the autonomous robot included:

- Fast and smooth movement around the track without oscillations
- Estimating the length of a track very accurately which resulted in the Track Length Estimation Award

The problematic features of the buggy included:

- The size, which was quite big in the width and resulted in the buggy getting blocked on the side walls when reversing the direction with the turnaround. This influenced the weight of the buggy as well, which limited the maximum speed that it could be operated at.
- The wiring, although efficient, was messy and many times cables were knocked loose during the building stage. This resulted in having to solder new wires instead and maintained a high risk of failure in the Heats and the Final Race.

Overall, the development of the whole system was successful as the good features boosted its performance and the problematic ones did not disrupt the successful navigation around the track and the turnaround. The efforts of the team during the year were fruitful.

6. Summary

Overall, the project is considered a success for the team and a meaningful learning experience. The initial plan was implemented which shows great teamwork and communication and all the objectives were accomplished on time.

7. References

[1] Available: https://www.forbes.com/sites/kevinkruse/2017/02/06/want-to-get-more-done-try-taking-more-breaks/#35f62f216db4 , [Accessed: 02- May- 2019]

[2] Available: https://www.gov.uk/national-minimum-wage-rates , [Accessed: 05- May- 2019]