## Contents

# 1. Executive Summary and Introduction

## 1.1. Executive Summary

This comprehensive report presents a detailed account of the development, challenges, and outcomes associated with the creation of an autonomous line-following buggy. The project is an integration of mechanical, electronic, and control systems to engineer a compact, efficient, and highly adaptive buggy capable of precisely following a white line.

The buggy boasts an innovative dual-tier chassis design which separates critical components; the upper tier houses the microcontroller and battery pack, while the lower tier accommodates the motors and motor drive board. This layout not only facilitates ease of access for debugging and maintenance but also enhances the overall stability of the buggy by evenly distributing weight. The chassis also incorporates a sensor board mounted underneath the lower tier to detect the track line accurately. Emphasis on compact design has significantly enhanced the buggy's agility and allow it to execute a 180° turn without risk of collision with the track. Furthermore, meticulous attention to cable management and component security, using Velcro and screws, ensures durability and reliability during operation.

At the heart of the buggy's line detection capability is the TCRT5000 sensor package, consisting of an infrared LED and a phototransistor to detect variations in light reflection off the track surface. This is incorporated onto a custom PCB along with a ULN2003an Darlington Array, which efficiently manages the switching of high-current LEDs. The layout facilitates rapid and accurate sensor readings, enabling the buggy to navigate with exceptional precision.

The control system employs a finely tuned PID (Proportional-Integral-Derivative) algorithm, chosen for its effective self-correcting behaviour, which ensures the buggy remains on track even under varying environmental conditions. The PID settings were optimised through rigorous testing, resulting in a system that adjusts steering and speed dynamically, greatly improving the buggy's handling on the track. These control parameters were extensively tested to find the balance that prevents overshoot and instability during manoeuvring.

The software architecture deviated from the initial plans to enhance operational efficiency and responsiveness. A critical development was the implementation of a ticker system, which ensures that sensor readings, motor control, and steering adjustments are updated simultaneously and at a high frequency. This integration has markedly improved the buggy's response times in real-time testing scenarios. Additionally, the incorporation of Bluetooth connectivity allows for telemetry data to be sent directly to a BLE terminal, facilitating remote control, with start, stop and turn around functionality, as well as PID gain adjustments and debugging.

The project was challenged by several unforeseen technical issues, including motor and sensor malfunctions, which necessitated multiple rounds of troubleshooting and component replacements. These challenges were met with a proactive and collaborative approach from the team, ensuring that project milestones were ultimately met despite initial setbacks. The team's ability to adapt to these challenges was facilitated by effective communication and robust planning, as evidenced by the detailed Gantt chart which helped in tracking progress and adjusting timelines. This allowed the buggy to successfully meet all its requirements within the given time frame.

The base cost of the buggy was maintained as projected; however, additional unforeseen expenses due to component failures increased the developmental costs. Despite these challenges, the financial analysis confirmed the project’s economic viability in a mass production scenario, projecting a robust profit margin of 15.69% and demonstrating the potential for scale, with a retail price of £379 and a cost of £327.59 per unit, assuming 1000 units produced.

The buggy demonstrated exceptional performance during the initial testing phases, particularly in the heats where it achieved a time of 11.28 seconds, 5$^{th}$ fastest in the heats. The success can be attributed to the judicious considerations of weight distributions, compact design, and the efficiency of the control algorithms. The iterative tuning and testing process was critical, with each adjustment meticulously documented and analysed to ensure optimal performance under a variety of conditions.

This project has successfully demonstrated the process of designing and building a high-performance autonomous line-following buggy through team collaboration and innovative engineering. The methodologies developed have set a solid foundation for future enhancements and potential commercial production. Continued development and testing are expected to refine the buggy's capabilities further, solidifying its performance in more competitive and complex environments.

### 1.2. Introduction

The aim of the embedded systems project (ESP) is to create a buggy that can follow a line around a track without human intervention. The buggy should stop if it detects a significant line break, should be a capable of climbing an incline of 20° and should execute a 180° turn upon receiving a Bluetooth command. The buggy should also be optimised for speed as Technical Demonstration C was timed. This report will explain in detail how this buggy was designed, why those decisions were made and how it fared in practice.

The final components summary will explain the final technical details of the buggy, and how the chosen option is advantageous to the others. The team management section will detail which systems are in place to ensure a balanced workload and that disputes are resolved, in addition to comparing the Gantt chart with the real-world timeline of the project. In the budget section, the actual cost of the budget will be compared with the forecast in the proposal document. Any alterations will be discussed, and discrepancies explained. Finally, as the heats are the final deliverable of the buggy, these will be analysed, and the performance of the buggy will be evaluated.

## 2. Final System Components Summary

### 2.1. Mechanical

The buggy's construction incorporates a dual-tier chassis design. The upper tier houses the microcontroller and battery pack, while the lower tier accommodates the motors and motor drive board. This configuration facilitates the debugging process due to the accessibility of the top-tier components, namely the microcontroller and battery pack. A sensor board is mounted beneath the bottom chassis which detects the white line on the track.

Compact design was prioritised to enhance navigation and prevent collisions on the track. The wiring was meticulously crimped and secured at optimal lengths, with excess wires neatly bundled using cable ties to avoid entanglement in the wheels, as illustrated in Figure 1. The battery pack was secured to the chassis using Velcro, and both the microcontroller and motor drive board were firmly attached with screws. Additional Velcro was also attached on the sides of the battery pack to avoid any sideways movements it could have made. A firmly attached battery pack ensured that the buggy's control did not become unstable due to a shifting weight.

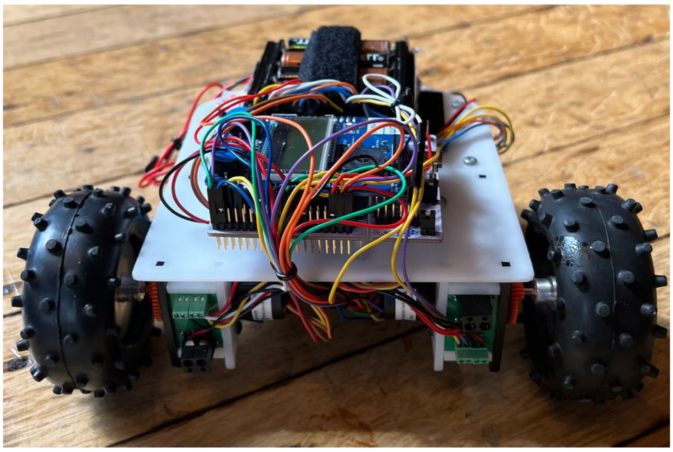


Fig. 1. Wires are secured firmly

Figure 2 shows that the physical buggy closely matches the CAD model shown in Figure 3, except for the wiring, Velcro, switch, and shroud. The shroud was added to block ambient light from influencing the sensors' performance. The shroud's bottom edge was cut slightly to mitigate the effect of friction should it come into contact with the track. A gearbox with a gear ratio of 10.8, Gearbox 2, was selected for the buggy, ensuring a suitable balance of speed and torque for which would be necessary for climbing the ramp.

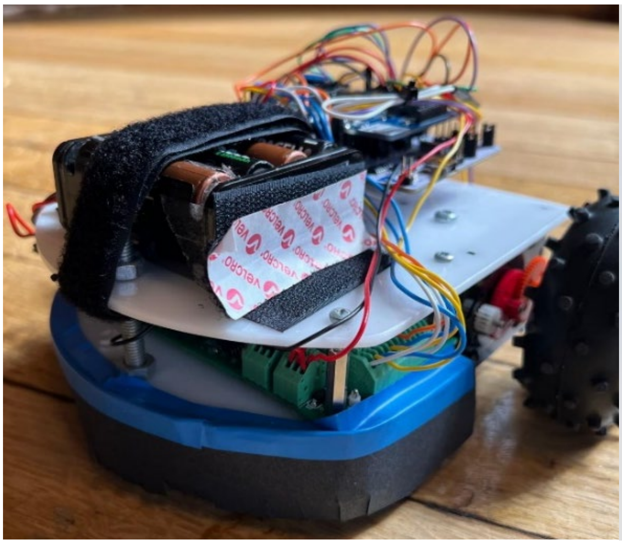


Fig. 2. Picture of completed buggy

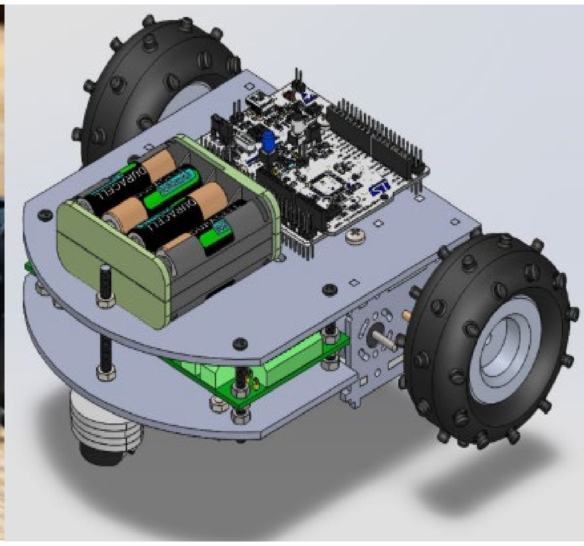


Fig. 3. CAD representation of buggy

## 2.2. Electronic

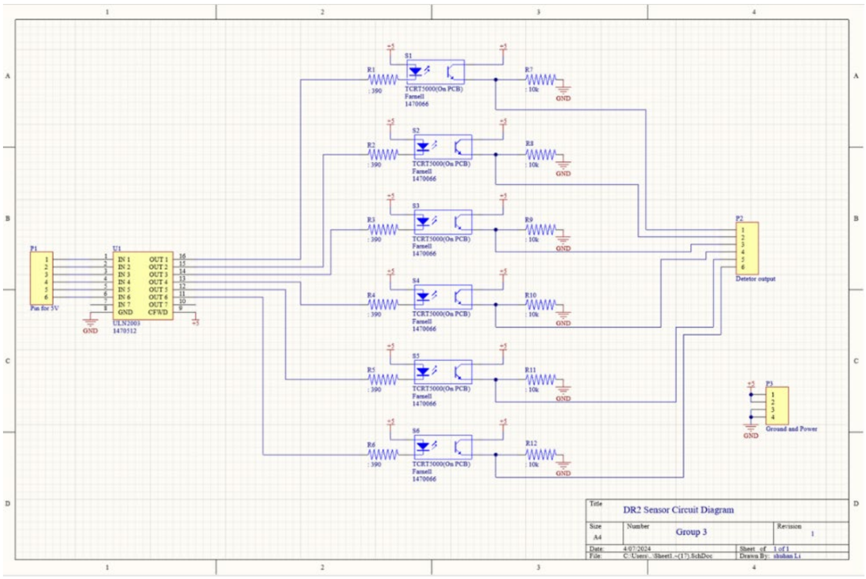



Fig. 4. Schematic diagram of the sensor array PCB

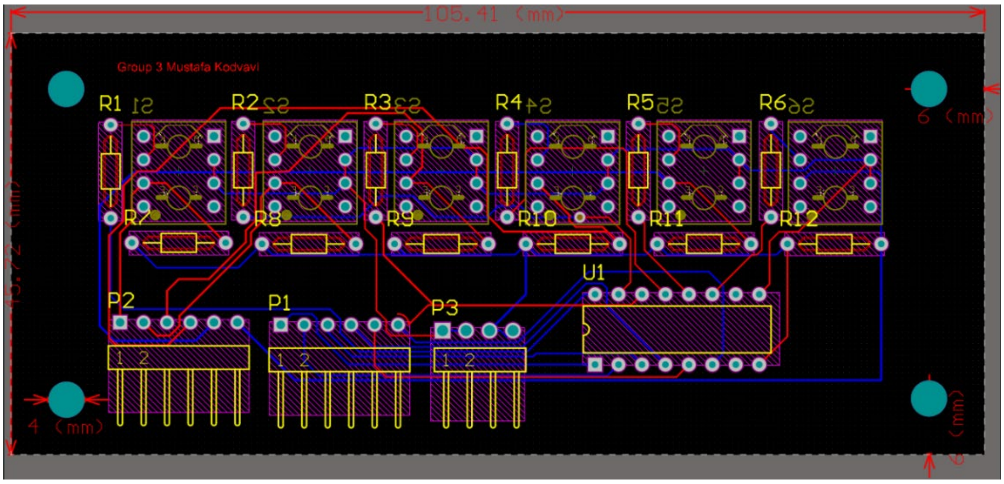



Fig. 5 Altium CAD diagram

The TCRT5000 sensor package used to detect the line comprises of an infrared LED and a phototransistor. The infrared LED emits light, and the phototransistor's collector-emitter current increases depending on how much light it receives. In this particular use case, the amount of current will be dependent on the proportion of white to black track in the field of view of the sensor, as a higher proportion of white track will cause more light from the LED to be reflected into the phototransistor. This means the collector–emitter current directly relates to its position over the white line and can be used in conjunction with other sensors to find the position of the buggy.

On the PCB, each sensor package is connected to an LED switching circuit and a current measurement circuit. The ULN2003an Darlington Array is an array of transistors that allows the high current LEDs to be switched on and off. By connecting the LED in series with a pin on the Darlington Array, the microcontroller can provide a voltage to a pin on the opposite side of the package, completing the circuit and switching on the LED without having to provide a high current. A 390 Ω resistor is connected in series with the LED to limit the current to around 10 mA. To measure the current through the phototransistor, a 10 kΩ resistor is connected in series with the phototransistor alongside the detector output header, which connected across the resistor. Thus, when the microcontroller pin is programmed as an AnalogIn, it can read the voltage across the resistor, essentially reading the current through the circuit. The layout for the sensor circuits is displayed in the schematic diagram Figure 4.

The advantages of flashing the LEDs were minimising potential crosstalk between the sensors and reducing the current draw of the sensor PCB. Two maintain a high reading frequency, alternate sensors were read at the same time; the 1st, 3rd and 5th sensor and then the 2nd, 4th and 6th. This proved successful, sending consistent and accurate position data to the microcontroller. However, a potential improvement would be grouping the tracks that turn on the LEDs under one header, which would reduce the number of wires required for the sensor board by 4.

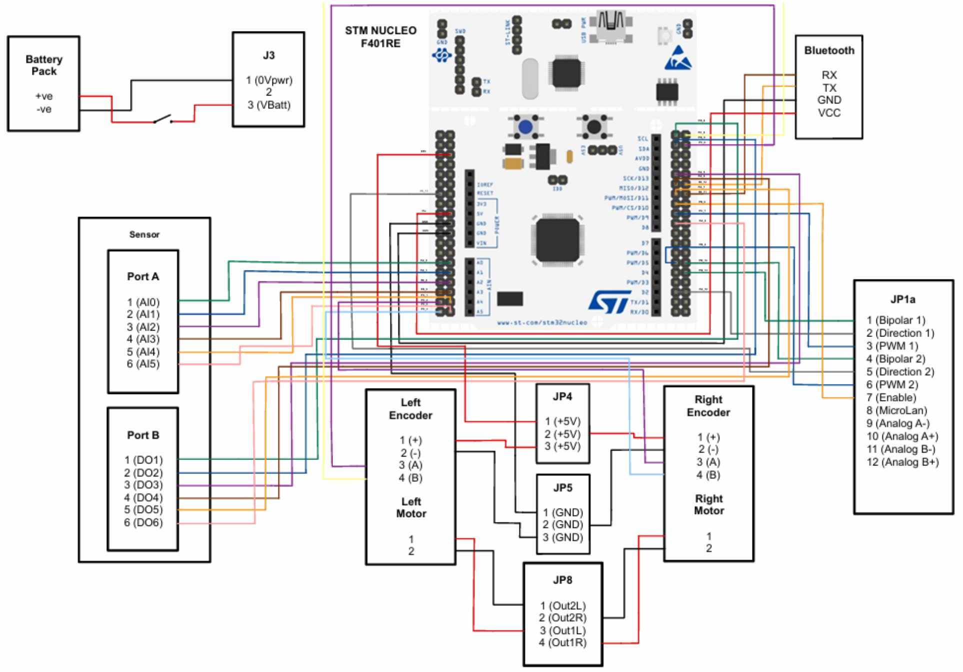



Fig. 6. Wiring diagram of the buggy

## 2.3. Control

The control algorithm is integral to steering the buggy along the white line, enabling the buggy to correct itself when deviating from the line. Therefore, the selection and design of the control algorithm are crucial for the buggy. PID control was ultimately chosen for its self-correcting behaviour and reliability. Additionally, bang-bang was considered and implemented but was abandoned due to its instability and oscillatory motion.

A PID (Proportional-Integral-Derivative) Controller function, Figure 7, was developed to control the buggy's movement. This implementation used a simplified version of PID control [2] eliminating complex calculations that would be computationally taxing for the microcontroller.

Selecting the appropriate motor control proportional gain, $M\_k_p$, was crucial for improving the buggy's stability and performance. An insufficient $M\_k_p$ value would prevent the buggy from responding quickly enough to changes in its position, resulting in delayed adjustments during turns. Conversely, a high $M\_k_p$ value would cause over correction and significant initial overshoot. After extensive testing, final parameters were established to ensure optimal performance in various track scenarios. The tuned values allowed the buggy's motors to change speeds responsively, enabling immediate speed adjustments on ramps and turns.

Next, the steering proportional constant, $S\_k_p$, and steering derivative constant, $S\_k_d$, were tuned. An insufficient proportional constant would lead to inadequate turning capabilities, while an excessively high proportional constant would lead to large oscillation and a large initial overshoot, as is characteristic with high $k_p$. However, proportional control alone exhibited significant oscillations, thus a derivative constant was added to dampen the system and thus prevent oscillations. The final constants, shown in Figure 8, were established after extensive testing.

By carefully tuning the PID controller parameters, the buggy's movement and steering were optimized, ensuring responsive and stable performance on the track.

```
float PID_controller(float error, float integral, float previous_error, float dt, float Kp, float Ki, float Kd)
{
    integral += error * dt;
    float derivative = (error - previous_error) / dt;
    float output = Kp * error + Ki * integral + Kd * derivative;
    previous_error = error;

    return output;
}
```

Fig. 7. PID control algorithm used in the buggy program

```
volatile float RM_Kp = 0.0002; // Proportional gain of right motor
volatile float RM_Ki = 0;         // Integral gain of right motor
volatile float RM_Kd = 0;         // Derivative gain of right motor
volatile float LM_Kp = 0.0002;  // Proportional gain of left motor
volatile float LM_Ki = 0;         // Integral gain of left motor
volatile float LM_Kd = 0;         // Derivative gain of left motor

volatile float S_Kp = 735; // Proportional gain of steering
volatile float S_Ki = 0;  // Integral gain of steering
volatile float S_Kd = 0.005;  // Derivative gain of steering
```

Fig. 8. PID parameters: RM and LM for motor values and S for steering values

## 2.4. Software

The software functionality changed drastically from the proposed case diagram in Design Report Two [3]. To ensure that each control algorithm was being updated at the same time, all sensor reading, motor control and steering control was called on a ticker. This dramatically improved the responsiveness of the buggy in testing and provided much greater control over the sampling rate. The error value used in the steering control algorithm was calculated by subtracting the sensor readings of the sensors on the right of the line from those on the left - with a biasing multiplier for the outermost sensors. This value was then sent to the steering control algorithm which would set appropriate target speeds for each motor depending on the steering direction. Finally, the motor control algorithm would use the encoders to calculate the motors' current speeds and adjust the PWMs until their targets were met. Therefore, the only functions in the main loop were related to the initialisation of the buggy and receiving Bluetooth interrupts for buggy manoeuvres, including the 180° turn, start and stop commands. Due to time constraints, the decision was made not to include the battery monitor in the code instead acquire a second set of batteries to use once the original set required charging. Additionally, for debugging purposes, Bluetooth commands were implemented that would send telemetry data to a BLE Terminal app instead of the LCD screen, as the AnalogIn pins used by the peripheral board made the sensor values unpredictable.

An extra feature that was added during the coding of the buggy was a synchronisation state that was called by a Bluetooth command. During this state, the buggy would record the minimum and maximum voltage values from each sensor. The maximum voltage value was used to normalise the sensor readings, making up for physical inconsistencies that would cause the sensors to respond differently. The minimum value was used in the threshold calculation to stop the buggy at the end of the white line. The code for the sync state is shown in Figure 9.

```
sLEDOn(5);
if (syncstate){
    if (sensor2Val > s2MaxVal){s2MaxVal = sensor2Val;}
    if (sensor4Val > s4MaxVal){s4MaxVal = sensor4Val;}
    if (sensor6Val > s6MaxVal){s6MaxVal = sensor6Val;}
    if (sensor2Val < s2MinVal){s2MinVal = sensor2Val;}
    if (sensor4Val < s4MinVal){s4MinVal = sensor4Val;}
    if (sensor6Val < s6MinVal){s6MinVal = sensor6Val;}
}
```

Fig. 9. Code for the synchronization state of the buggy

As has been stated previously, the decision was made to replace the peripheral board with a BLE Terminal App. This sent debugging values to the user's phone, but other commands were coded in that allowed the user to change the steering control algorithm without reuploading code to the microcontroller, significantly reducing the amount of time required to tune PID control values.

## 3. Team Organisation and Planning

### 3.1. Objectives

The main objective of the project is to design and build an autonomous line following buggy, but in order to achieve this, there are several prerequisites, comprising of both design reports and technical demonstrations; Design Reports 1 and 2 (DR1 and DR2) characterise the motors and sensors implemented on the buggy whereas technical demonstrations display the abilities of the buggy.

While all of the deliverables were achieved, TDB was delayed by three days due to a series of hardware problems with the motors and motor drive boards, further explained later in this section. This set the project back and necessitated updating the plan as the buggy had to be rewired and reprogrammed for the new motor drive boards. The biggest issue this presented was the significant impact on tuning as during this time, access to the track in the lab was revoked until the heats.

## 3.2. Timeframes

One issue was one member of the group soldering every component on the PCB on the wrong side, a problem in engineering that dates back at least as far as Murphy's law, and the resulting tests to determine why the board was not working caused delays to the project. When the replacement PCB was correctly soldered, one of the sensors was not responding. This was rectified but again, the culture within the group was to fix the problems, not assign blame or descend into unproductive arguments, and the objective of the group remained the same, to deliver an autonomous line following buggy.

Figure 13 shows the Gantt chart from the proposal document [4]. Through the Gantt chart, the order of the tasks was accurately predicted, although some of the timescales were optimistic. The buggy assembly took longer than expected so more time should have been allowed for this. However, as the buggy was still ready for TDA, this is an acceptable delay. The biggest discrepancy between the Gantt chart and the project was the line sensing and control section which took significantly longer than projected.

The first indication that something was wrong was a routine test of the motor control. The wheels of the buggy were moving at a constant speed both on the track and in the air but the buggy was unable to climb up the slope. The cause was identified as an issue with the motors and the gearboxes both of which were replaced – the motor drive board was also replaced for good measure. This delayed the project by about 5 days as the cause of the issue was not immediately obvious and the motors were not the first things that were checked. Furthermore, the new motors had issues with the shaft slipping on the gears which also had to be repaired by the technicians.

By the time this was fixed, Technical Demonstration B was less than 24 hours away and the power supply header on the new motor drive board became increasingly temperamental to the point of being unusable. The decision was made to make another visit to the technicians to ask them to resolve the problem - the technicians provided a third motor drive board, and this was fitted to the buggy. The design of the buggy is such that the buggy needed to be part disassembled and rewired to install the motor drive board, the buggy was dismantled and rebuilt three times within one week due to those hardware faults. The latest motor drive board was active low where the previous one had been active high, so the code had to be adjusted. When this was tested, only one wheel was functional. This time the cause was a loose wire in the gearbox and so a final trip to the technicians was made before they closed for the day.

The motor speed control of the buggy was tested on the incline and while this was successful, only a couple of attempts could be completed before the lab closed for the evening. Testing continued on a homemade track, shown in Figure 10, in a last-ditch attempt to complete the functionality before TDB the following morning. At this point, an extension to TDB was requested and granted due to the component failures of the preceding week. TDB was pushed back to the Friday from the Tuesday and this later deadline was met, with the buggy achieving most of its deliverables.

Fig. 10. Picture of makeshift home testing track

As most of these issues were unforeseen, they prevented work on the other areas of the project including this report.

### 3.3. Teamwork

The split of the workload during this intense period of testing was the entire team dedicating all their available time on the buggy to get it track ready. The fast-paced nature of the problem solving also entrenched previous roles, in that the group members with sensor expertise focused primarily the sensor issues, the people with stronger knowledge of the wiring led the repeated reassemblies and the people that understood the software best did the most coding. In some respects, this was a roughly equal division of labour but with more time, the project would have benefitted from every team member diversifying their skillsets, and thus providing fresh perspectives.

However, the time pressures did focus the group during disputes, as the problems needed to be solved quickly, rather than superfluously arguing over potential solutions. This also created an ethos where members were willing to investigate any theory that showed promise of solving the problem. Furthermore, the learning styles `described in the proposal document morphed as some `group members took on more pragmatic roles, arguing for restraint so as to test individual hypotheses while other members preferred a more “activist” approach. Ultimately, it is the compromises between these two positions, as well as the commitment of all the team members to step up when others had to step back, that led to the successful outcome. Despite the artificially short turnaround period between TDB and TDC, due to the extension, the interim period reflected a period of relative calm where the entire group took turns to tune the buggy to optimise it for the heats and the race.

### 3.4. Documentation

The fastidious record keeping was another factor that was integral to the overall success of the project. Starting with effective software versioning, the extensive use of one drive folders gave every team member access to every test result. The addition of the tuning spreadsheet prevented the duplication of tests and underpinned the rolling system of different team members testing at different points. Were this project to be carried out again, agreeing those formats earlier may have sped up that part of the process.

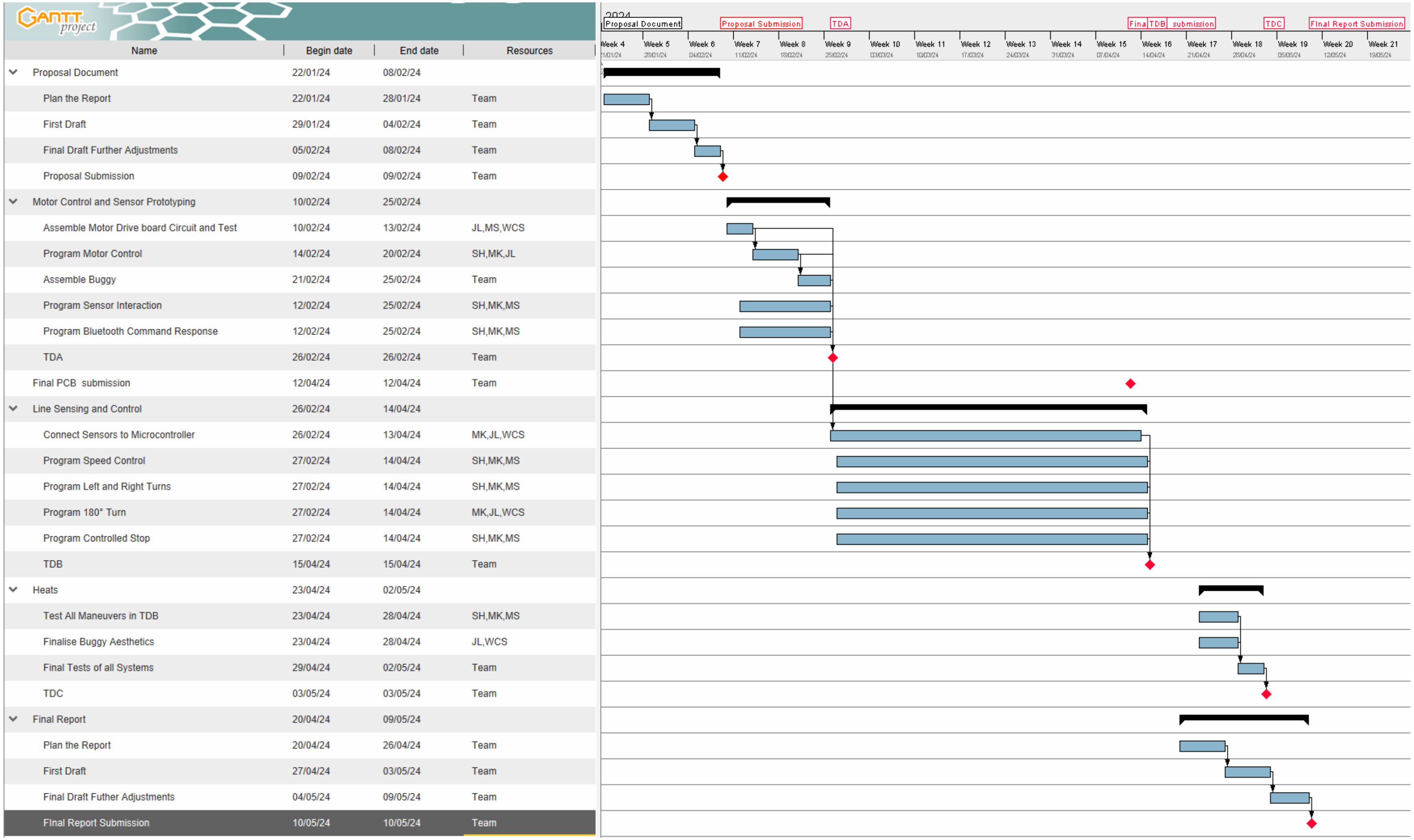
| | | | |
|---|---|---|---|
| ExperimentalCode.cpp | | 28/04/2024 16:59 | C++ Source |
| TDB_Final.txt | | 26/04/2024 11:38 | Text Document |
| TDB_NewHWTesting.cpp | | 21/04/2024 15:20 | C++ Source |
| TDB_SennyTesting.cpp | | 05/03/2024 12:03 | C++ Source |
| TDB_V3.0.cpp | | 09/04/2024 11:09 | C++ Source |
| TDB_V3.0_Motor.cpp | | 11/04/2024 19:33 | C++ Source |
| TDB_V3.0_OnlySensor.txt | | 09/04/2024 12:48 | Text Document |
| TDB_V5.txt | | 13/04/2024 13:37 | Text Document |
| TDB_V6.cpp | | 16/04/2024 12:52 | C++ Source |
| TDB_V7.cpp | | 16/04/2024 11:17 | C++ Source |
| TDB_V8.0_Motor.cpp | | 22/04/2024 10:43 | C++ Source |
| TDB_V8.cpp | | 18/04/2024 10:36 | C++ Source |
| TDB_V10.cpp | | 29/04/2024 12:21 | C++ Source |
| TuningSheet.xlsx | | 06/05/2024 03:05 | Microsoft Excel W... |

Fig. 11. Software versioning system, and the tuning document.

Other efficiency increasing documents include the pinout diagram which allowed problems to be diagnosed more quickly and gave a visual representation of which pins were available for functions.

| Name1 | Name2 | Free? | Use | Name1 | Name2 | Free? | Use |
|---|---|---|---|---|---|---|---|
| PA_14 | | No | Sensor 1.3 | +3V3 | | | |
| PA_15 | | No | Sensor 1.4 | +5V | | No | Bluetooth Power |
| GND | | | Bluetooth Ground | GND | | No | Micro Ground |
| PB_7 | | No | | GND | | | Bluetooth Ground |
| PC_13 | | No | Sensor 1.5 | VIN | | | |
| PC_14 | | No | Reserved | NC | | | |
| PC_15 | | No | Reserved | PA_0 | A0 | Maybe | Pot1 |
| PH_0 | | No | Reserved | PA_1 | A1 | Maybe | Pot2 |
| PH_1 | | No | Reserved | PA_4 | A2 | Maybe | UpJoystick / RightMotorEncoderA |
| VBAT | | | | PB_0 | A3 | Maybe | DownJoystick / RightMotorEncoderB |
| PC_2 | | No | Sensor 2.5 | PC_1 | A4 | Maybe | LeftJoystick |
| PC_3 | | No | Sensor 2.4 | PC_0 | A5 | Maybe | RightJoystick / Sensor 2.6 |

Fig.12. Section of the Excel PinOut table




| Name | Begin date | End date | Resources |
|---|---|---|---|
| Proposal Document | 22/01/24 | 08/02/24 | |
| Plan the Report | 22/01/24 | 28/01/24 | Team |
| First Draft | 29/01/24 | 04/02/24 | Team |
| Final Draft Further Adjustments | 05/02/24 | 08/02/24 | Team |
| Proposal Submission | 09/02/24 | 09/02/24 | Team |
| Motor Control and Sensor Prototyping | 10/02/24 | 25/02/24 | |
| Assemble Motor Drive board Circuit and Test | 10/02/24 | 13/02/24 | JL,MS,WCS |
| Program Motor Control | 14/02/24 | 20/02/24 | SH,MK,JL |
| Assemble Buggy | 21/02/24 | 25/02/24 | Team |
| Program Sensor Interaction | 12/02/24 | 25/02/24 | SH,MK,MS |
| Program Bluetooth Command Response | 12/02/24 | 25/02/24 | SH,MK,MS |
| TDA | 26/02/24 | 26/02/24 | Team |
| Final PCB submission | 12/04/24 | 12/04/24 | Team |
| Line Sensing and Control | 26/02/24 | 14/04/24 | |
| Connect Sensors to Microcontroller | 26/02/24 | 13/04/24 | MK,JL,WCS |
| Program Speed Control | 27/02/24 | 14/04/24 | SH,MK,MS |
| Program Left and Right Turns | 27/02/24 | 14/04/24 | SH,MK,MS |
| Program 180° Turn | 27/02/24 | 14/04/24 | MK,JL,WCS |
| Program Controlled Stop | 27/02/24 | 14/04/24 | SH,MK,MS |
| TDB | 15/04/24 | 15/04/24 | Team |
| Heats | 23/04/24 | 02/05/24 | |
| Test All Maneuvers in TDB | 23/04/24 | 28/04/24 | SH,MK,MS |
| Finalise Buggy Aesthetics | 23/04/24 | 28/04/24 | JL,WCS |
| Final Tests of all Systems | 29/04/24 | 02/05/24 | Team |
| TDC | 03/05/24 | 03/05/24 | Team |
| Final Report | 20/04/24 | 09/05/24 | |
| Plan the Report | 20/04/24 | 26/04/24 | Team |
| First Draft | 27/04/24 | 03/05/24 | Team |
| Final Draft Futher Adjustments | 04/05/24 | 09/05/24 | Team |
| Final Report Submission | 10/05/24 | 10/05/24 | Team |

Fig. 13. Gantt Chart for the project

# 4. Budget vs Outturn

The overall cost of the buggy closely aligns with the preliminary estimate of £291.49 outlined in the proposal document [4], as the design did not incorporate any new components. This figure represents the baseline valuation of the buggy, presuming that component costs remain constant irrespective of production scale. Nevertheless, various hardware malfunctions and extensive testing necessitated the replacement of multiple components, thereby incurring additional expenses. These extra components, which are not factored into the total cost of the buggy, are detailed in Table 1. They are classified into either "Research & Development (R&D)" or "Replacement" categories. The latter indicates parts substituted due to hardware defects—such as motors and motor drive boards—or accidental damage.

Table 1. Additional components purchased for the development of the buggy

| Components | Category | Amount | Cost per Unit (£) | Total Cost (£) |
|---|---|---|---|---|
| Batteries | R&D | 8 | 2 | 16 |
| Electrical tape | R&D | 1 | 1 | 1 |
| Motors | Replacement | 2 | 3.92 | 7.84 |
| Motor drive board | Replacement | 2 | 30 | 60 |
| TCRT5000, reflective optical sensor | Replacement | 6 | 0.72 | 4.32 |
| 4-way SIL header, 2.54 mm pitch | Replacement | 6 | 0.27 | 1.62 |
| Custom PCB | Replacement | 1 | 12.36 | 12.36 |

The total expenditure for these additional components was £103.14. This amount is included in the developmental cost of the buggy. Moreover, the developmental costs encompass the human capital involved in research and development. Assuming each team member dedicated 200 hours throughout the development period at a rate of £12 per hour, the comprehensive cost analysis for one buggy is presented in Table 2.

Table 2. Cost analysis for one buggy

| Description | Cost per Unit (£) | Units | Cost (£) |
|---|---|---|---|
| Material Cost | 291.49 | - | 291.49 |
| Manufacturing Hours | 12 | 2 | 24 |
| R&D material cost | 103.14 | - | 103.14 |
| R&D Labour Cost | 12 | 1000 | 12000 |
| Total | - | - | 12418.63 |

As per Table 2, £12,103.14 of the total cost is attributable to fixed costs. In a mass production scenario, this would be distributed across the number of units produced. Table 3 delineates the per-unit cost, incorporating profit margins, assuming 1,000 units are manufactured.

Table 3. Per unit cost and profit margins

| Description | Amount |
|---|---|
| Variable Cost per Unit (£) | 315.49 |
| Fixed Cost per Unit (£) | 12.1 |
| Total Cost per Unit (£) | 327.59 |
| Selling Price per Unit (£) | 379 |
| Profit per Unit (£) | 51.41 |
| Profit Margin (%) | 15.69% |
| Total Revenue (£) | 379000 |
| Total Profit (£) | 51410 |
| Profit per Member (£) | 10282 |

In conclusion, the analysis above provides a detailed breakdown of the costs associated with the development and potential mass production of the buggy. While the initial cost estimates were closely adhered to, unforeseen hardware issues and the requisite for additional testing led to increased developmental costs. Nevertheless, the comprehensive financial overview presented indicates that with scale, the production becomes economically feasible, offering a promising profit margin of 15.69%.

## 5. Analysis of Heats

### 5.1. Pre-heat testing and calibration

As detailed in the software section, the buggy's motor speed and steering were regulated through a PID controller. While this algorithm facilitates smooth, self-correcting operation, it requires extensive tuning, a time-consuming iterative process. Therefore, meticulous record-keeping was essential to maximise efficiency. A spreadsheet was developed to log each adjustment made in the code and its corresponding performance on the test track (small section shown in Table 4), ensuring no repetition of tested values.

Initially, the speed PID controller was adjusted to manage the motors' response times effectively during turns or when modifying the PWM signal for changes in incline. Since the steering mechanism operates by varying each motor's speed, proficient speed control was crucial for successful navigation. Inadequate speed regulation could impair steering, making the tuning of the turning algorithm more complicated and unpredictable. Therefore, tuning began with the motor speed control, aiming to maintain consistent motor speeds regardless of surface incline, decline, or friction. Once the buggy could adjust motor speed responsively, the steering PID control could be tuned.

During testing, it was immediately obvious that as speed increases the difficulty of tuning the control algorithm also increases. Thus, a low-speed version of the program was developed and tuned, which was intended to be used in Technical Demonstration B and as a safe version of the program for the heats. However, unlike Technical Demonstration B, the heats were timed, thus a high-speed version of the program was tuned. Additionally, to minimise the guesswork in high-speed tuning, slow-motion videos (folder shown in Figure 15) of the buggy's turns were analysed to determine necessary adjustments.

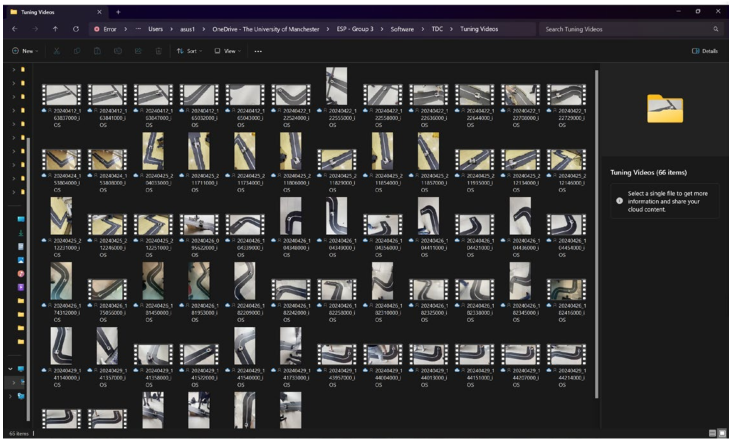


Fig. 15. Folder containing videos of buggy tests

Testing in various lighting conditions revealed a significantly greater impact on performance than initially anticipated. To address this, a shroud was designed and crafted from dark paper to standardise sensor lighting conditions, as time constraints precluded 3D printing. This makeshift shroud, featuring brush-like bristles for flexibility, effectively mitigated friction with the track, enhancing consistency across different lighting conditions.

In addition to the shroud, a new feature was incorporated into the buggy: a BLE command that activates a calibration mode. During this mode, the sensors continuously capture data to establish maximum and minimum values. These values are then used to normalise the sensor readings, effectively eliminating the impact of sensor inconsistencies. This enhancement significantly increases the buggy's adaptability, allowing it to perform consistently across different track conditions.

### 5.2. Heat Attempt I

After comprehensive tuning, the buggy performed exceptionally well in the initial heats, clocking a time of 11.28 seconds, and qualifying for the race. It demonstrated excellent line-following precision, agility, and speed maintenance on inclines and declines. The Bluetooth turnaround command effectively realigned the buggy with the track, and it stopped reliably at the track's end. However, the run displayed oscillations across the line on descents and highlighted the need for further tuning. Additionally, a significant portion of time was lost due to the slow 180-degree turnaround, although this was deliberate to prevent overturning due to the motors not stopping effectively after the turn. This could have been remedied by active braking, but it was not implemented due to time constraints.

### 5.3. Heat Attempt II

Although the buggy had already set a qualifying time, the buggy had the opportunity to try again to further improve its time. Thus, for this heat, the buggy used experimental high-speed tuning (different experimental parameters in software). It successfully negotiated the first 90-degree turn but failed to adjust in time for the second turn, resulting in understeering and track departure, thus failing the attempt. This attempt demonstrated potential for improved performance but underscored the necessity for further tuning and testing.

### 5.4. Analysis of performance

Reflecting on the performance during the heats, the strengths and weaknesses of the buggy's design and features were clearly observed.

**Most Successful Features:**

- **Compact Design:** The compact nature of the buggy contributed significantly to its enhanced manoeuvrability and ease of control.
- **Low and Central Centre of Gravity:** This design aspect substantially decreased the risk of overturning during sharp turns and high-speed manoeuvres.
- **Efficient Algorithm:** The algorithm was optimised to boost the buggy’s line-following precision and agility.
- **Efficient Tuning Value Adjustment via BLE (Bluetooth Low Energy):** This feature enabled real-time adjustments to be made remotely, eliminating the need for directly reuploading code.
- **Shroud:** The custom-designed shroud standardised lighting conditions for sensors, minimising the impact of variable external lighting and enhancing sensor reliability.
- **Brush Bristle Shroud:** The shroud's brush-like bristles enhanced its flexibility and reduced friction when making contact with the track, improving handling.
- **Height Adjustable Sensors:** These sensors provided crucial flexibility, allowing for adjustments to different track conditions which enhanced the buggy’s responsiveness and accuracy.
- **Calibration Mode:** A BLE-triggered calibration mode allowed sensors to recalibrate on the fly, capturing maximum and minimum readings to normalise the data, thereby enhancing the buggy's adaptability to various track surfaces and conditions.

**Areas for Improvement:**

- **Poor Sensor Distancing:** Inadequate sensor spacing hindered effective line detection.
  - **Solution:** Redesign the PCB to be wider to accommodate for increased spacing between each sensor.
- **No Sensor Hot Swap Capability:** The inability to quickly replace or modify sensors during testing limited operational adaptability. As broken sensors could only be replaced through desoldering which was time consuming and had potential to damage the PCB.
  - **Solution:** Design a mounting mechanism for sensors which does not add significant height to the sensors, perhaps using the clips on the sensor package.
- **Lack of Battery Monitor:** The absence of a battery monitoring system overlooked potential enhancements in power management as well as making it impossible to predict if the state of charge of the batteries had any effect on the performance.
  - **Solution:** Integrate a real-time battery monitoring system to track

voltage levels and battery health, improving power management.

- **Nondurable Shroud:** While effective in maintaining consistent lighting conditions, the paper-made shroud lacked durability and was susceptible to damage under various environmental conditions.
    - **Solution:** Replace the paper shroud with a flexible 3D printed one.
- **Poor cable management:** Disorganised cable routing led to potential entanglements and disconnections during operation, posing risks of sudden power loss and sensor failures.
    - **Solution:** Design a dedicated cable routing system using cable clips and conduits to organise and protect wiring.
- **Difficult to access MDB:** In an effort to make the chassis as compact as possible, it came at the expense of accessibility to the motor drive board, which made accessing the test points very difficult, thus delaying hardware testing.
    - **Solution:** A hinge system could be incorporated into the top tier of the chassis, allowing it to fold upward and provide complete access to the motor drive board.

Each feature's performance, both strengths, and weaknesses, were meticulously documented and analysed, providing valuable insights that will guide future improvements and tuning efforts to elevate the buggy's performance.

Table 4. Section of table detailing PID values from different test runs

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| LM_Kp | 0.00007 | 0.00009 | 0.0001 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| RM_Kp | 0.00007 | 0.00009 | 0.0001 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| S_Kp | 685 | 685 | 685 | 685 | 700 | 750 | 750 | 735 |
| S_Kd | 0.0035 | 0.0035 | 0.0035 | 0.0035 | 0.0035 | 0.0035 | 0.004 | 0.005 |
| S_Ki | 0 | 0 | 0 | 0.0002 | 0 | 0 | 0 | 0 |
| PPS | 4500 | 4500 | 4500 | 4500 | 4500 | 4500 | 4500 | 4500 |
| mps | 1.063048686 | 1.063049 | 1.063049 | 1.063049 | 1.063049 | 1.063049 | 1.063049 | 1.063049 |
| time1 | 10.9 | 10.8 | 10.6 | 10.7 | 10.6 | 10.5 | 10.5 | 10.4 |
| time2 | 11.2 | 11.2 | 11.1 | 11.1 | 11.0 | 10.9 | 10.9 | 10.9 |
| time3 | 10.9 | 10.9 | 10.8 | Failed | 10.7 | 10.7 | 10.6 | 10.6 |

Overall, the performance of the buggy was incredible, achieving a time of 11.28 seconds, the 5$^{th}$ fastest time in the heats. With the solutions to the problems mentioned, above as well as new information learnt from the heats, the buggy has the capacity to further reduce its time substantially.

## 6. References


[1] ESP Group 3, "Design Report 1," University of Manchester, Nov. 2023.

[2] University of Manchester, *Embedded Systems Project EEEN21000 Technical Handbook*, 2023/24 ed

[3] ESP Group 3, "Design Report 2," University of Manchester, Dec. 2023.

[4] ESP Group 3, "Proposal Document" University of Manchester, Feb. 2023.