MANCHESTER 1824

The University of Manchester

Department of Electrical and Electronic Engineering

Embedded Systems Project 2023-24

DESIGN REPORT #2

Title: Technical Characterisation

Group Number: 48

Group members:	ID Number	I confirm that this is the group's own work.
Mohamad Amierul Hakeem	11098269	☒
Teethabhumi Tripatarasit	11114360	☒
Jiexi Lu	11488522	☒
Yanhua Zheng	11047457	☒
Sarah El-Mahmoudi	10864355	☒

Tutor: Yin, Wuliang

Date: 01/12/2023

Contents

1. Introduction

This project aims to create a small autonomous vehicle (buggy) capable of navigating a specific route. The buggy is driven by two wheels mounted at the rear and automatically decides its route based on the white lines on the ground. While ensuring the reliability and high speed of the buggy, the project delves into multiple technical areas, including software design, analysis of line sensor characteristics, circuit design, application of non-line sensors, and development of control algorithms and chassis design.

The "Software Design" section examines methods for microcontroller programming to address encountered problems, including the origin and possible destinations of messages from peripherals and the microcontroller, as well as the potential structure of these messages. Functions are broadly reviewed, showing their possible modes of operation and potential inputs/outputs.

The "Line Sensor Characteristics" section is a key part of the report, analysing and comparing the test results of three different sensors to ultimately determine the most suitable sensor type. This sensor will be installed as an array to precisely identify the track line, optimizing navigation performance on the track.

The "Circuit Diagram for Line Sensors" section designs a schematic and PCB layout around the selected sensor, enabling it to function and connect to the microcontroller. The types of interfaces and their locations on the microcontroller and PCB are detailed. This schematic is then transformed into a compact PCB design that can be manufactured and attached to the chassis.

The 'Non-Line Sensors' section explains the capabilities of other sensors on the buggy, the plan of how to use them, the pins for connecting to the microcontroller and the signal type (digital or analogue), as well as discussing whether additional laser sensors are needed.

The "Hardware Overview" section proposes a chassis design for component mounting, considering their sizes, weights, and any flexing of the chassis itself. The material of the chassis is also considered.
Additionally, the report provides a detailed introduction to the functions and applications of other sensors on the buggy, discusses the importance of control algorithms in ensuring the buggy travels along the optimal path, and details the chassis design. These details play a crucial role in the final performance of the buggy.

2. Software

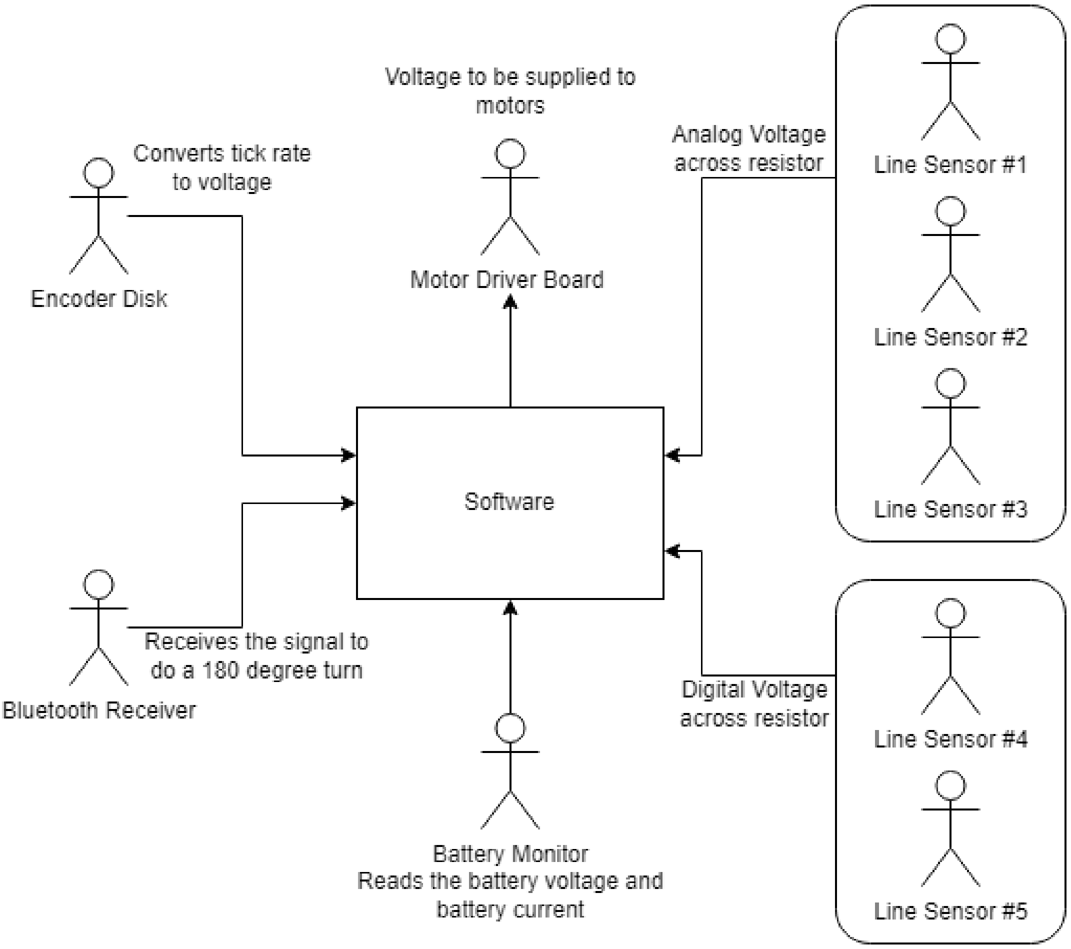
- **Functional summary**
  - The buggy needs to be able to detect the line on the track and keep its center along the line.
  - Able to ignore the gap in the line of length x.
  - The buggy can change directions by turning left and right depending on the direction of the line.
  - Able to stop, start and do a 180 turn when given Bluetooth input signal.
  - Move at a constant speed by increasing or decreasing voltage across the motor depending on whether the buggy is travelling uphill or downhill.

- **List of possible software constraints**

- Clock speed should be considered so that the polling rate doesn't clog up the whole system and that there are still room for calculations at required rate to keep the system relatively stable.
- Code cannot exceed the STM32 available total memory.
- Code must be limited to the number of pins available on the STM32.
- The ADC resolution is finite but should be big enough for the buggy to perform without any noticeable instability or 'shake'.

- Context diagram

The context diagram is to briefly show the main peripheral components that will be connected to and controlled by the STM32 microcontroller which is represented as the software. The motors are connected to the Motor Driver Board and are considered to be connected as one component in the diagram.


Voltage to be supplied to motors
Converts tick rate to voltage
Analog Voltage across resistor
Line Sensor #1
Line Sensor #2
Line Sensor #3
Encoder Disk
Motor Driver Board
Software
Receives the signal to do a 180 degree turn
Bluetooth Receiver
Digital Voltage across resistor
Line Sensor #4
Line Sensor #5
Battery Monitor
Reads the battery voltage and battery current

- Table of messages

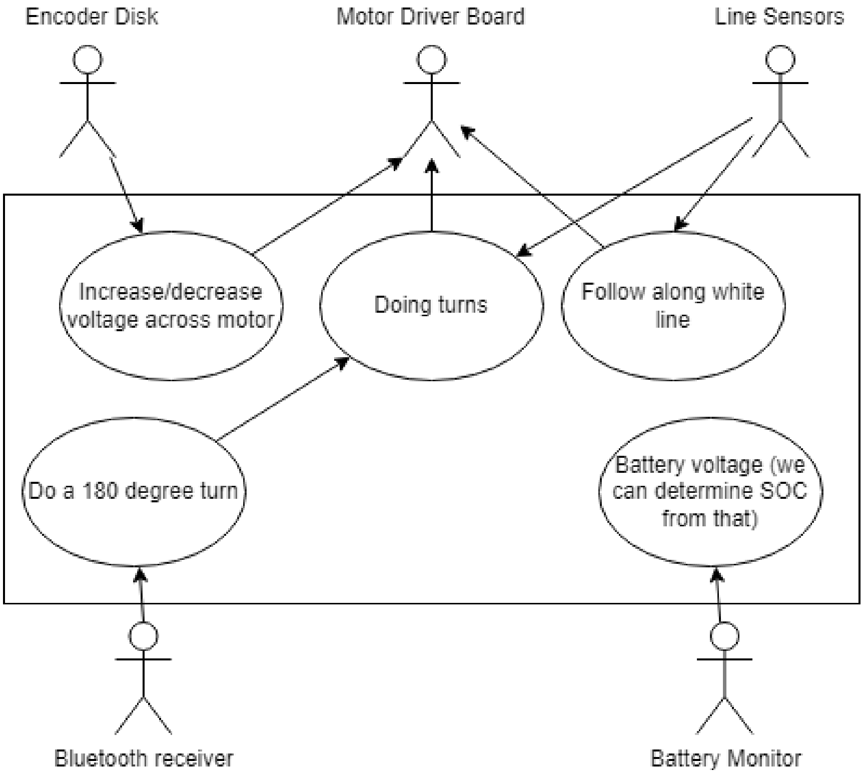
The table below shows the list of information that will be sent or received by the STM32 microcontroller when communicating with the peripheral components connected to the Nucleo board in the buggy.

Message	Input/Output	Data Content	Arrival Pattern	Data Size
Voltage across sensor resistor	Input	voltage	continuous	TBD
Battery voltage	Input	voltage	continuous	TBD

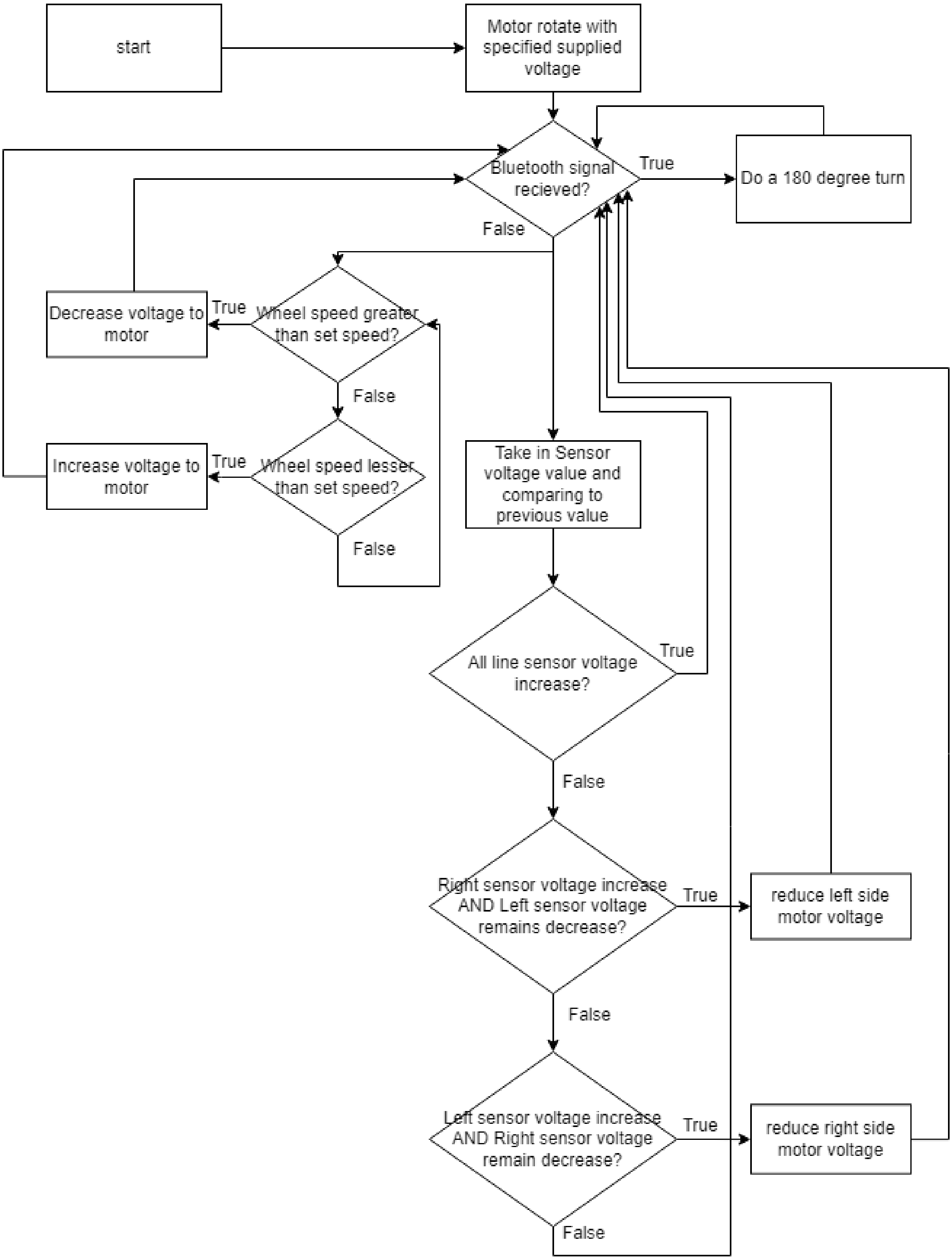
Motor speed	Input	Voltage from encoder tick rate	continuous	TBD
PWM voltage across motor	Output	Simulated DC voltage with AC	periodic	TBD
Signal to turn 180	Input	Bluetooth signal	aperiodic	TBD

- **Use Case diagrams**

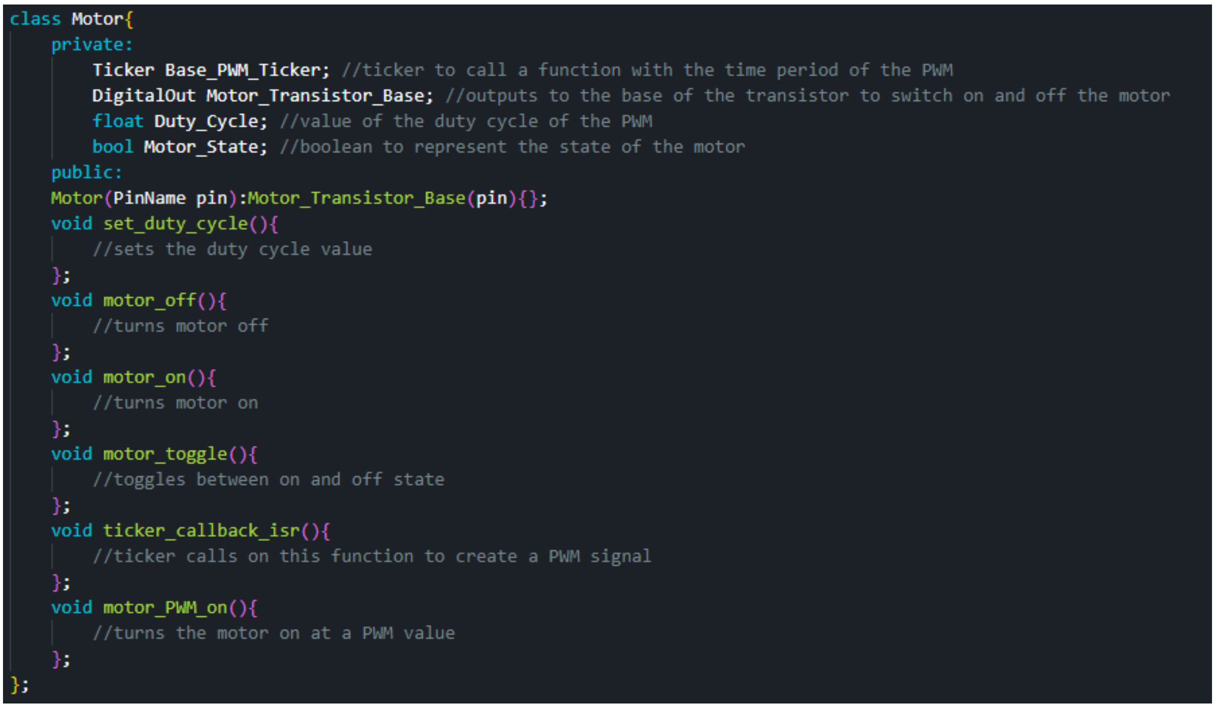
  Below is a diagram showing the signals from or to the peripheral components in the buggy and how they will be used by the software.



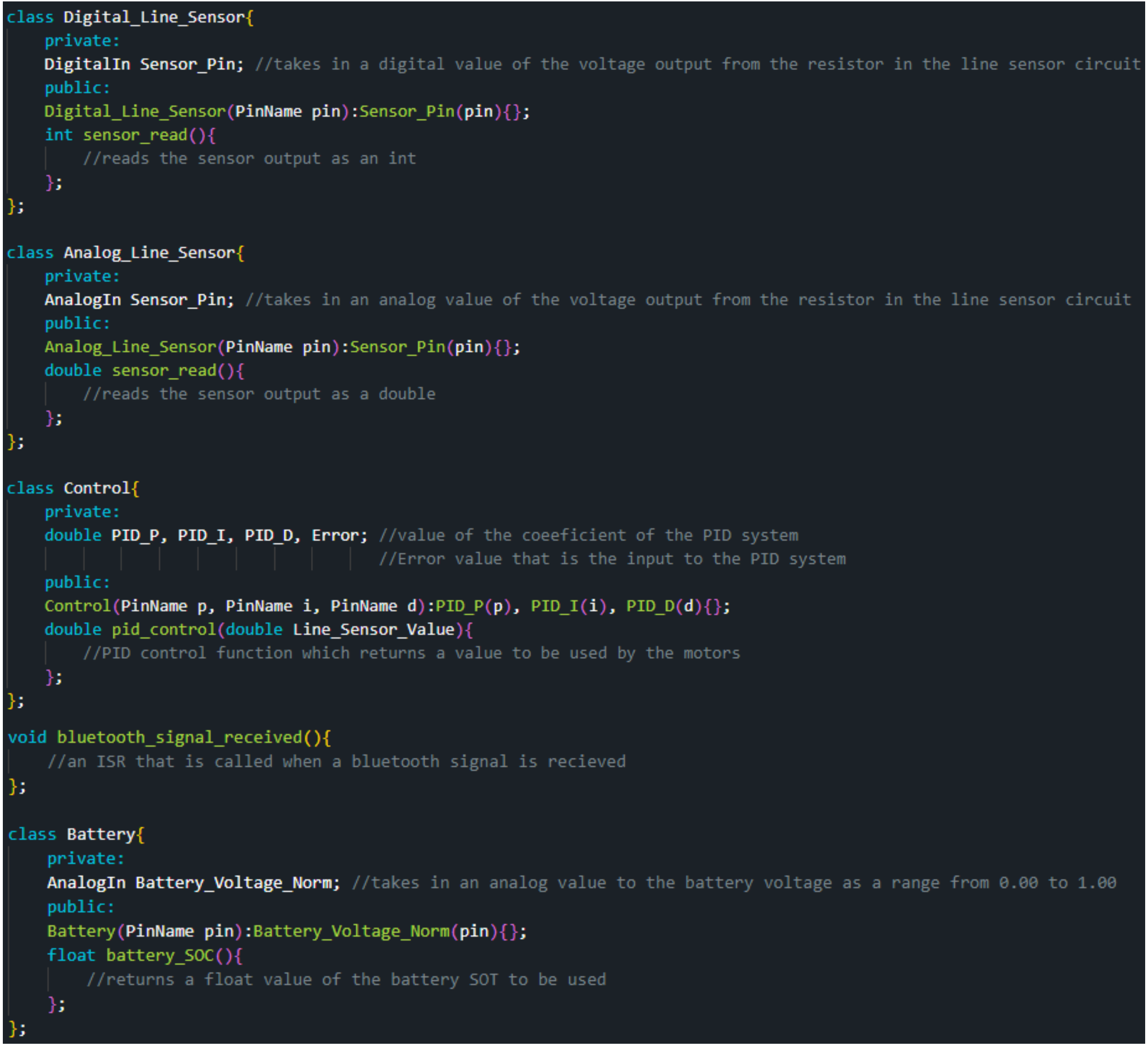
- **Use Case descriptions**


start
Motor rotate with specified supplied voltage
Bluetooth signal recieved?
True
Do a 180 degree turn
False
Decrease voltage to motor
True
Wheel speed greater than set speed?
False
Increase voltage to motor
True
Wheel speed lesser than set speed?
False
Take in Sensor voltage value and comparing to previous value
All line sensor voltage increase?
True
False
Right sensor voltage increase AND Left sensor voltage remains decrease?
True
reduce left side motor voltage
False
Left sensor voltage increase AND Right sensor voltage remain decrease?
True
reduce right side motor voltage
False

- Object specifications



```
class Motor{
    private:
        Ticker Base_PWM_Ticker; //ticker to call a function with the time period of the PWM
        DigitalOut Motor_Transistor_Base; //outputs to the base of the transistor to switch on and off the motor
        float Duty_Cycle; //value of the duty cycle of the PWM
        bool Motor_State; //boolean to represent the state of the motor
    public:
    Motor(PinName pin):Motor_Transistor_Base(pin){};
    void set_duty_cycle(){
        //sets the duty cycle value
    };
    void motor_off(){
        //turns motor off
    };
    void motor_on(){
        //turns motor on
    };
    void motor_toggle(){
        //toggles between on and off state
    };
    void ticker_callback_isr(){
        //ticker calls on this function to create a PWM signal
    };
    void motor_PWM_on(){
        //turns the motor on at a PWM value
    };
};
```



```
class Digital_Line_Sensor{
    private:
    DigitalIn Sensor_Pin; //takes in a digital value of the voltage output from the resistor in the line sensor circuit
    public:
    Digital_Line_Sensor(PinName pin):Sensor_Pin(pin){};
    int sensor_read(){
        //reads the sensor output as an int
    };
};

class Analog_Line_Sensor{
    private:
    AnalogIn Sensor_Pin; //takes in an analog value of the voltage output from the resistor in the line sensor circuit
    public:
    Analog_Line_Sensor(PinName pin):Sensor_Pin(pin){};
    double sensor_read(){
        //reads the sensor output as a double
    };
};

class Control{
    private:
    double PID_P, PID_I, PID_D, Error; //value of the coeeficient of the PID system
                                       //Error value that is the input to the PID system
    public:
    Control(PinName p, PinName i, PinName d):PID_P(p), PID_I(i), PID_D(d){};
    double pid_control(double Line_Sensor_Value){
        //PID control function which returns a value to be used by the motors
    };
};

void bluetooth_signal_received(){
    //an ISR that is called when a bluetooth signal is recieved
};

class Battery{
    private:
    AnalogIn Battery_Voltage_Norm; //takes in an analog value to the battery voltage as a range from 0.00 to 1.00
    public:
    Battery(PinName pin):Battery_Voltage_Norm(pin){};
    float battery_SOC(){
        //returns a float value of the battery SOT to be used
    };
};
```

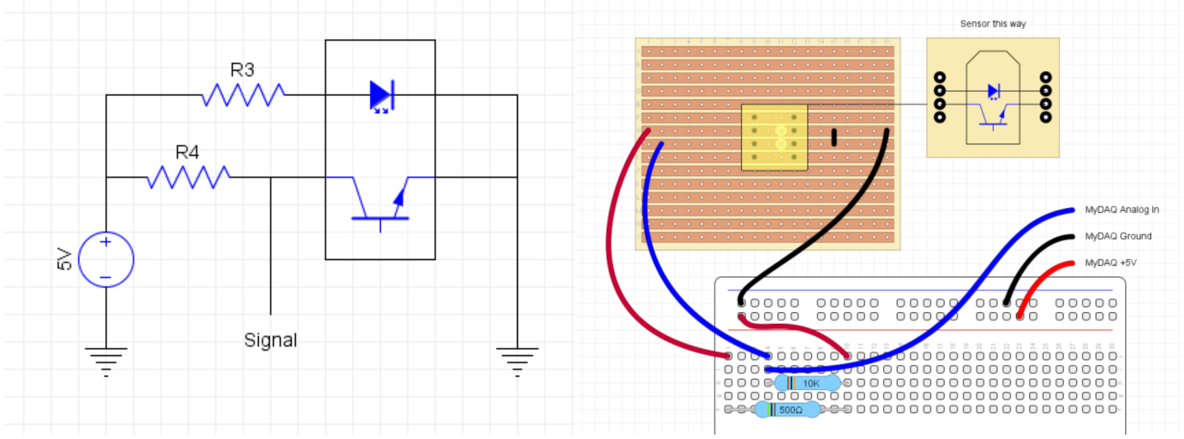
3. Line Sensor Characterisation

Determining the characteristics of the sensors is essential to many design choices in the buggy. The sensors should be able to differentiate between black and white surfaces and the transition at the boundary between them. This allows the buggy to determine the path of the line and turn when necessary. A good sensor is one that is sensitive to changes between black and white surfaces, is minimally effected by external factors that may change on race day, consumes minimal power and is easy to troubleshoot. Three tests were deemed most relevant to the sensor selection and the buggy design overall were included and discussed.

		TCRT5000	TEKT5400S & TSHG6400	SFH203
	Forward voltage	1.25V	1.5V	
	Peak wavelength	940nm	850 nm	
	Max forward current	60mA	100mA	
	Spectral bandwidth	N/A	850nm to 980nm	
	Collector current	1mA	4mA	
	Dark current	10nA	1nA	
	Angle of half sensitivity	+-22	+-37	

Characteristics of each individual emitter and receiver that must be taken into account prior to testing. Firstly the wavelength of the emitter must be within the range of the phototransistor's spectral bandwidth – the closer it is to the peak sensitivity wavelength the better the pairing. The angle of half sensitivity must be considered as it will determine the function of the signal as it passes across the white line. The values for both maximum current for the photodiode and the collector current for the phototransistor will be used to calculate resistor values respectively.

To carry out this experiment the circuit should be set up as shown in diagram x. During the experiment stripboard with a 8pin DIP can be used to slot into different heights and with different sensors as shown in diagram x. In this arrangement the phototransistor's collector and the photoemitter's cathode are connected to to Vcc via the respective resistors. The phototransistor's emitter and the photoemitter's anode are tied to ground. This setup gives a high signal when the track is dark and a low signal when the track is white. This active high signal arrangement is done to easily detect if there is loose connection or if there is a fault in our PCB because the default output should be high signal.



Test 1

To determine how the sensors output signal varies with height tests were conducted using a test rig with regular height intervals from both black and white surfaces, using various resistor values to vary the current. The difference between them was calculated as shown in graph. This is relevant to the sensor selection as the difference between the light and dark surfaces must be large enough to ensure variations in noise and sunlight (effects of sunlight will be discussed in greater detail later on), do not have a significant effect on the analog reading by the microcontroller or the threshold voltage in a digital reading. In addition to aiding the selection of the sensor used, this experiment also aims to provide information as to which resistor value to use, and the height of the sensor board on the chassis.

Each reading was taken 5 times, resetting the sensor each time. This was used to calculate standard deviation and therefore the standard error for each measurement which is represented using the error bars. A digital multimeter was used instead of a mydaq for increased precision. Based on small error range, the results appear to be robust and reliable for our application. This was also repeated at varying current values by using different biasing resistor values. One of the test rigs slots were faulty and there was no alternative, which is why there is a missing measurement in graph x . This did not have a significant impact on the final results or design and there is sufficient data to interpolate the missing data point .

First set of Graphs

For the TCRT sensor the voltage difference between the light and dark surfaces was greatest at 5-15mm heights for all current values tested. In this context sensitivity is defined as the voltage difference between dark and light surface measurements. Varying the resistor value, varies the current, which impacts the sensitivity of the sensor. At lower heights the sensor has the greatest sensitivity at lower currents, whereas when the height is increased the higher current values produce the greater sensitivity.

For the TEKT and ---- sensor the sensitivity at current values of --- and – appears to rise with distance upto a height of --- where is platos and begins to decrease. At a current value of --- it follows a different pattern whereby the sensitivity decreases as the distance decreases. This is likely due to the current being well below the maximum value.

For the ----- add third sensor

Compare all three sensors together

Test 2

The second test was done by measuring the output voltage of the sensor as it passes over the white section to the black to determine the output signal at the boundary. This was done at varying heights, which is shown in graph x. A 100 ohm resistor and 10k was used for TCRT. A 62 ohm resistor and 8k was used for ----. A --- ohm resistor------third sensor. The aim of this experiment is to assess the suitability of each sensor to be used as an analog sensor. For a PID algorithm to be applied we must be able to characterise the transition between the dark and light surfaces so that the buggy can turn proportional to the deviation from the white line.

Graphs

For TCRT at both 5mm and 10mm heights the transition is clear and can be detected across 6-8mm on either side of the white line.

Test 3

Sunlight on the day of the race can significantly effect the sensors ability to differentiate between the white and black surfaces. The wavelength of sunlight passing through the atmosphere can be anywhere between --- to --- nm , reference. Effects of external light was tested by disconnecting the photoemitter and measuring the difference in the output signal when a bright artificial light is directed at the photoemitter Five readings were taken to calculate average and standard error. The difference for the TEKT5400s was 1.803v +-0.287v, for the TCRT500 it was ----,

This method does completely characterise the effect of sunlight because the wavelength of the flashlight is at the ---nm range, where as sunlight covers the – range. Which means it will inherently effect any of the sensors used. The best option to minimise the effect of sunlight in the case would be to use other means such as filters, physical barriers around the sensor board or variable thresholds that can be altered on the day.

Pencil markings on both black and white track had an impact on the output voltage by approximately 400mV. As these are significant deviations, care was taken to ensure that all tests were carried out under the same conditions.

Add greater comparison and conlcusion

Based on these results we decided to use a TCRT500 sensor with 100ohms biasing resistor. Using a resistor value of 62 ohms provided the most sensitivity however due to the high current the resistor became hot when operated for long periods, which is why our final design uses 100 ohms resistor. The TRCT5000 also has a daylight filter (reference datasheet) .

--Mark scheme ---

To properly explain the experimental method, you need a reference to the ESP Handbook and also to provide any extra information required to replicate your results.

- Which line sensors have you chosen and why?
- A table comparing relevant sensor characteristics.
- Is there a reasoning on how you paired your emitter and detector?

- Will your line sensors work in direct sunlight?
- How will it cope with line-breaks etc?
- Provide test results for your chosen sensor to show that you have fully characterised it and to justify your preferred sensor compared with the other options. You may wish to show comparative test results to demonstrate this.
- Show error bar(s) on the graphs. A brief description should be provided to explain how you have justified the magnitude of your error bar(s)

4. Line Sensors Circuit Diagram

Maximising the flexibility of the circuit has been the priority in this design. 5 sensors in a straight line perpendicular to direction of the buggy have been used. Power to the emitter of each sensor is via a darrlington array, the sensors can be switch on or off using a digital signal from the microcontroller. Each sensor module has an output to analog pin and an output to a comparator through a shorting link. The comparator's reference voltage is set by a variable resistor (trimmer) and outputs to a digital pin. The comparator's output is open drain which is why it has a pullup resistor to power. The analog vs digital configuration of each sensor can be altered by either inserting or removing the shorting link and using the corresponding pin to the microcontroller. This versatile design allows us to experiment with different combinations of analog and digital sensors on our real buggy to maximise speed and reliability. It will also allow us to experiment with other non-line analog sensors if we discover more than two or three analogue sensors do not significantly impact performance.

- Full sensor circuit diagrams: Schematic, wiring, and stripboard/PCB-layout diagrams.
- These should include power supply connections.
- Show how your peripherals will interface to the microcontroller(s) (pin number/pin type)
- An explanation of the above.

5. Non Line Sensors

Non-line sensors enable the buggy to monitor the surrounding environment and respond to certain special situations, allowing the buggy to control its speed or detect whether the voltage is at the appropriate level.

Encoder

An encoder installed on the motor shaft can measure the rotational speed of the wheel as the buggy moves forward and use this to calculate the buggy's translational velocity and angular velocity. To get the velocity, the AEAT- 601B-F06[2] quadrature encoder should be mounted on the motor shaft. There are 3-channel outputs with the resolution of 256 CPR (Counts Per Rotation), and STM32 can measure these pulses. The encoder can generate two square wave output signals in channel-A and channel-B. By determining whether Channel A leads Channel B by 90 degrees, the direction of the shaft's rotation can be ascertained. If leading, it rotates clockwise; if lagging, it rotates counterclockwise. Meanwhile, the microcontroller treats the square wave signals from

the left wheel as digital inputs at PB_0 and PB_1, and those from the right wheel as digital inputs at PB_2 and PB_3. Set a timer to generate an interrupt when the rising edge appearing, then through stm32 built-in quadrature Encoder Peripheral TIM2 and TIM5 determine the wheel's angular velocity by calculating the time difference between each interrupt.
The velocity can be calculated as (V is velocity,ω is angular velocity):

$$V\ of\ Wheel = \omega\ of\ Wheel \times Wheel\ Radius$$

$$V\ of\ Buggy = \frac{V\ of\ Wheel\ Right + V\ of\ Wheel\ Left}{2}$$

$$\omega\ of\ Buggy = \frac{\omega\ of\ Wheel\ Right + \omega\ of\ Wheel\ Left}{Wheelbase}$$

$$Turning\ Radius = \frac{Wheelbase}{2} \times \frac{\omega\ of\ Wheel\ Right + \omega\ of\ Wheel\ Left}{\omega\ of\ Wheel\ Right - \omega\ of\ Wheel\ Left}$$

Battery Monitor

Dallas DS2781 is a cell integrated circuit on the Motor Drive Board that can be used to measure the cumulative voltage and the cumulative current of the battery. This sensor sends analogue input to Board through PB_4. Then, STM32 is used to acquire and process its outputs to implement the battery monitor. Its function is to ensure that the battery voltage remains relatively stable.

Bluetooth Module

Bluetooth will utilize a short-range wireless connection for data exchange with the microcontroller STM32. It is characterized by extremely low energy consumption, rendering the voltage loss negligible. The PA_2 on the microcontroller will be used to as an output to send signals, the PA_3 will be used to as an input to receive signals. The buggy will take corresponding actions based on the information received from the Bluetooth module.

Motor Current Sensor

ACPL-C78A-000E [3] is an isolation amplifier optocoupler which engineered for current sensing in electric motors. the device plays a crucial role in monitoring and controlling electric currents in motor applications. A constant PWM input is necessary to calibrate the current sensor. After connecting with stm32, the sensor can measure the current of each motor as an input through feedback looping and ultimately determine different voltages to be allocated to the motors based on this data.

Additional Sensor

A laser sensor, such as the AFBR-S50LV85D [4], might be purchased and can be used to detect obstacles ahead. It will send a digital signal to the STM32 for processing. The advantage of this is the potential reduction in colliding obstacles, preventing unexpected incidents that could prolong the time to complete the race. The downside, however, is the possibility of misidentifying slopes as obstacles which could impede the progress of the buggy and lead to defeat.

6. Control

6.1. how to control the speed & direction of the buggy

If the buggy wants to know where the white line is being located, an infrared sensor is needed. As described in the previous section, when the infrared sensor is powered, the infrared emitting diode emits infrared light, which is reflected into the receiving diode when it hits the surface of the object. Due to the photoelectric effect of semiconductors, the resistance drops when infrared light is received. The resistance is minimum when on the white line and maximum in the black area. From the measured resistance the voltage value can be returned to know where the cart is located on the white line.

For controlling the speed and direction of the buggy, a motor driver board is required to control the voltage input to the motor terminals and a quadrature encoder is provided on each motor to detect the speed of the

wheels and convert it to buggy speed.

The motor driver board has an H-bridge inverter that uses power semiconductors as switches. The motor drive voltage switches back and forth between high and low levels and using PWM (pulse width modulation) the digital output can be converted to an analogue output. The average voltage of the rectangular waveform is the input sent to the motor by the motor driver board, this is called the ‘duty cycle’ and this value is calculated from the pulse width and period using the following formula:

There are two modes, ‘unipolar operation’ and ‘bipolar operation’, where the bipolar operation can be easily changed to forward or reverse by adjusting the duty cycle as it uses all the switches.

Changing the direction of the buggy requires changing the speed of each of the two wheels, for example, if the buggy wants to turn left, it needs to make the speed of the right wheel higher than the left wheel. There are 5 line-sensors used on the buggy, one of them is in the middle and has an analogue output, so if the buggy has been travelling smoothly along the white line, the output voltage of this sensor should be the smallest. This is shown in figure (A). calculating the difference between the output voltages of the fourth and second sensors can tell the exact position of the white line and adjust it according to the difference, which is given in the following equation:

Once the difference is known, the speed of the two wheels is adjusted by P control with the following formula:

Where Kp represents the proportional gain, u(t) is the manipulated variable. This keeps the buggy directly above the white line. If in an uphill or downhill situation, another P controller is needed to keep the buggy speed. The error value is the difference between the speed detected by the speed sensor and the speed that the buggy should reach.

6.2. PID and bang-bang controller

Bang-bang controllers are binary controllers that output a maximum or minimum control signal only in extreme cases of deviation from the setpoint. This type of control is typically found in systems that do not require precise regulation in order for the system to quickly return to setpoint. This control strategy is relatively simple and easy to adjust, but because of the discrete outputs, it is prone to oscillations around the setpoint. As shown in figure (c).

In the case of proportional controllers (P control), the output of the sensor is a continuous value (figure (D)). The main function of the P controller is to reduce the steady state error of the system and the P control is the simplest form of PID control and is easy to adjust. But P controller may lead to overshoot and oscillation in the system. When the proportional gain is set too large, the cart may overshoot when it reaches the set value, causing the cart to oscillate near the white line. After reaching the set value the system still reaching the set value the system still by the possibility of static errors that cannot be completely eliminated.

I-control completely eliminates steady-state errors, improves stability, and reduces oscillations, but has the disadvantage of still having the overshoot and a slow response time. D-control reduces overshoot, but at the same time， D-control is sensitive to noise in the sensor measurements, which can introduce instability.

PID control has comprehensive advantages, but parameter tuning is difficult and does not work with linear systems.

6.3. Algorithm selection and implication

In this project the direction and speed of the buggy is to be controlled. For the control of direction, the best algorithm is 'Proportional Integral (PI) controller'. Compared to simple P controller, PI controller eliminates the possible steady state offset and keeps the cart directly above the white line. The formulae used are as follows:

In addition to this the buggy needs to maintain a constant speed in all situations, so another P-controller is needed to adjust the speed of the wheels according to the measurements of the quadrature encoder to keep the buggy at a constant speed.

The precision and accuracy of the sensor affects the choice of algorithm. Higher accuracy sensors allow the use of more complex algorithms such as PID controllers, whereas lower accuracy sensors are better candidates for simpler algorithms such as PI controllers because of the need to account for potential noise and inaccuracies. The rate at which the sensor processes the data will also affect the choice of algorithm. If the sensor is providing output at a high rate, the algorithm must be able to process this data quickly to maintain a real-time response.

6.4. Sensor implementation

After selecting the PI controller, the selection and placement of sensors need to be designed according to the situation. PI controller needs to get the specific output values of the sensors, so five sensors are placed in a line at the front end of the buggy as shown in Fig. (). The first and fifth sensors output type is digital output, because these two sensors only need to detect 90 degree turns, so they do not need to be set to continuous analogue output. The middle three sensors are set to analogue output, and only two sensors with analogue output are sufficient to measure the position of the white line. The middlemost sensor is used to detect whether the white line is in the middle of the buggy, because the output voltage is smallest when the sensor is directly above the white line.

Applying the results of the controller to the buggy requires quadrature encoders to detect the rotational speed of each wheel. A quadrature encoder is installed separately on each motor. All the quadrature encoders need to be connected to a microcontroller in order the system to respond quickly and implement the algorithm. Since bipolar operation is chosen, changing the PWM signal changes the direction of travel of the buggy. The pin value of the motor driver board connected to the microcontroller needs to be changed depending on the operation selected.

6.5. implement the control algorithm

In this project, all the algorithms for the buggy will be programmed through Mbed and implemented on the microcontroller STM32F401RE. Mbed will control the PWM signal output to control the speed of the buggy, while Mbed will control the blinking of the LEDs in order to detect the data from the sensors and reduce the noise.

6.6 solution for sunlight & line breaks.

For the high intensity of sunlight, the approach on the physical plane is to either use an infrared filter on the buggy to filter the ambient light, or add a black sheet of shade around the sensor. Another approach is to adjust the threshold based on the ambient light, which can be automatically adjusted after using differential measurements. For a broken line, set the three sensors in the middle of the trolley so that when the middle sensor fails to detect the route, the trolley will turn to the side where the route is detected by one of the sensors on either side..

7. Hardware Overview

7.1. Chassis Design & Drawings

The design of the buggy was made using SolidWorks which helped to plan out the layout of the buggy and ensure every component fits in the final design.

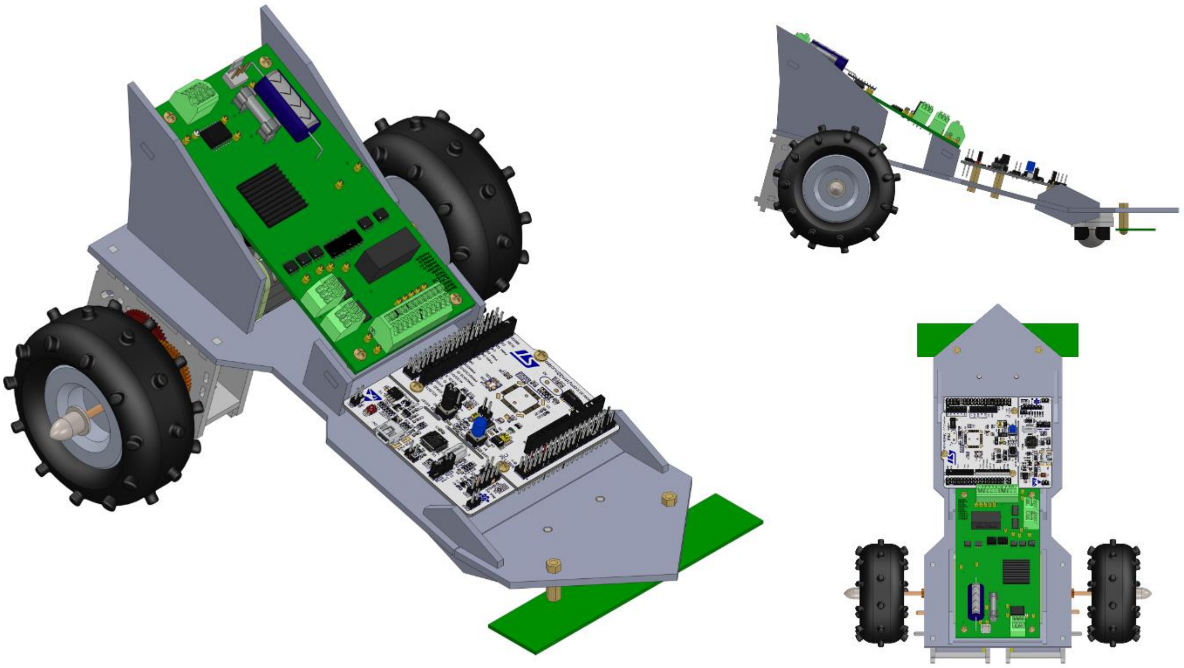


Figure 7.1.1: 3D buggy CAD design from different views

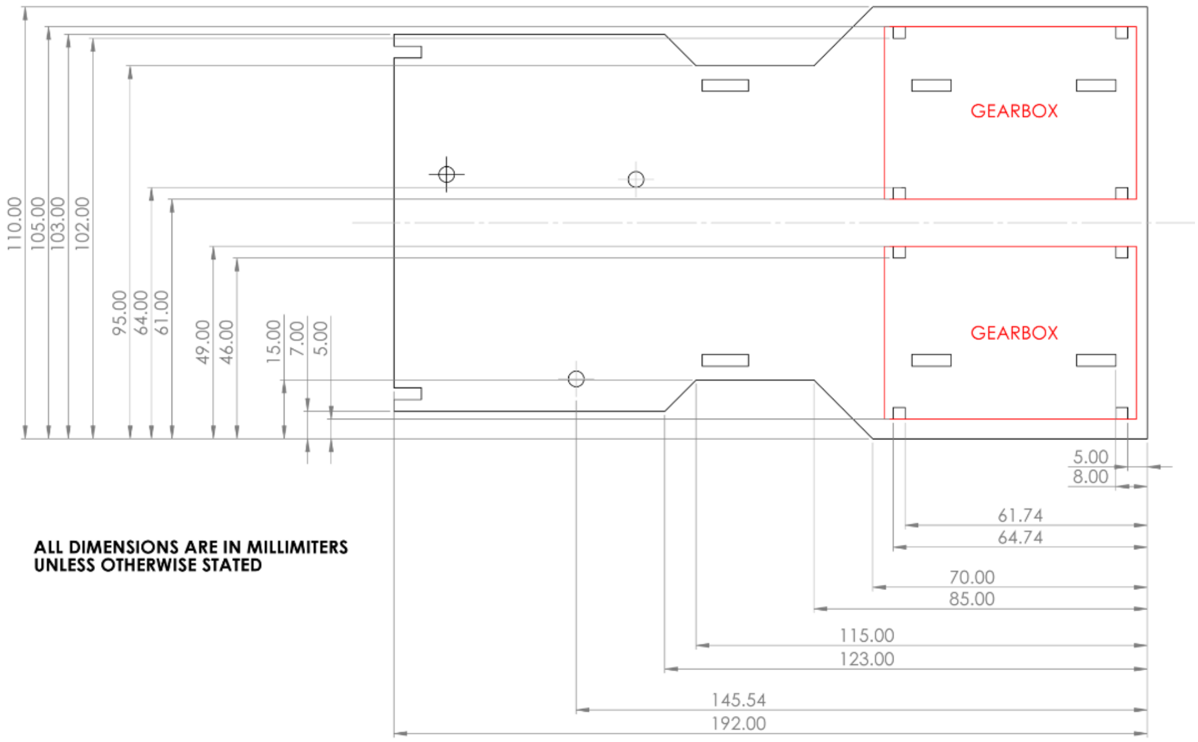


Figure 7.1.2: Main chassis 2D dimensions

7.2. Chassis Design Choice

The buggy design prioritizes to achieve the lowest weight possible by incorporating a minimal design with just a single chassis layer. This is crucial to lower the inertia and frictional force so it can move at a higher speed. In addition, the main chassis of the buggy was angled at a 15° degree downwards to lower the centre of gravity and therefore increase stability.

In terms of layout, extra effort was made to ensure the wires connecting each board as short as possible so that the wires are organised and avoid the long wires getting tangled. This is done by ensuring each connection point are close to each other. For instance, the microcontroller board has the most connections so, it was placed in the middle. The boards are exposed from the top view which allows for easy access and troubleshooting.

The option of front wheel drive layout was also considered. However, this means that the sensor will be at the back of the buggy which will cause the buggy to be in front of the line that is being detected. Thus, the buggy might hit a wall before detecting a turn on the sensor. Plus, there was not much of an advantage of a front wheel drive layout, so rear wheel layout was chosen.

Furthermore, rotational inertia is minimized by placing the heaviest components, such as the battery pack, near the centre of rotation, located at the midpoint between the wheels. This design choice enables the buggy to turn with reduced force.

Moreover, positioning the line sensors farther away from the centre of rotation increases the responsiveness to the change of angle detected by the sensor. This is because, a larger distance from the centre of rotation will result in larger distance travelled for a given angle of rotation. Therefore, the line sensors are placed in front of the front wheel instead of behind it to maximize the distance from the centre of rotation.

Finally, the spacing distance between the rear wheels is a critical consideration. A greater distance is advantageous, as it results in less turning for a given wheel rotation angle, allowing for more precise angle correction. However, it is crucial to strike a balance, as making the buggy too wide presents challenges in navigating tight corners. Hence, a careful equilibrium was sought in determining the optimal rear wheel spacing to ensure both precision and manoeuvrability.

7.3. Chassis Material Choice

With the design of chassis completed, the other crucial aspect of the chassis is the material choice. Choosing the best material for the chassis is crucial as each materials have different properties that would affect the performance of the buggy. Choosing the right material is also critical to the ensure the chassis does not fail and break during a race.

Therefore, 4 sheet materials were considered, acetyl, Glass-Reinforced Laminate (GLARE), aluminium, mild steel. The main characteristics that were examined for each material are flexural strength, ultimate tensile stress, young’s modulus, density, price, and ease of manufacture.

Young’s modulus describes the material’s ability to deform elastically under load. Ultimate tensile stress is the maximum stress that a material can withstand while being pulled before the specimen’s cross-section starts to significantly contract. Flexural Strength measures the maximum stress a material can withstand while being bent. In the context of our buggy's use case, a higher value for these characteristics is preferable.

The values shown in the comparison figure below are an approximation and may vary depending on the material's grade, processing methods and supplier. Therefore, they are used only for rough comparison between materials. The values for density in the graph have been multiplied by 10 to better fit the scale.

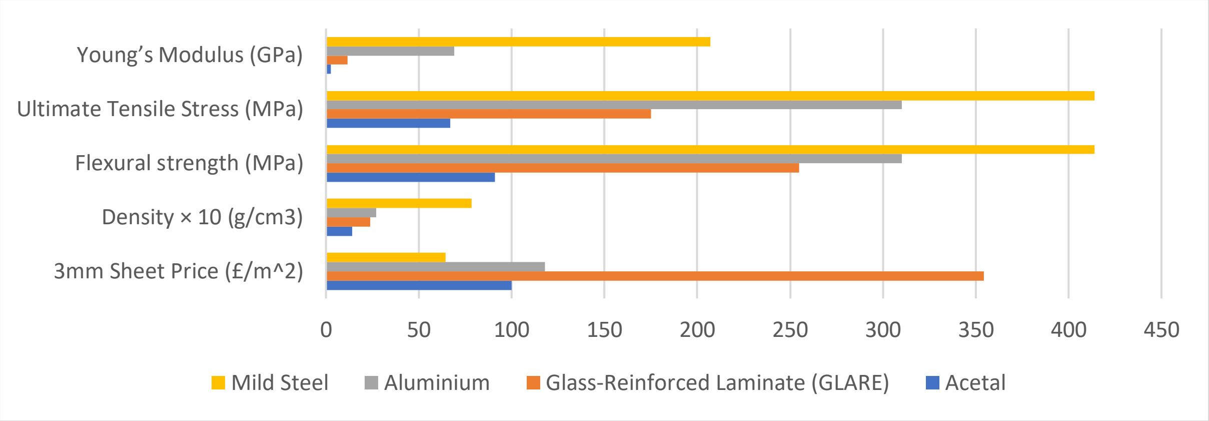


Figure 7.3.1: Material Characteristics Comparison

GLARE is a material commonly used in aerospace field, composed of layers of metal and glass fibre. However, the inherent risks associated with handling glass fibre, which can be harmful to eyes, skin, and lungs, pose an additional concern in manufacturing. Moreover, the material is at least 3 times more expensive than the other alternatives. Considering these factors, the additional risk and higher cost of GLARE are not justified by its other characteristics, leading to the decision not to use this material.

Mild steel, while the strongest and most cost-effective option, is also the heaviest, weighing at least twice as much as alternatives. Considering the density value, the chassis alone would weigh 436 g, nearly half the expected buggy weight calculated in DR1 (1141 g). Since weight significantly impacts the buggy's maximum speed, mild steel was not chosen for the final material.

Aluminium offers a lighter alternative to mild albeit at a higher cost and slightly reduced strength. Similar to mild steel, manufacturing the buggy with aluminium requires a high-power metal laser cutter. This introduces additional costs and complexity to the manufacturing process. Moreover, Aluminium's good electrical conductivity poses a risk of unintended electrical shorts that could potentially damage the electronic components. Considering these added risks, complexities, and costs, aluminium was not selected.

In contrast to the metal materials, acetal can be cut using a relatively low power CO2 laser cutter. It is also cheaper than aluminium. acetal can be cut using a lower-power CO2 laser cutter, making it a cost-effective choice. Not only is acetal more affordable than aluminium, but it is also easy to handle and non-conductive. Its standout feature is its low density, being the lightest with a weight 41% less than aluminium. Given that density significantly influences performance by affecting frictional force and maximum speed, the decision to choose acetal was primarily influenced by its lightweight characteristic.

Nevertheless, acetal comes with the drawback of being the weakest material among the options. To address this limitation, one approach is to use a thicker acetal sheet. To ensure the strength of the acetal is sufficient for the normal load the buggy will encounter, a Finite Element Analysis (FEA) simulation was conducted on the main chassis. The simulation explored various thicknesses of acetal sheets to determine the optimal thickness, taking into consideration that a higher thickness also increases the mass. In simulating the worst-case scenario, a concentrated force equivalent to the expected weight of the buggy (11.19 N) was applied at the weakest point on the chassis, located between the gearbox mount point and the front mount point.

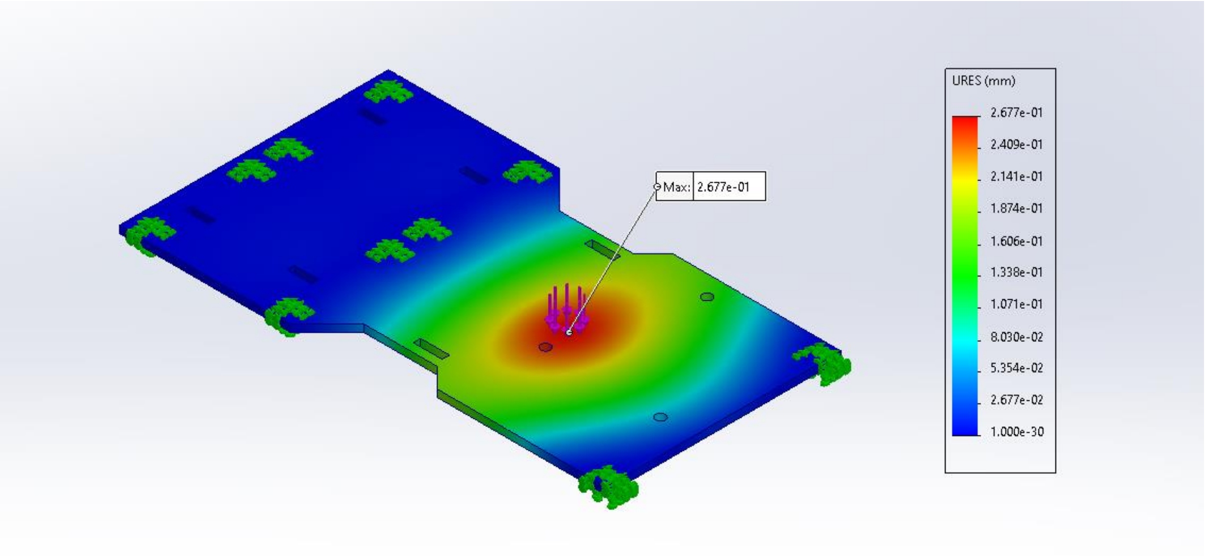


Figure 7.3.2: SolidWorks FEA simulation on 3mm acetal chassis

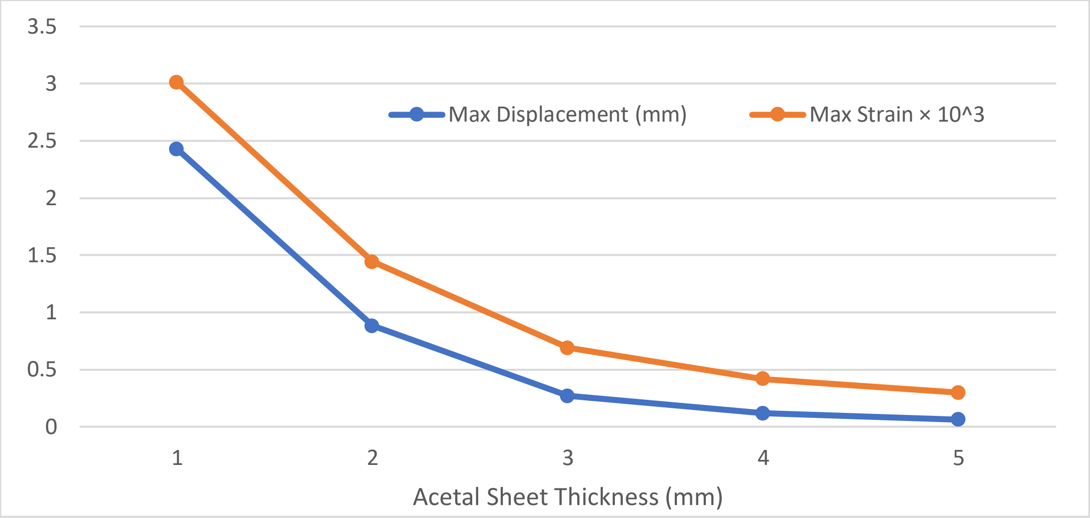


Figure 7.3.3: FEA simulation results of different thickness of acetal chassis

Due to acetal's low flexural strength, it's crucial to prevent excessive chassis bending, which could lead to breakage. The extent of bending is assessed by measuring the maximum displacement of a point from its initial position before applying a load. Maximum strain gauges the highest deformation at any point due to the applied force. A lower value for each metric indicates a better chassis performance.

Analysis results indicate that increasing sheet thickness yields diminishing returns, plateauing at 3 mm. At 2 mm, the maximum displacement reaches an unsafe 0.881 mm. Therefore, to balance low displacement without unnecessary thickness and added weight, 3 mm was selected as the final thickness for the acetal sheet.

8. Summary

Summary lol

9. References

[1] N. F. Abdullah, A. A. Wahab, and M. A. H. Rasid, “Study of temperature rise in small brushed DC motor components under different speed,” *IOP Conference Series: Materials Science and Engineering*, vol. 469, no. 1, p. 012098, Jan. 2019, doi: 10.1088/1757-899X/469/1/012098.

[2] Broadcom-Limited Accessed: Nov. 26, 2023. [Online]. Available:AEAT-601B-F06 Broadcom / Avago | Mouser United Kingdom

[3] Dallas semiconductor. *DS2781, 1-Cell or 2-Cell Stand-alone, Fuel Gauge IC rev. 062708*. (2020). Accessed: Nov. 27, 2023. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS2781.pdf

[4] Broadcom-Limited Accessed: Nov. 26, 2023. [Online]. Available:ACPL-C78A-000E Broadcom / Avago | Mouser United Kingdom

[5] Broadcom-Limited Accessed: Nov. 26, 2023. [Online]. Available:AFBR-S50LV85D Broadcom / Avago | Mouser United Kingdom

[6]