MANCHESTER 1824
The University of Manchester

Department of Electrical and Electronic Engineering

Embedded Systems Project 2022-23

DESIGN REPORT #2

Title: Technical Characterisation

Group Number: 12

Group members:	ID Number	I confirm that this is the group's own work.
Andrew Slater	10908845	✔
Kynard Tey Kai Min	10915992	✔
Tingxuan Wang	11174443	✔
Aqil Kamaruddin	10922046	✔
Fotis Kougionas	10727014	✔

Tutor: Andrew Forsyth

Date: 1/12/2022

Contents

1. Introduction

This report describes the various design considerations and component analysis used to create an effective response system for the buggy to react to various stimuli present on the track. This aligns with the objectives of the project, which is to design an autonomous buggy that can overcome obstacles, as without an effective response system, the buggy will not move autonomously. To accomplish this, software is implemented on the microcontroller of the buggy. The software takes inputs from line sensors, performs a set of arithmetic operations and then outputs into the motor driver board which will then drive the buggy motors. There are various obstacles present on the track such as line breaks and turning lines, which needs to be addressed for the buggy to be successful, thus line-sensors characterisation and analysis is necessary. There is a need to know how the sensors would behave under normal conditions so that the program may identify when the buggy meets said obstacle. Additional non-line sensors are also necessary to monitor other important parameters that play a part in the software, such as the encoders to monitor the speed of the wheel, which links to controlling the forward and steering motion of the buggy. It is also crucial that a suitable algorithm should be used to create an effective and efficient response system, as an ideal algorithm would allow the buggy to closely follow the line. The behaviour of the motor drive board is also studied so that the forward and backward motion of the buggy can be controlled. Additionally, a chassis design of the buggy is to be made to provide an overview of the buggy's body and to discover additional improvements that can be made for the buggy, to be able to go through the track in a smoother manner. The design of the sensor PCB also plays a crucial part to the sensing operation of the buggy, as different geometry and orientation could affect the input result significantly, thus careful examinations are taken to design an ideal sensor PCB that performs as well as possible.

2. Software

2.1 Functional Summary and Constraints

The system used in the buggy will maintain the correct trajectory for the buggy to follow the track’s white line. It will repeatedly read and process values given by the line sensors to allow the buggy motors to be controlled accordingly.

When the system is switched on, the microcontroller will send an electrical signal to the ULN2003 Darlington Buffer, which sends a signal to each sensor in the sensor array, turning on the LEDs. The microcontroller will read the voltage from each sensor’s output resistor. These voltages are then processed by the microcontroller to calculate the position of the white line and the correct output signal is sent to the motor driver which is then used to alter the speed of the wheels. The buggy’s direction is then changed, realigning it with the white line. The rotation speed is measured by the encoders and sent to the system, allowing it to maintain a steady speed or increase/reduce the speed to the required level.

When the buggy reaches a break in the track it will calculate its previous trajectory using previous sensor values and maintain that change in trajectory until it reaches the white line again. If the buggy needs to slow down it calculates its current speed and reverses the current to the motors, reducing the speed of the wheels. Similarly, the voltage can be increased when moving up the sloped area of the track to increase torque.

The main software constraints that could arise are the memory of the STM32F401RE microcontroller and the timing used in the steering system. According to the STM32F401RE datasheet [9], the microcontroller has 512 kB in flash memory and 96 kB of SRAM. This should be sufficient for the code used in the ESP project and therefore should not impact the way the code is written. The timing is a more significant constraint as the timing between reading sensors and outputting the correct signal must be short so the buggy reacts to changes in the track's white line. If the timing is too slow the buggy may react incorrectly, reducing its maximum speed and increasing the likelihood that it fails to follow the line. Due to this, there should be a change in motor speed within 50 ms of receiving sensor data to allow the buggy to react fast enough to change direction correctly, whilst also leaving a reasonable timeframe for the sensor's values to be processed.

2.2 Context Diagram

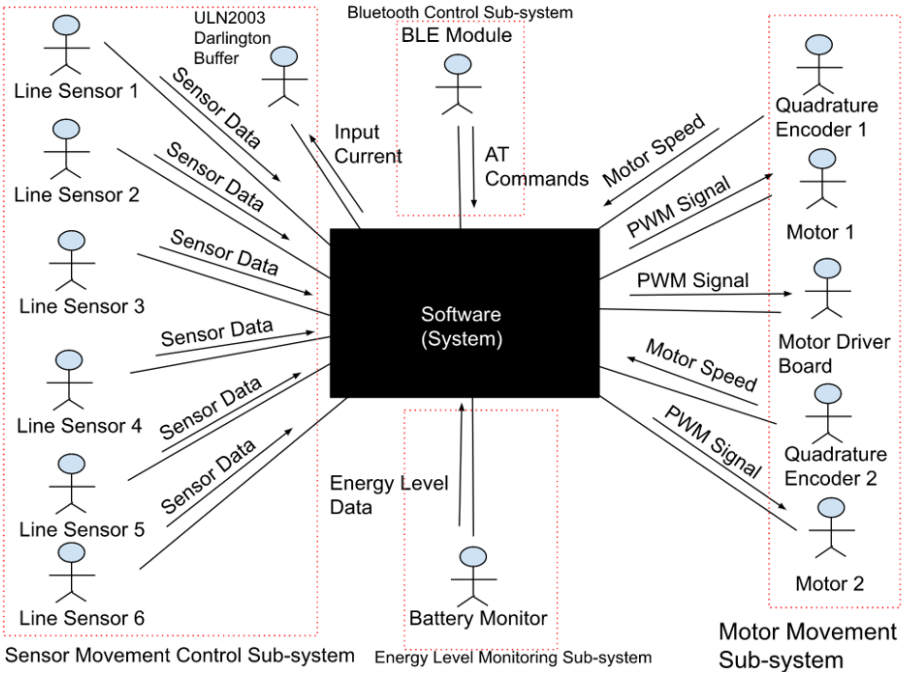


Figure 2.1: Context diagram of software system

2.3 Table of Messages

Table 2.1: All messages included in the context diagram sent/received by the system

Message Name	Data Content	Source	Destination	Arrival Pattern	Size
Sensor Data	The voltage value of the line sensor	TCRT5000 Line Sensors	System	Continuous	0.25-2.5 V at about 10 mA input current.

Energy Level Data	The energy level of the batteries	Battery Monitor	System	Continuous	-
Motor Speed	The speed of the wheels	Quadrature Encoders	System	Periodic	-
PWM Signal	The PWM signal going to the motors	System	Motors	Periodic	0-1.4 A
AT Commands	Various commands from the BLE module	BLE Module	System	Aperiodic	Up to 31 bytes
Input Current	Current into Darlington Buffer to power LEDs	Motor Driver Board	Darlington Buffer	Continuous	10 mA per sensor

2.4 Case Diagram

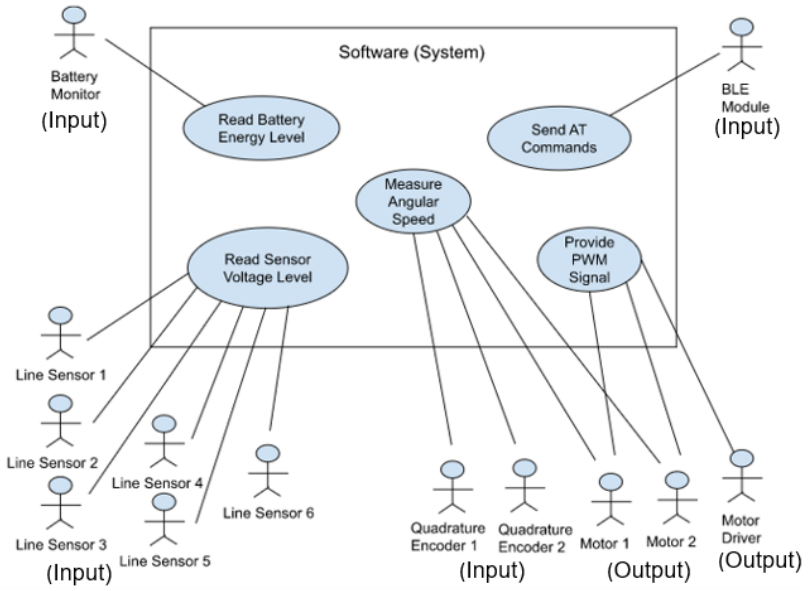


Figure 2.2: Case diagram of software system

2.5 Case Descriptions

- Read Battery Energy Level:

Externals: Batteries, Battery Monitor.

Description: The batteries are connected to the battery monitor and the system, with the monitor also connected to both. The monitor measures the energy level of the batteries and then transmits the result to the system.

Preconditions: Batteries are powering the monitor and system.

Postconditions: System has received power level reading.

- Read Sensor Voltage Level:

Externals: Line sensor array (6 line sensors).

Description: Each line sensor is connected to the system, incorporating a resistor for both the photodiode and phototransistor. The voltage over the phototransistor's resistor is measured by the system to be able to decide if the line sensor is detecting the white line or black track.

Preconditions: The system is powered and sensors are receiving power from the system.

Postconditions: System has received 6 sensor voltage readings.

- Provide PWM Signals:

Externals: Motor driver, 2 motors.

Description: The system outputs a PWM signal to control the motors to a motor driver which distributes the signal, including the correct amount of current, to each motor. This allows the motors to change speed and direction.

Preconditions: The system is powered and the line sensor voltage readings have been received.

Postconditions: The motors rotate in the desired direction at the desired speed.

- Measure Angular Speed:

Externals: 2 motors, motor encoders.

Description: The rotating motors' angular speeds are measured by the motor encoders, which then transmit these values back to the system to be processed further.

Preconditions: The system is powered and the motors are rotating.

Postconditions: The system is aware of the motor's angular speed.

- Send AT Commands:

Externals: BLE Module

Description: The BLE module receives AT commands and transmits them to the system to be processed by the microcontroller.

Preconditions: The buggy is turned on.

Postconditions: The buggy has reacted to the command sent to the BLE module.

2.6 Object Specifications

Table 2.2: Read Battery Energy Level

Property	Name	Quantity	Parameters/Declarations	Description
Data	energyLevel	1	Float value	Holds the value for the energy level in the batteries.
Function	measureEnergyLevel()	1	energyLevel	Measures the batteries' energy level.
Function	getEnergyLevel()	1	Void	Returns value energyLevel.

Table 2.3: Read Sensor Voltage Level

Property	Name	Quantity	Parameters/Declarations	Description
Data	sensorVoltage	6	6 float values	Holds the values for the line sensors in an array.
Function	measureSensorVoltages()	1	Void	Measures all line sensor voltages.
Function	getEnergyLevel()	1	Void	Measures the energy level in the batteries and returns it as energyLevel.

Table 2.4: Provide PWM Signals

Property	Name	Quantity	Parameters/ Declarations	Description
Data	motorFrequency	2	2 float values	Holds the frequency values for the motors.
Data	onDuty	2	Float value	Holds the value for the on-duty part of the duty cycle for both motors.
Data	offDuty	2	Float value	Holds the value for the off-duty part of the duty cycle for both motors.
Function	calculatePWM()	1	sensorVoltage	Calculates the on-duty and off-duty for the PWM signal.
Function	getPWM()	1	Void	Returns the values calculated by calculatePWM().
Function	outputPWM()	1	motorFrequen	Outputs the PWM signal to the

			cy, onDuty, offDuty, output pins	motor driver/motors.

Table 2.5: Measure Angular Speed

Property	Name	Quantity	Parameters/Declarations	Description
Data	angularSpeed[2]	2	2 float values	Holds the value for each motors' angular speed.
Function	measureMotorSpeed()	1	angularSpeed	Measures each motor's angular speed.
Function	getEnergyLevel()	1	Void	Returns each motor's angular speed.

Table 2.6: Send AT Commands

Property	Name	Quantity	Parameters/Declarations	Description
Data	receivedCommand	1	2 float values	Hold the information contained in the AT command.
Function	getCommand()	1	Void	Gets the command received by the BLE module.

3. Line Sensor Characterisation

The choice of sensor is a key part of this project, as it is the basis of the software execution and control process, used to make sure the buggy follows the white line correctly. Therefore, the experiment took place to figure out the characteristics of the TCRT5000 sensor, along with several additional detectors and emitters which were given to build some sensor pairs to compare to the TCRT5000.

3.1 Experimental method

Before the lab, resistors needed to be in series with the LED and transistor of the TRT5000 and had been calculated by using equations:

$$R_{input} = \frac{V_s - V_{LED}}{I_f} = \frac{5-1.1}{0.010} = 390\ \Omega \quad (3.1)$$

$$R_{output} = \frac{V_s - V_{CE}}{I_c} = \frac{5-1}{0.001} = 4000\ \Omega \quad (3.2)$$

Where the V_s is source voltage, V_{LED} is voltage across the LED, I_f is forward current, V_{CE} is voltage across the transistor and I_c is collector current.

The reason the forward current is chosen as 10mA is that the current transfer ratio has almost reached its maximum and other relevant values are in an acceptable

range. The actual resistor used in the sensor lab is 330Ω for the emitter and 4.7k Ω for the transistor, which are the closest value resistors we have available.

During the sensor lab, a TCRT5000 sensor was soldered on a small PCB transfer into an 8-pin DIL plug, matching with the 8-pin DIL socket soldered on the stripboard. Only 4 pins in the middle are working since the TCRT5000 is a 4-pin component. And two additional line sensors were built using discrete components. A testing circuit was created as the figure shows below.

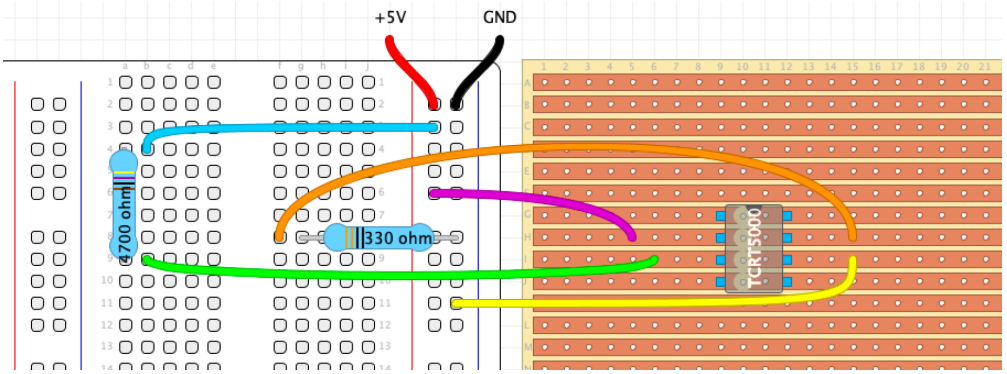


Figure 3.1: Configuration of the breadboard and stripboard during the experiment

The first experiment in the lab is the dark current of the phototransistor which is expected to be small (in nA). The track of the final race is composed of black and white areas, and the second set of experiments can give the sensitivity of sensors varied with height. The third section is to plot the line spread function which shows how the sensor examines the white line. All the measurements help with defining an optimum separation distance between each sensor and height of the sensor PCB.

3.2 Experimental results

More detectors and emitters are provided and can be matched as a combination. The table below summarises a few important parameters of these devices.

Table 3.1: Summarising data of devices from datasheets

Device	Wavelength peak (nm)	Wavelength Range (nm)
TCRT5000	950	
Detector		
BPW17N	825	620 to 960
SFH203P	880	400 to 1100
TEKT5400S	920	850 to 980
Emitting Diode		
TSHA4401	875	-
TSHG6400	850	-

From the table above, two combinations can be made according to the wavelength peak and range [10] [11] [12] [13] [14] [15], one is the TEKT5400S paired with the TSHA4401, the other is the TEKT5400S paired with the TSHG6400. The other detectors have not been used as they may be affected by visible light, whilst the TEKT5400S detector has a sun light filter. To deal with the line-break issue, it can be easily fixed because all the sensors work in an area where a small gap can be

ignored. a good arrangement of the line sensor board can guarantee the buggy will overcome these issues.

3.3 Sensor Lab Results

3.3.1 Dark Current and Background Illumination

Dark current is the current through the phototransistor when the LED is turned off and the phototransistor is blocked off from any light. This current is measured by taking the voltage across the resistor and calculating the current, the TCRT5000's being the lowest. The voltage across the transistor at this time is the offset voltage and may require an amplifier to subtract the offset voltage before the microcontroller measures it. The background illumination current is the current coming from the phototransistor with the LED off when a light source is held 1 m away from the phototransistor. This current is significant as it shows how the sensor pair is affected by other light sources which may exist during track day. For our experiment a phone torch was used.

Table 3.2: Dark Current and Background Illumination Results

Sensor Pair	Dark Current (nA)	Transistor Voltage (mV)	Maximum Current (nA)
TCRT5000	21.2	100.0	63.8
TSHA4401 + TEKT5400S	21.2	100.4	42.4
TSHG6400 + TEKT5400S	106.4	115.3	21.2

3.3.2 Variation with Height

Each sensor was tested at a range of heights, with the sensor pair being measured for both the black part of the test rig's test track and its white part. For each sensor pair the voltage over the phototransistor's connected resistor was measured, providing graphs relating the voltage with the distance from the ground for both coloured surfaces. Vertical error bars are not used because the noise from the measured voltage is negligible. Horizontal error bars have not been included as they are imperceivable on the graph.

TCRT5000:

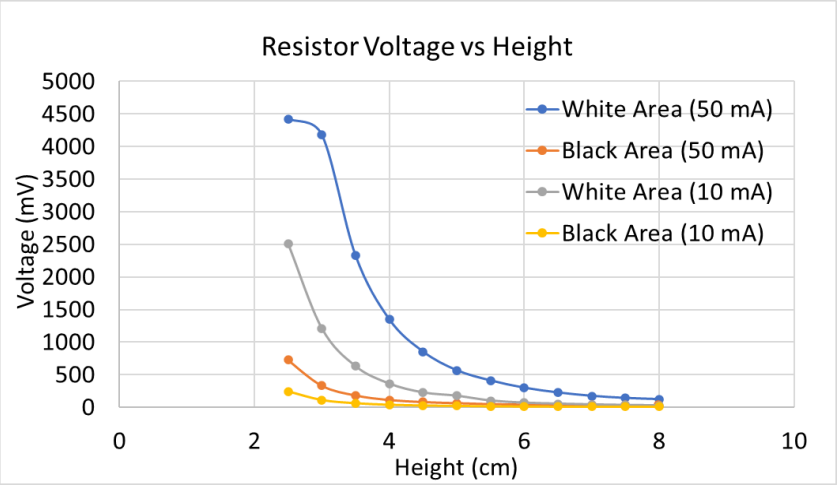


Figure 3.2: Voltage variation with height

As shown in Figure 3.2, the voltage over the phototransistor's resistor increases as the sensor gets closer to the track. This is due to the phototransistor passing more current as more light enters it. By having the transistor at a lower height, more of the light emitted by the photodiode is received by the phototransistor so more current flows through it, causing the voltage across the transistor to decrease. As the resistor is connected in series with the phototransistor, its voltage increases as the phototransistor voltage decreases, creating the group of curves shown above. The voltage does not reduce significantly from 5 cm to 8 cm for the black and white track sections, however there is a relatively large difference between the white and black area due to the white area reflecting more light back into the sensor. Larger currents produce a larger voltage over the resistor due to the increase in brightness of the photodiode and therefore increases the current passing through the transistor.

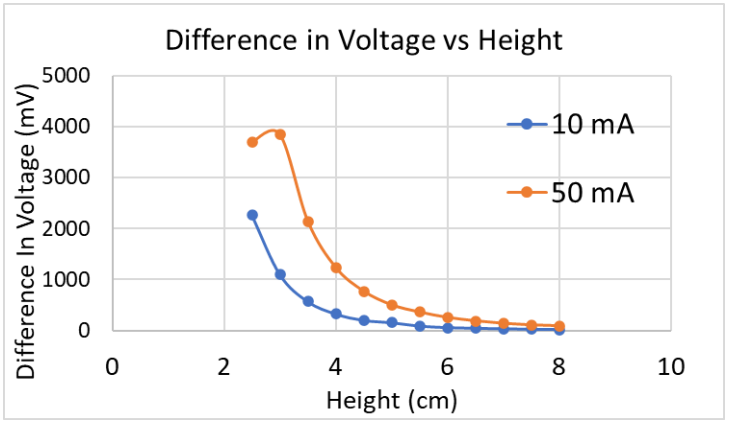


Figure 3.3: Sensitivity graph (difference in voltage vs height)

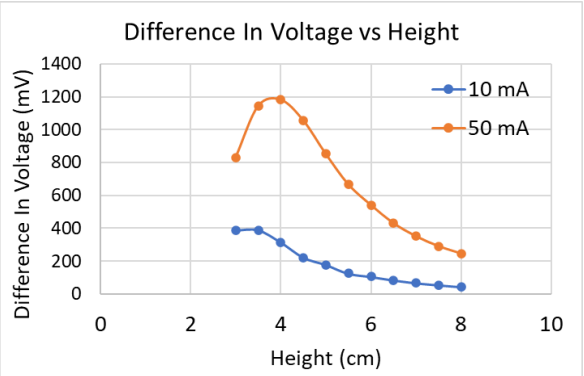
Figure 3.3 shows the difference in voltage vs distance between the white and black area, for both 10 mA input current and 50 mA input current - it is an essential characteristic of the sensor. This essential difference is the sensitivity of the sensor - a larger sensitivity is ideal for the buggy as it provides a clearer distinction between the two parts of the track to the microcontroller, allowing for more accurate motor speed adjustments and line following as it is easier for the microcontroller to classify each sensor as being above the white line or black track. At 10 mA at a distance of 0.03 m there is a difference of 2266 mV, whereas at 50 mA at the same distance the voltage difference is 3846 mV, indicating that a higher input current provides a higher sensitivity. In addition, it is clearly shown that the sensitivity is largest at a distance of around 3 cm and reduces as the distance from the track increases, however the sensitivity starts to decrease if the sensor is too close to the track. This relates to Figure 3.2 as the difference between the black and white area voltages is increased by the higher input current, meaning a higher current may be more ideal for the buggy, if the sensitivity increase is worth the increase in power consumption.

Comparative Sensitivity Results:

Table 3.3: Comparative Sensor Results at 10 mA Input Current

Sensor Pair	Sensitivity at 3 cm (mV)	Sensitivity at 5.5 cm (mV)	Sensitivity at 8 cm (mV)
TCRT5000	1100	361	19.4

TSHA4401-TEKT5400S	387	125	43.8
TSHG6400-TEKT5400S	667	190	54.9



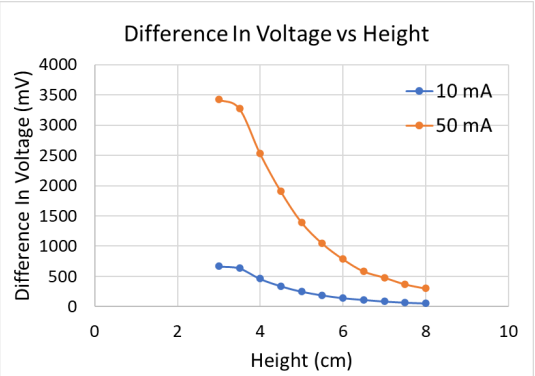


Figure 3.4: Sensitivity graph (TSHA4401) Figure 3.5: Sensitivity graph (TSHG6400)

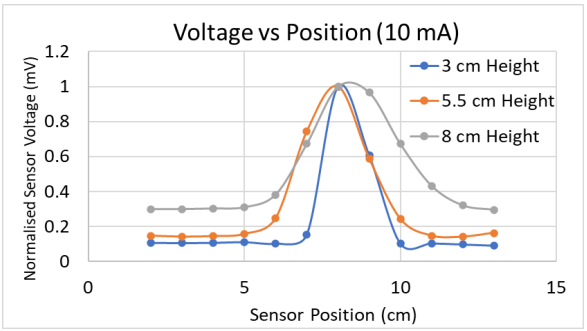
Corresponding with the previous figure, the difference in voltage for the other two sensors are lower overall in comparison with the TCRT5000, indicating that the TCRT5000 will be more reliable during technical tests and the race day as there is a decreased chance of the microcontroller mistaking the white line as part of the black track. It demonstrates that the TCRT5000 has a larger sensitivity to the white line.

3.3.3 Line Spread Function

In this part of the characterisation stage, each sensor was tested at three different heights (3 cm, 5.5 cm and 8 cm) for their behaviour in the line spread function test. A black and white test track was passed under each sensor, providing a clear plot of the sensor's reaction to the white line and the severity in distinction between the white line and the rest of the track. This provides data which can be used to decide which sensor is most sensitive to change between the black track and white line, and conversely which sensor blurs the line with the track the most.

All data has been normalised to the same maximum value as a higher maximum voltage does not mean the sensor will blur the white line less - by scaling the figures to the same maximum voltage, each plot will have a more distinct and easier to analyse curve, abstracting the differences in maximum voltage which are not necessary in this stage of characterisation.

TCRT5000:



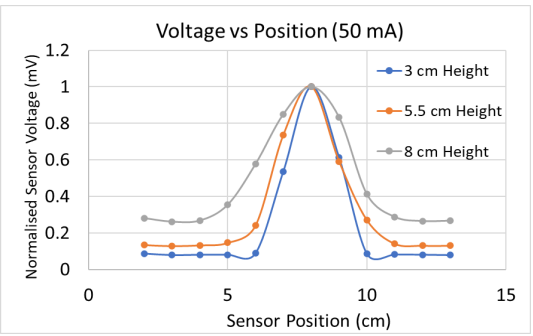


Figure 3.6: Line spread function at 10 mA Figure 3.7: Line spread function at 50 mA

It is made clear that the TCRT5000's voltage increases rapidly as it passes over the white line and then decreases rapidly as it reaches the black track again. The change in voltage is large over a small distance, indicating the sensor reacts greatly towards the change in reflecting surface. At larger heights the line is blurred with the track more for both 10 mA and 50 mA input currents, whilst at smaller heights the amount of input current did not affect the blurring of the line as much.

Comparative Results:

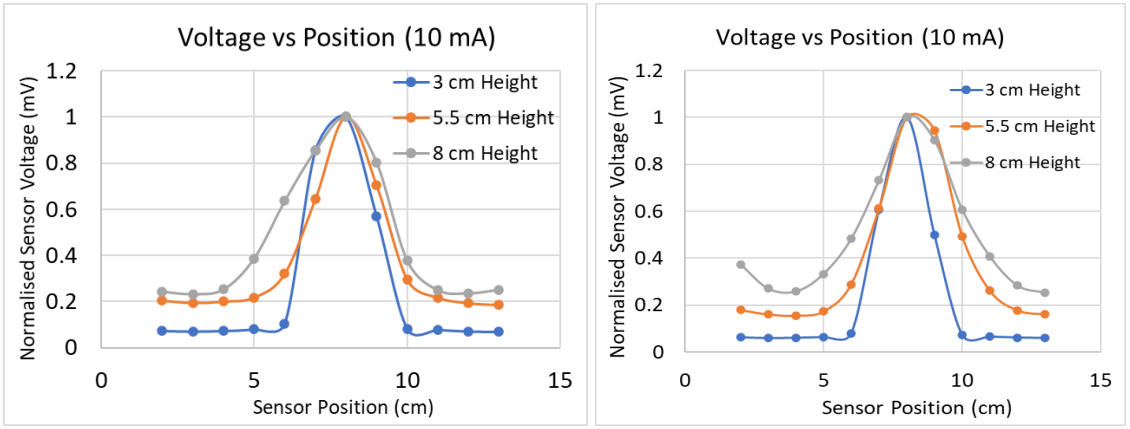


Figure 3.8: Spread Function (TSHA4401) Figure 3.9: Spread Function (TSHG6400)

The trend in the line remains the same as the TCRT5000 for the other two sensor pairs, however there is less distinction made between the white line and black track. As before, the current had a greater effect on clarity of the white line at larger heights. The TCRT5000 has a sharper transition between the black track and white line at a height of 3 cm than the other two sensor pairs as the distance between the two points at which the voltage starts to increase is smaller than the other two sensor pairs, demonstrating that the TCRT5000 will not blur the white line with the black track as much as the other sensor pairs. Although it blurs the line with the track more than the other sensor pairs at larger heights, the sensor will be operated at 3 cm from the ground, making the TCRT5000 the ideal choice of sensor.

As shown through this characterisation, the TCRT5000 is still the best choice compared with the other two sensor pairs soldered during the lab. It is not only most sensitive to the white line, but also easiest to use as it requires less assembly, reducing the chance of part failure. It blurs the line less at the ideal height, making it superior in every significant characteristic to the other sensors.

3.4 Sensors reaction to direct sunlight and line-breaks

In the presence of sunlight, the sensors would likely respond with a high voltage if light was reflected into it. This means that the sensor would return a high voltage as if the line was under the buggy regardless of where the line actually is. This poses a problem, as the sensors are practically rendered useless. However, it can be solved via several conditional statements in the software section.

In the presence of line-breaks, the sensors would return with a low voltage, as they do not detect the lines. This may confuse the buggy; however, it could be solved in a similar fashion to sunlight interference via conditional statements in software.

4. Circuit Diagram

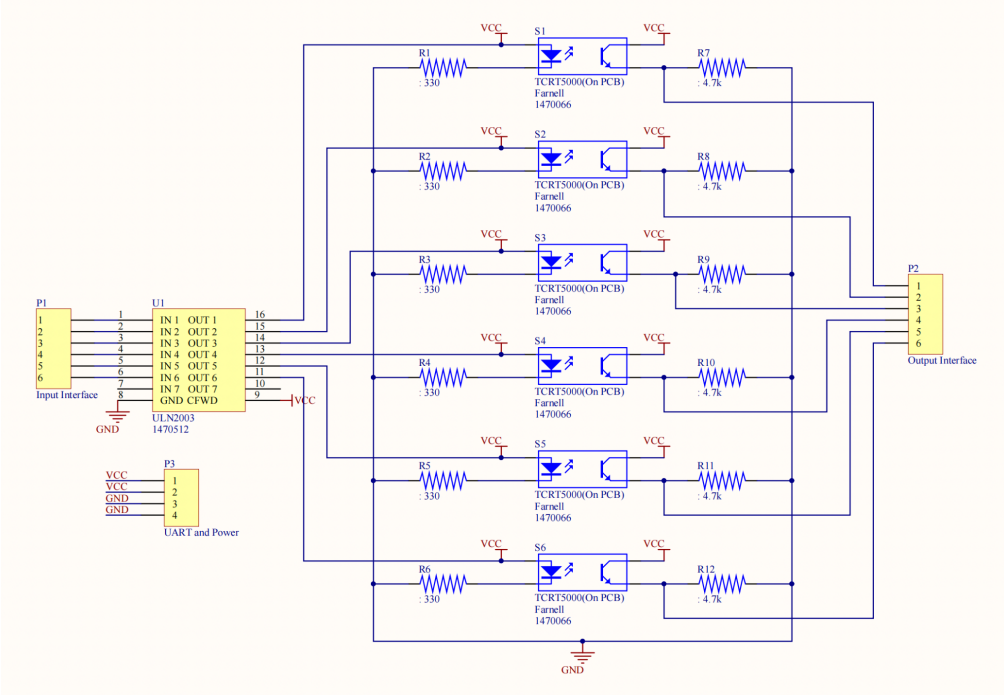


Figure 4.1: Schematic diagram of line sensor board

For the schematic diagram, the sensor board and microcontroller are connected through three ports P1, P2 and P3, where P1 conveys a digital signal from the microcontroller to the LED of TCRT5000, P2 transmits the output of the phototransistor to the microcontroller and P3 is the power supply for the sensor board. U1 is a ULN2003 Darlington Array which is used to amplify the signal of the digital I/O of the microcontroller [6] so that it can reach the required current to switch the LED of each TCRT5000 on. The value of the two resistors is calculated to ensure a satisfactory sensitivity is achieved and all the components are not damaged.

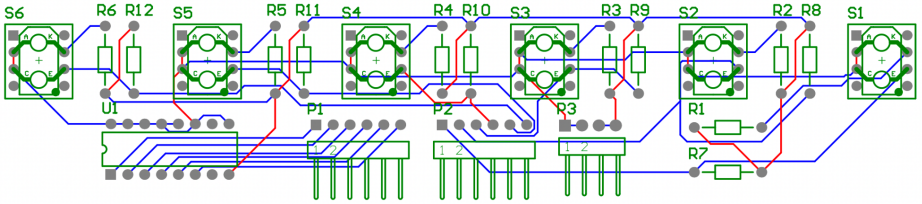


Figure 4.2: PCB-layout diagram

For the PCB layout, the route is automatically arranged by the Altium Designer based on the schematic diagram. And all six sensors are aligned in a row with a centimetre gap between each other.

For the wiring diagram, it shows how the microcontroller connects with other main components, furthermore it is divided into two main parts: the line sensor board and motor board. The input of the line sensor is a type of digital I/O which is supported by

all pins on the microcontroller [7]. The output of the sensor is an analog signal which has to be hooked up with pins that support analog I/O, shown on the diagram above. And the power supply of the line sensor board connects with the +5V and GND pin on the Nucleo Board.

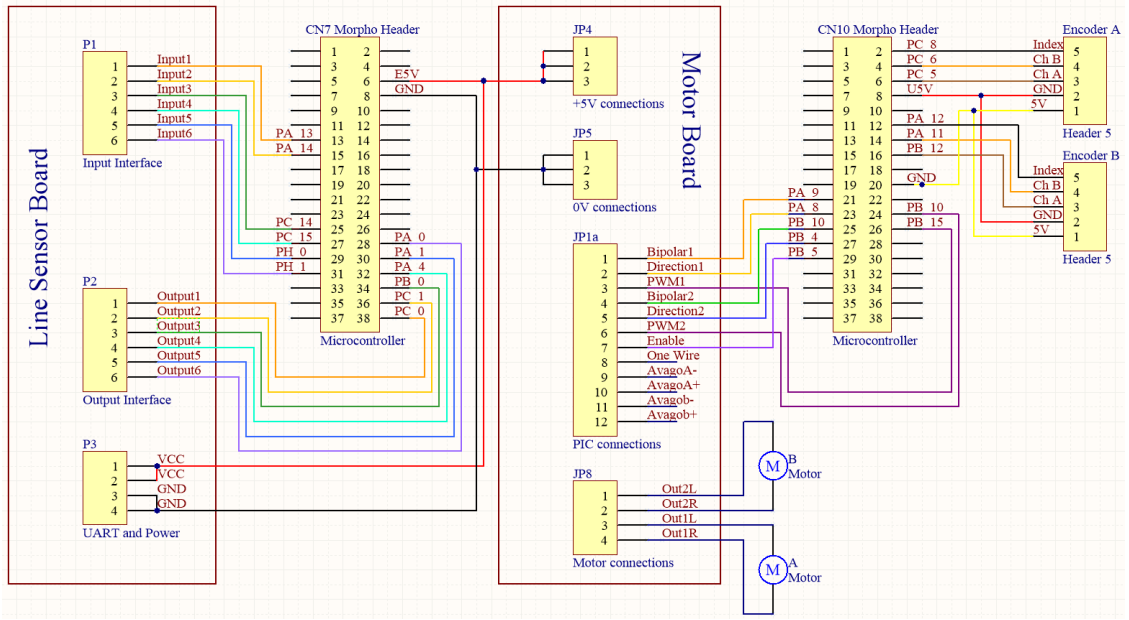


Figure 4.3: Wiring diagram

With regard to the motor board, power supply by the ports JP4 and JP5 connected with +5V and GND on the Nucleo Board respectively [8]. JP1a deals with the signal used to control two motors while the motor is connected with the motor drive board through JP8[8]. Specifically, the PWM signal applied to the motor must be conveyed to the pins (PB_10 and PB_15) [7] that the PWMOut is usable in code.

5. Non-Line Sensors

One of the non-line sensors provided for the project are the quadrature encoders. Quadrature encoders are used to measure the speed of the buggy and give the direction of the buggy. The encoders can be measured with the microcontroller using GPIO interrupts, a Timer Peripheral in Input Capture mode or using the built-in Quadrature Encoder Peripheral. The Nucleo can count the encoder at higher precision by doubling or quadrupling the resolution of the encoder or count per revolution (CPR). The encoders can be interfaced with the microcontroller using the mBed platform, either by the Quadrature Encoder Interface Library or GPIO Interrupt-In using Timers/Counters [6].

The current sensing circuits provided on the motor drive board are used to convert the current in a system such as the battery monitoring system, and convert them to a measurable voltage. The current sensing capabilities can be calibrated to produce a more accurate reading of the voltage.

The battery monitoring circuit is used to measure the voltage and the accumulated current of the battery. The non-line sensors are implemented to the buggy using the MBED and interfaced to the microcontroller. This demonstrates how much energy is used by the buggy to move and enables us to make any adjustments to the systems in the buggy so that the buggy can finish the race with the provided battery.

6. Control

Control method is undeniably the most important part of the entire buggy design, as it dictates the action the buggy would take based on the inputs from its sensors. It forms the basis for the buggy's motions and without it, the buggy is incapable of functioning. There are various types of control algorithms available to program the actions of the buggy, each with its own advantages and disadvantages to take into consideration. This affects the sensor and circuit design of the buggy, with extended algorithms to further reduce or eliminate errors.

6.1 Line sensing & buggy actions

The location of the line relative to the buggy can be known via the six TCRT5000 sensors located at the front of the buggy, which returns a value to the microcontroller. Each sensor returns a voltage between 0-1 V back to the microcontroller, with the higher voltage indicating the closer the line is to that sensor. The values returned by the sensors can then be used to calculate the accurate location of the line relative to the buggy via the use of equation 6.1 [5].

$$E = \frac{\sum_{i=1}^{N} (i)Vi}{\sum_{i=1}^{N} Vi} - \frac{N+1}{2} \qquad (6.1)$$

Where N is the number of sensors, V_i is the voltage measured by the sensor at distance i from the centre in V, i is the distance from the centre of the buggy in m and E is a variable that describes the position of the line to the buggy in m.

The buggy's forward and backward motions can be controlled via the implementation of a H-bridge, powering the motors which drive the wheels. There are two operations that can be performed on the motor drive board - the unipolar and bipolar operations.

The buggy's steering can be controlled by applying a differential gain on one wheel upon detecting the buggy is no longer centred to the line. By applying a differential gain to one wheel, one wheel will be faster than the other, which results in a turn. The speed of the wheels can be tracked using an encoder.

6.2 Pros and cons of proportional and bang-bang controllers

There are two kinds of controllers that can be implemented on the buggy, the proportional and bang-bang controllers. A bang-bang controller only cares about whether the line is to the left or right of the buggy, it then applies a pre-defined differential gain on the opposite wheel causing one wheel to speed up in a turn. The pros and cons of a bang-bang controller is shown below:

Table 6.1: Pros and cons of bang-bang controller

Pros	Very simple to implement, as there are no additional calculations that need to be performed to find out the differential gain.	Faster response, as the microcontroller wouldn't have to wait for additional calculations, but this may be insignificant.	-
Cons	Zig-zag motion due to differential gain on wheels are not exact to realign the buggy back to line, it usually overshoots.	Less power efficient, as differential gains on the wheels are usually more than necessary. More voltage is needlessly used.	A faulty sensor would cause the entire buggy to steer off in the wrong direction.

A proportional controller however calculates the exact distance between the line and the buggy. It then applies an exact differential gain to one wheel to steer the buggy the other way. The pros and cons of a proportional control is listed below:

Table 6.2: Pros and cons of proportional controller

Pros	Exact differential gain leads to buggy realigning itself to line very quickly, the buggy motion is smooth.	More power efficient, as differential gain is exact. Less extra voltage is wasted.	Better response to a faulty sensor.
Cons	Algorithm design is more difficult as more equations need to be included.	Longer processing time of microcontroller, may lead to slightly slower response.	Need to come up with a way to determine the exact differential gain.

6.3 Pros and cons of extended variables of PID controllers

A PID control aims to reduce the errors that can be found in a normal proportional control using methods such as applying an integral term and differential term to the equation. Pros of a proportional term (P) in a PID controller are that exact differential gains can be found and will be applied to one wheel, leading to swift realignment to the line in a smooth motion. However, the buggy would never be exactly centred on the line. It usually is to the side of the line by a small distance, the small distance between the line and the buggy is referred to as the steady-state error. Pros of an Integral term (I) in a PID controller are that by adding an integral term to the equation, the steady-state error would be eliminated, allowing the buggy to follow the line accurately without offset. The cons of an integral term is that the integral term has to be found exactly. If the Integral term is too big, then the buggy would be very insensitive, while if the integral term is too small, then the buggy would be very sensitive. Pros of a differential term (D) in a PID controller are that the differential term acts as a damping factor to the turning of the buggy, allowing it to return back to line quicker when critically damped. However, like the integral term, the differential term also has to be appropriate. Too large of a differential term would lead to overdamping, causing the buggy to oscillate about the line. Too small of a differential term would lead to underdamping, which results in the buggy not returning to line as fast. Additionally, if the sensor measurements are not as accurate, then the differential term would amplify that error, which is not ideal.

6.4 Control algorithm and control parameters

The things that are to be controlled are the H-bridge to control the forward and backward motion of the buggy. The speed of each wheel is altered to control the steering of the buggy. For the forward and back motion of the buggy, a bipolar mode design will be used to control the H-bridge. This is done for several reasons. It is easier to program bipolar mode than unipolar mode as the PWM of the microcontroller would only have to supply one signal. Secondly, there are not many instances when the buggy has to brake or reverse, thus the switching loss of a bipolar mode is negligible so bipolar mode is used. For the steering, a proportional integral differential controller design will be used to control the voltage supplied to the motors, as it solves a lot of the constraints within the steering of the buggy. The biggest problem that lies within the steering, in using a difference in wheel speed, is that it is unstable, as the buggy would most likely turn more than necessary. By using a PID controller, the increase towards wheel speed will be more precise and allow

the buggy to align to line faster, resulting in smoother steering motion.

6.5 Choice of algorithm relationship to sensor design

The choice of algorithm severely affects the performance and expectations of the sensors. An algorithm that is severely prone to error propagation should not be used on a sensor that is sensitive and inaccurate as it would amplify the error, causing undesirable results. The TCRT5000 sensor provides an accurate and not overly sensitive measurement thus a proportional integral differential (PID) design is used for its advantage as stated above.

6.6 Effect of algorithm choice on sensor, circuit design and microcontroller interface

A PID controller requires the sensors' measurements to be very accurate as the differential term may further amplify the error. Thus, borders around each sensor are needed so that the sensors do not interfere with each other. In addition to borders, the microcontroller could also sample the measurements more frequently and averaged to obtain a more accurate result before it is used in the algorithm.

6.7 Sensor implementation

The sensor design will include Six TCRT5000 sensors each placed equally distanced from each other at an angle that is parallel to the buggy, at 3 cm above the ground. Six TCRT5000 sensors are to be used, as that is the maximum amount of sensors that is allowed in this project. In addition, an even number of sensors are to be used so that the line's position relative to the buggy can be better known, as compared to using an odd number of sensors. This is because the centre sensor of an odd numbered sensor buggy will not be able to differentiate which side the line is closer to. Placing the sensors at equal distance allows for easier calculation and also reduces the error an outlier measurement brings. The sensors are placed parallel to the buggy as there is no need to place them at an angle which may bring complications. The sensors are placed at 3 cm above the ground, so that the sensors will not touch the ground when the buggy is going up the slope. In addition, this height allows for an accurate measurement without being too sensitive, which due to the PID controller, would amplify the error.

6.8 Control algorithm implementation

The control algorithm will be programmed into the microcontroller then output from the microcontroller to the motor breakout board to control the buggy. A set of analog input voltage between 0-1 V would be read into the microcontroller. The microcontroller then performs arithmetic based operation using equation 6.1 as stated above to calculate the relative distance of the line to the buggy, with positive values indicating right side of buggy and negative values indicating left side of buggy. The magnitude of the value produced will determine the distance between the line and buggy. If the value calculated is 0, the buggy is on top of the line and the buggy will go straight by applying a positive PWM signal to the motor drive board and operating in bipolar mode. In the case of non-zero values, the buggy would have to turn to align itself to the line, a differential gain would have to be found using equation 5.2 [6] which is given as:

$$u(t) = K_p e(t) + K_i \int_0^t e(t^*)dt^* + K_d \frac{de}{dt} \qquad (6.2)$$

Where u(t) is the manipulated variable, K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, e(t) is the error.

The differential gain will then be applied to the opposite wheel so that the buggy will turn in the desired direction. A set of encoders will be used to monitor the speed of the wheels. When the speed of the wheels are decreased to lower than the default speed of the buggy going straight, the buggy is climbing a slope. Voltage/current will be increased using the microcontroller to the motor breakout board to supply the torque necessary in climbing the slope.

6.9 Solutions to line breaks and direct sunlight interference

The dimensions of the Line Breaks are given in the ESP technical handbook. The control algorithm can be programmed to make the buggy continue its previous action for a certain amount of time when all six sensors are not able to pick up the line. The time can be calculated via a simple equation of the dimension of the line break/ current average speed of the buggy. This can be done by simply implementing a wait statement in the code. If the buggy is still not able to pick up a line after the estimated time, A reverse command will be implemented, as this means that the buggy has reached the end of the track.

Sunlight interference will cause multiple sensors to be lit up at once, the algorithm can then be programmed to ignore such measurements when more than 2 sensors are receiving a voltage above a certain point, such as 0.5 V. In addition, the program can also ignore the new measurements if the sensor's measurements are at least two sensors apart, as there should only be two sensors side by side having more than 0.5 V at once.

7. Hardware Overview

7.1 Chassis

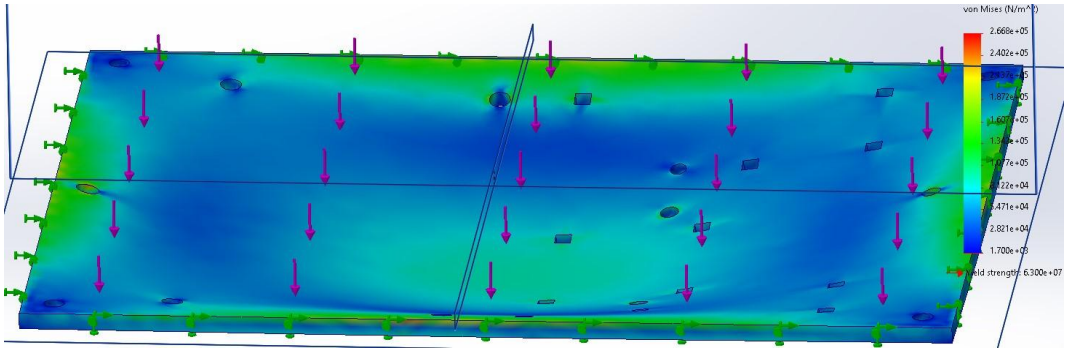


Figure 7.1: Stress analysis of the chassis

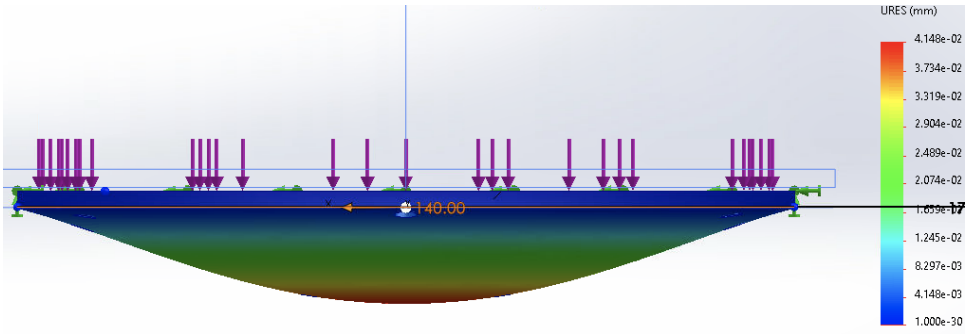


Figure 7.2: Deflection analysis of the chassis (Front)

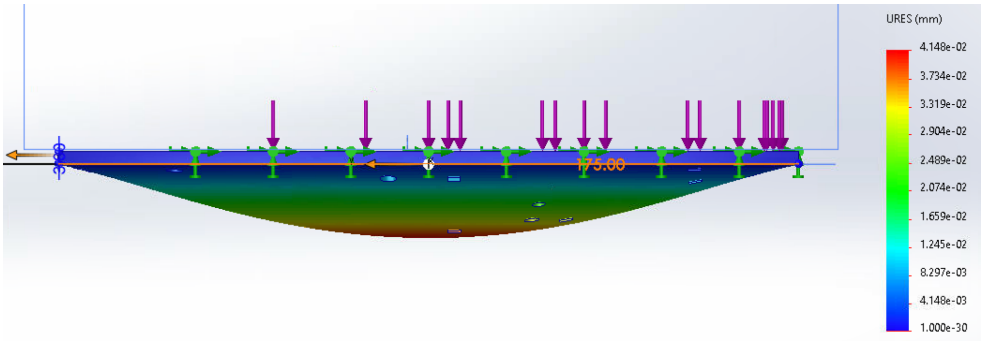


Figure 7.3: Deflection analysis of the chassis (Side)

The initial dimensions for both acetal layers are: 175 mm (length) x 140 mm (width) x 3 mm (height)

Deflection is an important consideration for the buggy. The appropriate calculations must be made when considering the stress the buggy will have to withstand to prevent the board from breaking in construction or testing. Even though deflection might not be fully visible, it shouldn't be neglected. Different materials have different structural strengths and maximum deflection [1].

Maximum Deflection Calculations

Table 7.1: Mass of components loading the chassis

Components	Nucleo board	Motor Board	Battery pack	Motor + gearbox + encoder (2x)	Total
Mass (g)	70	53	265	266	654

$W = 654 \times 10^{-3} \times 9.81$ = 6.41 N where w is the concentrated load and 9.81 is the gravitational acceleration.

Solidworks deflection calculations:

$$x_{acetal} = 41.4\ \mu m,\ x_{GRP} = 2.1\ \mu m,\ x_{aluminium} = 1.06\ \mu m,\ x_{steel} = 0.59\ \mu m$$

7.2 Chassis Material Choice

The goal is to have a very robust, rigid and efficient buggy. The buggy needs to be lightweight, therefore a chassis material that matches the requirements needs to be considered. When looking at the suggested materials, Acetal is the best choice for the material in terms of price and mass. Acetal doesn't have the ability to withstand the most stress but is the most lightweight, which is very important when considering the speed of the buggy. Mild steel is more rigid but also heavier than acetal which is against the chassis design goals. Aluminium is very flexible and has a higher maximum deflection than acetal. Nevertheless, it is also heavier than acetal which goes against the chassis requirements. Rigidness is important when considering the material choice. A chassis that isn't rigid won't be able to withstand the necessary load and risks breaking if subjected to excessive load. The most rigid material is steel but it is also the heaviest which automatically excludes it. The acetal chassis will be constructed with another layer of acetal over it. The two layers can be connected using bolts and screws, significantly increasing the stress the chassis can withstand, and reinforcing it. When considering material price acetal and fibreglass are the most cost-efficient. Chassis thickness is another crucial part of the chassis design. More thickness leads to more rigidness and less deflection, but also more mass. Less

thickness leads to the chassis being more lightweight and having a higher risk of breaking.

7.3 Ease of Manufacture

Out of the 4 suggested materials, (acetal, fibreglass, aluminium and mild steel) aluminium looks to be the easier to manufacture. After aluminium comes mild Steel, then acetal, and finally fiberglass.

When comparing the safety of manufacturing, steel and aluminium seem to be safer to manufacture, because it takes a lot of labour and expertise to manufacture plastics, and it isn't a simple process since not all plastics are the same.

Aluminium and Steel also tend to be more sustainable than Plastics. Plastics dont have such a high melting point, which leads to many harmful chemicals being released during the melting process. Steel, aluminium and fiberglass all have a very high melting point, meaning high temperatures in manufacturing aren't a concern.

Although aluminium and mild steel are preferable in terms of the simplicity of manufacturing, sustainability and environmental issues, these aren't the main points that need to be considered for the chassis construction and the ESP scope overall. For all the reasons listed in section 6.2, acetal remains the material of choice [2] [3] [4].

7.4 Weight and layout

The load will mainly be distributed around the front of the chassis. This will prevent the buggy from rolling behind when climbing a slope. The chassis will have 2 layers of acetal. The bottom layer (Figure 6.3) will include the motor control board on it. Under the bottom layer will be the Sensor board, castor wheel, and both gearboxes. The battery back and the Nucleo Board will be on the top layer (Figure 7.4) of the chassis. Both layers will be connected by 4 screws placed on the corners.

7.5 Dimensioned 2D drawings

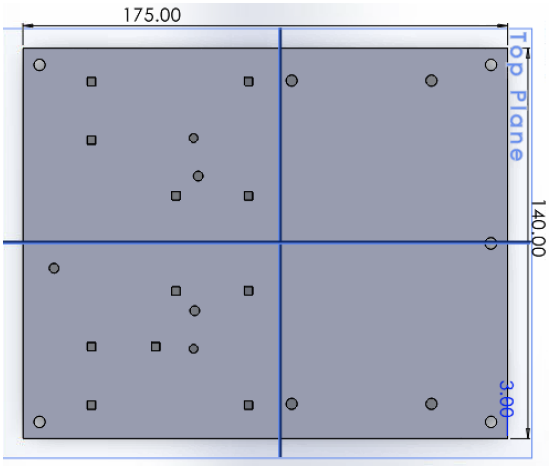


Figure 7.3: Bottom layer (left)

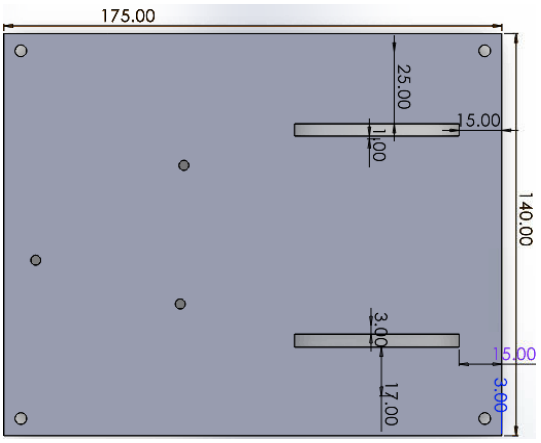


Figure 7.4: Top layer (right)

Both gearboxes will be fitted on to the back of the bottom layer (assuming front is where the castor assembly is connected).

7.6 Other Hardware thoughts

Length and width of the Buggy have to be considered to avoid the buggy from crashing into a wall when turning at the track. Length of the chassis also affects the

turning time of the buggy. 3 wheel control allows the buggy to have a smaller turn radius due to the castor ball being able to rotate in any direction, which is ideal for the restricted space of the track.

8. Summary

To summarise, the report's aim was to design a responsive system that controls the behaviour of the buggy so that it may tackle the obstacles present on the track. A key point of interest is that the TCRT5000 provides superior characteristics in all practical aspects. Various key design decisions are shown in the table below:

Table 8.1: Key Design Decisions

Sensor choice	Algorithm choice	H-bridge control mode	Material for chassis	Dimension of chassis	Additional component
TCRT5000	PID controller	Bipolar operating mode	Acetal	175 x 140 mm	None

9. References

[1] https://www.designingbuildings.co.uk/wiki/Deflection [Accessed November 28, 2022]

[2] https://thisisplastics.com/plastics-101/how-are-plastics-made/ [Accessed November 28, 2022]

[3] https://dielectricmfg.com/knowledge-base/acetal/ [Accessed November 28, 2022]

[4] https://www.wshampshire.com/wp-content/uploads/acetal_grades.pdf [Accessed November 28, 2022]

[5] Lecture Slides - PWM and Motor Control [Accessed November 28, 2022]

[6] ESP Technical Handbook – EEEN21000 Embedded Systems ... (manchester.ac.uk) [Accessed 29 November, 2022]

[7] STM32F401RE Nucleo Platform Datasheet [Accessed November 29, 2022]

[8] Motor Drive board connections [Accessed November 29, 2022]

[9] STM32F401xB/C and STM32F401xD/E advanced Arm®-based 32-bit MCUs Datasheet [Accessed November 29, 2022]

[10] TEKT5400S Datasheet [Accessed November 30, 2022]

[11] TCRT5000 Datasheet [Accessed November 30, 2022]

[12] BPW17N Datasheet [Accessed November 30, 2022]

[13] SFH203P Datasheet [Accessed November 30, 2022]

[14] TSHA4401 Datasheet [Accessed November 30, 2022]

[15] TSHG6400 Datasheet [Accessed November 30, 2022]