MANCHESTER 1824
The University of Manchester

School of Electrical and Electronic Engineering

Embedded Systems Project

DESIGN REPORT #2

Title: Technical Characterisation

Group Number: 37

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

Tutor: Dr Emad Alsusa

Date: 29/11/2018

Contents

1. Introduction

This report discusses the technical characterisation and acts as a guide for the construction of the buggy in the next step. All the different aspects that will support the movement are explicitly analysed and the features that will make the buggy competitive are examined. The overall aim of the report is to shape a clear image of the final design.

More specifically, the software part discusses the structure of software in pseudo code which inspects the input response from the sensors and makes critical decisions for the movement of the buggy. Also, the ability of software to do this is strongly dependent on the line sensor responses. Therefore, the selection of the line sensors is crucial in order to achieve accurate sensing of the white line around the track. Different emitters and detectors are paired and their response is evaluated to select the best one. Additionally, the interface of these to the microcontroller is demonstrated by circuit diagrams, which clearly show how the proposed sensors are going to be connected and how the data is evaluated by software.

Depending on these data, the algorithm run by software will adjust the orientation and the speed of the buggy. For this to be achieved, different sensors are used, which output useful quantities, like speed, current and battery voltage. All of these are discussed in the non-line sensors section.

The control of the system is made possible by using a particular type of controller. All of the options offer different advantages, which are compared in the control section to decide the most efficient one. Furthermore, the control algorithm pseudo code used to implement the movement is provided.

Finally, all of the above combine to give the first representation of the buggy base with hardware design. The components which are crucial for movement and the communication of data between them are taken into consideration to design the chassis of the buggy. The deflection analysis ensures that there will be no significant bending or fracturing and the ease of manufacture of the design is also examined.

Overall, the report is valuable for the progress of the Embedded Systems Project because it deals with the “intelligence” of the buggy. The controller is the “brain” that ensures the buggy will always stay on the line. Sensors are the “eyes” of the buggy, detecting the position of the line. Software supports the algorithm for controlling the motors depending on the sensors’ response. Circuitry and hardware provide the “inner body” of the buggy, making the movement possible. All of the above act as one to create a competitive buggy.

2. Software

2.1. Functional Summary

The software the buggy uses will monitor the position of the buggy. Also, it will send signals to the motors to change the steering of the buggy by changing the speed of the wheels. This will be done using various other sensors used to monitor the initial speed of the motors.

2.2. Constraints to the System

There are various constraints a system will encounter that will limit its functionality. These limits must be considered before writing a program. The main constraints placed on the embedded system software would relate to timing, memory and cost. Each algorithm has different execution times therefore when deciding specific algorithms, the timing should be fast so the motors act within reasonable time to turn a corner. Therefore, it is an important factor in deciding which algorithm is chosen. The programme written must fit onto a memory constraint of 512 KB of flash and 96 KB [1] of SRAM otherwise it will not be executed as the microcontroller cannot store the programme. This is not a huge issue as the microcontroller has plenty of memory for the purpose of our program. Finally, the project has a £40 budget which will limit what extra sensors or gadgets can be added to the buggy that will maximise its functionality.

2.3. Diagrams

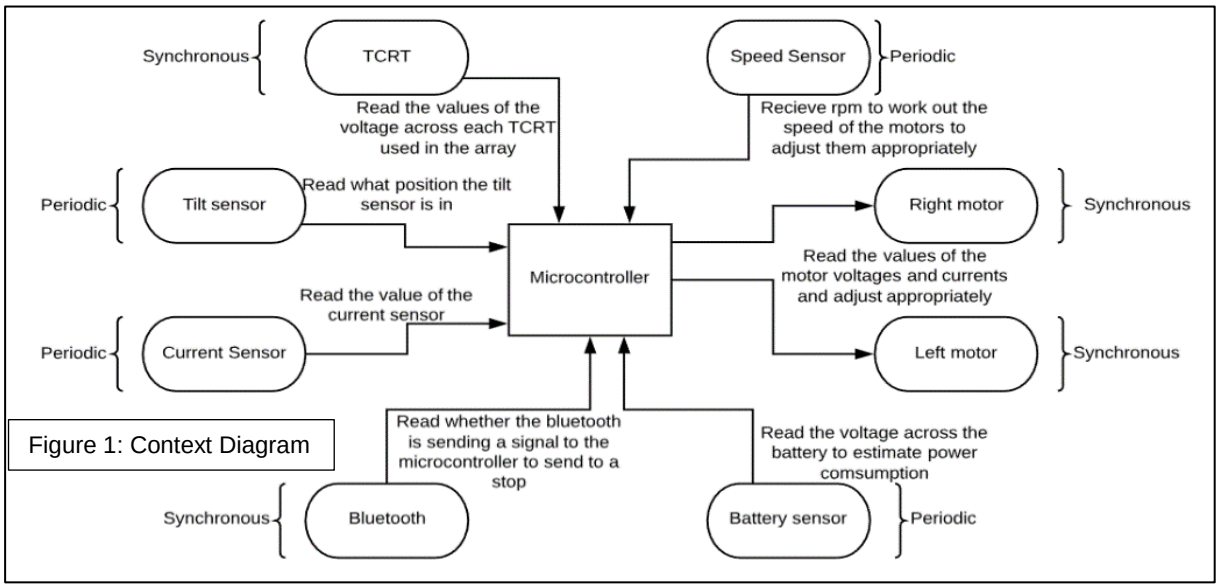


Figure 1: Context Diagram

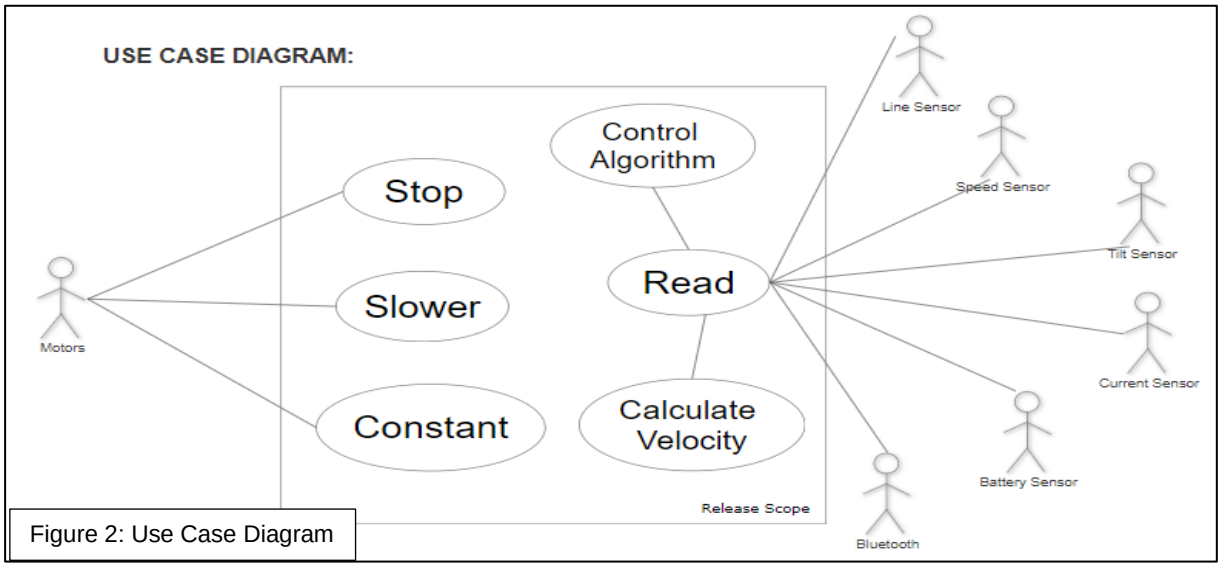


Figure 2: Use Case Diagram

The Context Diagram in Figure 1 shows the messages exchanged between the external objects and the arrival pattern of the messages. It also shows the direction of messages as it passes to or from the microcontroller.

The Use Case Diagram in Figure 2 shows the relationships between the system functions and the entities that are involved in the system. It allows designers to think further into the flow of messages, different scenarios the buggy may encounter and the data and member functions that will be required.

2.4. Use Case Descriptions

Main function: This pseudo code shows the ultimate functionality of the software

```
while (true){
read sensor voltage values
if (on the line){
  PWM.left=constant
  PWM.right=constant}
  else {
  adjust the buggy steering };
}
```

Hill/step: Shows the flow of the code that will be used when the tilt sensor show that the buggy is moving up or down a terrain

```
if      (tilt      sensors      =
up/down hill){
  reduce PWM
  increased torque}
  else {
    PWM = constant
  }
```

Re-finding the line: shows the code applied when the buggy comes off the line and no reading is being found from the sensor

```
if (completely off line){
  PWM=0;
  read(left most sensor)
  if (found value){
    steer left
  }
  else {
    steer right }
}
```

Bluetooth: shows the flow of the code that will be issued when a Bluetooth signal is applied and the buggy must turn around at the end of the track

```
if (bluetooth = true){
  PWM=0
  PWMleft=0
  PWMright=constant
}
refindtheline
```

Sharp corner: shows the code applied when a more complex manoeuvre is required

```
if (sharp corner ahead){
  PWM =reduced
  read(values from sensors)
  if (veering left){
    PWM left = reduced
  }
  else {
    PWM right = reduced
  }
}
```

Steer: shows the general steering code that will be used when the buggy approaches a corner

```
if (left corner){
     PWM left = reduced }
     else if (right corner){
           PWM right = reduced}
else PWM = constant }
```

Stop: Shows the code that will be used when the buggy is at the very end of the track.

```
If (reading high across all sensors)

   Wait until if (reading 0 voltage across all sensors){

   PWM=0; }
```

Table 1: Use Cases

2.5. Object Specifications

Object	Motor	TCRT5000	Bluetooth	Control Algorithm	Speed sensor	Velocity	Battery
Data members	speed direction	sensorArray	Visible	KP Offset TP Error turn	speed counter timer	Ticks sampleTime	Voltage sampletime
Member functions	On/off ApplyPWM SetDC getDirection getSpeed	On/off getVoltage contrAlgorithm	On/off	getLeftsens getRightsens Power1 Power2	getTime StartTimer StopTimer	tickRate wheelV TransV AngularV	getCurrent getEnergy

Table 2: Objects / DMs / MFs

3. Line Sensor Characterisation

3.1. Introduction

In order to decide on the most suitable sensor to use for the buggy the following sensor pairings were tested:

Emitter	Detector
TCRT5000 LED	TCRT5000 Phototransistor
OPE5685 Infrared Emitter	SFH203P Photodiode
OVL5521 White LED	VT90N2 LDR
Red LED	BPW17N Phototransistor

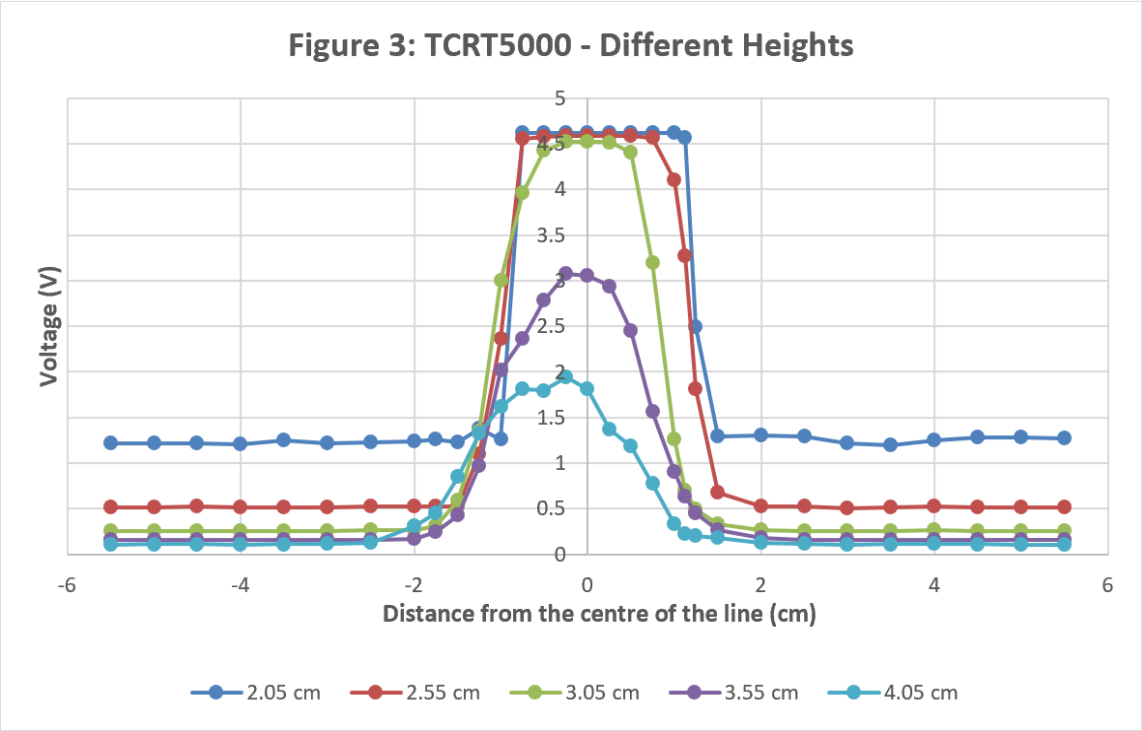
Table 3: Emitters-Detectors Pairings

First, the emitters with the matching detectors as well as the 8-way plugs were soldered onto a small PCB. In between them, a black barrier was placed, in order to block the direct access of light from the emitter to the detector. That is because the desired response of the detector is only the deflection of the light on the white line. That way, the system correctly decides whether the buggy is on the line. After that, the 8-way socket was soldered on the stripboard and the track next to the pins used was cut. Then, the PCB was placed on the stripboard by connecting the plugs to the socket. Next, the circuit of the sensor was created and the appropriate current-limiting resistor, according to the datasheet of the emitter, was calculated and placed on myDAQ. Four wires were soldered on the stripboard that connected it to the power supply, resistors and the ground according to the circuit diagram. The output voltage was measured across the resistor of the detector (10 kΩ) by connecting two wires to an analogue channel of the myDAQ. By using the oscilloscope from the Instruments Launcher for the myDAQ the voltage in the analogue channel was measured. The stripboard containing the sensor was fixed in the test rig at different heights and the output voltage was measured. The track was placed underneath. The distance between the centre of the line and the sensor was the varying quantity ranging from -5.5 cm to 5.5 cm. That way, the output voltages were measured and recorded in an Excel Sheet, where the graphs were plotted.

3.2. Sensors Measurements

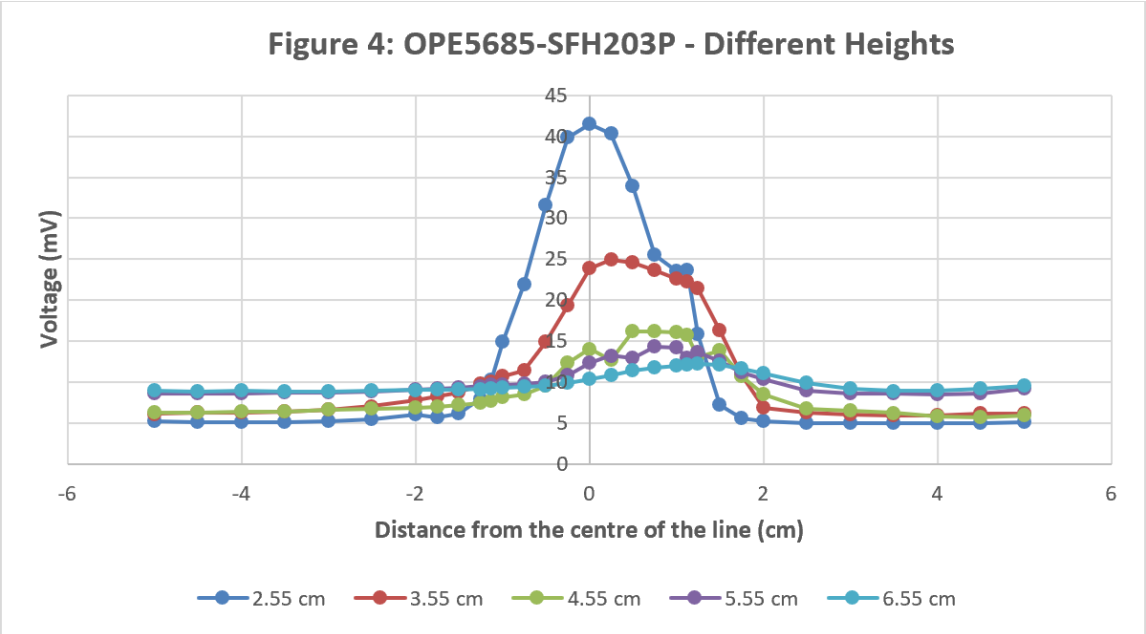
- **TCRT5000 Sensor**

At first, the TCRT5000 sensor with integrated LED and phototransistor was tested. According to Figure 3, the voltage at distances from -5.5 cm to -1 cm and from 1 cm to 5.5 cm (black part of the track) is constant ranging from 0 to 1.5 V depending on height. When it comes to the distance between -1 cm and 1 cm (white line), each curve increases suddenly and tends to stabilise at about 4.5 V. When the sensor is lower, the output response over the white line is maximum (2.05 cm,2.55 cm, 3.05 cm). From these three heights the most suitable is 2.55 cm as we get the most rapid increase between the white and the black line. Also, intermediate voltages are measured when the sensor is transitioning from the white to the black part. Hence, it can be seen that TCRT5000 is sensitive to detecting the white line.



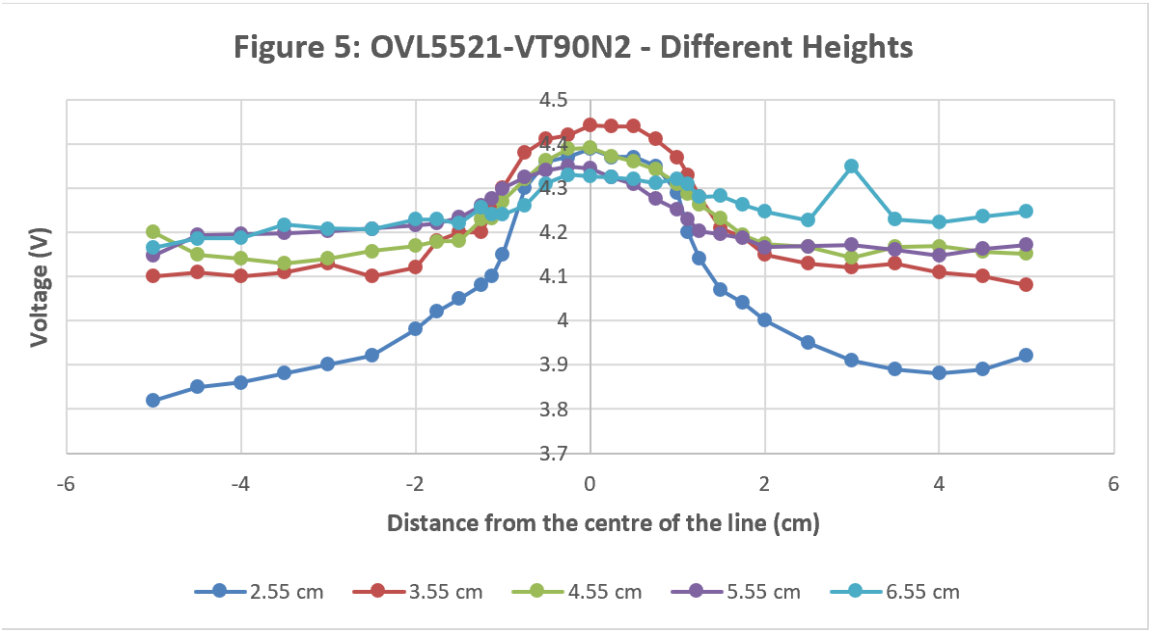
- **OPE5685 Infrared Emitter – SFH203P Photodiode**

As it is shown in Figure 4, the OPE5685-SFH203P sensor is sensitive in the black section because the voltage curve is almost flat below 10 mV when it is at distance from -5.5 cm to -2 cm and from 2 cm to 5.5 cm. However, when it gets closer to the white line, the voltage increases rapidly at a height of 2.55 cm but is not stable when it is above the line. Meanwhile at the other heights, the voltage increase is not obvious as the increase of the response of the white line is only 10 mV to 20 mV.



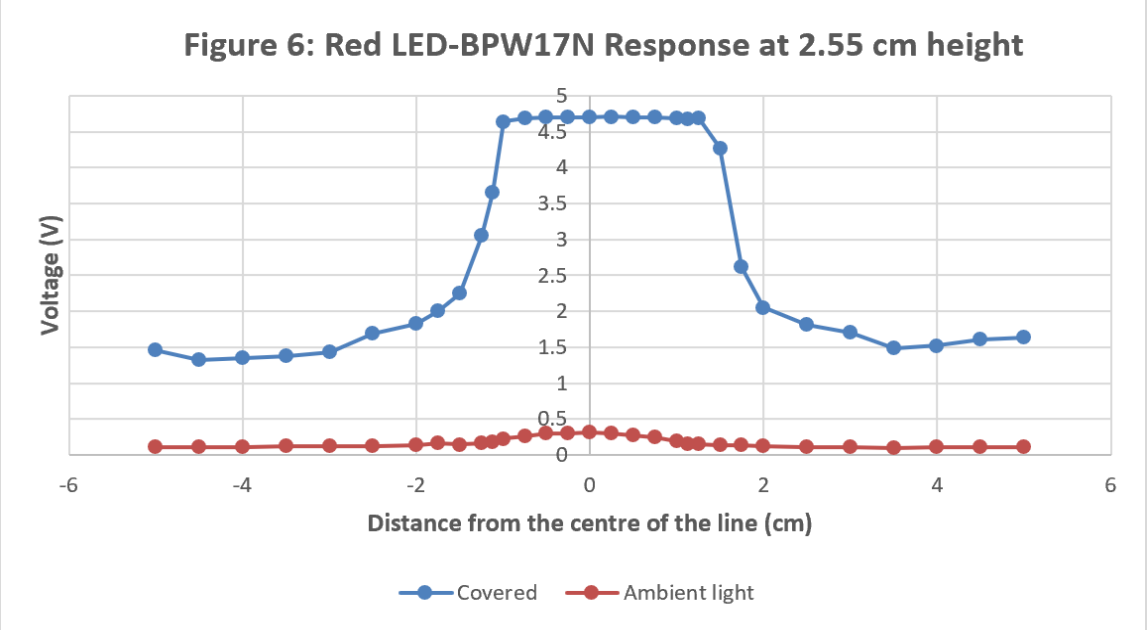
- **OVL5521 White LED – VT90N2 LDR**

From Figure 5, the voltage curves start rising when the sensor is still on the black part. Therefore, the increase is not rapid and the buggy will not have a clear understanding of where the white line is. As the height increases, the difference between the low and high responses decreases. The high response value is not stable when the sensor crosses through the white line. This sensor would not be suitable, as the system requires a rapid change between the black and the white part and a stable value on the white line, to interpret the response of the sensors correctly.



- **Red LED – BPW17N Phototransistor**

Through this experiment, it is found that the Red LED - BPW17N pair is affected by ambient light. In Figure 6, when it is uncovered, the voltage has almost no change between the two parts. Once it is covered by a black mat, the voltage shows a clear increase when it detects the white line and is stable at about 4.7 V. However, this is not practical as no cover will be used. Hence, this is not a good sensor to use in ambient light.



3.3. Sensors Characteristics

The wavelength range of the emitters and the detectors was the main factor considered when pairing them. More specifically, the wavelength of both should be compatible, so that when the emitter emits light of specific wavelength, the detector will be able to detect it after reflection on the white line. For example, the pairing of the OPE5685 Infrared Emitter and the SFH203P Photodiode is based on the fact: the emitter emits lights of 850nm wavelength (Table 4) and the photodiode detects wavelengths in the range 400 nm – 1100 nm (Table 5). Therefore, the wavelengths are compatible. The same fact applies for the other pairs according to the tables.

The datasheets give information on the characteristics:

Emitters	TCRT5000 LED	Green LED	Yellow LED	Red LED	OVL-5521 White LED	OPE5685 Infrared Emitter
Wavelength Range/Peak	950nm	550nm-570nm	570nm-590nm	610nm-760nm	390nm-760nm	850nm
Suggested Current	60 mA	25 mA	25 mA	25 mA	30 mA	50 mA
Beam half angle at half intensity	-	-	-	-	+15°	±22°

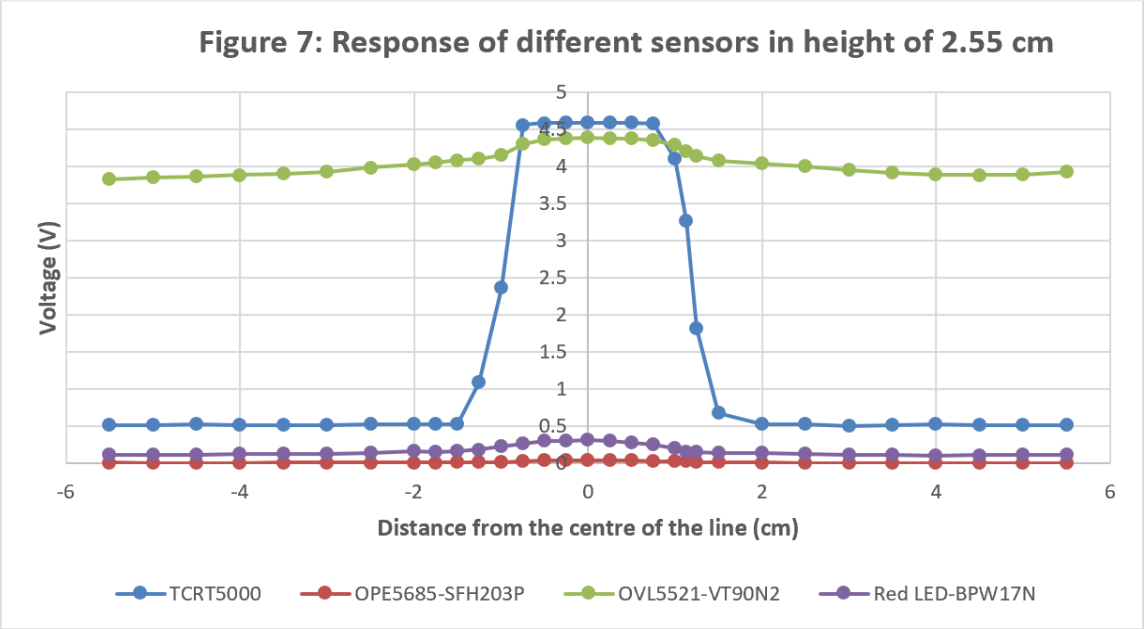
Table 4: Emitters Characteristics

Detectors	TCRT5000 Phototransistor	VT90N2 LDR	BPW17n Phototransistor	SFH203P Photodiode
Wavelength range/peak	950nm	390nm-760nm	450nm-1040nm	400nm-1100nm
Suggested PSU voltage or current	1 mA	-	1 mA	-
Expected output range	0-5 V	0-100 V	-	0-1.3 V
Response time	-	78 ms	-	5 ns
Beam half angle at half intensity	-	-	±12°	±75°

Table 5: Detectors Characteristics

3.4. Sensors comparison

By comparing at the height of 2.55 cm between 4 sensors in Figure 7, it can be seen that TCRT5000 gets a stable and large value when it detects the white line. Meanwhile it has a rapid change when the sensor transitions from the black to the white section. These values are useful for software to get a more accurate steering pattern. For these reasons, the TCRT5000 is the final choice.

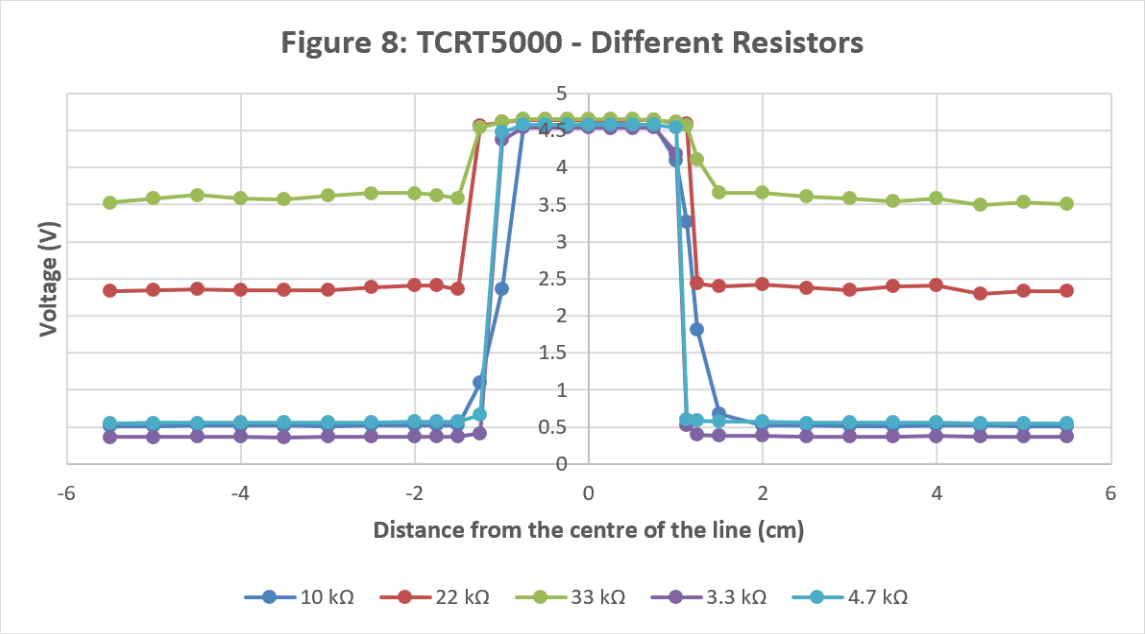


3.5. Sensor Characterisation

After repeated comparisons, the final decision of the height for the TCRT5000 sensor to detect the white line is at 2.55 cm. When the sensor gets closer to the white line, the voltage increases rapidly and then tends to keep a stable value.

The appropriate value of output resistor also needs to be determined. By taking measurements for different resistor values the graph of the response can be plotted as in Figure 8. For resistor values smaller than 10 kΩ, the response of the sensor is

rapid between the black and the white part, and in-between values are not provided. On the other hand, for resistor values bigger than 10 kΩ, the difference between the low and high output response is smaller and in-between values are not visible as well. When it is connected to the 10 kΩ resistor however, the voltage increase is more evident than the others and voltage values in-between the white part and black part are provided. As a result, the most suitable resistor value is 10 kΩ.



3.6. Direct sunlight, line-breaks and other irregularities

- For sunlight, TCRT5000 has a filter to block ambient light, therefore it can get better results than other sensors, because the direct sunlight cannot have a big effect on the output result of the sensor.
- To deal with line-breaks, the sensors will cover a wide range of the track. Through discussion, 5 sensors in an array are going to navigate the buggy around the track.
- The reflectivity of the white line will not significantly affect the output voltage, because the line will still be white enough to reflect most of the emitted light. The variation might produce an unstable output voltage, however it is still in the range of the high output voltage. Therefore, the movement of the buggy will not be affected.

4. Circuit Diagrams for Proposed Line Sensors

4.1. Schematic Diagram

Figure 9 shows the schematic diagrams which represent how the TCRT5000 works internally with an LED named X1 and Phototransistor named X2 and they are connected to R1 of 66.8 Ω and R2 of 10 kΩ. Also the second diagram uses the schematic representation of a my-DAQ.

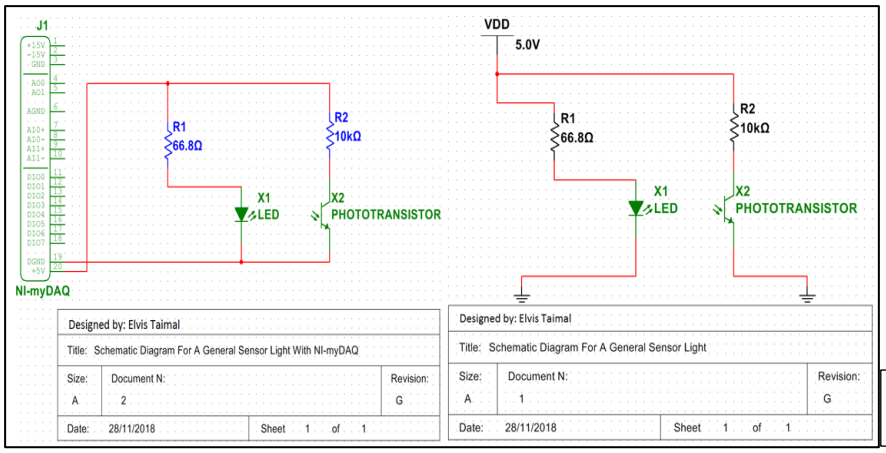


Figure 9: Schematic Diagrams for TCRT5000

Figure 10 shows the third diagram which uses an 8-pin integrated circuit to represent the TCRT5000 sensor making it easier to identify the pins.

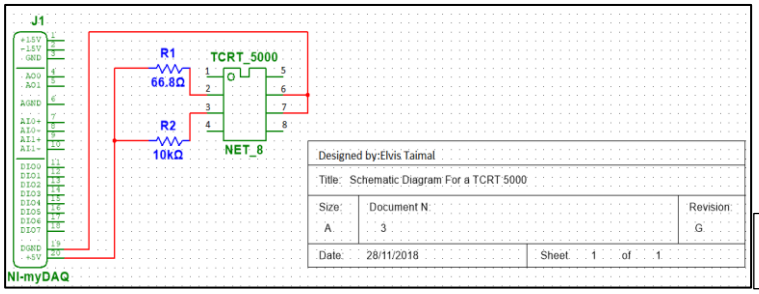


Figure 10: Schematic Diagram with 8-way plug

Similar diagrams apply for all the sensor pairings but with different current-limiting resistors with the emitters and detectors mentioned in Section 3.1.

4.2. Wiring Diagram

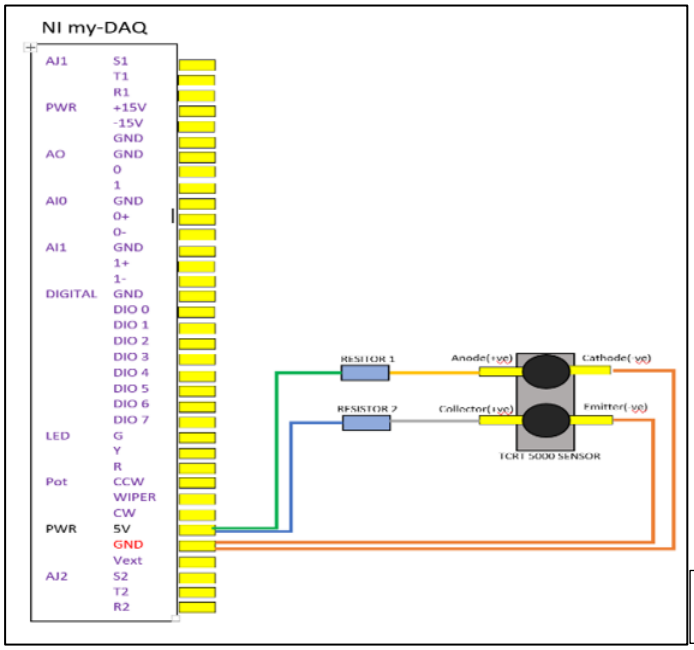
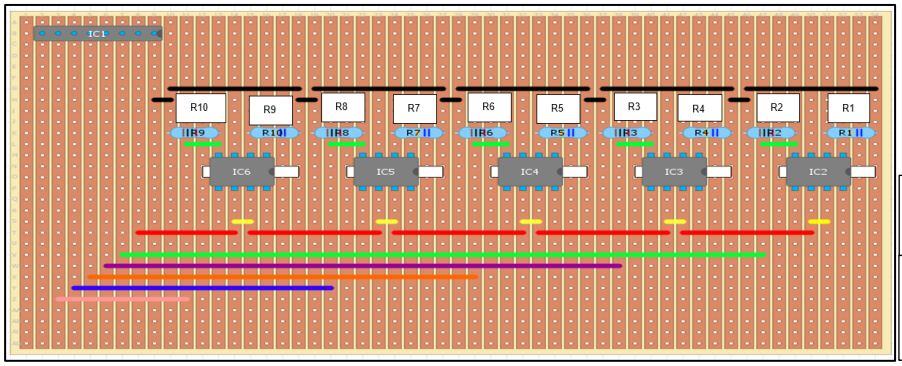


Figure 11: Wiring Diagram for TCRT5000

The wiring diagram shows how the devices are connected with each other without knots or unions between cables with colour-coded wires.

4.3. Stripboard Diagram



*White rectangles showing resistor labels

Figure 12: Stripboard Diagram for TCRT5000 Geometry

The stripboard diagram shows how the 5 TCRT5000 sensors will be used for the buggy and how they will be connected to the 5 V source and to the microcontroller through a socket. The track will be cut between the sensors to avoid short circuits. The components representation on the diagram is:

Table 6: Connections and values

Resistor	R1 66.8 Ω	R2 10 kΩ	R3 66.8 Ω	R4 10 kΩ	R5 66.8 Ω	R6 10 kΩ	R7 66.8 Ω	R8 10 kΩ	R9 66.8 Ω	R10 10 kΩ
TCRT	IC2	IC3	IC4	IC5	IC6	IC7				
Socket	IC1									

The socket will be of a single line package where pin 1 is connected to the 5 V source, pin 2 is connected to the GND and pin 3-7 to the collectors of the respective line sensors (IC2-IC6). These connections give the voltages across the 10 kΩ resistors. These values are required for the control algorithm to control the position of the buggy.

4.4. Connections with Nucleo-F401RE Microcontroller Board

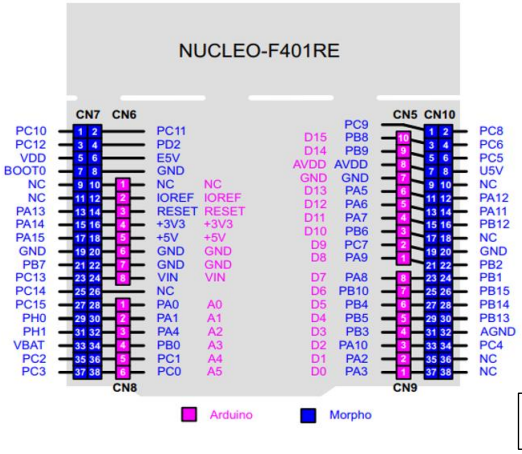


Figure 13 [2]: Nucleo Board

The buggy will use the Arduino connector CN6 for 5 V power-supply (pin 5) and GND (pin 6). Arduino connector CN8 will be used to connect the sensor outputs to the microcontroller analogue-in channels using the pins 1-6 which link to the ADC. The voltage read by these pins will be the voltage across the 10 kΩ resistors of the phototransistors' collectors.

5. Non-Line Sensors

5.1. Introduction

There are 5 types of sensors which will be used for the buggy to improve functionality and competitiveness.

5.2. Speed Sensor

Firstly, it is possible to use two speed sensors, an encoder or a quadrature encoder. The quadrature is more advanced as it tells the speed and the direction of the buggy whilst the encoder only measures speed. Each gearbox is fitted with this sensor and outputs a pulse every full rotation the wheel makes. However, the index channel must be fitted individually. The best way of implementation is using a start/stop timer to measure the time of a rotation as well as a counter to count how many times it has rotated. This will have to be a near continuous examination of speed as the speed will change when moving across other terrains. There is a built-in Quadrature Encoder Peripheral (TIM2 (CH1, CH2) and TIM5 (CH1, CH2)) that can be used to interface the sensor with the microcontroller with resolutions of x2 or x4 [3].

5.3. Battery Monitor

Secondly, a battery monitor using a Dallas integrated circuit can measure voltage and accumulated current of the battery. By sampling the current at specified intervals and summing them together (presuming voltage is constant) the result is given by $E = \int V(t)I(t)dt$ [4]. To implement this in the microcontroller the code provided on blackboard will be used.

5.4. Current Sensor

Thirdly, the current across the motors can be measured to understand how much torque is being applied and whether to adjust the current if going up and down a hill. There is an isolating circuit on the motor driver board used to measure the current in each motor, amplify and send the outputs to the microcontroller to work out the total output voltage measured between test points TP18 and TP20 for motor A and TP19 and TP21 for motor B [5].

5.5. Bluetooth

Finally, Bluetooth will be used to aid the turnaround of the buggy when it reaches the end of the track. It is a short distance module that operates at 2.4 GHz [6]. GAP defines the role of the BLE to decide whether it is visible to other devices during its running. To implement this to the microcontroller the BLE module has a UART serial port [7].

5.6. Tilt Sensor

Another sensor that could aid the competitiveness of the buggy is the tilt sensor. It consists of a rolling ball and a conductive plate. Essentially when the terrain changes, the ball rolls onto the conductive plate allowing current to pass through [8]. This is used when going up and down a hill and tells the microcontroller what incline the buggy is in. This can make the buggy go up the hill at a more competitive speed by increasing the duty ratio of the PWM. This sensor being used in junction with the

current sensor allows proper calculation to be made of how much torque needs to be applied to get up the hill in the fastest time. The advantages of this sensor are the “built-in (1 ms) hardware debounce” [9] arrangement. To interface with the microcontroller it needs to be connected to a digital-in input pin as it outputs logic high and logic low [10]. However, it is polarity sensitive therefore it must be connected properly or will damage the circuit. Two of these sensors will be used with opposite implementation for uphill and downhill respectively.

6. Control

6.1. Introduction

The position of the line is specified by the difference of light that the sensors detect on each side of the buggy. For example, if the sensors on the right side of the buggy detect more light than the left ones, that means there is a right turn ahead. In that case, a controller is needed to supply less power to the right motor to make the buggy turn right and stay on track.

6.2. Comparison of Different Controllers

Three types of controllers are taken into consideration. The first is Bang-Bang Controller, which is a feedback controller that switches abruptly between two states [11]. Although it is simple to use, it cannot provide smooth shift between different states, which will cause the vehicle to zigzag across the line [12]. Therefore, this is not ideal for the buggy.

The second is Proportional Controller. Compared with the former one, it is more effective because it can offer rapid corrections and responses.

The last one is the Proportional-Integral-Derivative Controller (PID). Theoretically, this is the best one because it has a proportional(P) term to correct the current errors, integral(I) term to correct past errors and derivative term(D) to correct future errors. However, it will amplify any noise on the sensor measurements [13].

Based on the above consideration, Proportional Controller is chosen for the buggy.

6.3. Proposed Sensor Implementation

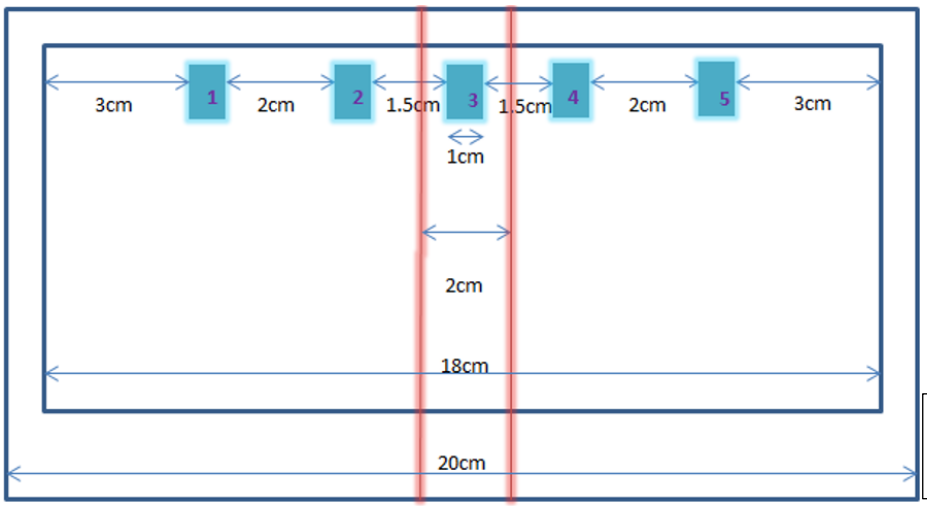


Figure 14: Sensors Geometry

Figure 14 shows the proposed geometry of sensors on the buggy and their position to the line. The five blue rectangles (number 1 to 5) represent the five TCRT5000 line sensors and the two red lines are the edges of the white line. Sensor 3 is put in the center to make sure the buggy is on the line. Sensors 2 and 4 are responsible for detecting the direction change of the line. And sensors 1 and 5 are used to prevent the effect of line breaks since they are far from the line. Since we use TCRT5000 line sensors and they have filter to block sunlight, the effect of sunlight is not considered. As discussed in Line Sensor Characterisation section, the sensors will be placed 2.55 cm above the white line. The sensors are connected to the microcontroller through pins PA1 to PA5.

6.4. Proposed Control Algorithm

As mentioned before, Proportional Controller is used to control the direction of the buggy. The controller continuously gets analogue voltage values from the sensors and uses proportional transition to convert it into power change for the motors. Using the control algorithm, the PWM of the motors will be adjusted. The proposed algorithm is shown below:

```
set value of Kp, offset, Tp;
//Kp is proportionality constant
//offset is the average voltage value across the sensors
//Tp is the power of each motor when error=0
start loop forever
VoltageValueLeft = read left line sensors;
VoltageValueRight = read right line sensors;
VoltageValue = average of all sensors outputs in a single sample
Error = VoltageValue - offset;
Turn = Kp*Error;
Power1 = Tp + Turn;
Power2 = Tp - Turn;
If VoltageValueLeft > VoltageValueRight
then Power1 -> RightMotor,
    Power2 -> LeftMotor;
If VoltageValueLeft < VoltageValueRight
then Power1 -> LeftMotor,
 Power2 -> RightMotor;
end loop forever;
```

7. Hardware Overview

7.1. Introduction

This section provides the hardware specifications of the buggy. The aims are to design the chassis, considering how all the components will fit on as well as ensuring robustness and effective communication between components.

7.2. Chassis

- **2D Chassis Design**

The chassis is designed on SolidWorks 2017 Edition and is shown in Figure 15.

As it is shown, the chassis is composed of one layer of Acetyl. The semicircle allows the adjustment of the components and wiring on the chassis without getting cut on a sharp edge. The holes connect it to components like the microcontroller board/break out board, the motor drive board and the gearboxes.

The mass of the chassis, according to the SolidWorks Mass Properties is expected to be 109.11 grams.

This design is simple and has sufficient surface area to hold all the components required for the movement of the buggy both on the upper-side and the underside. Additionally, other holes can be drilled later in case something else needs to be attached on it. The choice of two layers of material has been avoided due to increased mass and therefore reduced speed.

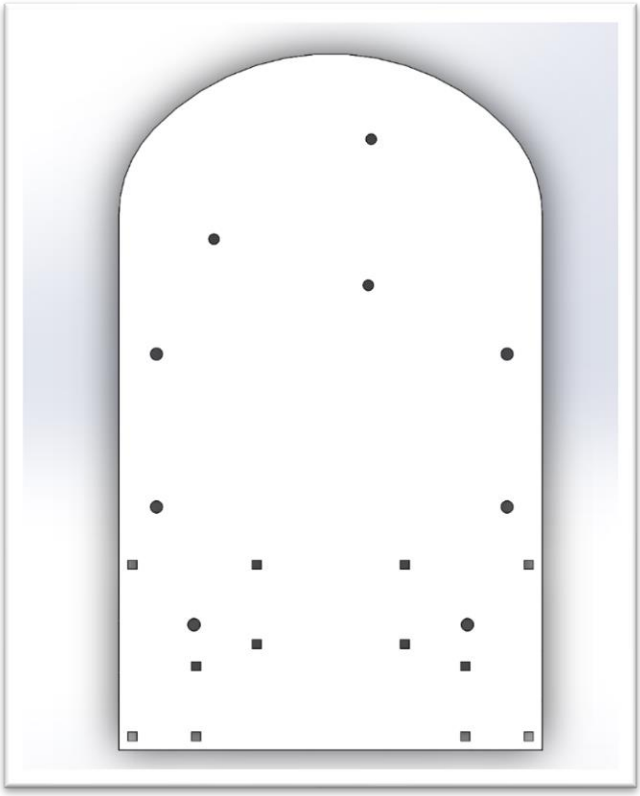


Figure 15: 2D Chassis Design

- **2D Drawings**

The detailed 2D drawing of the chassis, including all the dimensions in mm and in scale 1:2, is shown in Figure 16.

The track specifications [14] pose limitations on the design dimensions. Therefore, the width of the chassis is 14 cm, so that it will be able to make turns without touching the barriers.

The chassis is connected to the gearboxes with screws through the 4.1 mm holes and they will be attached with the help of the square holes. The microcontroller will be attached with screws through the 3.48 mm holes and the motor drive board with screws through the 4 mm holes. A thickness of 3 mm is chosen, as it will reduce the weight. The robustness is evaluated using stress analysis.

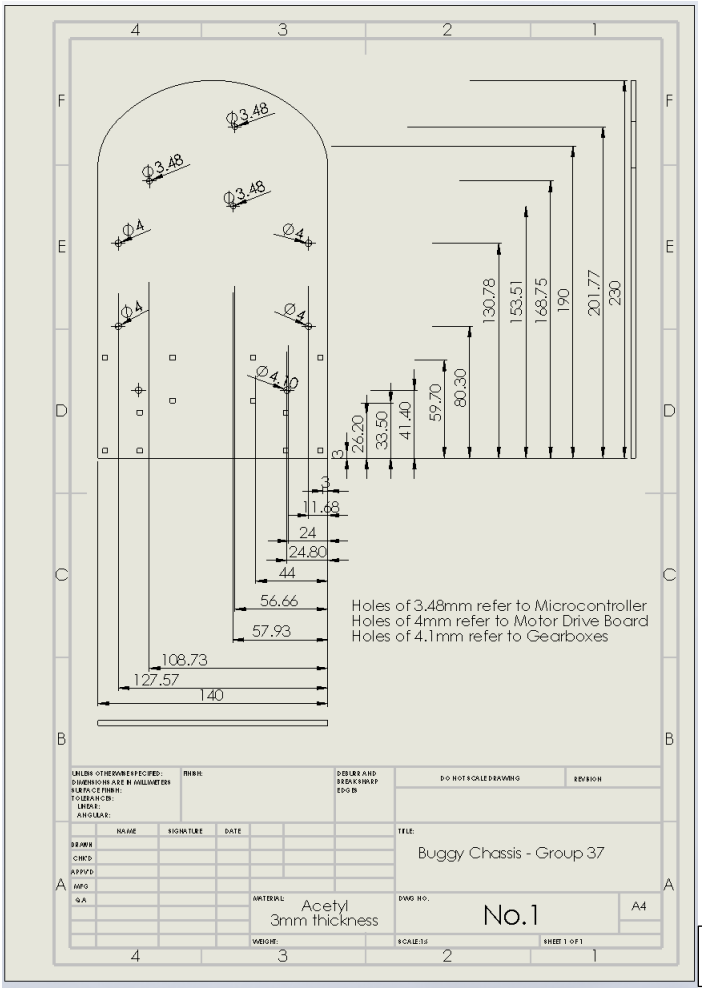


Figure 16: 2D Drawing

7.3. Deflection / Stress Analysis

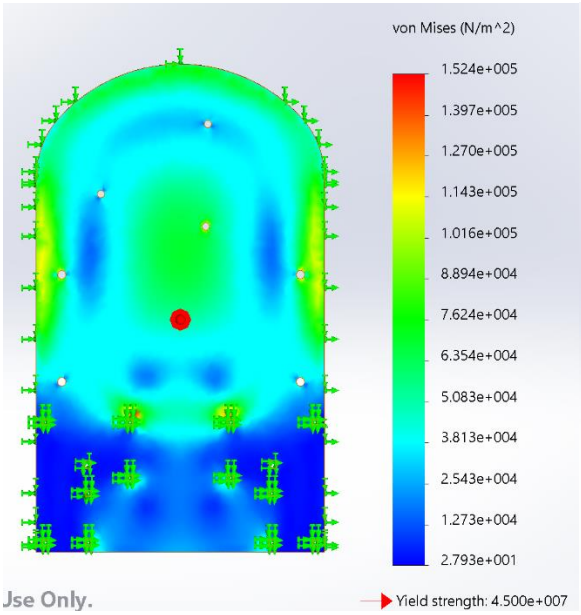


Figure 17: Stress Analysis

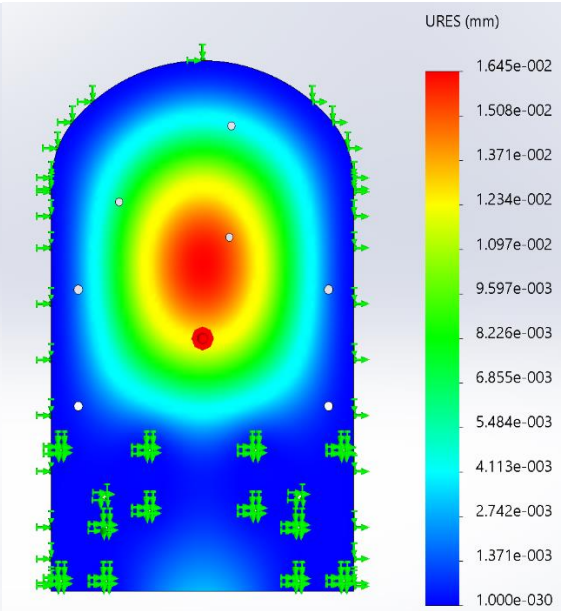


Figure 18: Deflection Analysis

Deflection is an important aspect when considering a hardware design like the chassis of the buggy. The load will cause displacement and stress on the base. It is crucial that the deflection is insignificant when compared to the thickness and that the maximum stress is smaller than the yield strength. If not, the chassis is likely to significantly bend or break.

The loading that the base is subject to is 491 g, which includes the battery pack (265 g), the motor drive board (53 g), the microcontroller (35 g), its breakout board (38 g) and wiring (100 g). The weight on the base will be uniformly distributed, so the μC board will be placed at the front, the battery pack underneath and the motor drive board in the middle. The analysis is performed with a distributed mass of 0.491 kg.

In Figure 17, the value of maximum deflection is low, i.e. 0.01 mm and indicates that the chassis is not likely to bend significantly or break. In Figure 18, the maximum stress value is much smaller than the value of yield strength, so the material is not likely to yield.

The stress analysis results are the expected as they are influenced by the properties of the chassis, like Acetyl's high Young's Modulus, high area moment of inertia [15] but also by the light load on it. In the values derived, there is a percentage of error, as in the real chassis the load cannot be perfectly distributed and the overall load might be slightly heavier, however even this will still not produce enough stress to yield the chassis. Hence, the design will be robust, light and will not break when subject to the components' load.

7.4. Material Choice

- **Comparison of Different Materials**

The different material choices are evaluated in the following table:

Material / Material Characteristic	Acetyl	Glass-Reinforced Plastic Laminate	Aluminium	Mild Steel
Cost (£ per cm^3) [16]	0.019	0.077	0.025	0.045
Density (g per cm^3) [16]	1.8	1.9	2.7	7.8
Flexural strength (MPa) [16]	91	255	310	414

Table 7: Materials characteristics for comparison

Acetyl is generally easily accessible and is mass produced by many suppliers around the UK. This means that its cost is low (£0.019 per cm^3) compared to the other alternatives. Furthermore, Table 7 shows that Acetyl has the lowest density and that means it can be cut more easily with lower power consumption than the other materials. Also, for the same volume of chassis it will be the lightest material to use.

As far as the technical characteristics are concerned, the flexural strength gives the maximum load that the base can be subject to.

For the flexural strength [17]:

$$\sigma = \frac{3W_{max}L}{2bd^2} \quad (1)$$

Where σ is the flexural strength (Pa), W_{max} is the maximum load (N), d thickness (m), b width (m), L distance (m) between the two gearboxes which act as support bases.

Hence, the maximum load that the 3mm Acetyl base can cope with is:

$$W_{max} = \frac{2\sigma bd^2}{3L} = \frac{2 \times 91 \times 10^6 \times 230 \times 10^{-3} \times 9 \times 10^{-6}}{3 \times (115.2 - 24.8) \times 10^{-3}} = 1389\ N\ or\ 141\ kg$$

Therefore, a base made from Acetyl enables all the components to be put on it without any yielding. As a result, a higher flexural strength is not required.

- **Material Selection**

In conclusion, the accessibility, low cost, low density and high flexural strength of Acetyl make it the best choice for the chassis. With this material it is ensured that the chassis will be able to hold all the components required and will be robust and light, which will increase the speed.

7.5. Ease of manufacture

The manufacture of the chassis does not pose any particular difficulty as the design is relatively large and without minute details or sharp edges, which might cause inaccuracies. Therefore, the laser-cutting machine can identify the dimensions easily and produce the required results quickly. The material does not add any difficulty to the manufacturing process, as it is the most common to laser-cut. The small size of the holes can also be achieved by the accuracy of the machine.

7.6. Drivetrain / Layout

The drivetrain of the buggy is rear-wheel drive because the power from the motors is transferred to the wheels at the back. At the front, the ball castor supports the steering. Rear-wheel drive will make steering more efficient and will also improve acceleration and braking [18]. The layout of the wheels is shown in Figure 19.

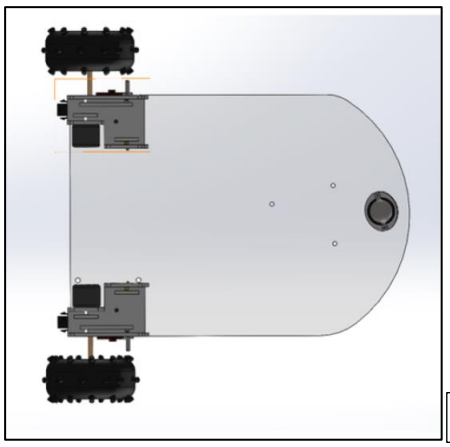


Figure 19: Wheels Layout

7.7. Further Hardware Thoughts

A final point refers to the wheels position. They will have the same distance from the center of the chassis so that the steering is smooth in both right and left direction.

Also, the wheels are attached close to the chassis so that the distance between them is minimum. That way, the steering is faster than in a case where the wheels are placed further apart [19].

8. Summary

Concluding this report, important findings were derived which characterise fully the functionality of the buggy. The results and the conclusions give a clear image of how the buggy will look but also how each aspect of the structure participates to make successful movement around the track possible. Each section is summarised:

- Software: The constraints of the system do not pose any serious limitations on the progress of the project. All the possible functions and objects for the movement are now known and provide a useful guide for writing the actual code.
- Line Sensor Characterisation: Following the comparison of the different sensors, the TCRT5000 is the selection of the line sensor. A full characterisation of the proposed sensor includes the most suitable height, which is 2.55 cm and also the appropriate output resistor of 10 kΩ. The integrated daylight blocking filter will deal with direct sunlight and change in reflectivity of the white line. The configuration of the sensors (5 in a straight array) will deal effectively with the line breaks.
- Circuit diagrams: The diagrams provide a useful guide to connect the sensors to the microcontroller in the next step. The pins used are PA1 to PA5, which are analogue-in channels.
- Non-line sensors: The implementation of the non-line sensors is now known to be taken further in the next step. The buggy will be more competitive with the use of tilt sensors which will sense when it runs on an incline, in order to adjust the torque.
- Control: The rapid correction and response of the Proportional Controller make it the most suitable option for controlling the speed of the motors and the direction of the buggy. The control algorithm is now completely defined and ready to be taken to the next level. The proposed geometry includes 5 TCRT5000 sensors in a straight array.
- Hardware: The final design of the chassis is now ready for manufacture. The dimensions of the chassis adapt with the specifications of the track. A fully-dimensioned drawing is produced. On the base, all the components will be attached with screws through holes. The stress analysis indicates a maximum deflection of 0.1 mm and a small value of stress, which ensures the robustness and low weight of the chassis. The material is chosen to be Acetyl, as this will pose no difficulty in the process of manufacture. The drivetrain of the buggy is characterised as rear-wheel drive and the wheels will have the same distance from the centre of the chassis with minimum distance between them. This will ensure effective control of the buggy.

All the above conclusions demonstrate that the buggy is now fully characterised and a good performance around the track is expected.

9. References

[1] ST life Augmented. (2015, Jan). *STM32F401xD - xE*. Available: https://www.st.com/resource/en/datasheet/stm32f401re.pdf , p.1
[2] ST life Augmented. (2017, Dec). *UM1724 User Manuals*. Available: https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf , p.30
[3] Podd. F, Apsley. J, "Measuring Encoders with the STM32F401RE", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.66
[4] Podd. F, Apsley. J, "Battery Monitor", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.75
[5] Podd. F, Apsley. J, "Current Sensing", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.74
[6] Podd. F, Apsley. J, "Bluetooth", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.69
[7] Podd. F, Apsley. J, "HM-10 BLE module", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.72
[8] EL-PRO-CUS. (2018). *Understanding about Tilt Sensor, Types and Working with Applications*. Available: https://www.elprocus.com/tilt-sensor-types-working-principle-and-its-applications/
[9] Abra Electronics (2018). *SENS-39 Golden Angle Ball Switch Tilt Sensor Module - SW520D*. Available: https://abra-electronics.com/sensors/tilt-sensors/sens-39-one-way-tilt-sensor-sens-39.html
[10] HK Shan Hai Group Limited (2018). *Tilt Sensor Module*. Available: https://components101.com/sites/default/files/component_datasheet/Tilt%20Sensor%20Datasheet.pdf
[11] Wikipedia (2018). *Bang-Bang Control*. Available: https://en.wikipedia.org/wiki/Bang%E2%80%93bang_control
[12] Podd. F, Apsley. J, "Bang-bang with Deadzone", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.77
[13] Podd. F, Apsley. J, "PD or PID in Detail", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p.81
[14] Podd. F, Apsley. J, "Track specification", in *Embedded System Project: Procedures Handbook*, University of Manchester, 2018-2019, pp. 6-7
[15] Podd. F, Apsley. J, "Chassis Construction", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, pp. 84-85
[16] RS Components, Metals4u. (2018, Nov). Available: https://uk.rs-online.com/web/p/solid-plastic-sheets/0824654/ https://www.metals4u.co.uk/mild-steel/c6/sheet/c2240/sheets/c55/3mm-thick/p2501
[17] L. Johnson. (2018, Nov. 05). *How to Calculate Flexural Strength*. Available: https://sciencing.com/calculate-flexural-strength-5179141.html
[18] Buy Here Pay Here USA. (2017, Aug. 31). *Front-Wheel Drive vs. Rear-Wheel Drive | Pros & Cons.* Available: https://www.buyherepayhereusa.com/blog/front-wheel-drive-vs-rear-wheel-drive-pros-cons/
[19] Podd. F, Apsley. J, "Calculating Velocity", in *Embedded System Project: Technical Handbook*, University of Manchester, 2018-2019, p. 68