Capstone Project Report

Autonomous Driving based on Line Following and Obstacle Detection

by
Yap Jia Yung
(15046824)

BSc (Hons.) in Computer Science

Supervisor: Dr. Yap Kian Meng
Date: June, 2019

Project title : Autonomous driving base on line following and obstacle detection.
Date : 21st June 2019
Student : Yap Jia Yung
Supervisor : Dr Yap Kian Meng

Abstract

Autonomous driving is the manoeuvre of a vehicle without human interaction. Is a vehicle that is able to guide itself along the road. The goal of this project is to develop an autonomous driving system based on line following and obstacle detection. The main objective is to analyse, design, implement and analyse the line following using camera and obstacle detection system using an ultrasonic sensor. A Lego prototype vehicle will be used to implements the system and tested on a model course. The research began with a literature review on various researches on the line following and obstacle detection. An interview were also conducted with line following experts to understand how a line follower works and line follower that is available in the market. The system developed was based on the findings of existing technologies used for line following and obstacle detection. The results have shown that using camera and ultrasonic sensor is able to develop a system to follow a set of path drawn using line on the surface as well as avoiding obstacle blocking the path. The conclusion can be drawn that camera outperforms majority of the line follower in the market that use Infrared (IR) sensor.

***Keywords**: Autonomous driving, Line following, Obstacle detection, EV3, OpenCV, Java*

Table of Contents

1 Introduction

1.1 Introduction

This project is an implementation of an autonomous driving base on lane following and obstacle detection system on a model vehicle using digital image processing and Lego EV3 building kits, a software platform produced by Lego for the development of programmable robots based on Lego building blocks. The system targets Sunway Bus Rapid Transit (BRT) line that is part of the Klang Valley Integrated Transit System servicing the suburbs of Petaling Jaya, Malaysia. The BRT is driven by a human on an elevated guideway that is not shared with other road traffic to cater to the high volume of commuters. Since the bus services is a closed system and traveling in a single direction guideway repeatedly, an autonomous driving system can be used to replace human for doing a repetitive task. The system also intends to reduce the risk of an accident caused by human error because human tends to make much more mistake compare to a technical error. According to the National Highway Traffic Safety Administration (NHTSA), they have found that 94% of a major accident are caused by human error, with the other 2% and 2% are caused by environment error and vehicle error respectively, followed by unknown causes for the rest 2% [1]. Due to the fact that autonomous driving is still at an infant stage, the system will only be tested on a scale model as a proof of concept. The model is build using Lego and will be tested on a model course as a representation of BRT path with line marking on the surface for the vehicle to follow the line and pattern marks to indicate the system to stop. Pattern marks are used to represent a bus stop for the line follower to stop, same as how the BRT needs to make a stop at every bus station. Besides, the obstacle will be placed around the path to test for obstacle detection. This paper will discuss various problem faced in existing line following method and a comparison of different obstacle detection technology, proceed with developing a line follower using a camera and, results and discussion of this system

1.2 Problem Statement

Line following system that is available nowadays suffers from blurry and incomplete lines which will affect the reading from the camera as input for the system [2]. For example, most line follower nowadays uses infrared (IR) sensors which is sensitive towards the quality of line marking and any contamination on the line will affect the accuracy of the system because they depend on the reflectance on the surface [3]. Apart from that, line follower that use IR sensors to detect lines only works with a black line on a white surface or vice versa because they depend on the reflectance value on the surface to identify the position of the robot corresponding to the line [4]. A surface with the same reflectance value such as red and green will not return an accurate value to differentiate which is the line and which is the background for IR sensors because they both have similar reflectance value. Besides, to get an accurate reading for the IR sensors, the line on the surface need to be drawn with a certain width, wide enough so that the sensors will not overlap between the contrast of the two edges on the line [5]. Moreover, infrared sensors are easily affected by environmental factors such as fog, smoke, and sunlight [6]. Therefore, a line following system using image processing that is robust against incomplete lines, different colour of lines and, lines with different width is developed to solve the problem encounter in IR sensors for line following.

1.3 Goals

The goals of this project are to develop an accurate line following and obstacle avoidance system for a vehicle that is able to detect and follow the line on the surface and stop when encountering any obstacle. The system should also be able to recognise the pattern on the surface and stop when a certain pattern is recognised.

1.4 Objectives

This research involves the development of line following system that is based on image processing that would require the input of the surrounding from the camera and ultrasonic sensor. The objectives to achieve this system is listed below:

- Design a system that is able to follow a line drawn on the surface.
- Design a system that is able to detect and avoid obstacle.
- Design a system that is able to stop and wait if an indication(wide line) is detected
- Implement the line following and obstacle avoidance system in a model car.
- Test and evaluate the system in a model course.

1.5 Scope of Work

- The system must be able to follow a line drawn on the surface.
- The system must be able to follow the line with a different width (<5cm).
- The system must be able to make different degrees of turns.
- The system must stop when it encounters an object that block it's the path.
- The system must be able to follow line even if there is breaks or blurry lines.
- The system must be able to follow any colour of the line.
- The system must be able to detect for pattern on the surface and make a temporary sstop.

2 Literature review

2.1 Introduction

In recent years, autonomous driving technology had been a popular research area for automotive and robotic companies to develop a self-driving vehicle such as Tesla, BMW, EV3, etc. To develop a fully autonomous vehicle, several components are required to construct the system and one of them includes line following [6]. Hence in this section, several lines following techniques and obstacle detection techniques are being reviewed for the different approaches used and a comparison of their advantages and disadvantages are made at the end of this section.

2.2 Overview of line following technologies

2.2.1 Path planning of line follower robot

In this paper, the author presents the development of a line follower wheeled mobile robot based on infra-red line sensors to give a fast, smooth, accurate and safe movement [7]. In general, the line follower robot is a self-operating mobile vehicle that will follow a line drawn on the surface. The line on the surface can be either black and white or vice versa.

To detect the line on the surface, a quadratic line detection algorithm is used by the author. In this algorithm, eight reflective optical sensors were used to determine the position of the robot. To get the position of the robot in regards to the line, three consecutive sensors with higher output readings than the other five needs to be located as shown in Figure 1.

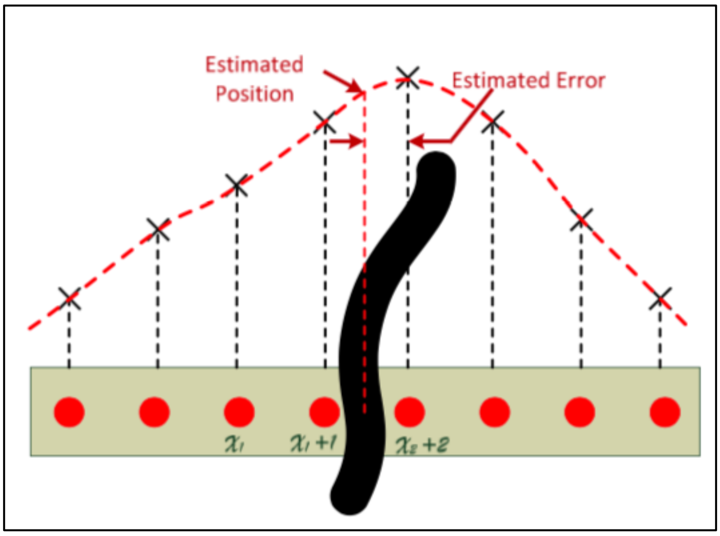

Figure 1 *Line detection algorithm via quadratic interpolation*

Assume that the three sensors are $x, x+1, x+2$ and the true shape of sensor output values are in range of $[x, x+2]$ which can be approximated by a quadratic curve. Based on the quadratic curve, we can then derive the relationships between coordinated of sensors and output values as:

$$y_1 = ax^2 + bx + c \tag{1}$$
$$y_2 = a(x+1)^2 + b(x+1) + c \tag{2}$$
$$y_3 = a(x+2)^2 + b(x+2) + c \tag{3}$$

The coordinate of true position of the line is where the quadratic curve is at maximum and by using calculus, the coordinate value can be derived with:

$$x = -\frac{b}{2a} \tag{4}$$

$$a = \frac{y_1 + y_2 - 2y_3}{2} \tag{5}$$

$$x = y_1 - y_2 - 2ax_1 - a \tag{6}$$

With the assumption that the coordinate for the centre position of the line following robot is 0. Thus, the offset position(error) between the line and centre position of robot is:

$$e = 0 - x = -x \tag{7}$$

With the error calculated, the controller of the robot then calculate the distance between the error and current position of the robot. The controller then commands the motors whether to take a hard turn or small turn if the error is either high or low. In general, the magnitude of the turn taken will be proportional to the error.

To test the line follower, a robot programmed with a simple on/off control is used as a comparison purpose to evaluate the performance of Dynamic PID algorithm using a quadratic line detection algorithm. The results are shown in Figure 7.

Criteria	Dynamic PID algorithm	Simple (on/off)
Time to complete one whole circuit	47.6s	71.4s
Line tracking	Smooth	Not so smooth
Velocity	0.2m/s	0.14m/s
Tendency to astray from line	Low	High

***Table 1** Experimental result for Line Following Robot*

From the results, it shows that the quadratic line detection algorithm scores better in terms of time, line tracking, velocity and, the tendency to astray from the line. Despite the accuracy, the main disadvantages of this line detection are the path needs to be either black line on a white surface or vice versa.

2.2.2 Black Line Tracking Robot

In this paper, the author presents a line following robot that will track line in black/dark line on a lighter surface depending on the contrast using two Infrared (IR) sensors in a linear position [5]. The IR uses the reflectance on the surface to estimate the position of the robot from the line with the basic criteria that black line has lesser reflectance value than lighter surface around it because black absorbs light as shown in Figure 2. The low reflectance value is the parameter used to indicate the position of black line whereas high reflectance value indicates the surface around the line. Thus, from the input of both linear position IR, if either the left/right IR sensors gather low reflectance value from the surface, it indicates that the black line is moving towards the left/right side of the robot correspondingly. Knowing the black line is moving towards the left/right, a controller then control the motor to go in the same direction with the line shown in Figure 3.

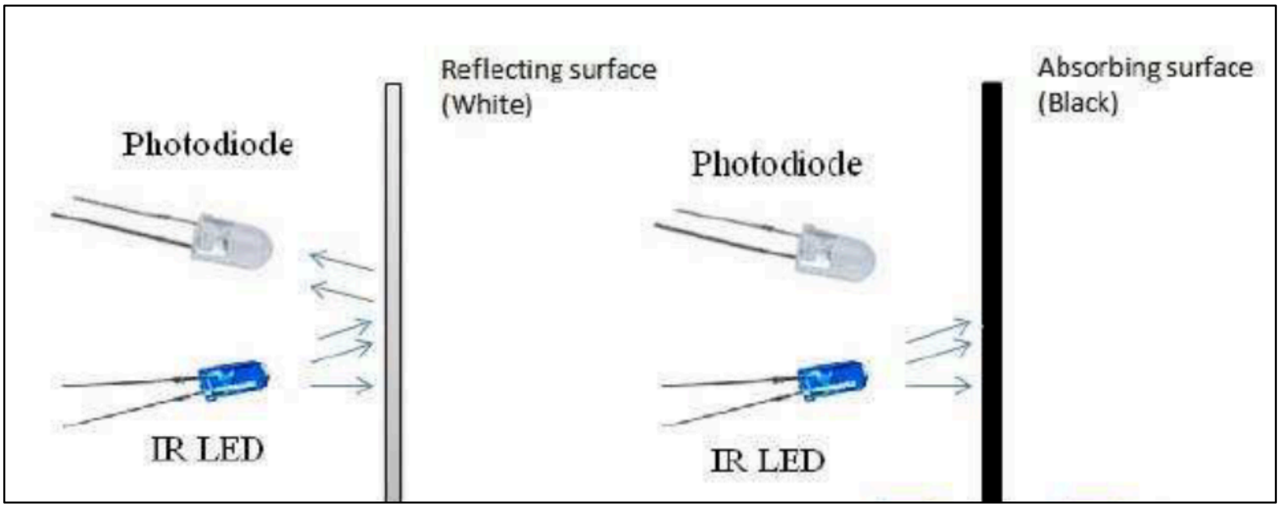

Figure 2 *IR sensor working principle*

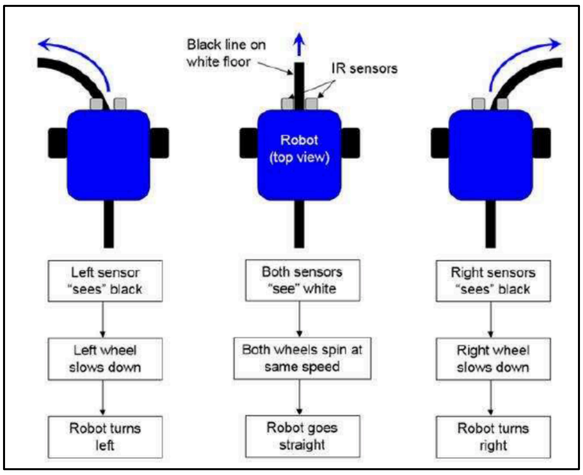

Figure 3 *Robot principle*

To test the robot, the author has used a very low reflection colour(black) as a guiding colour and white with a high reflection value for the base colour. The guiding line(black) needs to be a minimum of certain width and 200mm is used by the author to prevent the two sensors from overlapping. The path used for testing is shown in Figure 4.

Figure 4 *Path for testing*

The results show that the black line follower is capable of navigating through the lines with no difficulties even under ambient lighting. Despite the accuracy of this method, the line follower base on IR needs to operate under a certain condition which is the path can only be either black or white for the IR rays to reflect from the path. Apart from that, the lines need to be about 1 or 2 inches in width on the white surface to prevent overlapping of IR rays. To get a more accurate value, the IR sensors also need to be position very close to the surface which will affect the ground clearance of the robot. The robot also suffers from slow speed and instability on different thickness of lines and hard angles.

2.2.3 Analysis of Line Sensor Configuration for the Advanced Line Follower Robot

In this paper, the author presents a line follower robot using a line sensor configuration. A simple set of experiments are conducted on sensor positioning and also controlling strategy to enable a 90-degree turn accuracy. This method is tested on a test robot shown in Figure 5 and evaluated on a test pitch shown in Figure 6.

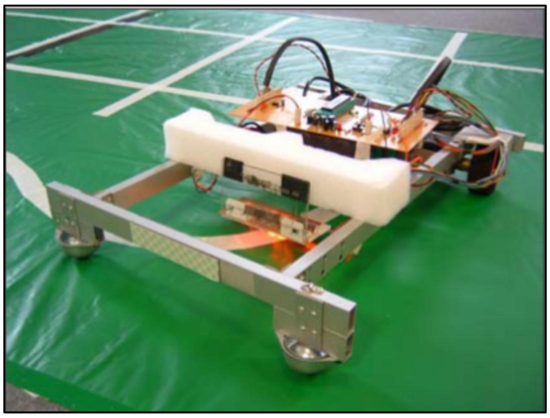

Figure 5 *Test robot*

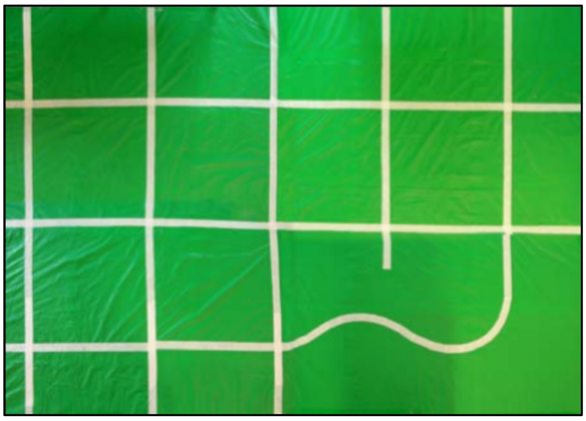

Figure 6 *Test pitch*

The sensors used in the line follower robot consist of ultra bright red LED combined with a light dependent resistor (LDR). The LDR has a variable resistance that changes with the light intensity that falls upon it and in this case, red is chosen because according to [8], red has high spectral intensity compared to the other colour. The basic line sensor circuit of using red LED and LDR is shown in figure 8. The sensor senses line, the output of the sensor circuit will be logic or vice versa. However, due to the fact the sensors rely on the spectral intensity of the LED, the different ambient light condition will affect the logic output from the circuit and user would need to repeatedly fine tune the circuit. Due to the similar threshold between line and background, the line follower is affected by different ambient of light. Figure 7 shows the line following algorithm flow chart.

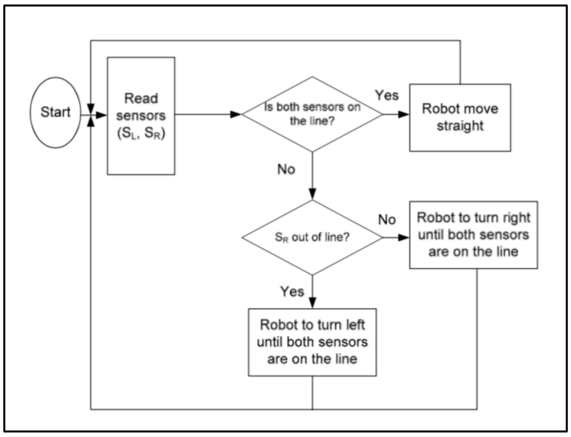

Figure 7 *Line follower flow chart*

Because the algorithm used in this line follower only utilize two middle sensors (SL and SR) for navigation shown in Figure 8, the robot tends to overshoot when following the line due to the sensors locating too close to the surface to get an accurate reading. Thus, another set of algorithm is implemented to search for line.

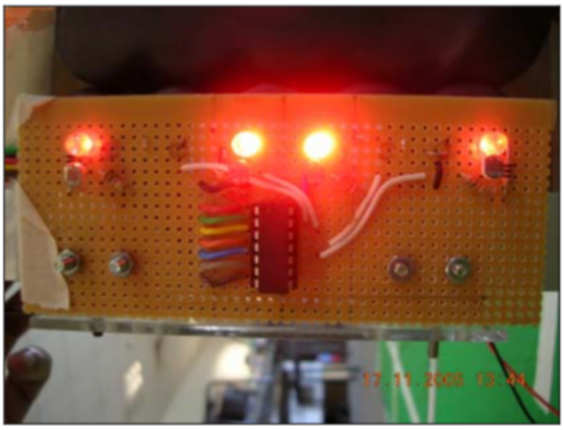

Figure 8 *Auto tuning sensor array*

In conclusion, the method used by author to track lines using LDR yields successful result in line following but some navigational problems will occur at higher velocities due to the sensor not able to pick up line marking on the floor.

2.2.4 Autonomous Mobile Robot with GPS Navigation and Ultrasonic Obstacle Avoidance System

In this paper, the author presents an autonomous robot that will navigate base on GPS navigation to determine the position of the robot [9]. The aim of this project is to identify the GPS navigation for the robot based on the waypoint that is pre-set to the GPS module and the sonar sensor, which then send input to the microcontroller to control the motor for navigation as shown in figure 9. With the pre-set waypoint, the autonomous mobile robot can navigate through desired waypoint and at the same time apply obstacle avoidance rules.

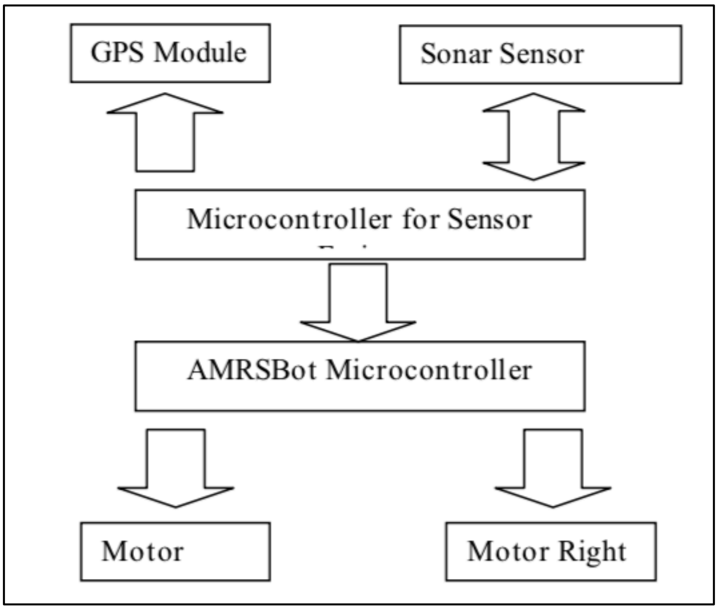

Figure 9 *Block diagram for microcontroller input and output.*

The GPS module used by the author is Garmin E-Trex Vista to coordinate the autonomous robot. Many consideration was done to choose the suitable GPS and among them are power requirement, which is 3v that needs longer power for localization and navigation. Besides, the chosen GPS is small in size making it easy to handle and be used in rugged terrain. The accuracy of the GPS is up to +/- 2 with +/- 5 degrees extreme northern and southern latitudes and 1-degree resolution.

The data that is suitable to be used to conduct in this experiment are GPRMC, GPGGA, HCHDG to gather the coordinate of the robot based on latitude, longitude and, bearing. The data is used to navigate the robot is shown in Figure 10.

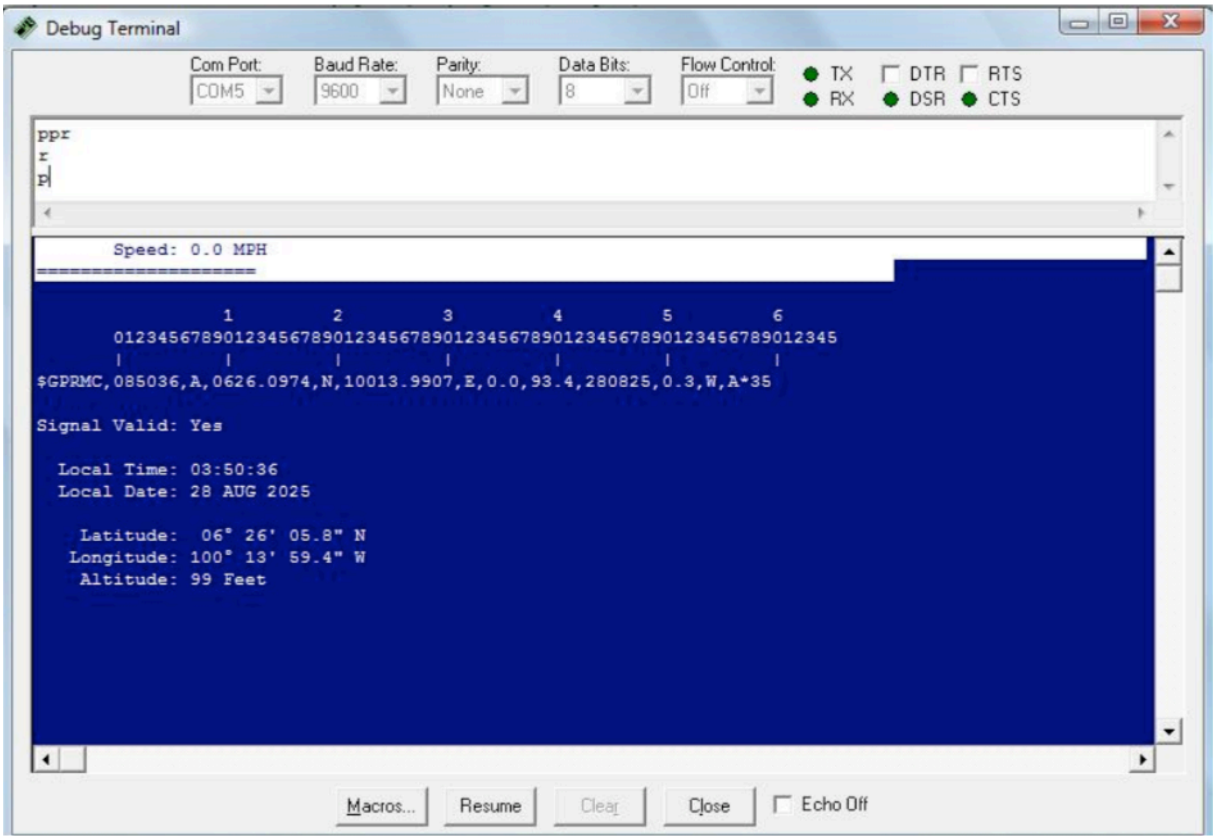

Figure 10*GPRMC data parsing*

Based on the data gathered from the GPS module, a microcontroller is used for decision making to control the DC motor of the robot. From the gathered data, the microcontroller can decide whether to move left or move right to correct the error of the system.

The robot is tested according to a pre-set waypoint with obstacle avoidance implemented. Based on the testing, the accuracy of the robot is heavily dependent on the signal from the satellite with variation between 4 to 12 input satellites. Thus, the accuracy of hitting the mark can drift from 10 to 12 feet. If the waypoint from start to end is within 12 feet, the start/end will be recognised as 1 point due to the accuracy of GPS is only up to 20 feet, any point within 20 feet will not be recognised by the GPS. Apart from that, the GPS module will not work if the robot is located underground or places with weak signal resulting in the robot to lost track from the predefined waypoint.

2.3 Comparison among different line following technique

Line follower method	8 IR and Quadratic line interpolation	2 linear IR black line follower	Ultra-bright red led and LDR	GPS
Accuracy	Very Accurate	Accurate	Accurate	Average
Surface with low reflectance value	Inaccurate	Inaccurate	Inaccurate	Not affected
Follows coloured lines	Inaccurate	Inaccurate	Inaccurate	Not affected
Line need to be a certain width	Yes	Yes	Yes	Not affected
Follow line under poor lighting	Weak	Weak	Weak	Not affected
Works when there are breaks in line	Inaccurate	Inaccurate	Inaccurate	Not affected
Works under covered area	Yes	Yes	Yes	No

Table 2 *Comparison of different line following techniques.*

Based on the literature reviewed, most of the line follower implements IR sensors for line detection. The achieve better accuracy, some method uses up to 8 IR sensors to get better coverage of line to calculate the distance the robot deviates from the center position. Whereas some approach uses an ultra-bright red led and a light dependent resistor to get a better reading of the reflectance value from the surface. Despite the amount of IR is being used, the main disadvantage of IR sensor is that is suffering from the condition of the line and it depends a lot on the type of surface. If the surface has a low reflectance value, the reflected value will not be accurate enough for the sensor to differentiate between the line from the surface. Due to IR rely heavily on the reflectance value, the colour of the line with same reflectance value with the surface will also give a false reading for the sensor such as a red line with a green surface because they both have similar reflectance value. Apart from that, the line follower that implements IR sensor tends to astray from the incomplete line and blurry line because the breaks between line will reflect a false and incorrect value for the IR sensors. Besides, for IR sensors to get an accurate reading, the line on the surface needs to be a certain width so that sensors will not detect the two edges present in a straight line.

The other method being reviewed is through GPS to plan a path for the robot to follow. Instead of an actual line, a virtual line is pre-set for the robot to follow based on the coordinates. Although this method does not suffer from the condition of line, but the accuracy of the destination and start point needs to be in a certain distance for the satellite to distinguish them. Finally, GPS will not work under close area such as a tunnel due to weak signal from the satellites.

Due to the limitation of IR sensors and GPS, I have decided to implement a line following system that is based on camera input. The camera will be a good substitution of IR sensors because the camera does not depend on the reflectance value which is easily affected by the condition of line and surface. Whether a line if blurry, incomplete or no reflectance value, as long as a line is distinguishable from the surface, appropriate image processing techniques can be done to detects the line presence on the surface

2.4 Overview of obstacle detection technologies

2.4.1 Light Detection and Ranging (LIDAR) sensors

LIDAR is a remote sensing method that adopts the principle of light in the form of a pulsed laser to measure ranges [10]. LIDAR detects obstacle by firing a pulse of laser light at a surface ¬up to 150000 pulses per second and uses a sensor on the instrument to measure the time taken for the pulse to bounce back from the targeted surface [11]. Because the speed of light can be measured and light travels at a constant rate, the distance between the LIDAR and the target can be derived in (1) where distance equals to speed multiply by time. The distance is divided by 2 because the time involves the propagation forward and backward from the target.

$$Distance = \frac{Speed\ of\ Laser\ Light\ x\ Time\ Taken}{2} \tag{1}$$

According to [12] LiDAR can acquire precise data collection and accuracy making it one of the most preferred remote sensing technologies. Despite the advantages, LiDAR comes with a high cost in some applications, affect by environment factor such as rain and, the performance degrades under high sun angles and reflections because the laser pulses depend on the principle of reflection [12].

2.4.2 Radio Detection and Ranging System (Radar)

In [13] RADAR is a system use to detect location and distance of an object using electromagnetic waves. Energy is radiated into space to receive echo or reflected signal from the objects to identify its location. Microwaves is used in RADAR due to the short wavelengths that is able to reflect at small objects. The distance is derived from the product of speed of electromagnetic waves and time take divided by 2 round trip in (1).

$$Distance = \frac{Speed\ of\ Electromagnetic\ Waves\ \ x\ Time\ Taken}{2} \tag{1}$$

According to [14], RADAR is good when comes to getting the exact position of an object and can penetrate mediums such as clouds, fogs, mist, and snow. Besides, RADAR can also track several objects in a different location. Despite the advantages, RADAR can be interfered by several objects and mediums in the air (other signals), and it comes with high cost.

2.4.3 Ultrasonic Sensors

An ultrasonic sensor is a device used to measure the distance between an object from the source using sound waves [15]. Ultrasonic sensors detect obstacle by emitting an ultrasonic wave via the transmitter and receives the wave reflected back from the target to the receiver. According to [15] the distance is derived from the product of the speed of sound and time taken between emission and reception in (1)

$$Distance = \frac{Speed\ of\ Sound\ \ x\ Time\ Taken}{2} \tag{1}$$

According to [16], the ultrasonic sensor has the capability to sense all types of material, not affected by environment factor such as dust, rain, snow, etc and can work in adverse conditions. Despite the advantages, Ultrasonic sensor is very sensitive towards different temperature and does not get an accurate reading from soft, curve, thin and small objects.

2.4.4 Infrared sensors

According to [17] Infrared sensor is an electronic device that emits and detect infrared radiation to sense the surrounding. The infrared signal is basically emitted by a Light emitting diode(LED) and the infrared signal that bounces from the surface is received by the infrared receiver. The distance is then derived by the speed of the infrared signal and time taken over 2 round trip in (1).

$$Distance = \frac{Speed\ of\ Light\ \ x\ Time\ Taken}{2} \tag{1}$$

According to [18] infrared sensor detect motion in either night time or day time, provides secure communication because of point to point communication and, can measure soft objects that are hard to be detected by the ultrasonic sensor. Despite the advantage, the infrared sensor only works at a shorter range and the performance degrades if the object is far away. Besides, infrared frequencies can be affected by smoke, dust fog or sunlight.

2.5 Comparison among different obstacle detection technique

In [19] shows the comparison among the different type of obstacle detection technologies.

Features	LiDAR	RADAR	Ultrasonic Sensor	Infrared Sensor
Type	Laser Light	Radio Wave	Sound Wave	Infrared Radiation
Range	~ 100m	~0.15 – 250m	~5m	~5m
Proximity	Poor	Good	Very Good	Very Good
Performance under snow/fog/ rain	Average	Very Good	Very Good	Weak
Performance under bright light	Very Good	Very Good	Very Good	Weak
Performance under dark light	Very Good	Very Good	Very Good	Good
Cost	$70000	$50 ~ $200	<$50	<$50

Table 3 *Comparison of different sensor technologies [19]*

Based on table 3, it provides valuable insights for me to choose the type of technologies I am using for my system. Out of the 4 reviewed technologies, they all use the same formula to derive the distance from objects which is the product of speed and time divided by two round trips. The main difference is the type of medium each technology used to propagate towards the target which is laser light, radio wave, sound wave and, infrared radiation respectively. Different medium performs differently in different condition. Under certain environment condition such as fog/snow/rain, infrared sensor perform the worst because infrared frequencies will easily get affected. In terms of range, LiDAR sensor and Radar sensors can be used to measure long distance object whereas Ultrasonic Sensor and Infrared Sensor work very good in short distances. Besides, Infrared sensor performs the worst under bright light because of the electronic spectrum of infrared are sensitive against light. Under dark light, all technologies perform well. Finally, the LiDAR sensor cost the most at around $70000 whereas the rest is within the range of $100. Due to financial constraint, LiDAR will not be considered in this small project although it performs very well from the rest. Out of the 4, I have chosen Ultrasonic Sensor for obstacle detection in my system because it is the cheapest among the others and it works well under different conditions. Although the range of Ultrasonic sensor is short as compare to LiDAR and RADAR, the short distance measurement is enough to implement in my small project using a Lego prototype model.

2.6 Conclusion

In conclusion, multiple methods for line following and obstacle detection are being compared. Based on the findings, an approach using camera for line following and ultrasonic sensor for obstacle detection will be used for developing the system in this project.

3 Development Methodology

The main objective of this project is to develop a line following vehicle and to determine whether the line following using a camera is a better approach compared to other methods such as IR sensors and GPS. Apart from line following, the system also comes with various functions such as obstacle detection and a stop and wait procedure if a certain line pattern(wide line) is detected. The stop and wait procedure is a representation of a bus making a stop at the station which the system targets to implement. The specific objectives are (i) Design a system that is able to follow line drawn on surface (ii) Design a system that is able to avoid obstacle (iii) Design a system to stop and wait if a certain line pattern is detected (iv) Implement the line following and obstacle avoidance system in a model car (iv) Test and evaluate the system in a model course. In this section, detail task and description about what is done at each stage to achieve the objectives are described in detail including development technologies, development environments and, development models involved in conducting the project.

3.1 Requirement gathering

The requirement gathering methodology I have used for this project is making research and study existing research projects related to line following, IR sensors, and, existing systems. Other forms of requirement gathering such as interviews were also conducted at a Lego EV3 gathering event for a better understanding of how a typical line follower works in the Lego world.

An interview was conducted with Yip Kau Kean, a Lego EV3 expert who specialises in developing a program for a robot using Lego EV3. The interview was conducted to gather information and data on the approaches towards developing a robust line follower robot. According to him, he agrees with me to use a camera instead of IR sensors like the majority of the approaches because based on his experience, IR sensors are sensitive and only work under certain condition. Apart from him, an interview was also conducted with Ahmad Sahar, a teacher who runs a robotic workshop specialised in Lego Mindstorms (EV3, NXT). He also provided me some of his insights on using a camera to follow line will achieve better results as well as using a third-party firmware that is compatible with EV3 such as leJOS and EV3 dev because they provide more flexibility in programming my robot.

3.2 Software Development Methodology

The software development methodology I have used is the Modified Waterfall Model. The development stages include Requirement Analysis, System Design, Implementation, Testing and, Development and Maintenance. Unlike the traditional Waterfall Model where it cannot go back to the previous stage once development has started, making the development process not flexible, I have used a Modified Waterfall Model as shown in figure 11. Modified Waterfall model enables the development phases to overlap and made able to split it each stage into subproject to allow changes to be made during development. This model is easy to understand and use because each phase has specific deliverables with the review process and deadline, making my system development to be manageable and delivered on time. Besides, it is also easier to implement this model for a small project like this where the requirements are very well understood.

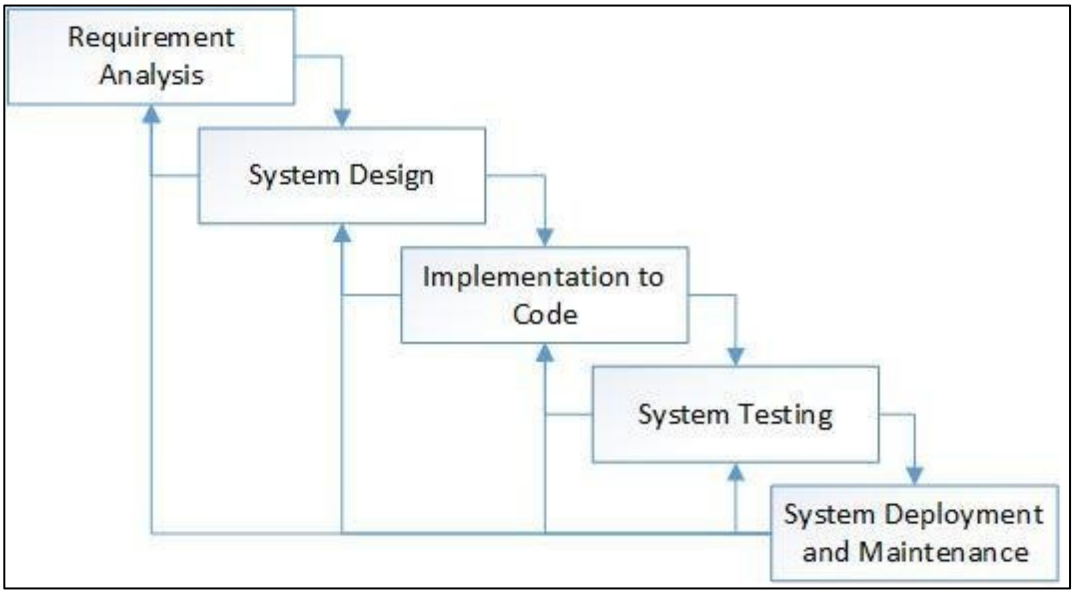

Figure 11 *Modified Waterfall Model*

3.3 Development Technologies

This section describes the detail and specification of the development technologies I have used to develop the line following and obstacle detection system. The development technologies can be categorised into Hardware components and Software components.

3.3.1 Hardware Components

EV3 intelligent brick

The Lego Mindstorms EV3 intelligent brick gathers information coming from sensors such as ultrasonic sensors, colour sensors and, gyro sensors. The information from the sensors will be processed and sends commands to the actuators [20]. The EV3 brick can also communicate with computers, smartphones via Bluetooth or Wi-Fi because it comes with a USB port. Among what is available to me which is Arduino and EV3, the reason I have chosen Lego EV3 to develop my system is because it comes with motor and sensors ready to plug in and play which I need not solder them separately such as Arduino and Raspberry pie. Besides, EV3 also supports different firmware such as leJOS which gave me the flexibility to develop the system using Java programming language. Finally, EV3 also have a much higher RAM compare to Arduino which is 64MB and 16MB (Arduino with highest RAM) respectively.

Figure 12 *EV3 intelligent brick*

Specification:

- ARM9 microcontroller
- 16MB flash memory
- 64 MB RAM
- 2.0 USB interface for Wifi connection
- SD memory slot
- 4 input ports and 4 output ports for sensors and motors
- Open source Linux OS

EV3 regulated Large Motor

The EV3 large motor is the most powerful motor among the other motor comes in the EV3 development kit. It uses tacho feedback for precise control of the motor rotation within one degree of accuracy. The motors come with a built-in rotation sensor to get the degree of rotation and made align with other connected motors.

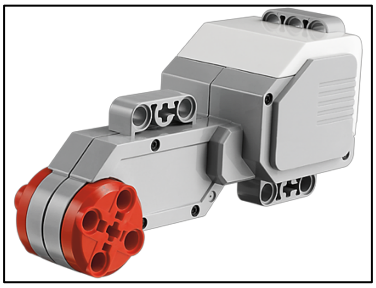

Figure 13 *EV3 Large Regulated Motor*

Specification:

- 160-170rpm
- Torque: 20N/cm
- Tacho feedback to +/- 1 degree of accuracy

EV3 regulated Medium Motor

The EV3 medium motor has the same feature as the large motor except that it has a higher rotation speed and lesser torque.

Figure 14 *EV3 Medium regulated motor*

Specification:

- 240-250rpm
- Torque: 8 N/cm
- Tacho feedback to +/- 1 degree of accuracy

EV3 ultrasonic sensors

The EV3 ultrasonic sensor generates sound waves and the receiver inside the sensor reads the echoes to detect and measures distance from objects in front.

Figure 15 *EV3 ultrasonic sensor*

Specification:

- Measures distance between 1cm and 250cm
- +/- 1cm accuracy
- Constant front illumination of ultrasonic waves

Logitech C170 webcam

The Logitech C170 webcam is used as my line follower vision to detect lines on the surface. I am using a third party camera because the Lego EV3 development kit does not come with their own camera.

Figure 16 *Logitech C170 webcam*

Specification:

- Video Capture XVGA up to 1024 x 768 pixels
- High Resolution 5 MP Photos
- USB 2.0

WI-FI dongle

Wi-Fi dongle is a device that connects smartphone, laptop, etc and allows me to access the internet. A Wi-Fi dongle is needed to connect EV3 brick to a router(wirelessly) to stream video capture from the Logitech camera to my laptop to get a real-time view and made easy for me to debug and perform digital image processing of the line drawn on the surface.

Specification:

- IEEE 802.11n: 150Mbps downlink and uplink
- USB 2.0

USB extension port

A USB extension port is used to plug in the WI-FI dongle and Logitech C170 webcam because the EV3 intelligent brick only comes with one USB port by default.

Specification:

- USB 2.0
- Data transfer speed: 480Mbps
- 4 port

SD Card

An SD card is a small flash memory card designed to provide high-capacity memory in a small size. The reason I have used a microSD card is to install leJOS firmware to be used with the EV3 intelligent brick. Through this method, the EV3 brick runs the firmware on this SD card whereas the original EV3 firmware can still persist in the brick.

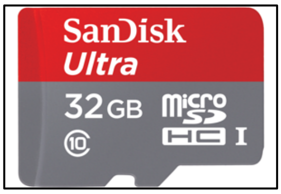

Figure 17 *32GB microSD CARD*

Specification:

- 32GB storage

Router

A router is used to allows communication between my PC and EV3 brick. Both my PC and EV3 brick are connected to the same router and the router forwards data packets from the EV3 that act as a server to my PC to display the video captured from the webcam.

Specification:

- TP-LINK
- 1200Mbps wireless speed
- Dual band 2.4Ghz and 5 Ghz

3.3.2 Software Components

leJOS firmware: version 0.9.1

leJOS is a firmware replacement for Lego Mindstorms programmable bricks that includes Java virtual machine for the system to be programmed in Java programming language. The reason I have to use leJOS firmware to replace with the default EV3 firmware is that leJOS promotes flexibility to program my system as compare to EV3 which use drag and drop feature to program. Besides, EV3 do not support image processing library such as OpenCV and JavaCV, thus, leJOS firmware is necessary to replace EV3 firmware to do image processing.

OpenCV: version 2.4.1

OpenCV (Open Source Computer Vision) is a library of programming function for real-time image processing. OpenCV supports multiple programming languages such as Python, Java, C++, C. It’s a vast application which includes object tracking, facial recognition that comes with various features provide assistance for me to develop the line detection. For this system, I have chosen to use OpenCV 2.4.1 instead of 4.0(latest) is due to the EV3 brick not being very powerful and thus leJOS only support until version 2.4.1.

Java: version 1.7

Java is a popular programming language that provides a system for developing application software and deploying it in a cross-platform computing environment. The reason I have used Java to program the line following system is that leJOS firmware only supports Java programming language. Besides, Java also provides their own core class libraries which makes it easier for developers. Java version 1.7 is used instead of the latest version is due to compatibility issue.

Eclipse IDE: version 4.9

Eclipse is an integrated development environment used in programming especially Java. Eclipse IDE is used to code the line follower because it has an easy navigation system due to its package class hierarchy view, error highlighting, and, class method outline view allows users to navigate along with the classes without the need to going through all the folders.

3.4 System requirements

3.4.1 Introduction

This section discusses the Software Requirements Specification for the line following and obstacle detection system. Requirement analysis is an important aspect of software development because it encompasses all the tasks that determine the needs or conditions to meet the objectives of a system. A poor requirement analysis will result in developing an incomplete software with poor functionality. Thus, to gather all my requirements, research through literature review, existing system, journals, etc are conducted. Apart from that, interviews and discussion are also conducted with people who are expert in the field of line following and EV3 to gather their feedback and comment in developing of a line following and obstacle detection system. All the requirement of the system is defined in this section.

3.4.2 User Requirements

The user requirements or known as users' needs describes what the users does with the system (i.e. interaction with the system). The users requirements are listed down below:

No.	Description
UR1	The system has a button to calibrate the wheels for users to make the steering align.
UR2	The system has a button to stop the vehicle.

Table 4 *User Requirements Specifications*

3.4.3 Functional Requirements

The functional requirements defines what a system should do(i.e. technical functionality of a system). The aim of this project is to investigate if line following based on camera is better than IR sensor. A line follower with obstacle detection will be developed and the functional requirements are listed down below:

No.	Description
FR1	The system will detect any colour of lines drawn on surface.
FR2	The system will detect any obstacle in front within the range of 1cm to 25cm.
FR3	The system will command the steering motor to steer left/right according to the position of the line on surface.
FR4	The system will command the drive motor to move forward.
FR5	The system will command the drive motor to stop upon detecting any obstacle in front.
FR6	The system will resume driving operation when the obstacle in front has cleared.
FR7	The system will stop for a period of time and continue moving forward if a line pattern is detected. The line pattern use in this case is a wide line.

Table 5 *Functional Requirements Specification*

3.4.4 Non-functional Requirements

The non-functional requirements are the requirements that contain specific criteria used to define the operation of a system. Often referred as quality attribute, the non-functional requirement of line following system are as follows:

Type	Description
Reliability	The line following system should execute all the programmed operation properly without being aborted.
Performance	The system should be able to responds quickly to the change in direction of line and obstacle that suddenly appear in front. The system should react within 2 seconds when any changes from the input is detected.
Flexibility	The line following system should be able to be modified to enhance the performance and also adapt with different environments.
Interoperability	The line following system should be able to work with other firmware such as Arduino and Raspberry Pi.

Table 6 *Non-Functional Requirements Specifications*

3.5 System Analysis

3.5.1 Introduction

This section describes the analysis to develop the system. The section is further divided into system analysis and system design. The system analysis involves the process of collecting facts and interpreting facts about the system to be developed and identify the goals and purpose of the system. The system design involves the process of designing the layout(architecture, modules interface, etc) and flow of the system during execution. Often, it involves prototyping from low fidelity to high fidelity until a system ready to be deployed for development is achieved.

3.5.2 Block diagram for Line Following and Obstacle Detection System

This section depicts the overview of line following and obstacle detection system using a block diagram. A block diagram is a representation of a system where components are represented by blocks connected by lines that show relationships of blocks.

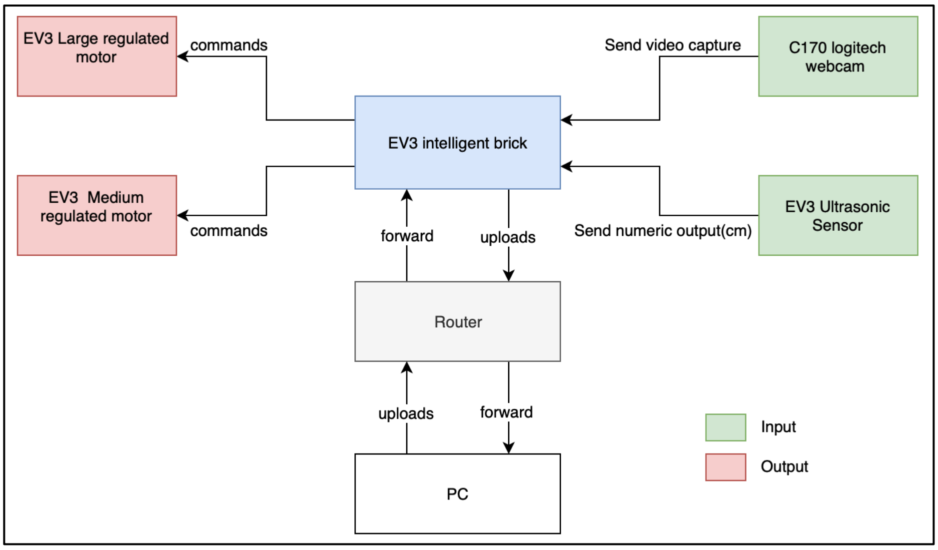

Figure 18 *Block diagram of the system*

Figure 18 shows the interaction between different components in the system. Based on the figure, the PC uploads executable Jar file to router and router forwards it to the EV3 brick. Upon receiving the Jar file, EV3 brick will execute the code and get input from Logitech webcam and EV3 ultrasonic sensor that is both connected to the brick. The webcam and ultrasonic sensor will send video capture and numeric value(cm) as input for the EV3 brick respectively and the EV3 brick will process the input. Based on the processed input, the EV3 brick then commands the motor to rotate corresponding to the position of the robot based on the input to follow the line. During the execution of the program, the EV3 brick will also upload streams of data(video capture) acquired from the webcam which will be forwarded to the PC to display real-time view from the line follower on my PC.

3.5.3 Flow Chart Diagram for Line Follower and Obstacle Detection

A flow chart is a visual representation of a workflow which includes the sequence of steps and decisions needed to perform a process. It is useful to help readers in understanding how each step is linked to one another during execution of a program. The flow chart shown in figure 19 represents the workflow of line follower and obstacle detection system. The program begins when user press the start button.

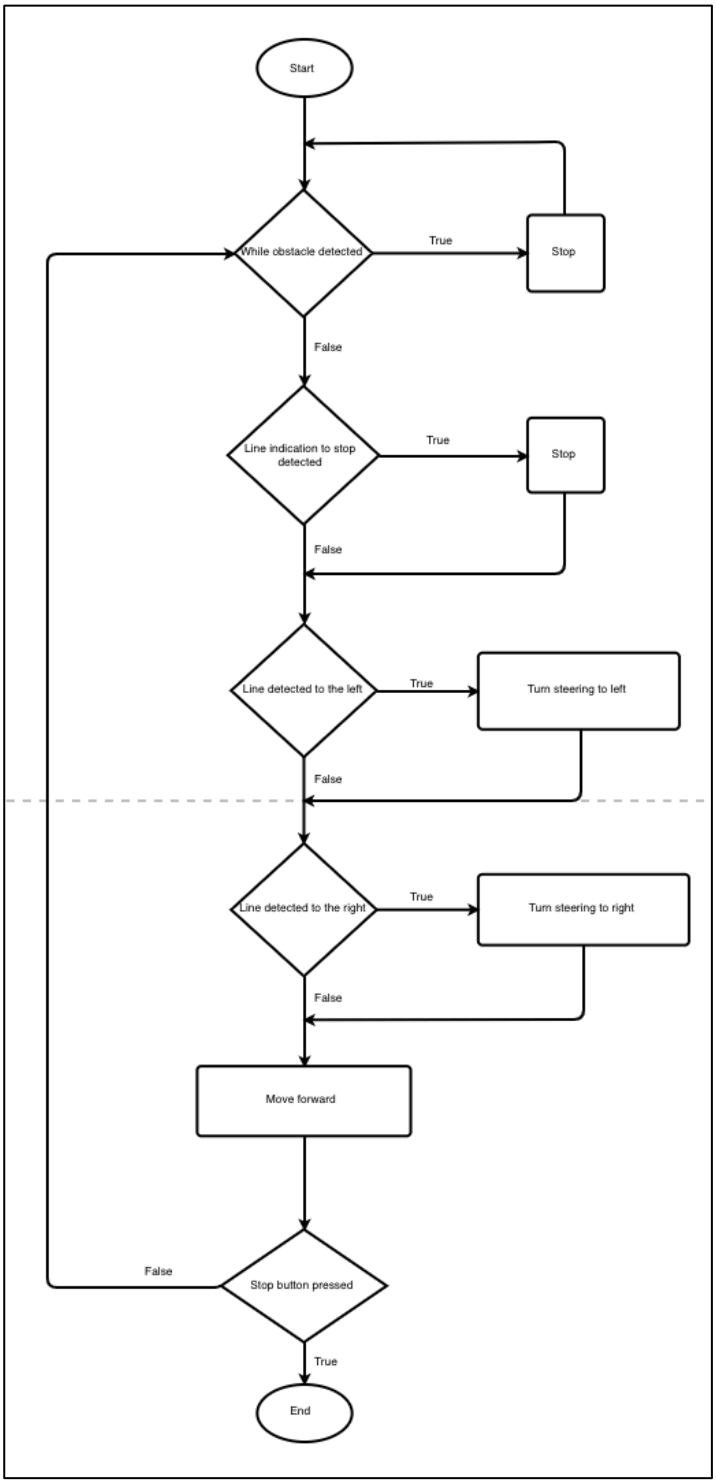

Figure 19 *Flow Chart diagram for line follower and obstacle detection*

The first action of the flow chart involves the user to run the program from the IDE. Upon running the code, the PC will upload an executable Jar file to the EV3 brick to execute the program. The system then checks if there is any obstacle in front via the ultrasonic sensor. If an obstacle is detected within the pre-set range, the system will stop the drive motor from moving forward and checks again until the obstacle in front has cleared. If no obstacle is detected, the system then commands the drive motor to move forward and start to detect line on the surface via images captured from the camera mounted above the vehicle. During the line detection, the system then checks if the line has any indication(represented with a wide line) for the vehicle to stop, if detected the system will stop the drive motor from moving forward and wait for 5 seconds before continue moving forward. If no indication to stop is detected the vehicle will continue to move forward. Apart from that, if the line is moving towards the left, the steering motor then steers the vehicle to the left or vice versa. If the line is neither moving to the left or right, the drive motor will continue to move the vehicle forward and repeat the above process again until a stop button is pressed by the user and the line following system will end.

3.6 System Design

3.6.1 Introduction

This section describes about the design procedure taken to develop the system. The system design involves the process of designing of the layout(architecture, modules interface etc) and flow of the system during execution. Often, it involves prototyping from low fidelity to high fidelity until a complete design is achieved.

3.6.2 System Environment

To design the system environment, I have used Lego EV3 development kit to program the lane following system along with obstacle detection. Lego EV3 is a software platform for the development of programmable robots which comes with an EV3 intelligent brick that gathers information coming from the EV3 sensors, processes them and commands the actuators according to the way it is being programmed. For this system, two motors are used to drive and steer the vehicle, one ultrasonic sensor for obstacle detection and, a third party C170 Logitech camera to track lines
Because the system requires image processing which the original firmware by EV3 does not support, I have use leJOS firmware which supports OpenCV for me to do image processing.

To run leJOS firmware, I have downloaded leJOS into a 16 GB microSD card, the microSD card is then slotted into the SD Card port on EV3 brick. Once the EV3 brick load up, it will run the leJOS firmware downloaded in the SD card. By using an SD card, the EV3 firmware still persists and I do not need to overwrite it while I have an additional firmware available through SD card. With leJOS firmware available in the brick, I have installed the Eclipse integrated development environment (IDE) and leJOS eclipse plugin to program the system in Java programming language. Java version 1.7 is used due to compatibility issue.

With leJOS running in the brick and Eclipse IDE for me to program the brick, the written program needs to be uploaded to the brick in order to execute the program. To do so, I have used a router and Wi-Fi dongle which is slotted in the USB port on EV3 brick to make a connection between my PC and EV3 brick by sharing the same IP address. Once connected, the program in my PC is uploaded as a Jar file to the EV3 brick wirelessly to execute the program.

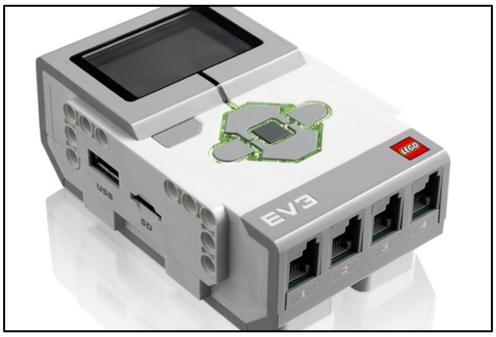

Figure 20 EV3 with their respective USB and SD slot on the side

After setting up the programming environment, I then installed OpenCV 2.4.1 on my PC to do image processing with leJOS firmware. To do image processing in the brick, a camera is needed to get feed input from the surrounding to the EV3 brick in order to process image. With this, I have used a Logitech camera to work with the EV3 brick. The Logitech camera is plugged into a USB extension port on EV3 brick to capture the lines on the floor. A USB extension port is used to plug in the wi-fi dongle and Logitech camera as shown in figure 21 because EV3 only comes with one USB port. To make it easier for me to debug and do image processing, I have stream video from the camera view to my PC by using a Java web server running on the robot and establish a TCP connection to exchange streams of data. The streams of data are open in a browser to display the captured view.

Figure 21 *USB extension port with wi-fi dongle and camera plugged in*

After setting up the camera for image processing, I proceed with connecting the hardware components from EV3 development kit to the EV3 brick. For the system, I have used a large motor to drive the vehicle, a medium motor to steer and an ultrasonic sensor to detect the obstacle. The components are plugged into their respective ports build into the EV3. The ultrasonic sensor goes into the input port for sensors whereas large motor and medium motor are plugged into the output port as shown in Figure 22.

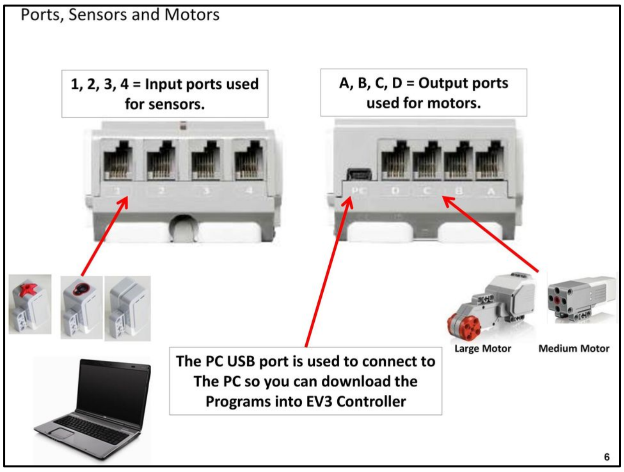

***Figure 22** Input and Output*

3.6.3 Real Time Video Display

Figure 23 shows the screen that display the camera view of the vehicle. A real time image captured from the moving vehicle will stream to the PC for user to see how the system track lines. Besides, the display also makes it easier for me to debug and program the line following system.

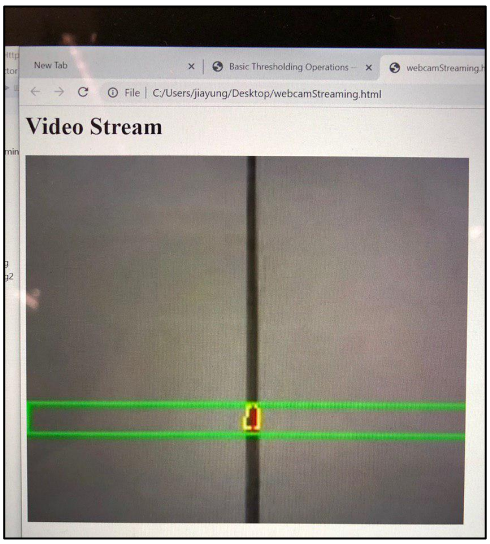

***Figure 23** Video stream of the line follower.*

3.6.4 Building the Lego model(Prototype)

A Lego model is necessary for the implementation of the system because the system is just a proof of concept, it will not be tested in a real-life vehicle. In this case, a prototype Lego model will be used as a replacement for a real-life vehicle. To build the model, I have used Lego technic parts which consist of Lego interconnecting plastic rods and parts to create an advanced model with various functionality. Lego technic is chosen due to the flexibility to scale my model and build any model that is able to fit in the EV3 module because they were made to combine with each other.

Figure 24 shows the Lego prototype model which adopt the concept of a vehicle with rear driving axle and turning steering angle as shown in Figure 25 and Figure 26 respectively. The Lego model consists of an EV3 Large motor for forwards driving, an EV3 Medium motor to control the front wheel for turning left/right, a camera mounted on top of the vehicle to capture a good line of sight of the line drawn on surface and, an ultrasonic sensor mounted in front of the vehicle to detect for any obstacle blocking the path while moving forward.

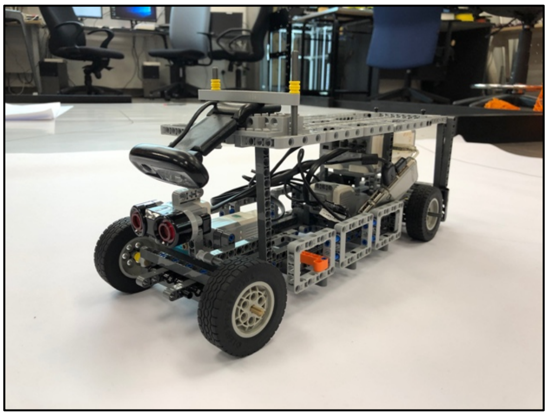

Figure 24 *Lego prototype*

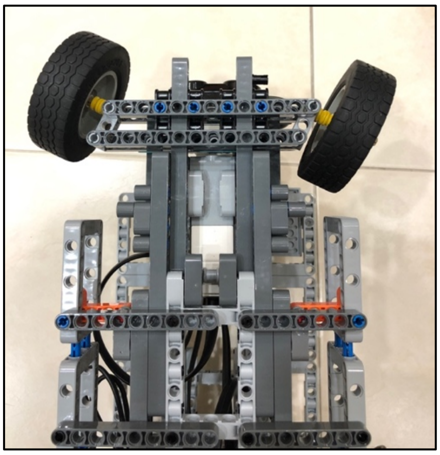

Figure 25 *Steering angle*

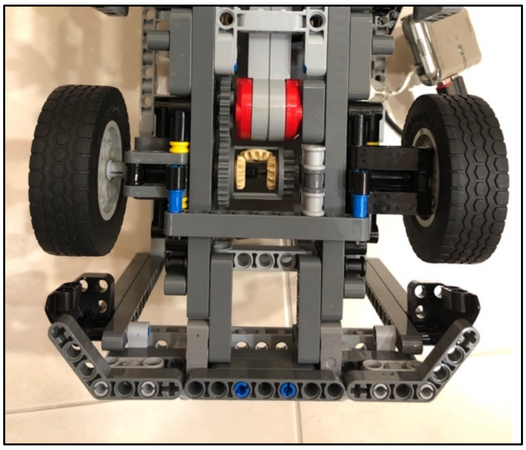

Figure 26 *Rear axle with differential mechanism*

For the turning function of the prototype Lego model, I have applied the concept of a real life vehicle steering angle as shown in Figure 27

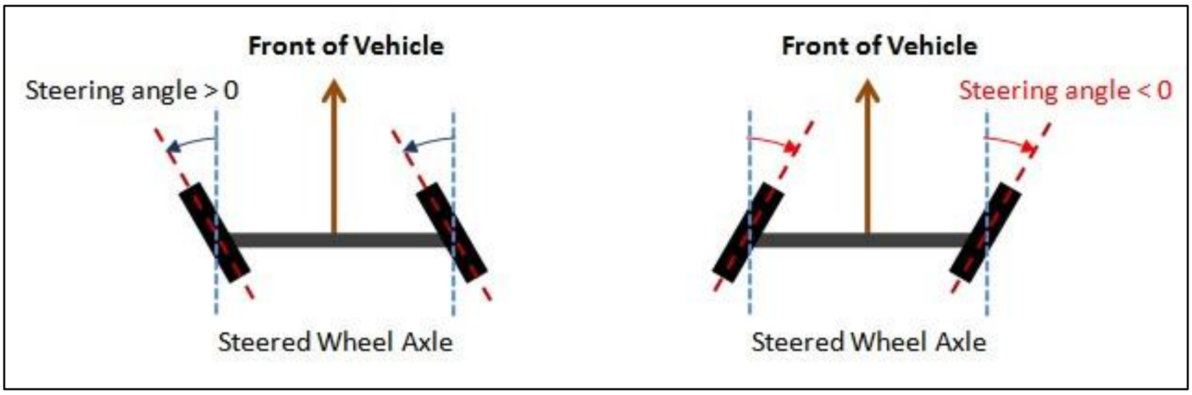

Figure 27 *Steering Angle*

As for the rear driving axle to drive the vehicle forward, I have adopted the concept of real-life vehicle by using a mechanism known as a differential. A differential is necessary to allow the model to turn left/right smoothly because the differential is designed to drive a pair of wheels while allowing them to rotate at a different speed while cornering. This is because the outer wheel will rotate faster to cover more distance while the inner wheel will rotate slower with a shorter distance to cover. Without a differential, the drive motor will force both wheels to rotate at the same speed even at cornering and causing the vehicle to swerve.

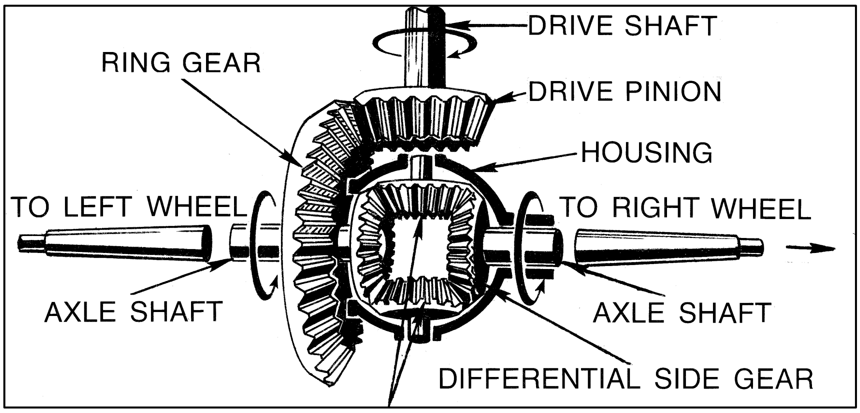

Figure 28 *Differential Mechanism*

3.7 System Implementation

3.7.1 Introduction

Implementation involves the actual coding of the line follower and obstacle detection. Java and OpenCV are the tools used to develop the program as explained in the development technologies. Eclipse IDE and leJOS firmware is the programming environment used for the development.

3.7.2 Line Detection

In order to detect any line present on the surface, the system uses a camera to get input from the environment. Using the input from the camera, every frame(images) captured by the camera is processed using OpenCV library function to detect for the presence of the line. Due to the limited processing power of the brick and to improve the performance, the image is converted to greyscale instead of RGB pane. At greyscale, images only need to be processed at one pane instead of the three color panes for RGB. To further improve the performance, only a region of the frame was processed instead of the entire frame which is done by cropping out the region to be processed known as a region of interest.

To get an accurate reading of the line, every frame was pre-processed to remove noise present in images, remove shadow and, fill in incomplete lines that will affect the detection rate. Blur function from OpenCV is used to remove noise that will cause random variation of brightness that degrades the quality of the image captured. Thresholding operation is performed to remove the shadow or any contamination on the surface or shadow due to illumination that will give a false reading where the system will detect it as part of the line. Because the line appears to be a darker colour compared to the surface, a threshold value is set to separate the contamination from the actual line. Any contamination that appears to be lighter than the line will be thresholded to white colour which is not part of the line. Morphological operation from OpenCV is also performed to fill in incomplete gaps in the line for better accuracy. Finally, the line is identified based on all continuous points along the boundary. Figure 29 shows the detected line with a yellow bounding box and a green box indicating the region of interest. The red line in between is the computed middle position of the yellow bounding box.

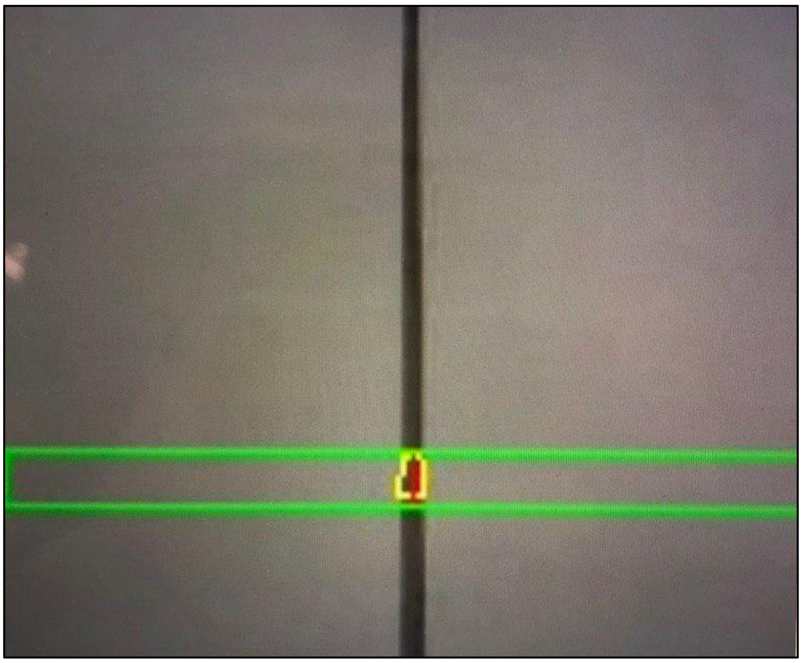

Figure 29 Line detection

3.7.3 Line Following

In order to make the system to follow the line, the system needs to send a command to the steering motor to turn the steering angle relative to the position of the line detected. The system will update the steering angle in every 10 frames to keep the vehicle centered based on the position of the line. If the line is turning to the right, the system then commands the steering motor to turn right or vice versa. The rotation angle the motor needs to turn is based on the distance of that line deviates from the center, the greater the distance, the greater the rotation angle of the steering motor. The steering motor will only be updated every 10 frames due to the limitation of processing power. If the motor is updated in every frame, there will be a delay in processing the image and updating the motor.

3.7.4 Obstacle detection and avoidance

In order to avoid the obstacle, the line follower will come to a stop if it detects any object blocking its path and continue moving forward if the obstacle no longer exists. To achieve this objective, I have used the ultrasonic sensor mounted in front of the vehicle to get the numerical reading of the distance between the sensor and obstacle. The transmitter of the ultrasonic sensor vibrates at a frequency above the range of human hearing to generate sound wave and receiver reads their echoes to detect and measure the distance from any objects in front between 1cm to 250cm [21]. Figure 14 shows the working principle of an ultrasonic sensor. From the numerical input of the ultrasonic sensor, the distance I had set to stop the vehicle upon detecting an object is 25cm.

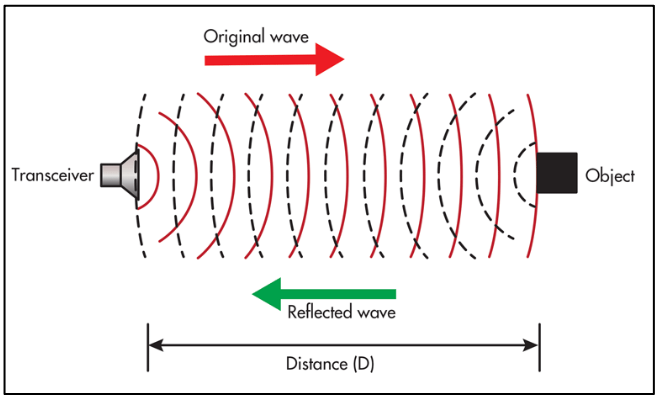

Figure 30 *Working principle of ultrasonic sensor*

3.7.5 Stop and Wait upon detecting pattern

In order to let the vehicle to stop and wait for a period of time when it comes to a line with wider markings, I have used image processing techniques to get the width of the line on surface. Based on the width, the section of the line which is wider will indicates the system to stop. The system then stop and wait on that section of line for a period of time before continue to move forward.

3.7.6 Real time video streaming

In order to provide an interface for the user to view the current state of line following system, I have make use of the EV3 embedded device as a web server that is connected to a router. The user PC will constantly make request for images from the web server, the web server then accepts the request and stream data bytes to the PC. Upon receiving, the PC then converts the data to images and display on the user web browser. The constant request of an image from the webserver will then generate a video like display for the user.

3.8 System Testing

3.8.1 Introduction

System testing is the testing of a complete system to evaluate the performance of the system and compliance against the specified requirements. During system testing, the functionalities are tested from end-to-end perspective which includes both functional and non-functional testing. The main objective of conducting system testing is to investigate whether line following using camera is able to follow line on surface and lesser restriction compare to IR sensors. The other objective includes

House of Multimodal Evolution Laboratory Lab (HOME LAB)

The system was conducted at HOME LAB situated level 3 Sunway University Building. The lab serves as a research area to provide novel and innovative multimodal applications and communications toward the technology world. This lab comes with a large open space area with good lighting to test my line following and obstacle detection system.

3.8.2 Overview

To conduct testing, a model course marked with line is used to evaluate the accuracy of the system. The course involves various turning angle, different line colour, incomplete section of line, obstacle place within the course and, a section with pattern(wider line) to evaluate the accuracy of line following, accuracy of obstacle detection and, accuracy of stop and go function respectively. Figure 31 shows the model course where the system will drive the Lego prototype model in the course for 20 rounds and Figure 32 shows the section mark with a wider line.

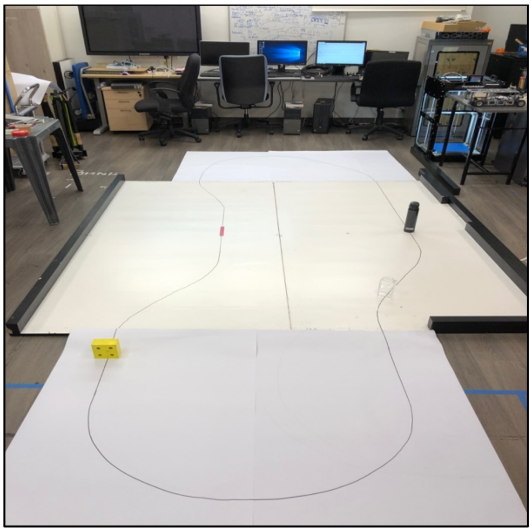

Figure 31 *Model Course*

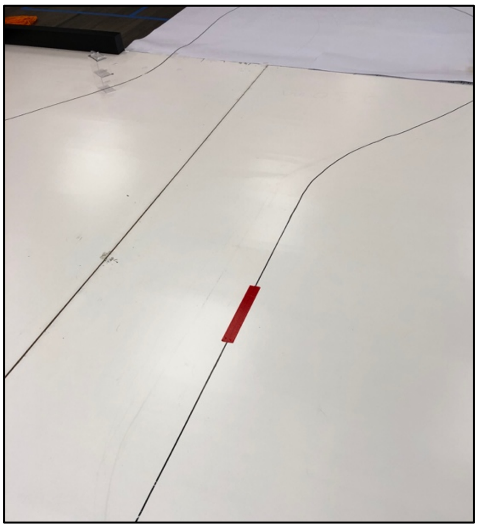

Figure 32 *A section mark with wider line as an indication for the vehicle to stop*

3.8.3 Line Following

To test the accuracy of line following, the model will be tested with 20 runs, and the accuracy will be derived from number of completed runs over total number of runs in (1). A run is consider complete if the line follower did not deviates away from the line during each run.

$$\text{Accuracy\%} = \frac{completed\ run}{total\ number\ of\ runs} \qquad (1)$$

3.8.4 Obstacle Detection

To test for the accuracy of obstacle detection using ultrasonic sensor, obstacle is placed around the course to find out if the system is able to detect for obstacle and stop the vehicle. Different material of obstacles which are opaque, transparent and, soft are used for testing to find out if the sensor reading is affected by the type of material as shown in Figure 33. The accuracy will be derived from the number of obstacle detected over the total number of obstacle in (1).

$$\text{Accuracy\%} = \frac{obstacle\ detected}{total\ number\ of\ obstacle} \qquad (1)$$

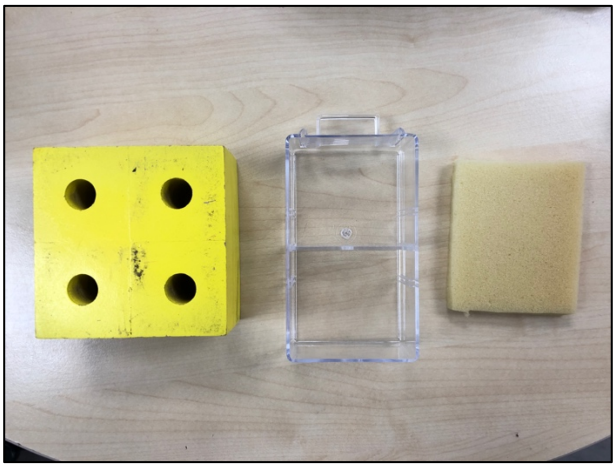

Figure 33 *Different type of obstacles. From left, opaque, transparent, cushion.*

3.8.5 Pattern Detection

To test the accuracy of recognising patterns on line that indicates a stop and wait procedure, pattern(wide line) will be marked on the surface to find out if the system is able to recognise it and make a temporary stop. The accuracy will be derived from the number of successful pattern recognised over the total number pattern in (1).

$$\text{Accuracy\%} = \frac{pattern\ recognised}{total\ number\ of\ pattern} \qquad (1)$$

4 Results and Discussion

4.1 Line Following

	Value
Total number of course	20
Number of course completed	17
Accuracy	85%

Table 7 *Results for accuracy of line following*

Based on the result of line following, we can notice that the line following method by using camera to track line is fairly accurate with an accuracy rate of 85%. Out of the 20 runs, there is 3 runs where the vehicle went away from the line due to the system not able to react fast enough when there is changes in direction of line, false detection of line where the image quality is affected by noise and, detection of line is affected by shadow. Despite the 3 incomplete run, the line follower is able to complete the rest 17 runs under blurry line condition, different colour of line and, line with different width.

4.2 Obstacle Detection

Transparent	Value
Total number of transparent obstacle	10
Number of transparent obstacle detected	10
Accuracy	100%

Table 8 *Results for transparent obstacles*

Opaque	Value
Total number of opaque obstacle	10
Number of opaque obstacle detected	10
Accuracy	100%

Table 9 *Results for opaque obstacles.*

Soft	Value
Total number of soft obstacle	10
Number of soft obstacle detected	4
Accuracy	40%

Table 10 *Results for soft obstacles.*

Based on the result of obstacle detection, we can notice that ultrasonic sensor is very accurate for measuring short distances object except for detecting soft object such as a cushion in this case. The detection rate of soft object scores only 40% compare to transparent object and opaque object, this is because the wave use by the sensor the reflect upon hitting any object is absorbed by the soft material and giving the sensor a false reading. Other than soft material, ultrasonic sensor works very well with transparent and opaque object by giving an accurate reading for the vehicle to stop.

4.3 Pattern Detection

	Value
Total number of pattern	20
Number of pattern detected	20
Accuracy	100%

Table 11*Results for pattern detection*

Based on the result of pattern detection on the surface, we can notice that the camera is quite accurate in detecting any changes of shape and size on the surface. Upon detecting, the system is able to carry out execution fast enough for the vehicle to stop on the surface mark with pattern to simulate BRT making a stop at bust stop while travelling on the BRT route.

4.4 Limitations

The first limitation of this line following and obstacle detection system is the speed of the vehicle. Due to the limited processing power, the line follower is not able to travel at a fast speed because the system is not able to process each images fast enough and sends command to the motor to take appropriate action. If the vehicle is moving too fast, the vehicle has a higher chance of moving away from the line.

Apart from processing power, the other limiting factor is environment factor. While detecting for the presence of line, environment factor such as poor lightening which results in shadow will cause the system to mistaken it as a line. Noises appear in the image also affect the quality of image

5 Conclusion and Future Work

5.1 Conclusion

In conclusion, the aim of this project is to design, develop and evaluate a line following and obstacle detection system that is implemented in a Lego prototype model. With autonomous driving becoming an increasing research topic, line following technology is one of the component to be researched in the autonomous field. The idea of developing a line follower and obstacle detection for a Lego model is to serve as a proof of concept that autonomous driving can be achieved based on line and using camera to achieve a better results in line detection compare to IR sensors. Although real life autonomous driving require much more advance system and resources, this project shows that a simple system can simulate how autonomous driving will work in real life. Besides targeting Sunway BRT, the system can also serve various application such as restaurants, industrial automated carriers, etc to replace human effort in today's technology.

5.2 Future Work

Although the requirements of the system developed has been met, extensive features of the system via image processing could not be implemented due to time constraints. Besides, the development of the system is further affected by the lack of processing power of the EV3 brick.

Although the line follower and obstacle detection system is working as it intended to be, there are still room extension and improvements because the accuracy of the current system is still far away from what is acceptable in the autonomous driving world that should be close to 100% to prevent any accident. Apart from that, line following is just part of autonomous driving out of all the data gathered from the vehicle which includes special awareness, communication with other vehicles on the road, recognising street board, etc.

The first improvement that can be done as part of my future work is to use a more powerful processing power so that my system is able to process images at a higher frame per second for the vehicle to react faster on any sudden changes of line on the surface. As the current processer only has 64MB of RAM, the amount of frame that can be processed per second is really limited. Thus, a more accurate and fast line follower can be achieved with a better processor.

The second improvement that can be done is to implement a better image processing method to eliminates shadow and contamination on the line because the current line follower is sensitive towards low ambient light and shadow present on the surface. The image processing method can only be done provided we have a better processor so that a more advanced and complex method can be applied to process the image.

As for the additional feature, I would like to add an accelerating and decelerating features. The vehicle will accelerate on a straight line and decelerate on cornering same as how human drive on road. With this feature, the BRT that implements this system will be able to reach the desired destination faster.

Through the improvements and additional feature, the system not only targets BRT but also has the potential to be applied on some other application such as industrial applications to carry equipment to the desired location at a factory, guidance application to provide path guidance to the public in museums, etc.

References

[1]"Human Error Causes 94 Percent of Car Accidents", *LawInfo Blog*, 2019. [Online]. Available: https://blog.lawinfo.com/2017/09/06/human-error-causes-94-percent-of-car-accidents/. [Accessed: 21- Jun- 2019].

[2]H. Rudra, P. Krishna and S. Pisipati, "A Simple and Efficient Lane Detection using Clustering and Weighted Regression", *Research Gate*, 2019. [Online]. Available: https://www.researchgate.net/publication/221324943_A_Simple_and_Efficient_Lane_Detection_using_Clustering_and_Weighted_Regression. [Accessed: 21- Jun- 2019].

[3]M. Mohammad, "A line follower robot from design to implementation: Technical issues and problems", *Academia*, 2019. [Online]. Available: https://www.academia.edu/972274/A_line_follower_robot_from_design_to_implementation_Technical_issues_and_problems. [Accessed: 21- Jun- 2019].

[4]"How to Make a Line Follower Robot in 10 Minutes | Arduino", *Maker Pro*, 2019. [Online]. Available: https://maker.pro/arduino/projects/make-line-follower-robot. [Accessed: 21- Jun- 2019].

[5]F. Zainab and A. Ulla, "Black Line Tracking Robot", 2019.

[6]"Lane detection for self driving vehicles", *mc.ai*, 2019. [Online]. Available: https://mc.ai/lane-detection-for-self-driving-vehicles/. [Accessed: 21- Jun- 2019].

[7]M. Engin and D. Engin, *Path Planning of Line Follower Robot*. 2019.

[8]Z. Baharuddin and I. Zainal Abidin, *Analysis of Line Sensor Configuration for the Advanced Line Follower Robot*. 2019.

[9]M. Abdul Hamid, A. Adom, H. Mohd and M. Fazalul Rahiman, *Autonomous Mobile Robot (AMRSBot) with GPS Navigation and Ultrasonic Obstacle Avoidance System*. 2019.

[10]"What is LIDAR?", *Oceanservice.noaa.gov*, 2019. [Online]. Available: https://oceanservice.noaa.gov/facts/lidar.html. [Accessed: 21- Jun- 2019].

[11]*Lidar-uk.com*, 2019. [Online]. Available: http://www.lidar-uk.com/how-lidar-works/. [Accessed: 21- Jun- 2019].

[12]"Advantages and Disadvantages of LiDAR", *LIDAR and RADAR Information*, 2019. [Online]. Available: http://lidarradar.com/info/advantages-and-disadvantages-of-lidar. [Accessed: 21- Jun- 2019].

[13]"RADAR - Introduction of RADAR Systems, Types and Applications", *ElProCus - Electronic Projects for Engineering Students*, 2019. [Online]. Available: https://www.elprocus.com/radar-basics-types-and-applications/. [Accessed: 21- Jun- 2019].

[14]"Advantages and Disadvantages of RADAR systems", *LIDAR and RADAR Information*, 2019. [Online]. Available: http://lidarradar.com/info/advantages-and-disadvantages-of-radar-systems. [Accessed: 21- Jun- 2019].

[15]"What is an Ultrasonic Sensor? | Sensor Basics: Introductory Guide to Sensors | KEYENCE", *Keyence.com*, 2019. [Online]. Available: https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp. [Accessed: 21- Jun- 2019].

[16]M. Lab, "Ultrasonic Sensor working applications and advantages", *Microcontrollers Lab*, 2019. [Online]. Available: https://microcontrollerslab.com/ultrasonic-sensor-working-applications-advantages/. [Accessed: 21- Jun- 2019].

[17]"IR (Infrared) Obstacle Detection Sensor Circuit", *Electronics Hub*, 2019. [Online]. Available: https://www.electronicshub.org/ir-sensor/. [Accessed: 21- Jun- 2019].

[18]"Advantages and Disadvantages of Infrared sensor", *Rfwireless-world.com*, 2019. [Online]. Available: http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Infrared-Sensor.html. [Accessed: 21- Jun- 2019].

[19]"From Google to Tesla, it’s a War of LiDAR or RADAR", *UnitedLex*, 2019. [Online]. Available: https://www.unitedlex.com/news/from-google-to-tesla-its-a-war-of-lidar-or-radar. [Accessed: 21- Jun- 2019].

[20]M. EV3, "31313 MINDSTORMS EV3 - Products - Mindstorms LEGO.com", *Lego.com*, 2019. [Online]. Available: https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313. [Accessed: 21- Jun- 2019].

[21]"EV3 Ultrasonic Sensor", *https://education.lego.com*, 2019. [Online]. Available: https://education.lego.com/en-us/products/ev3-ultrasonic-sensor/45504. [Accessed: 21- Jun- 2019].