ENHANCED ROBOTIC APPLICATION WITH TOPOLOGY FORMATION FOR SECONDARY SCHOOL RBT COURSE

BY

TAN KAI YING

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2020

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course

Academic Session: JAN 2020

I TAN KAI YING

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)	(Supervisor's signature)

Address:
9, Lorong Tanjung Indah 2,
Taman Tanjung Indah,
12300 Butterworth, Pulau Pinang.

Dr. GOH HOCK GUAN
Supervisor's name

Date: 18 April 2020	**Date**: 18 April 2020

ENHANCED ROBOTIC APPLICATION WITH TOPOLOGY FORMATION FOR SECONDARY SCHOOL RBT COURSE

By

Tan Kai Ying

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ENHANCED ROBOTIC APPLICATION WITH TOPOLOGY FORMATION FOR SECONDARY SCHOOL RBT COURSE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : TAN KAI YING

Date : 18 April 2020

ACKNOWLEDGEMENTS

I wish to express my sincere thanks and appreciation to my supervisors, Dr. Goh Hock Guan and Mr Teoh Shen Khang who has given me this splendid chance to take part in a robotic application based project. I am incredibly grateful and obligated to them for sharing expertise, and valuable guidance and encouragement stretched out to me.

I accept this open door to offer thanks to the entirety of the Department faculty members especially lab assistants for their assistance. I must express profound gratitude to my parents and my family for their love, support and nonstop encouragement throughout the course.

I also place on record, my feeling of appreciation to one and all, who have lent a helping hand, directly or indirectly, in this venture.

ABSTRACT

This project develops a simple, cheap and accessible robotic application with additional formation topology for secondary school RBT course. It will provide an affordable robotic platform built using low cost components so that it can be purchased in large quantities by secondary school. Suitable robotic application according to the RBT course syllabus will be made possible for students to learn with minimal effort. An additional formation topology is added to facilitate students' learning and make the learning process become more interesting. Although there are many different kinds of robotic application available on the market with various functionalities such as line-following function and obstacle-avoidance function, each of them still exists with some limitation, in which they are costly and secondary school cannot afford to purchase in large quantities. The application developed in this project is multiple robotic cars built using ESP8266 microchip as the main component. ESP8266 microchip is equipped with low cost Wi-Fi feature for easy access to the Internet and highly-integrated on-chip features that offers reliability, compactness and robustness. ESP8266 microchip is also equipped with Wi-Fi server and Wi-Fi client that allow server-client communication for movable group of mobile devices. The open source Arduino IDE is used as the software framework in this project. For smart obstacle avoidance, ultrasonic sensor modelled HC-SR04 is used to measure the distance of the obstacle, alongside with a buzzer to alert if the robotic car is approaching an obstacle. In the aspect of automatic lighting control, LDR sensor is used to measure the brightness of the room, and LED is used as the headlight of the car, which will turn on when it is dark and off when it is bright. A mobile application using online open source MIT App Inventor 2 is developed to control the motion for the master of the robotic car. While the master robotic car controls the motion of the subsequent robotic car through Wi-Fi server and Wi-Fi client features built in ESP8266 microchip. The project result shows the master robotic car is able to perform functions according to the commands from mobile application which is to move back and forth, turn left and right, while the subsequent robotic car is able to follow the commands sent by the master. The robotic car is able to detect obstacle, room brightness and react to the situations automatically. The end product is later tested for its speed in movements and the accuracy to detect obstacle and room brightness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
API	Application Programming Interface
DC	Direct Current
GPIO	General Purpose Input / Output
GUI	Graphic User Interface
IDE	Integrated Development Environment
IIC	Inter-Integrated Circuit
IP	Internet Protocol
IR	Infrared Red
LCD	Liquid Crystal Display
LDR	Light Dependent Resistor
LED	Light Emitting Diode
MYR	Ringgit Malaysia
PWM	Pulse Width Modulation
RBT	Reka Bentuk dan Teknologi (Design and Technology)
ROS	Robot Operating System
STEM	Science, Technology, Engineering and Mathematics
TCP	Transmission Control Protocol
USB	Universal Serial Bus
USD	United States dollar
VPL	Visual Programming Language

Chapter 1
Introduction

In this chapter, we present the problem statement and motivation of our project, our contributions to the secondary school, and the outline of the thesis.

1.1 Problem Statement and Motivation

As indicated by Anon (2018), Didier Roy, a specialist in learning advancement and mechanical technology intercession and maths educator having a place with the Flowers group claimed that giving students a chance to learn robotics is a compelling method to help them overcome their poor academic performance. It normally prompts to project-based learning and changes the instructive system by making it progressively adaptable and less oppressive, especially for students in need. It additionally includes dynamic research and encourages debate, which thus energizes students who are inconsistent with the conventional educational setting that does not suit their needs, express their thoughts. Consequently, educational robotic tools must be created in the manner that the devices are accessible, affordable and that can be broadly adopted by teachers and students in school.

Educational robotics is an interdisciplinary learning condition dependent on the utilization of robots and electronic components as the repeating theme to improve the development of abilities and capabilities in youngsters and adolescents. One well-known example of educational robotic application is the low cost educational robotic kit based on the UNO Arduino platform with both source code and project that are free and available (shown in Figure 1.1), which is a line-following robot. It has picked up prevalence because of its oversimplified idea in three measurements in the plan which are assembly, operation as well as maintenance and understanding and its affordability as far as one robot for one student in the class. However, there is restriction to the ability of this low cost educational robotic kits to be used in secondary school RBT course due to absence of functionalities that suits with the course syllabus. To enhance the robotic applications with the goal that it is reasonable to be utilized for secondary school RBT course, there are several factors to be concerned. The robotic application needs to have suitable functionalities such as obstacle-avoidance function, light automation function and so on as indicated by the course syllabus. The robot should be affordable enough to overcome the secondary school spending confinement issues and give each student opportunities to learn more effectively. The learning process ought to be simple, understandable and enjoyable for students and teachers with additional formation topology added such as multiple robots communication features. Therefore, in this project we would like to develop an enhanced, affordable and accessible robotic application that not only suits the course syllabus, and also aids students' learning as well as makes the student enjoy the interesting learning process.

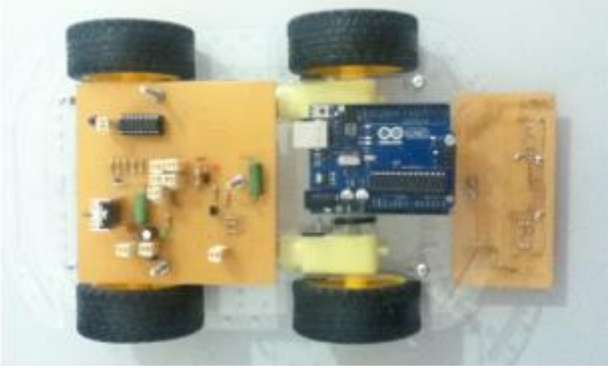


Figure 1.1 *Low Cost Robotic Kit*

1.2 Background Information

1.2.1 Robotics in General

In today's world, the unstoppable improvements of modern technologies have advanced the development of robotic technology at a very high speed. Along the same line, robots are being brought into different fields just as progressively creating key capacities that empower them to take care of issues that past human's ability, so far as to replace humans in everyday tasks. In fact, these are not simply essential tasks. As innovation progresses, these procedural systems can show improvement over natural ones. Robotic technology transformation from the past to present has changed the world around us, giving us new advances that can help with home errands, automobile assembly and numerous different assignments. From customary family apparatuses, for example, clothes washers and toasters to mechanical robots and artillery machinery, these gadgets, computers and engines are a unique and inherent part of our modern society. These invention of robots is not only to assist us, yet in addition empowered us to step out of the monotony of life and dare to explore and experiment new and risky things that human beings never dare to try but are eager to achieve. According to Innorobo Community (2016), since robotic technology is currently considered as a General Purpose Technology, implying that it can possibly upset society through its effect on existing monetary and social structures. In view of this, it is natural to talk about educational robotic as a really motivating key subject for the future in each nation.

1.2.2 Science, Technology, Engineering and Mathematics (STEM)

STEM – an education curriculum that in view of instructing students in four explicit subjects – science, technology, engineering and mathematics. STEM is an instructive development that is growing not only in the United States, however around the globe because of its permeation in each piece of our lives. For instances, Science is wherever in our general surroundings. Technology keeps on venturing into each part of our lives. Engineering is the essential structure of streets and bridges and the challenge of tending to worldwide environmental change and ecologically well-disposed changes in our homes. Mathematics exists in every profession and activity in our life. STEM-based learning programs are expected to expand students' enthusiasm in pursuing higher education and careers in these fields. As indicated by National Science Foundation, an independent federal agency in United States claimed that “In the 21st century, scientific and technological advancement have turned out to be progressively significant as we face the advantages and difficulties of both globalization and an information-based economy. To succeed in this new information-based and highly innovative society, students need to build up their abilities in STEM to levels much beyond what was viewed as adequate in the past.” (Kids, P.B.E.F., 2016). STEM education regularly utilizes another mixture learning model that joins conventional classroom instruction with online and hands-on learning activities. This blended learning model is intended to offer students the chance to encounter various methods for learning and taking care of issues. Obviously, many people are truly adept at utilizing these modern gadgets, however relatively few of them are keen on the manner in which these items are made. Thus teaching robotic to youthful ages is a decent presentation into science and technology, not just they will figure out how to plan and assemble a robot, they will likewise pick up an increasingly complete thought of the prerequisites for a creature to act in reality.

1.2.3 RBT (*Reka Bentuk dan Teknologi*)

In Malaysia, RBT (*Reka Bentuk dan Teknologi*) course – a design and technology course is one of the important elements that leads students towards the learning of the STEM-based learning program. RBT course for the secondary school is a continuation of the RBT contents that has been introduced to students during primary school. RBT course in secondary school replaces the KH (*Kemahiran Hidup*) course – an integrated life skills curriculum introduced in 1989. RBT course is an optional course that focuses on the production of technology-based products. The field of this design and technology is the focus of the national education system, which allows students to apply knowledge, skills and value in designing activities and producing effective products. RBT course covers areas such as public utilities, electricity, electronics, machinery, home sciences, agricultural sciences and financial management. Throughout the school, students will learn about the theory and practice in existing workshops. Teaching and learning activities will emphasize students' mastery of all areas of the RBT curriculum. The purpose of this RBT standard course is similar to the objective of STEM-based learning programs which is to equip students with the necessary knowledge and skills to understand the choice of interest and the areas of study when they reach a higher level.

1.2.4 Multi-Robot Formation or Swarm Robotics

The robot formation can be defined as branch of robotic system which studies the movable group of mobile robots to coordinate in an assured shape (A., B. & T., A., 2019). Swarm robotics is the use of various, autonomous robotics to accomplish a task through coordinating the robots' behaviours in a dispersive way (Rouse, M., 2017). The formation of robots are utilized to perform collective tasks include search and rescue, such as items transportation, environmental monitoring and field inspection. The multiple robot coordination is also used in distinct application fields which are mobile sensor network, nature positioning as well as medical operations. Distinctive control strategies are required to plan the design of robotic systems, such that leader follower approach. For instance, controlling the movable group of mobile robots to follow the simulated route while keeping the needed organized patron.

1.2.5 Component / Hardware Assembling

The point of assembly is to get together various individual components to frame an entire gadget, structure, or system. To accomplish this point, one must concentrate on the key actualities recorded, that is, arrangement of assembly, technique of joining, position of joints, interrelationship and distinguishing proof of parts, resistances, and protection of parts through checking of the components' datasheets.

1.2.6 Hardware Interface Programming

We all know that savvy devices improve our lives every day. And there are coders behind all of those gadgets. Hardware interface programming can also be called device driver programming. Since what we write in our software will either be utilized by the computer device for choices that may influence other appended hardware components, or our software will legitimately drive, control, enable or disable the components. Control law mathematics might be included, or it can be as straightforward as turning something on or off with a software command to the hardware through an input or output port, register, and others.

1.2.7 Mobile Application Development

Mobile application development is the plan of techniques as well as systems engaged with composing programming for small, remote processing gadgets. Like Web application development, mobile application advancement has its hidden establishments in progressively conventional programming advancement. One critical distinction, however, is that mobile apps are regularly composed explicitly to exploit the special features a specific cell phone offers.

1.3 Project Objectives

In general, this project aims to provide secondary school RBT course students and teachers a cheaper and accessible enhanced robotic application with topology formation which is a robotic car acts as a master, controlled using mobile application through Wi-Fi and followed by subsequent car. The end product will be able to assemble using low cost components with suitable functionalities according to the RBT course syllabus such as move back and forth as well as turn right and left which controlled using mobile application, autonomously response when it approaches an obstacle, when it is in dark or bright area. This is to overcome the budget limitation issues in secondary school so that it can be purchased in large quantities to allow each student has chance to use the product. This project facilitates students' learning and make the learning process more interesting through an algorithm designed to control the master robotic car and the subsequent robotic car will follow. The algorithm designed is also enable the robotic car to adjust their topology in order to line up become one in narrow walkway and expand when come to wider walkway.

This project will mostly focus on the assembling and programming of the robotic car, instead of mobile application. In order to obtain the end product which is a whole robotic car, hardware design such as block diagram is firstly needed, followed by collecting and assembling all the hardware components required. All the components have to joint carefully and correctly to the pins according to the components' datasheets so that we will not burn or damage the components. For example, before joining a LDR sensor to the ESP8266 microchip, we have to identify the input voltage of the pin on the microcontroller, and connect the sensor with resistor first to obtain the correct input voltage for the microcontroller so that the pin will not burn and damage the microcontroller. After assembling all the components correctly, hardware programming using C programming language will be focused on to test or control the functionalities of each components, and also developed a server-client communication for multiple robots formation. To be able to interact with the robotic car remotely, this project will focus on the mobile application development for motion control through Wi-Fi.

However, this project will have some limitations such as the battery life can only last longer for around two hours and have to recharge the batteries again due to the high power consumptions by the ultrasonic sensor and the microcontroller. Once the power is getting low, the ultrasonic sensor will function improperly and causes the robotic application cannot work well. The project also does not cover the interaction with uneven surface. If the surface is bulge, the ultrasonic sensor will detect it as an obstacle, whereas if the surface is dent, the ultrasonic sensor will not detect it and the robotic car might fell into it and causes the damage of the components. Besides that, the algorithm is also hardcoded for the topology adjustment which means that the position and distance between the subsequent robotic cars are fixed.

1.4 Project Scope

This project develops an enhanced robotic application with topology formation which is a master robotic car controlled using mobile application, and it controls the subsequent robotic car to follow it for education purpose in secondary school RBT course. Not only according to the syllabus of RBT course in secondary school, robotic car is also simple to learn and implement as well as the results or effects can be clearly observed by students. The robotic car is also able to be implemented in a low cost to overcome the budget limitation issues, and it is affordable in large amount to give every students chances to learn more effectively. Other than simple functionality on the robotic car such as moving back and forth, stop when detecting obstacle using ultrasonic sensor, turn on LED automatically when in dark area which can be detected using LDR sensor, so on and so forth, an additional leader follower approach and the robotic cars are able to line up and expand according to the walkway condition are added in order for students to learn about robotic application more appropriately and effectively as well as the students will enjoy the learning process.

1.5 Impact, Significance and Contribution

The innovation of this project is to build an affordable enhanced robotic application with topology formation compared to other robotic applications available on the market that can only perform line-following and obstacles-avoidance functions. This will definitely help to solve the budget limitation issues from secondary school and facilitate students' learning by providing every student the chances to use the application. The robotic application will be a robotic car which performs motion and light automation, while avoiding obstacles automatically as well as alerting the user by triggering the buzzer. Leader follower approach is added to control only the master robotic car and the subsequent robotic car will follow the light automation and motion of the master robotic car. The robotic application is also designed to adjust the robotic car topology in order to line up become one in narrow walkway and expand when come to wider walkway.

The hardware components for the robotic cars are cheap and available online. Everyone can easily buy the components online with low cost and start to build the robotic car themselves. Besides, the programming environment, either for the robotic car or for the mobile application is open-source as well, that means everyone can access to it without paying any fees to write their own codes for controlling and testing the robotic car. The Arduino IDE which is the programming environment to test the robotic car is well equipped with a wide variety of packages that is compatible with most hardware. This also means the project is highly expandable in terms of hardware by simply installing new hardware and controlling them with the open source packages found on the Arduino IDE. Hence, addition of new functionalities is possible by further utilising existing hardware available on the robotic car or by adding new hardware. Therefore, the robotic application we developed has more flexibility for expansion and is more customizable according to the task requirement when compared to existing robotic applications available on the market, at the same time keeping the price affordable.

1.6 Report Organization

The details of this research are shown in the following chapters. In Chapter 2, some related previous work are reviewed and compared. Then, a system methodology of the robotic application is presented in Chapter 3, which included the system development model, system and functional requirement, project milestone and estimated cost. Beside that, Chapter 4 describes a detailed system design such as system architecture, functional modules, system flow and design for the enhanced robotic application. Furthermore, Chapter 5 reports the system implementation in which the setup of hardware and software, the setting and configuration together with the system operation are clearly described. Moreover, in Chapter 6, the system evaluation such as system testing and performance metrics are discussed, followed by the project challenges and objectives evaluation. Lastly, conclusion and recommendation is provided in Chapter 7.

Chapter 2
Literature Review

This chapter includes literature review of related project, summary of the pros and cons of the projects and our proposed solution.

2.1 Literature Review

Throughout the years, numerous researchers have sought to provide robotic application that are engaging, natural and pertinent to be used in the STEM-based learning related course with the goals that instructors and facilitators can deliver high-quality, hands-on educational program that empowers imagination and critical thinking while at the same time fortifying the capacities and abilities required for students to be successful in the core classroom. Hence, let's us discuss about some similar products for education purpose and other robots that utilises similar technology with our proposed solution.

2.1.1 Mona

According to Arvin, F. et al., (2018), Mona is proposed as a low-cost, easy-to-use and adaptable robotic platform with open-source software environment and hardware components. It has been created to be compatible with various standard programming environments for robotic education. Mona robot is utilized for both teaching and research to make sure the students are being educated with the latest technology and provide an amazing pathway for those students who are keen on pursuing a research career.

The Mona robot allows students to embrace practical experiments on framework characterization such as actuation system, and movement arranging like obstruction identification and progressively complex swarm algorithms. The lasting swarm interface intended for Mona takes into consideration huge scope, long-term self-sufficiency and swarm situations to be studied. The Monas are additionally being utilized to investigate fault tolerant control of multi-robot frameworks, swarm behaviour based on pheromone communication and human-robot-interactions utilizing mixture of reality interfaces.

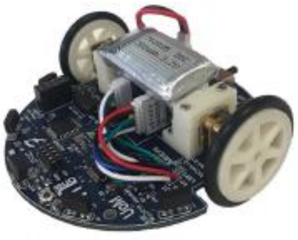


Figure 2.1 *Mona*

Mona utilizes an inexpensive ATmega 328 microcontroller as the core processor to build up the robot on account of the Arduino Mini/Pro design and to be compatible with Arduino's open-source software environment. The microcontroller has a few inner modules giving simple and reliable least framework to build up the Mona robot. It comprises of an interior timer module to generate pulses for the speed control of motors, eight analogue to digital converter (ADC) channels to connect the infrared red sensors for barrier distance estimation and battery level monitoring, a few serial communication techniques such as IIC for flash memory programming or external modules communication, as well as general purpose input output ports for LEDs and IR emitters connection. The microcontroller controls the motor driver directly and communicates with the computer utilizing its USB driver. Pulse-width modulation (PWM) is used to command the rotational speed for each motor and an H-bridge DC motor driver is used to control the motors. Movement of Mona robot is controlled using ROS server to send commands to the motors through Wi-Fi module.

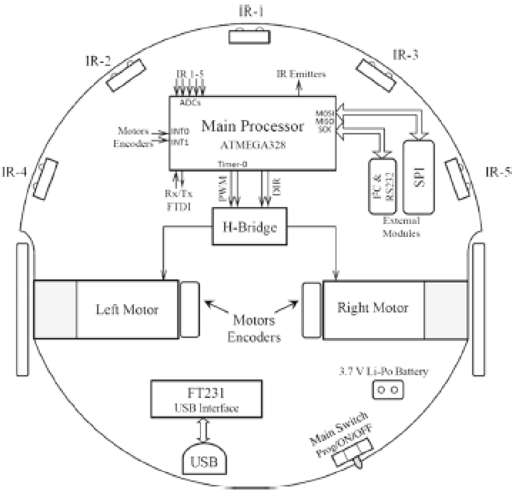


Figure 2.2 *Hardware architecture of a Mona robot*

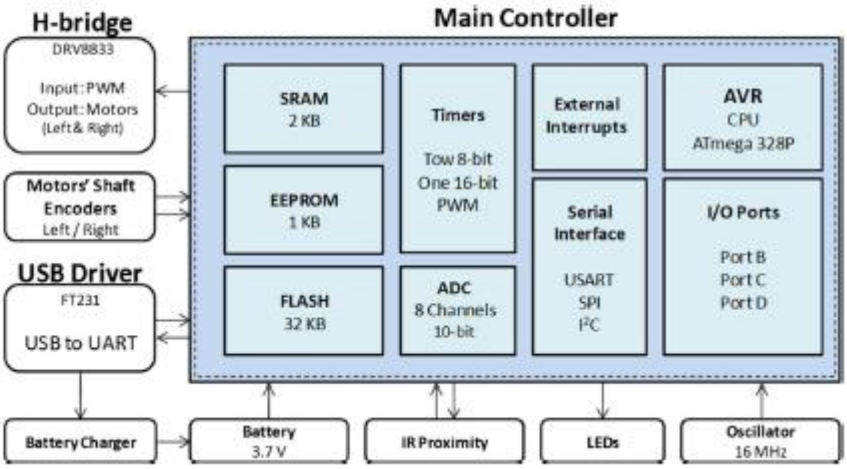


Figure 2.3 *Architecture of the main controller*

Arduino, one of the best open-source stages, was utilized to program Mona. This is because it is a relatively easy platform to use in contrast with other open-source platforms. It also provides a wealth of online forums and freely accessible libraries, as well as an assortment of Arduino compatible programming environments such as Mblock and Scratch, particularly for young students. Because of the popularity of the Arduino platform and the fact that the Arduino project is open source, Mona robot is all programmed in C language.

The advantages of using Mona robots for education is that multiple robots coordination can be studied. A state-of-the-art swarm aggregation algorithm, BeeClust was selected because of its simple implementation and programming. To perform multi-robot communication, a similar control mechanism was followed by all robots. For instance, a light source was placed on one side of the field as a gradient cue. The robots followed the algorithm to locate the ideal piece of the field.

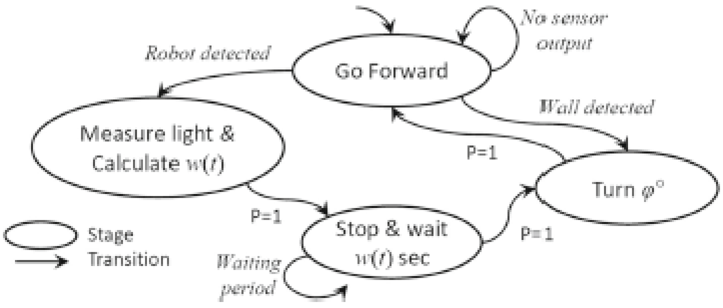


Figure 2.4 *Controller of the performed swarm robotic scenario*

Nonetheless, Mona is still not intended for secondary school education due to some lacking of functionalities required by RBT course syllabus such as light automation and the motion controlled by ROS (Robot Operating System) commands that is not familiar in secondary school. In order to master this operating system, additional course is needed to attend and secondary school teachers have to pay for the course in which the cost is at least MYR 880. This may burden the teachers and the school for providing the training, and it goes against our main objective which is to provide low-cost robotic application. Beside that, since the on-board battery controls the motors directly, any drop in voltage affects the robot's speed. Hence, the battery voltage must be considered in the kinematic model of the robot. Other than that, the infrared proximity sensor can be used for face-to-face communication between robots in multi-robot scenarios. However, due to the distance limitations of the modules used, they cannot provide high quality or fast communication. Thus, Mona requires an external module to offer communication between robots, distance estimation, and 360 degree to the robot's orientation.

2.1.2 HeRo

Other than Mona, Rezeck, P.A.F., Azpurua, H. & Chaimowicz, L., (2017) presented HeRo, a novel swarm robots platform that is affordable, open platform, simple to assemble with off-the-shelf components and is profoundly incorporated with open source robot operating system (ROS). The robotic platform also consists of 3D printing as well as open source programming that developed utilizing ROS with generic devices and abiding strictly by criterion to be simple incorporate with different sorts of projects or development.

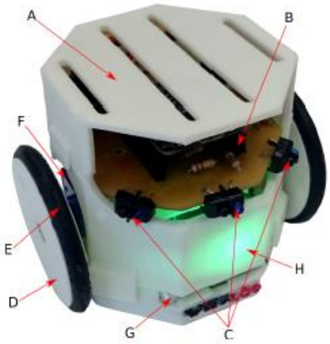


Figure 2.5 *HeRo platform with (A) 3D printer body, (B) Circuit board, (C) IR sensors, (D) Wheel, (E) Rubber O-ring, (F) Servo Motor, (G) Battery and (H) RGB LED*

HeRo robots are made dependent on three key elements which are high modularity, maximize the use of commercial components that are easy to produce and assemble as well as keep the price as low as possible without relinquishing processing and sensing power. The controller for the robot is a low cost ESP8266 microprocessor packaged into NodeMcu V3 board that has two functions. It controls all the low-level gadgets such as RGB LED, sensors and motor, as well as initiates a TCP/IP communication with the ROS platform that run on clients' computer. The controller also includes 16 pulse width modulation (PWM) channels for controlling sensors, motors and RGB LEDs, as well as 10-bit analog-to-digital converters for measuring the intensity of incident infrared light from infrared sensors.

Servo Motors are chosen for mobility in HeRo robot due to its good balancing between size, speed and control. HeRo uses a basic movement control technique that is differential-driven configuration to control the movable robots. Pulse-width modulation (PWM) is also used to control the motors' rotational speed. Moreover, the platform also uses only infrared proximity sensors for obstacles-avoidance purpose and it comprises of five fundamental 3D printer parts that is the top frame, the main frame, the middle frame, the board support and the wheel.

For programming environment, commonly used Arduino IDE is utilized to actualize the firmware for effectively fabricate an interface among ROS, the actuators and the sensors. The ROS package can run on an external computer and use the ROS serial node to communicate directly with the robot in TCP mode. This node connects to a pre-configured TCP port in the ROS server to answer all points created in the robot microcontroller. The HeRo swarm robots have such functions as cooperative block transportation, collision free autonomous navigation, road point following realization and collision free autonomous random walk based on vehicle information.

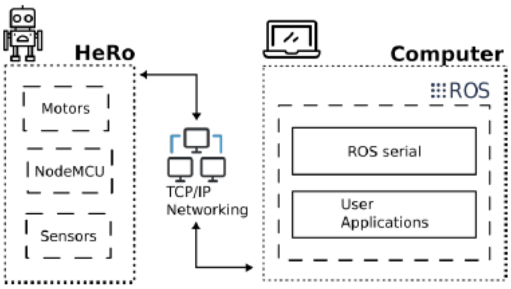


Figure 2.6 *HeRo robot connects by TCP to ROS to share control and sensor messages*

Nevertheless, similar with Mona robots, HeRo platform uses ROS that requires teachers and staffs in secondary school to set aside longer effort to learn and ace it before showing the students by attending classes that are costly to afford. Other than that, the HeRo platform is lacking of light detection sensor which is needed in the RBT course syllabus. Secondary school teachers have to self-investigate the sensor used and configure for the light detection purpose.

2.1.3 Thymio Robot

Another educational robotic application proposed by Mondada, F. et al. (2017) and Vitanza, A. et al. (2019） is a Thymio robot. The preferences for the Thymio robot are it is planned along seven fundamental axes: low expenses for clients; A lot of highlights appropriate for kids to grown-ups of the two sexual orientations and various age gatherings; Mechanical structure to advance innovativeness; A blend of actuators, sensors and programming features that encourage learning; A lot of prepared to-utilize programs that rapidly get to robot conduct; A programmable situation; And an open source network that adds to structure and communication. In spite of its simplicity and low cost, Thymio is well suited to group robotics experiments, which assume that complex self-organizing behavior emerges from low complexity as far as rules followed by every robots. In the swarm robot experiment, Thymio has a few infrared proximity sensors that used for communication, such as sending small messages to neighbors.

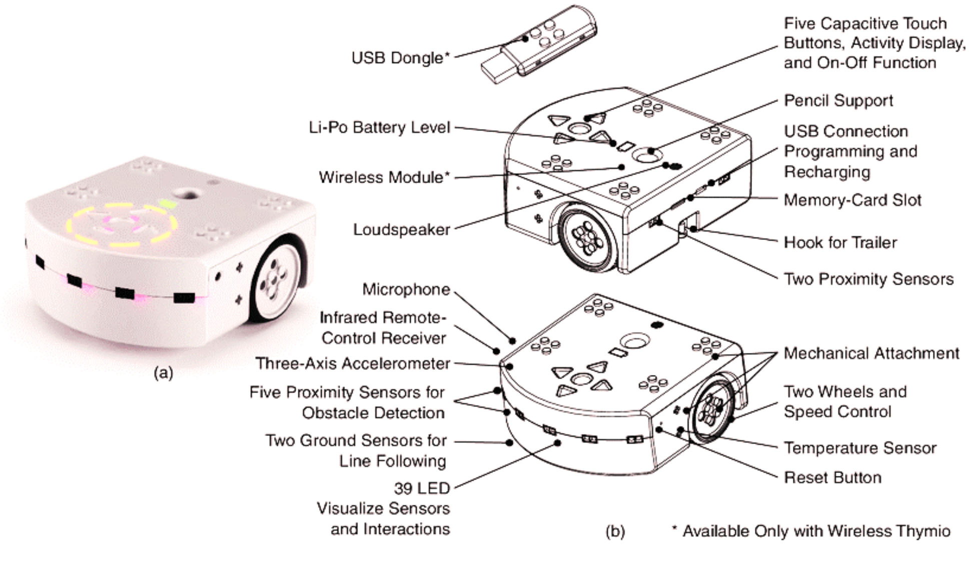

USB Dongle*
Li-Po Battery Level
Wireless Module*
Loudspeaker
Microphone
Infrared Remote-Control Receiver
Three-Axis Accelerometer
Five Proximity Sensors for Obstacle Detection
Two Ground Sensors for Line Following
39 LED Visualize Sensors and Interactions
Five Capacitive Touch Buttons, Activity Display, and On-Off Function
Pencil Support
USB Connection Programming and Recharging
Memory-Card Slot
Hook for Trailer
Two Proximity Sensors
Mechanical Attachment
Two Wheels and Speed Control
Temperature Sensor
Reset Button
(a)
(b)
* Available Only with Wireless Thymio

Figure 2.7 *Thymio robot and its main components for the wireless- and the USB-connected versions.*

The Thymio robot is a minor differential wheeled robot that appropriate for work area use. Thymio robot includes a translucent white body and a wide scope of actuators and sensors. It has an implanted rechargeable battery that can provide 3 to 5 hours of intensity. The robot comprises of five capacitive touch catches to shape an instinctive user interface that streamline the plastic body of the robot and make it stronger than the physical catches. Henceforth, it is powerful enough to be utilized by students since it can tumble from table without breaking. Thymio robot utilizes PIC24F as its microcontroller because it incorporates a USB port to drive the capacitive touch button straightforwardly so as to spare additional segments. This microcontroller controls all the sensors and actuators, aside from the interior lithium-particle battery charging rationale, which uses a particular chip for security reason.

Thymio robot can likewise be associated with numerous Thymio robots through programming, in order to facilitate multi-Thymio robotic structures. Thymio is running on Aseba which is an open source programming condition. Aseba is intended to empower beginners to program robots effectively. On the automated side, Aseba gives a lightweight virtual machine that keeps running on microcontrollers, for example, Thymio's inner PIC24F. A virtual machine permits momentary transfer and safe program execution. While on the work zone side, Aseba gives a coordinated improvement condition that incorporates a mixed language, Blockly, visual programming language (VPL) and a scripting language, for graphically collecting contents. The significant contrast between Thymio robot with other educational robotic application is that Thymio robot can demonstrate its operational practices right out of box without the necessities of collecting or arranging. In this manner, Thymio robot has six diverse available essential practices stored in flash forever which enable individuals to promptly interface with the robot. Despite the fact that individuals does not have to assemble the robot, they can still making developments over these fundamental practices utilizing paper manifestations.

Figure 2.8 *Extensions of the Thymio basic robot with paper or cardboard body extensions.*

Anyhow, there are some limitations in Thymio robot to be used as an educational robotic application. Firstly, since Thymio robot is already assembled, students probably would not get the chance to find out about the parts inside the robots and the right method to gather a robot. Moreover, Thymio robot which expenses USD130 each is viewed as costly to be purchased for educational purpose in Malaysia because the robot would costs around MYR530 each and it is exorbitant to get it in enormous sum. In addition, the programming environment for Thymio robot which is Aseba is incorporating with ROS, a software frameworks in robotic research. This joining permits running complex calculations, for example, concurrent limitation and mapping, related to Thymio robot and makes the Thymio robots progressively appropriate for university-level education instead of secondary school education.

2.1.4 Spiderino

On the other hand, Jdeed, M. et al. (2017) built up an incredible, inexpensive research robot based on small size spider toy which is called Spiderino. Spiderino is a solitary robot that accompanies a limited set of functions and sensors, which simplifies the programming interface, and secondly, hexapod mobility and spider design are probably going to bring up enthusiasm among kids. The low cost of a Spiderino would correspond to the very limited budgets of secondary schools for additional materials for science education.

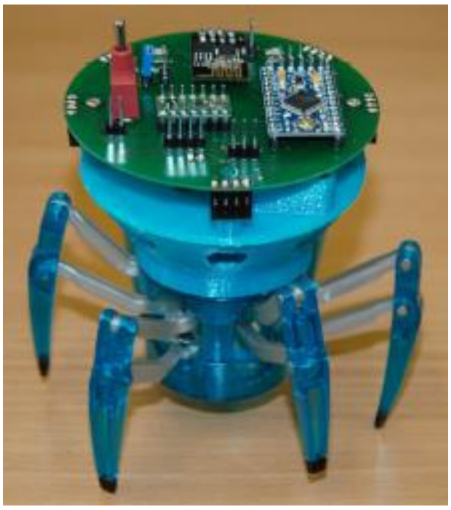


Figure 2.9 *Spiderino*

Spiderino's board is a round, double-sided PCB with an Arduino microprocessor socket, the motor driver PCB, and a Wi-Fi module. For client collaboration, the board includes two light-emitting diodes (LEDs), two jumper mode options and a switch/off/charge switch. It also contains six four-pin sensor interfaces, motors and battery connectors. In terms of fundamental electronic components, an Arduino Pro Mini is used with an ATmega processor, an ESP8266 Wi-Fi module, and a POLOLU-Motor-DRVDRV8835 that controls the Spiderino's motors. The Arduino Pro Mini has various facilities to link up with ESP8266, as well as serial communication.

The proposed robot design for Spiderino includes a six-worm spider motion system in which a 3D-printed adapter is appended. The physical parameters are chiefly educed from the hexapod spider. A mechanical, coordinated motion framework is provided to robots' legs and motors in order to coordinate the movement of the spider's legs simultaneously. One motor is utilized for rotational movement and the other for forward or backward motion, and Spiderino has to turn its head for altering the direction.

For software environment, Arduino Studio is used to program the Arduino microcontroller for controlling Spiderino's motors, and realizing the fundamental functions of walking such as moving back or forth, turning left or right as well as lighting of the two leds. Moreover, a software library written in C or C++ that can be easily imported into Arduino Studio to execute Spiderino's firmware is provided in order to control motor speed and read data from proximity sensors.

However, there are a few constraints in Spiderino robot to be used as an educational robotic application. Although it costs less than 70 Euro and it is considered as moderate in Europe nations, it is as yet required a high spending plan to buy in enormous amounts for secondary school education purpose in Malaysia. Furthermore, obstacle-avoidance function which is needed in RBT course syllabus is not mentioned in Spiderino robot, whether it is available or it has to append externally.

2.2 Critical Remarks of Previous Works

***Table 2.1** Summary of existing systems*

Existing System	Advantages	Disadvantages	Critical Comments
Mona	✓ Uses Arduino Studio as software environment ✓ Controlled through Wi-Fi ✓ Multiple robots coordination	× Uses ROS (Robot Operating System) × Lighting function × Obstacle-avoidance function	➢ A costly classes will need to attend to master ROS ➢ Do not fulfil requirement in RBT course syllabus
HeRo	✓ Uses ESP8266 as microcontroller with Wi-Fi module ✓ Uses Arduino Studio as software environment ✓ Obstacle-avoidance function	× Uses ROS (Robot Operating System) × Lighting function	➢ A costly classes will need to attend to master ROS ➢ Do not fulfil requirement in RBT course syllabus
Thymio Robot	✓ Firm and tenacious ✓ Facilitate multi-Thymio robotic structures ✓ Obstacle-avoidance function ✓ Lighting function	× Already assembled × Costly (USD130 = MYR530) × Uses Aseba incorporating with ROS as firmware	➢ Fulfil requirement in RBT course syllabus ➢ Too costly for secondary school to buy in large quantities ➢ Progressively appropriate for university-level education instead of

			secondary school education
Spiderino	✓ Spider design increases students' interests ✓ Uses Arduino Studio with C/C++ languages as programming environment ✓ Uses low cost ESP8266 as microcontroller with Wi-Fi module ✓ Lighting function	× Costly (70 Euro = MYR 330) × Obstacle-avoidance function	➢ Too costly for secondary school to buy in large quantities ➢ Do not fulfil requirement in RBT course syllabus

To fulfil the objectives of our project, this proposal presents a solution to develop a low cost, enhanced robotic application with formation topology that suitable for secondary school RBT course using affordable hardware and open-source software. Our project uses a low cost ESP8266 microchip as the microcontroller as the main components and open-source Arduino IDE as the software environment to program the robotic car. A mobile application that used to control the motion of the master robotic car is also developed using online open-source software. The master robotic car is then controlling the subsequent robotic car through Wi-Fi module packaged in ESP8266 microchip. Next, we are using an affordable ultrasonic sensor for the detection of obstacle and a LDR to detect the room brightness. We are also using DC motor to control the motion of the car such as moving back and forth as well as turning left and right. Instead of using servo motor, the proposed approach can minimize the cost of the robotic car and make the robotic car easier to assemble and program.

2.3 Concluding Remark

Although there are many different types of robotic applications available on the market, each of them still exists with some shortcomings, either it is expensive or the software programming language or the functionalities does not meet the RBT course requirements. As a result, the secondary school could not get a robotic application that is suitable for mass purchase and to be learnt by student so far.

Chapter 3
System Methodology

This chapter includes system development model used by the project, system and functional requirement, overall timeline of the project and the estimated cost to develop a robotic application.

3.1 System Development Model

The robotic application developed in this project is based on agile system development model, in which continuous repetition of development and testing is carried out. Agile development model is a blend of iterative and incremental procedure models with focus on procedure flexibility and consumer satisfaction by fast transport of working programming item. Agile model trusts that each assignment ought to be taken consideration differently and the current strategies should be custom fitted to best suit the project necessities. Iterative methodology is taken and working software build is provided after every repetition. Each build is incremental on the basis of features, and the final build contains all the features required by the consumer. In this project, the functionalities are divided into smaller tasks to deliver specific function of the release. For instance, the functionalities of movement, obstacle detection, light detection, remote controlled application are developed separately. After some repetition of testing on each tasks, each of these functionalities is then combined to form the final products that contains all the features needed.

3.2 System Requirement (Technologies Involved)

3.2.1 Hardware

ESP8266 Microchip

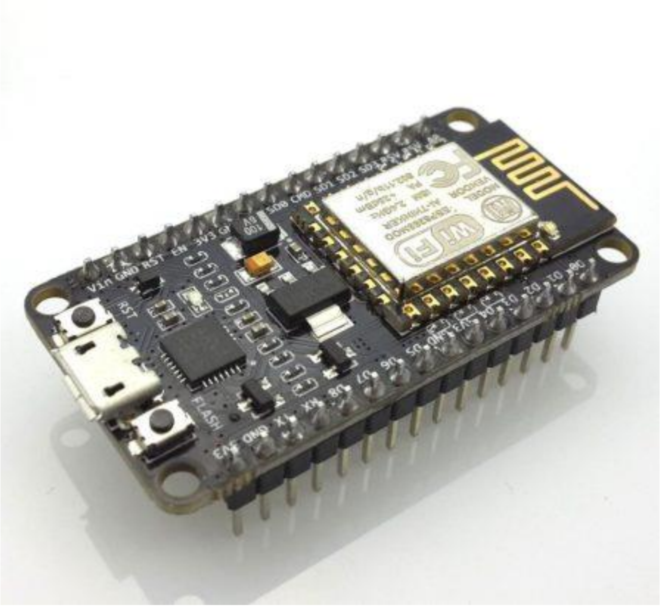


Figure 3.1 *ESP8266 Microchip*

ESP8266 is a cheap Wi-Fi microchip with full TCP/IP stack and microcontroller ability created by manufacturer Espressif Systems in Shanghai, China. The undeniable favorable position of ES8266 microchip over the Arduino or PIC is that it can promptly associate with the Internet by means of Wi-Fi. As per Pelavo, R. (2019), ESP8266 microchip looks like an Arduino Nano. Speaking of Arduino, another advantage of the microchip is that it can be connected straightforwardly to the personal computer and program it like an Arduino. Espressif Systems (n.d.) referenced that ESP8266 coordinates GPIO, PWM, IIC, 1-Wire and ADC all in one board and due to the exceedingly incorporated on-chip features, the microchip offers unwavering quality, compactness and robustness. ESP8266 is also coordinated with the most minimal cost Wi-Fi and simple to prototyping improvement unit.

L298N Motor Driver

Besides of ESP8266 microchip, the sensors and motors to be used in the platform is also low cost but capable. For example, the motor driver used in the platform is L298N motor driver. The L298N is a dual H-Bridge motor driver which permits speed and direction control of two DC motors simultaneously, or control one bipolar stepper motor easily (Pelayo, R., 2018). Speed control is also conceivable with L298N motor driver by feeding the PWM signals to the motor enable pins. The speed of the motor will fluctuate as indicated by the pulse width where the wider the pulses, the quicker the motor pivots.

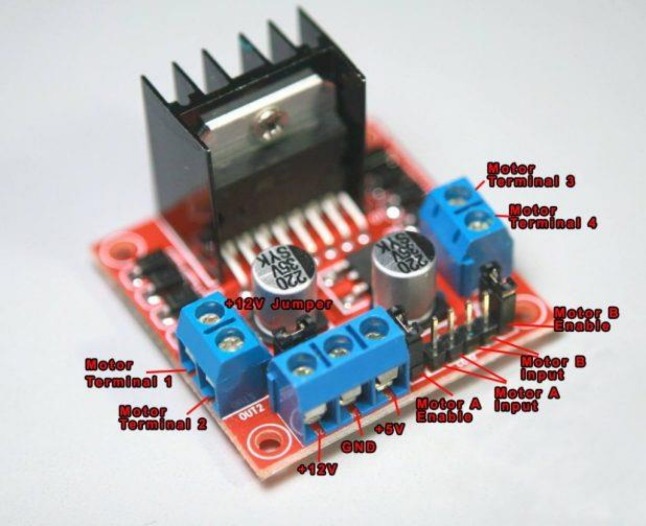


Figure 3.2 *L298N Motor Driver*

HC-SR04 Ultrasonic Sensor

Another low cost component used in the robotic platform is the ultrasonic sensor modelled HC-SR04. HC-SR04 is an ultrasonic sensor mostly used to identify the distance of the target object and it is commonly used with both microcontroller and microprocessor like Arduino and Raspberry Pie. The sensor is made dependent on the standard of echolocation utilized by creatures like bats and dolphins. Since ultrasonic sensors use sonar to decide their distance from an item, they work autonomously of daylight, spotlights and surface shading, which can influence the readings of any infrared distance sensor. According to Aqeel, A. (2019), the sensor estimates exact distance using a non-contact technology which is an innovation that includes no physical contact among sensor and item. Transmitter and recipient are the two primary pieces of the sensor, the former converts the electrical signal into ultrasonic, the latter proselytes the ultrasonic signal into electrical signal. Two ultrasonic sensors are required for the stage to distinguish obstructions in front and at the back.

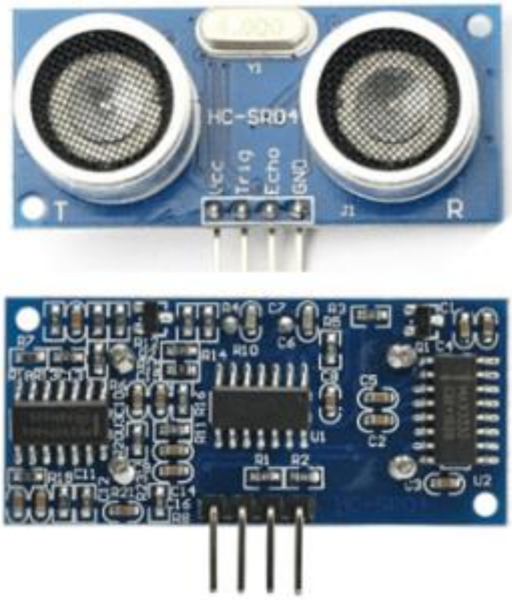


Figure 3.3 *HC-SR04 Ultrasonic Sensor*

Light Dependent Resistor

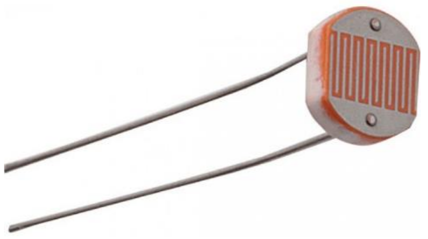


Figure 3.4 *Light Dependent Resistor (LDR)*

In order to perform brightness detection, a low cost yet powerful light dependent resistor (LDR) is used for this project. According to JOJO (2018), it is also called a cadmium sulphide (CdS) cell or a photo conductor or a photo resistor. It is essentially a photocell that deals with the standard of photoconductivity where the inactive segment is fundamentally a resistor whose resistance value diminishes when the intensity of light reduces. This optoelectronic gadget is generally used in light changing sensor circuit, and light and dull initiated exchanging circuits. LDR is cheap and promptly accessible in numerous sizes and shapes, just as it requires a little power and voltage for its activity.

3.2.2 Software

Arduino IDE

On the other hand, apart from the hardware, the software used for the prototype robotic platform is Arduino IDE which is an official, open source software introduced by Arduno.cc, that is predominantly utilized for writing, verifying and uploading the code into Arduino Module. Arduino IDE is effectively accessible for operating systems like Windows, MAC, Linux and run on the Java platform that comes with built-in functions and commands for troubleshooting, altering and accumulating the code in nature. The environment also supports both C and C++ languages. Arduino IDE environment comprises of two fundamental parts: an editor for writing the required code, and a compiler for compiling and uploading the code to a given Arduino module. Arduino IDE is suitable to be used for learning purpose since it makes code gathering so straightforward that even a typical individual with no earlier specialized information may fiddle with the learning procedure (Aqeel, A., 2018).

ESP8266 Libraries

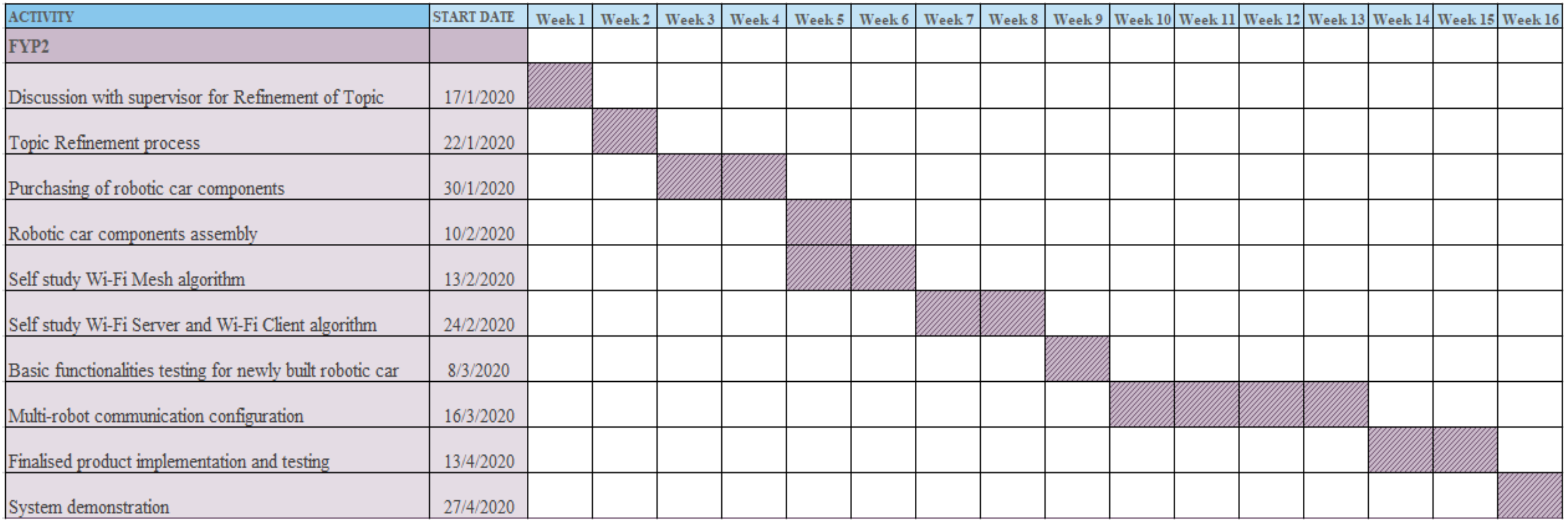
The ESP8266 libraries used in this project will be the ESP8266WiFi for Wi-Fi configuration and connection. In this project, ESP8266 microchip works as a soft access point, to set up its own Wi-Fi network. The ESP8266WiFi library gives a wide accumulation of C++ methods or functions and properties to configure and operate an ESP8266 module in soft access point mode. Next to that, ESP8266WiFi provides function calls that create clients to access services given by servers for sending, receiving and processing the data. With these function calls, mobile application developed can associate with ESP8266 microchip's Wi-Fi system and it serves as the client to send data to the microchip for processing. Moreover, WiFiServer and WiFiClient libraries are also used in this project to enable leader-follower approach. The master robotic car sends commands to an internal server using WiFiServer, and the subsequent car acts as the clients and retrieve data from the server by using function calls in WiFiClient library.

3.3 Functional Requirement

According to the RBT course syllabus in secondary school, the robotic car developed should have a few of functionalities. One of the important and basic function is to move the robotic car in four directions. The robotic car is able to move forward and backward, turn left and right, as well as stop. The movement of the robotic car is controlled using mobile application created via Wi-Fi. Beside of this, the robotic car is also an obstacle-avoidance car. It is able to detect an obstacle in front of it and trigger signal to alert the user by turning on the buzzer. Furthermore, light detection is another functionalities of the robotic car. The robotic car is able to detect the room brightness and response accordingly to the brightness. For instance, when the robotic car is in a dark area, it is able to trigger signal to turn on the LED, and vice versa. An additional formation topology is added in the robotic application in which one robotic car is acted as a master car that controls the motion of the subsequent car. The subsequent car is also able to move in a line in narrow walkway and expand when come across wider path.

3.4 Project Milestone

Table 3.1 *FYP Timeline*



ACTIVITY	START DATE	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16
FYP2																	
Discussion with supervisor for Refinement of Topic	17/1/2020	▨															
Topic Refinement process	22/1/2020		▨														
Purchasing of robotic car components	30/1/2020			▨	▨												
Robotic car components assembly	10/2/2020					▨											
Self study Wi-Fi Mesh algorithm	13/2/2020					▨	▨										
Self study Wi-Fi Server and Wi-Fi Client algorithm	24/2/2020							▨	▨								
Basic functionalities testing for newly built robotic car	8/3/2020									▨							
Multi-robot communication configuration	16/3/2020										▨	▨	▨	▨			
Finalised product implementation and testing	13/4/2020														▨	▨	
System demonstration	27/4/2020																▨

3.5 Estimated Cost

Table 3.2 The minimum costs required for one unit of proposed robotic application.

No.	Components needed	Price per unit (MYR)	Quantity needed	Total price (MYR)
1.	ESP8266 Microchip	19.90	1	19.90
2.	L298N Motor Driver	6.00	1	6.00
3.	HC-SR04 Ultrasonic sensor	3.90	1	3.90
4.	5 volts Buzzer	0.90	1	0.90
5.	LED	0.50 for 5 pieces	1 out of 5	0.10
6.	LDR	0.60	1	0.60
7.	Breadboard (8.5*5.5cm)	3.00	1	3.00
8.	Robotic Car Platform with DC Motor set	18.60	1	18.60
			Total Costs :	53

The price for each components is taken from the online shopping website Lazada Malaysia. The batteries that act as the power supply and the battery holders will not be included. If non-rechargeable batteries and battery holders are included, the batteries will cost MYR 15.95 for one package with 12 units and the battery holder will cost MYR 4.30. Hence, the total cost for one unit of robotic application will be MYR 73.25.

3.6 Concluding Remark

In this project, agile system development model is used for the development of functionalities of the robotic application, in which the functionalities developed is based on the secondary school RBT course syllabus. The system requirement of both hardware and software is also described clearly, together with the functional requirement. An overall timeline of the project and the estimated cost to develop a robotic application is also provided in this chapter.

Chapter 4

System Design

This chapter includes system architecture of the project, functional modules in the system and system design details.

4.1 System Architecture

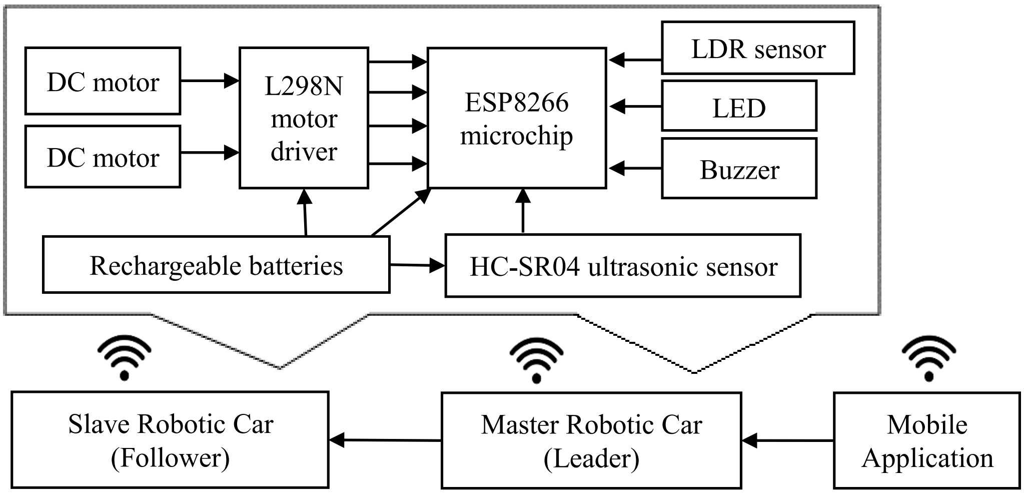


Figure 4.1 *Hardware block diagram*

4.2 Functional Modules in the System

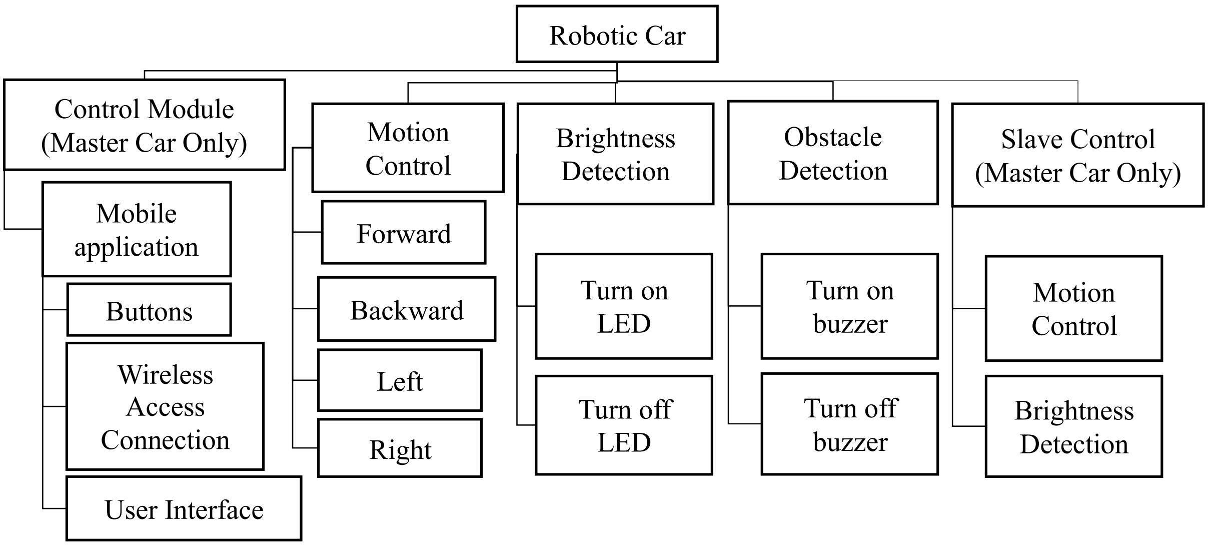
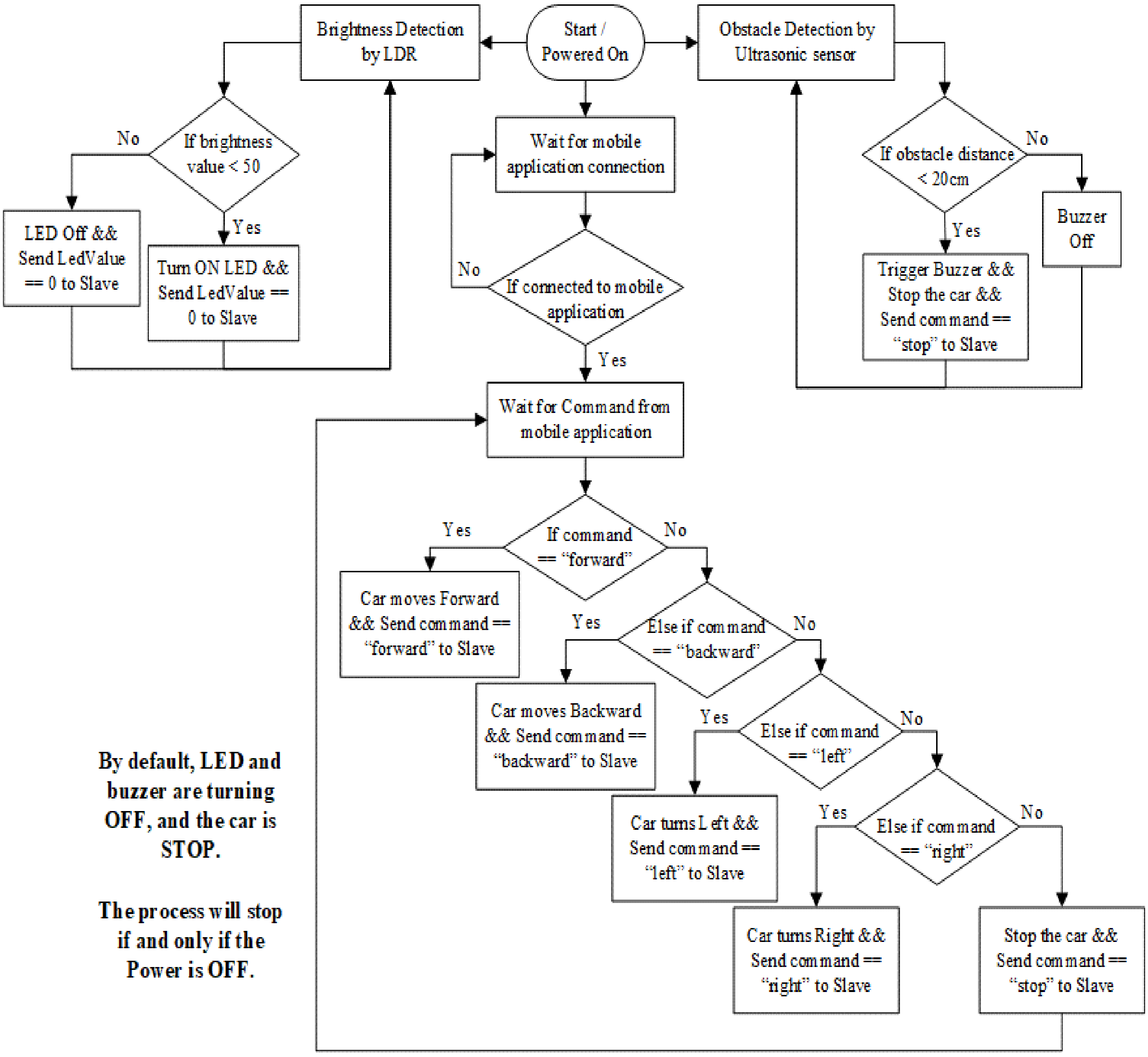


Figure 4.2 *Functional modules diagram*

4.3 System Flow

4.3.1 System Flow for Master Robotic Car

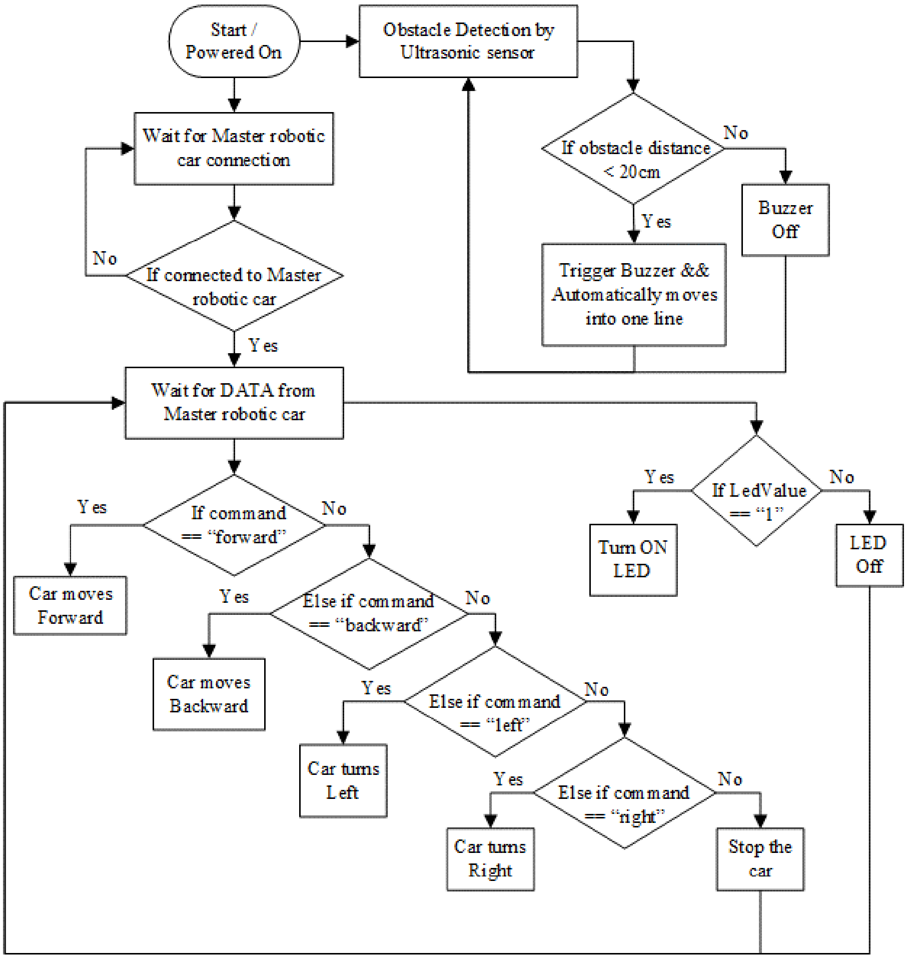

Start / Powered On
Brightness Detection by LDR
Obstacle Detection by Ultrasonic sensor
If brightness value < 50
No
Yes
LED Off && Send LedValue == 0 to Slave
Turn ON LED && Send LedValue == 0 to Slave
Wait for mobile application connection
If connected to mobile application
No
Yes
If obstacle distance < 20cm
No
Yes
Buzzer Off
Trigger Buzzer && Stop the car && Send command == "stop" to Slave
Wait for Command from mobile application
If command == "forward"
Yes
No
Car moves Forward && Send command == "forward" to Slave
Else if command == "backward"
Yes
No
Car moves Backward && Send command == "backward" to Slave
Else if command == "left"
Yes
No
Car turns Left && Send command == "left" to Slave
Else if command == "right"
Yes
No
Car turns Right && Send command == "right" to Slave
Stop the car && Send command == "stop" to Slave

By default, LED and buzzer are turning OFF, and the car is STOP.

The process will stop if and only if the Power is OFF.

Figure 4.3 *System flowchart (Master Robotic Car)*

4.3.2 System Flow for Slave Robotic Car


Start / Powered On
Obstacle Detection by Ultrasonic sensor
Wait for Master robotic car connection
If obstacle distance < 20cm
No
Buzzer Off
Yes
Trigger Buzzer && Automatically moves into one line
No
If connected to Master robotic car
Yes
Wait for DATA from Master robotic car
Yes
If LedValue == "1"
No
Turn ON LED
LED Off
Yes
If command == "forward"
No
Car moves Forward
Yes
Else if command == "backward"
No
Car moves Backward
Yes
Else if command == "left"
No
Car turns Left
Yes
Else if command == "right"
No
Car turns Right
Stop the car

By default, LED and buzzer are turning OFF, and the car is STOP.

The process will stop if and only if the Power is OFF.

Figure 4.4 *System flowchart (Slave Robotic Car)*

4.4 Algorithm Design

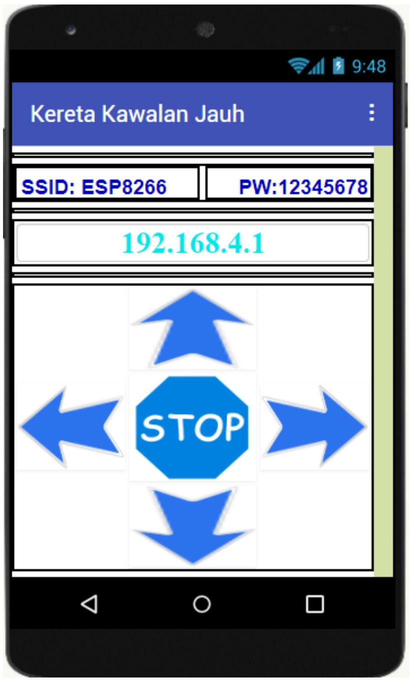
4.4.1 Pseudo Code for Master Robotic Car

```
Turn ON the power

Wait for connection from mobile application

REPEAT

     Brightness detection, Obstacle detection and Motion
     control run simultaneously

     IF brightness detected < 50

          THEN turn on LED && send ledValue = 1 to slave

     ELSE turn off LED && send ledValue = 0 to slave


     IF obstacle distance detected < 20cm

          THEN trigger buzzer && robotic car stop && send
               command == "stop" to slave

     ELSE off the buzzer


     IF mobile application connected

          THEN wait for commands from mobile application

               IF command == "forward"

                    THEN robotic car moves forward &&
                         send command == "forward" to
                         slave

               ELSE IF command == "backward"

                    THEN robotic car moves backward &&
                         send command == "backward" to
                         slave

               ELSE IF command == "left"

                    THEN robotic car turns left && send
                         command == "left" to slave
               ELSE IF command == "right"

                    THEN robotic car turns right && send
                         command == "right" to slave

               ELSE robotic car stopped && send command
                    == "stop" to slave


     ELSE wait for connection with mobile application

UNTIL Power is OFF
```

4.4.2 Pseudo Code for Slave Robotic Car

```
Turn ON the power

Wait for connection to Master robotic car

REPEAT

     IF connected with Master robotic car

          THEN wait for data from Master robotic car

               IF ledValue = "1"

                    THEN turn on LED

               ELSE turn off LED


               IF command == "forward"

                    THEN robotic car moves forward

               ELSE IF command == "backward"

                    THEN robotic car moves backward

               ELSE IF command == "left"

                    THEN robotic car turns left

               ELSE IF command == "right"

                    THEN robotic car turns right

               ELSE robotic car stopped
               IF obstacle distance detected < 20cm

                    THEN trigger buzzer && robotic car
                         automatically moves into one
                         line

               ELSE off the buzzer


     ELSE wait for connection to Master robotic car

UNTIL Power is OFF
```

4.5 GUI Design



Kereta Kawalan Jauh

SSID: ESP8266 PW:12345678

192.168.4.1

***Figure 4.5** Mobile application interface design*

4.6 Concluding Remark

The functionalities and the structure of the robotic application as well as the interface design of mobile application are briefly shown using diagram. The robotic application developed will have the function of brightness or light detection and obstacle detection. The master robotic car is controlled remotely using mobile application through Wi-Fi connection for the car's movement and these functionalities start to run once the robotic car is powered on. Whereas the slave robotic car connects to the master car through Wi-Fi and follows the command from the master car for brightness and motion control.

Chapter 5
System Implementation

In this chapter, we present about the setup, setting and configuration, as well as the system operation with screenshots.

5.1 Hardware Setup

ESP8266 Microchip

The main hardware used for developing the enhanced prototype robotic application which is the robotic car controlled by mobile application through Wi-Fi and abled to control another car will be ESP8266 microchip. ESP8266 microchip will be fixed on a breadboard and placed on the robotic car platform and it is ready to be joint and programmed.

L298N Motor Driver and DC Motors

The robotic application needs to move in four directions using two direct-current motors, controlled by a dual H-bridge motor driver which modelled L298N. L298N will be connected using female-to-female jumper wires to the ESP8266 microchip. The IN1 to IN4 pins on the L298N will be connected to the pins GPIO5, GPIO4, GPIO0, and GPIO2 on the microchip respectively. Whereas for DC motors on the robotic car platform together with two wheels, each DC motor is connected to the pins labelled OUT1, OUT2 and OUT3, OUT4 on the L298N using wires.

LDR Sensor and LED

The light-dependent resistor will be also used to measure the room brightness, and the response will be shown using LED in which it will turn on when it's in dark and vice versa. To provide current to the LDR sensor, it is connected to the pin 3.3V on ESP8266. And before connecting LDR sensor to ESP8266 pin ADC, LDR sensor will firstly connected to resistors to reduce the output voltage to less than 1 Volt through Voltage-Divider Rule. This is because ESP8266 pin ADC only with input voltage range from 0 to 1 Volt, so we need to cut down the voltage sent from LDR sensor to the pin in order to prevent the pin ADC from burning out. To show response after getting the brightness value from LDR sensor, LED is connected to the pin GPIO15 on ESP8266 with a 330 Ohm resistor.

$$Voltage\ Divider\ Rule\ Formula: Vout = Vin\ (\frac{R1}{R1 + R2})$$

$$1V = 3.3V\ (\frac{100\Omega}{100\Omega + 220\Omega})$$

HC-SR04 Ultrasonic Sensor and Buzzer

The robotic application will be using the ultrasonic sensor to detect the distance between the obstacle and the robotic car, and the buzzer will be used to alert the user when the robotic car is approaching the obstacle. HC-SR04 ultrasonic sensor is placed at the front of the robotic car platform like a pair of eyes and it is connected to ESP8266 pin GPIO14 with pin ECHO and connected to ESP8266 GPIO12 with pin TRIG. Buzzer will be connected to ESP8266 pin GPIO13 and it is programmed later to carry out its function.

Rechargeable Batteries

Moreover, rechargeable batteries are used to provide power to the motor driver, ESP8266 microchip and ultrasonic sensor. In this project, we separate the use of rechargeable batteries to provide 12V voltage for L298N motor driver and 5V voltage for ESP8266 microchip and HC-SR04 ultrasonic sensor. A total of eight 1.5V rechargeable batteries is connected to L298N motor driver to supply 12V voltage, and a total of four 1.5V rechargeable batteries is connected to the pin Vin on the ESP8266 microchip and to the pin Vcc on the HC-SR04 ultrasonic sensor. The pin Gnd on ESP8266 is connected in the breadboard as a common ground and every ground pin from each component is connected to it.

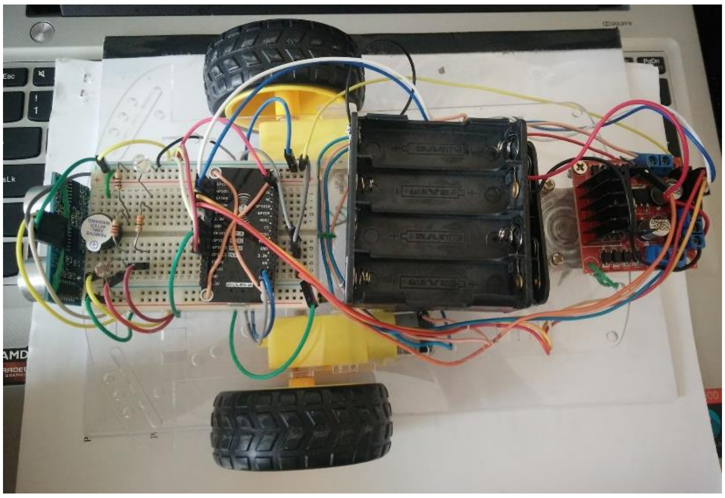
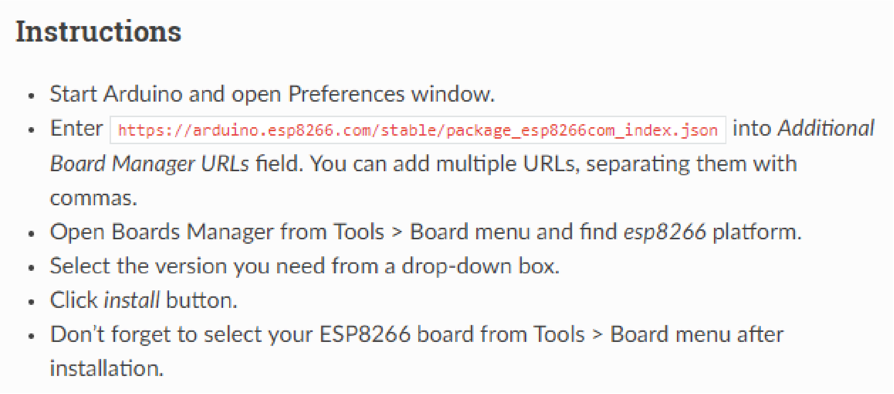


Figure 5.1 *Hardware Component Assembling*

5.2 Software Setup

Arduino IDE

The open-source Arduino IDE which will be used to program the ESP8266 microcontroller needed to be installed in personal computer or laptop. After installing it, the ESP8266 package needs to be installed into the board manager so that we can program our ESP8266 and use the libraries that is already given in the ESP8266 package. After installing the package, we can start to write our codes and upload it to ESP8266 to run the program.



Instructions

- Start Arduino and open Preferences window.
- Enter `https://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and find *esp8266* platform.
- Select the version you need from a drop-down box.
- Click *install* button.
- Don't forget to select your ESP8266 board from Tools > Board menu after installation.

Figure 5.2 *Instruction to install ESP8266 package*

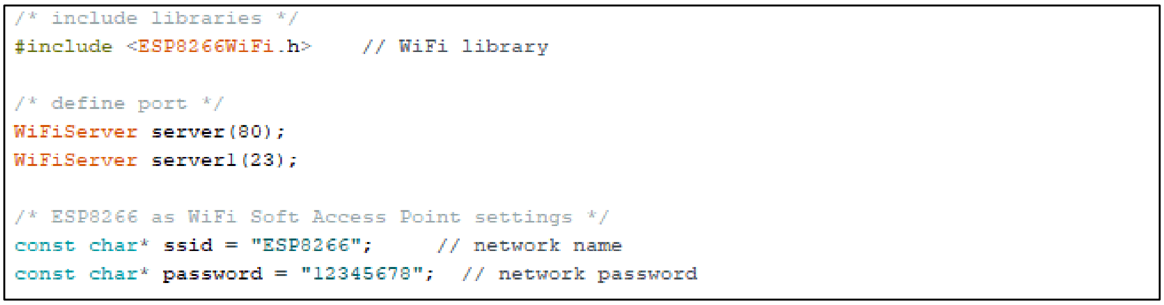
5.3 Setting and Configuration

5.3.1 Network Configuration

To connect both ESP8266 and mobile application through Wi-Fi, we developed the mobile application and configured the ESP8266 microchip as the soft access point. Besides that, to enable communication between the master robotic car and the slave robotic car, the server-client communication is established using WiFiServer and WiFiClient functions available in ESP8266WiFi library. ESP8266 microchip on the master car is configured as the server and ESP8266 microchip on the slave car is configured as the client.

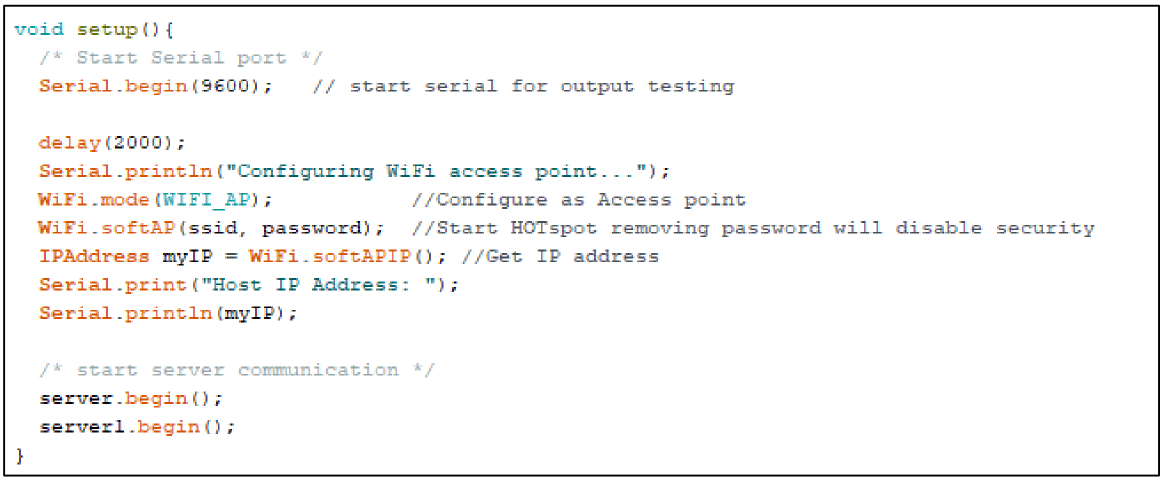
Configure ESP8266 as Soft Access Point and Server in Master robotic car

Since ESP8266 microchip consists of Wi-Fi feature, we configured it as the soft access point and connected to it using our mobile application and the slave robotic car. To configure it as access point, we included the Wi-Fi library for ESP8266. After that, we defined the port numbers for the Wi-Fi servers. In the robotic application, we defined port 80 as the server communication for mobile application and port 23 as the server communication for master ad slave robotic cars communication. Access point setting is also done by configuring the network name and network password for the ESP8266 microchip. ESP8266 microchip is setup as the access point by configuring the mode and the server communications are started to allow the connection.



```
/* include libraries */
#include <ESP8266WiFi.h>    // WiFi library

/* define port */
WiFiServer server(80);
WiFiServer server1(23);

/* ESP8266 as WiFi Soft Access Point settings */
const char* ssid = "ESP8266";     // network name
const char* password = "12345678";  // network password
```

Figure 5.3 *Configuration of server and soft access point in Master robotic car*

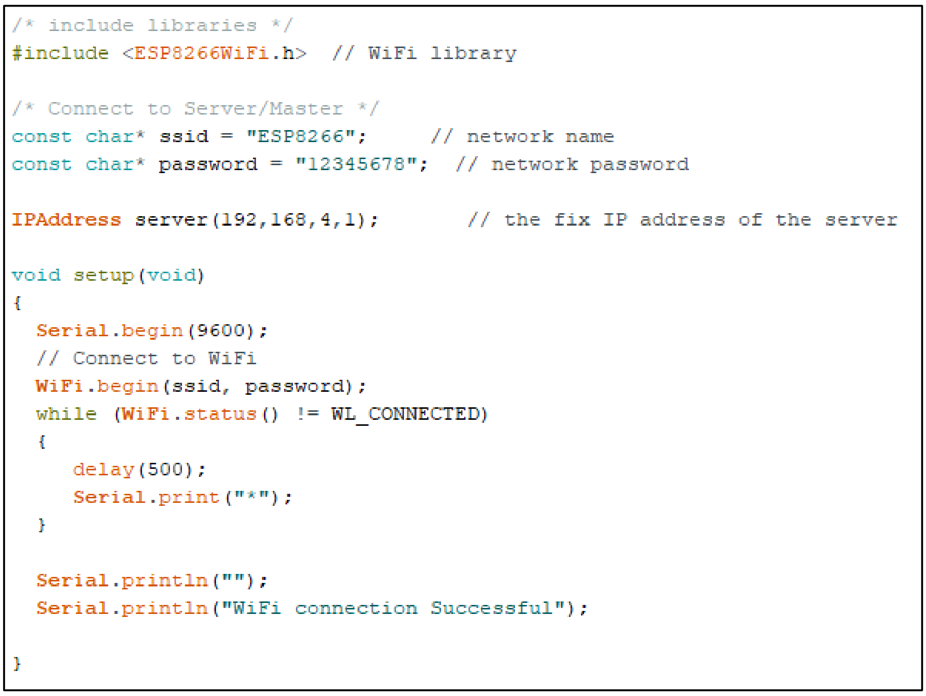


```
void setup(){
  /* Start Serial port */
  Serial.begin(9600);   // start serial for output testing

  delay(2000);
  Serial.println("Configuring WiFi access point...");
  WiFi.mode(WIFI_AP);           //Configure as Access point
  WiFi.softAP(ssid, password);  //Start HOTspot removing password will disable security
  IPAddress myIP = WiFi.softAPIP(); //Get IP address
  Serial.print("Host IP Address: ");
  Serial.println(myIP);

  /* start server communication */
  server.begin();
  server1.begin();
}
```

Figure 5.4 *Setup of ESP8266 as Soft Access Point and Start server communication*

Configure Wi-Fi and Server connection in Slave robotic car

Similar to the master robotic car, in order to enable Wi-Fi connection, Wi-Fi library for ESP8266 is included. The network name and password to connect are also defined together with the fixed IP address of the master robotic car. After the Wi-Fi connection is established by the master robotic car, the slave robotic car connects to the Wi-Fi automatically.



```
/* include libraries */
#include <ESP8266WiFi.h>  // WiFi library

/* Connect to Server/Master */
const char* ssid = "ESP8266";     // network name
const char* password = "12345678";  // network password

IPAddress server(192,168,4,1);       // the fix IP address of the server

void setup(void)
{
  Serial.begin(9600);
  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
     delay(500);
     Serial.print("*");
  }

  Serial.println("");
  Serial.println("WiFi connection Successful");

}
```

Figure 5.5 *Configuration and Setup of Wi-Fi connection in Slave robotic car*

Mobile Application Development

To remotely control the robotic car, mobile application is developed using open source MIT App Inventor 2. App Inventor is a cloud-based tool, apps can be built right in the web browser. App Inventor is programmed using block-based programming. We just have to drag and drop the buttons, layout, textbox, so on and so forth we wanted to use in designing the apps interface. Whereas to design the functionalities of the buttons, textbox and others, we drag and drop the function blocks and joint them together to form the complete function we want. The mobile application created has the functions to control the movement of the robotic car in four directions. To ensure that the mobile application is connected to the correct access point, IP address of the ESP8266 microchip is fixed in the application. User only needs to connect the mobile phone Wi-Fi to the access point we configured for ESP8266 microchip, then the mobile application developed is ready to use.

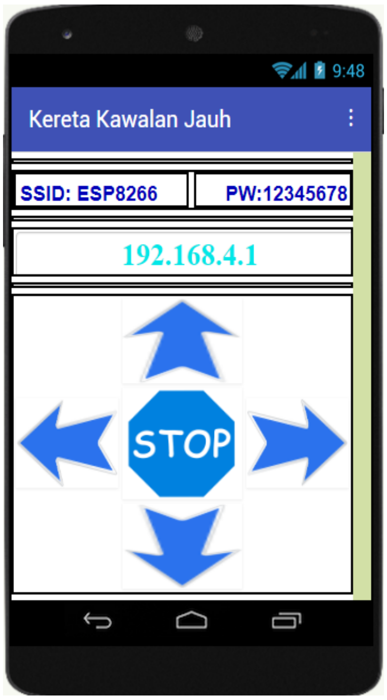


Figure 5.6 *Interface of the mobile application developed*

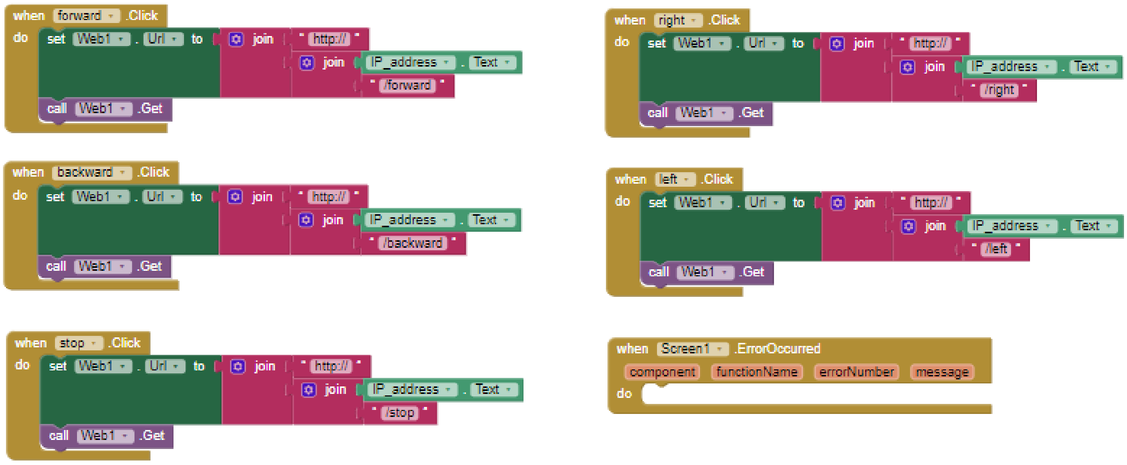
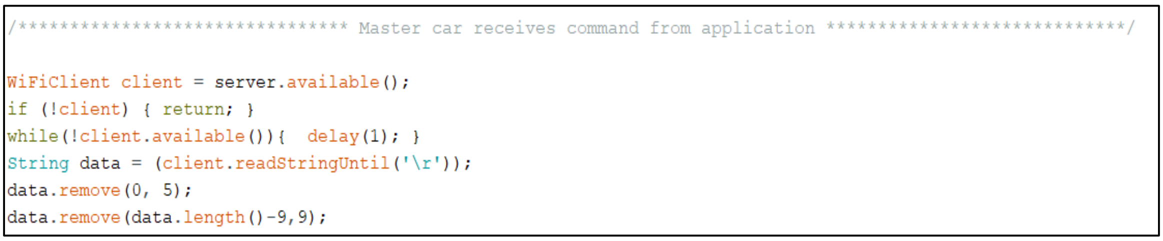


Figure 5.7 *Block-based programming in MIT App Inventor 2*

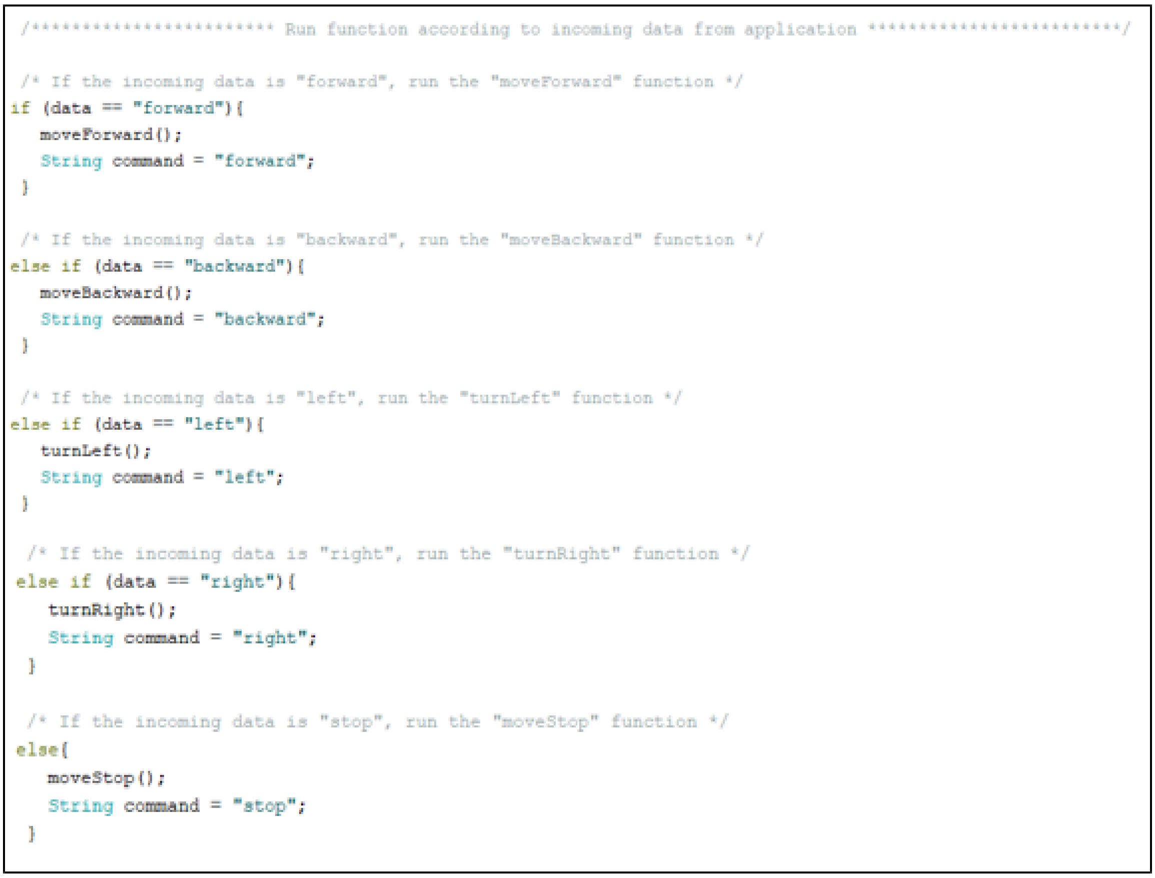
Program of ESP8266 in Master robotic car to Receive Command from Mobile Application Developed

In order to control the robotic car using mobile application, ESP8266 is programmed to receive the command from the mobile application and carry out the response accordingly. Client port is defined for server communication at port 80 and variables are created to receive the command from the mobile application and store commands that needed to send to the slave robotic car. After that, if the mobile application is connected to the ESP8266 access point, incoming data from the mobile application is received and the functions are run accordingly, at the same time, the master robotic car stores the command which needed to send to the slave robotic car to follow the functions carried out the master car. Otherwise, the master robotic car waits for the connection and the command entered.



```
/******************************** Master car receives command from application ****************************/

WiFiClient client = server.available();
if (!client) { return; }
while(!client.available()){  delay(1); }
String data = (client.readStringUntil('\r'));
data.remove(0, 5);
data.remove(data.length()-9,9);
```

Figure 5.8 *Master robotic car receives data from mobile application*

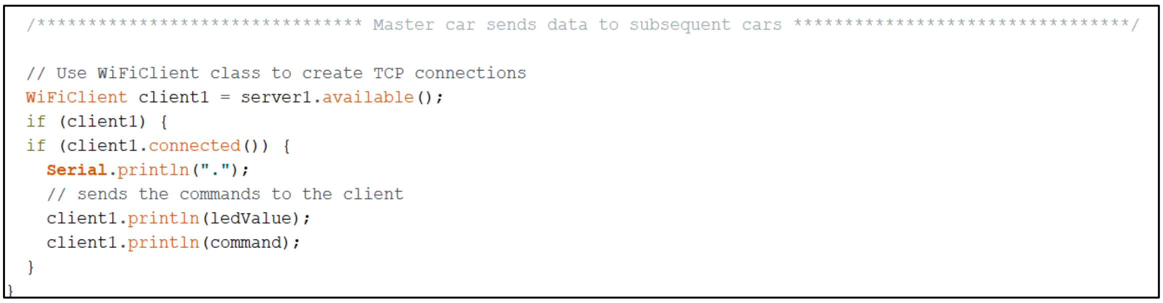


```
/************************ Run function according to incoming data from application ************************/

 /* If the incoming data is "forward", run the "moveForward" function */
if (data == "forward"){
   moveForward();
   String command = "forward";
 }

 /* If the incoming data is "backward", run the "moveBackward" function */
else if (data == "backward"){
   moveBackward();
   String command = "backward";
 }

 /* If the incoming data is "left", run the "turnLeft" function */
else if (data == "left"){
   turnLeft();
   String command = "left";
 }

 /* If the incoming data is "right", run the "turnRight" function */
else if (data == "right"){
   turnRight();
   String command = "right";
 }

 /* If the incoming data is "stop", run the "moveStop" function */
else{
   moveStop();
   String command = "stop";
 }
```

Figure 5.9 *Master robotic car responses and stores commands for slave accordingly*

Program of ESP8266 in Master robotic car to Send Command to Slave robotic car

ESP8266 is also programmed to send command to the slave robotic car in order to control the motion of the slave car. Another client port is also defined for server communication at port 23. After that, if the slave robotic car is connected to the server at port 23, the master robotic car writes the data stored in variables *ledValue* and *command* into the server using *println* function.

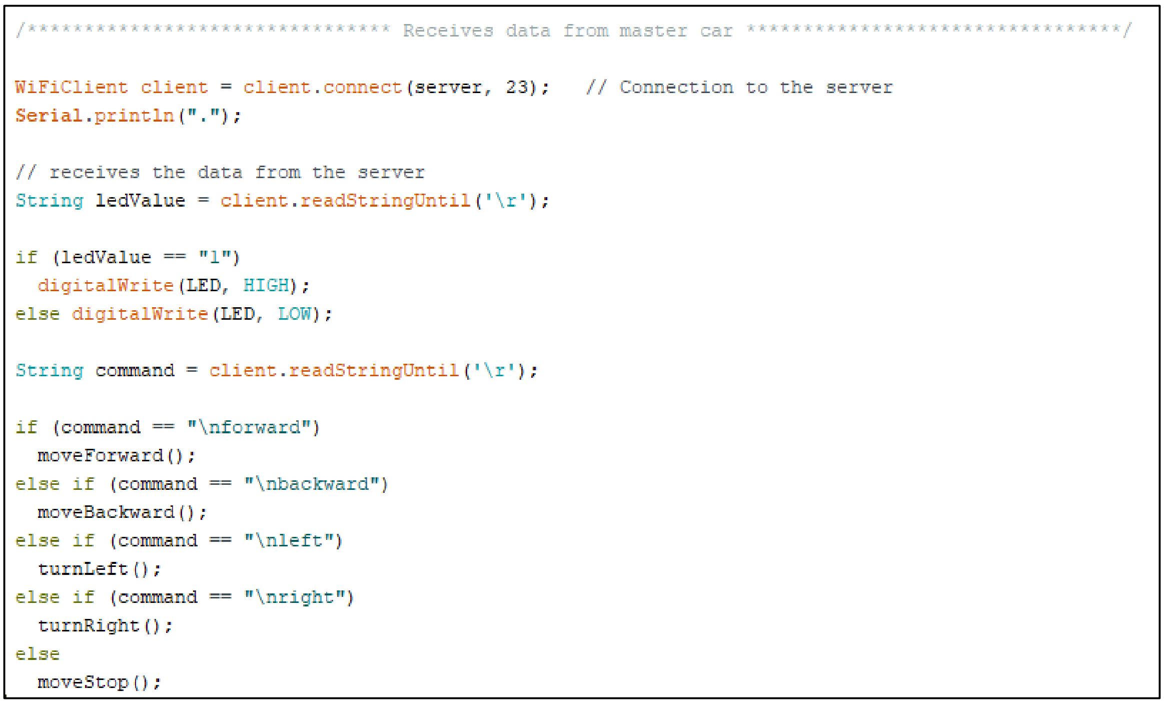


```
/******************************** Master car sends data to subsequent cars ********************************/

 // Use WiFiClient class to create TCP connections
 WiFiClient client1 = server1.available();
 if (client1) {
 if (client1.connected()) {
   Serial.println(".");
   // sends the commands to the client
   client1.println(ledValue);
   client1.println(command);
 }
}
```

Figure 5.10 *Master robotic car sends commands to the server*

Program of ESP8266 in Slave robotic car to Receive Command from Master robotic car through Server-Client communication

On the other hand, ESP8266 microchip on the slave robotic car is programmed to retrieve the command from the server and carry out the response accordingly. Client port is defined to connect the server at port 23 and different variables are created to receive different data from the server. After the slave robotic car is connected to the server, it reads the data from the server using *readStringUntil* function and performs the functions according to the commands received.



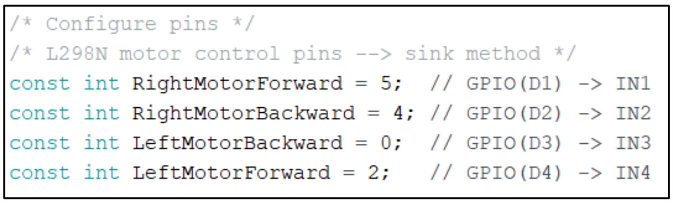
```
/******************************** Receives data from master car *********************************/

WiFiClient client = client.connect(server, 23);   // Connection to the server
Serial.println(".");

// receives the data from the server
String ledValue = client.readStringUntil('\r');

if (ledValue == "1")
  digitalWrite(LED, HIGH);
else digitalWrite(LED, LOW);

String command = client.readStringUntil('\r');

if (command == "\nforward")
  moveForward();
else if (command == "\nbackward")
  moveBackward();
else if (command == "\nleft")
  turnLeft();
else if (command == "\nright")
  turnRight();
else
  moveStop();
```

Figure 5.11 *Slave robotic car retrieves data from server and perform the functions accordingly*

5.3.2 Hardware Interface Programming

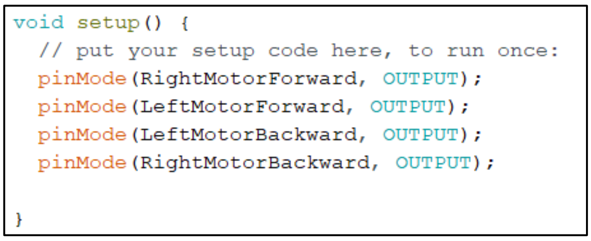
L298N Motor Driver and DC Motors

To program L298N motor driver which connected to ESP8266 microchip, we need to firstly define the pins used and setup the pin mode as output inside the setup function as shown in Figure 5.12 and Figure 5.13.



```
/* Configure pins */
/* L298N motor control pins --> sink method */
const int RightMotorForward = 5;  // GPIO(D1) -> IN1
const int RightMotorBackward = 4; // GPIO(D2) -> IN2
const int LeftMotorBackward = 0;  // GPIO(D3) -> IN3
const int LeftMotorForward = 2;   // GPIO(D4) -> IN4
```

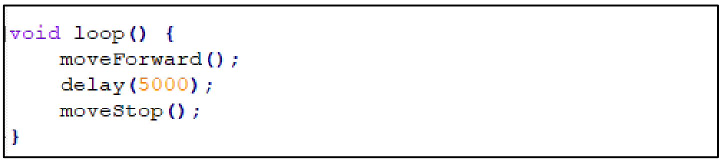
Figure 5.12 *Configuration of the pin numbers*



```
void setup() {
  // put your setup code here, to run once:
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

}
```

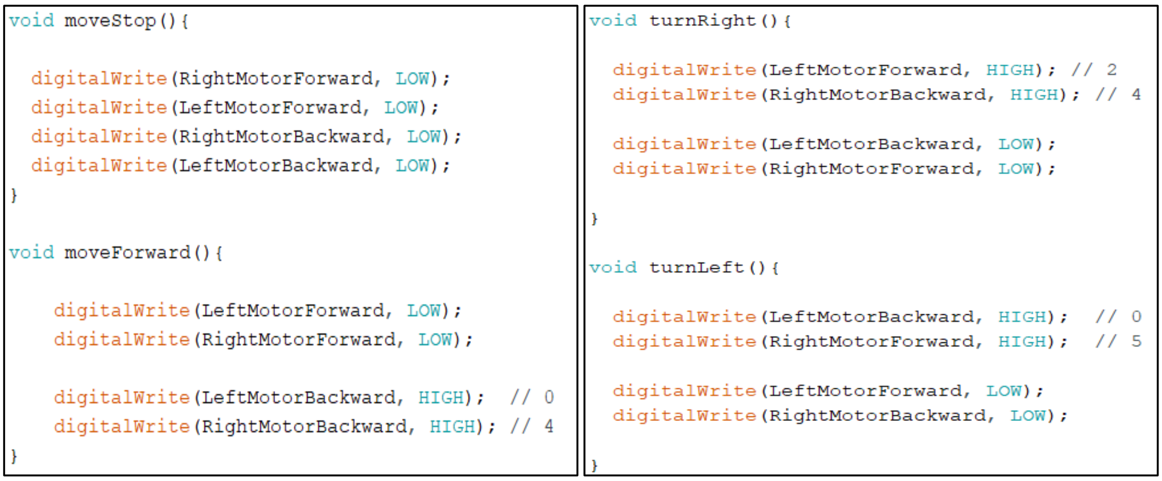
Figure 5.13 *Setup of the pin mode as output*

After that, we can test the function of the L298N to check if it can run the motor and make the wheels move by using *digitalWrite* function to configure the pins as low or high. The Figure 5.14 shows the code that enable the motor to run forward for 5 seconds and then stop, whereas the Figure 5.15 below shows part of the codes which enable the motor to run forward, stop, turn left and right using *digitalWrite* function.



```
void loop() {
    moveForward();
    delay(5000);
    moveStop();
}
```

Figure 5.14 *Motors move forward for 5 seconds and stop*

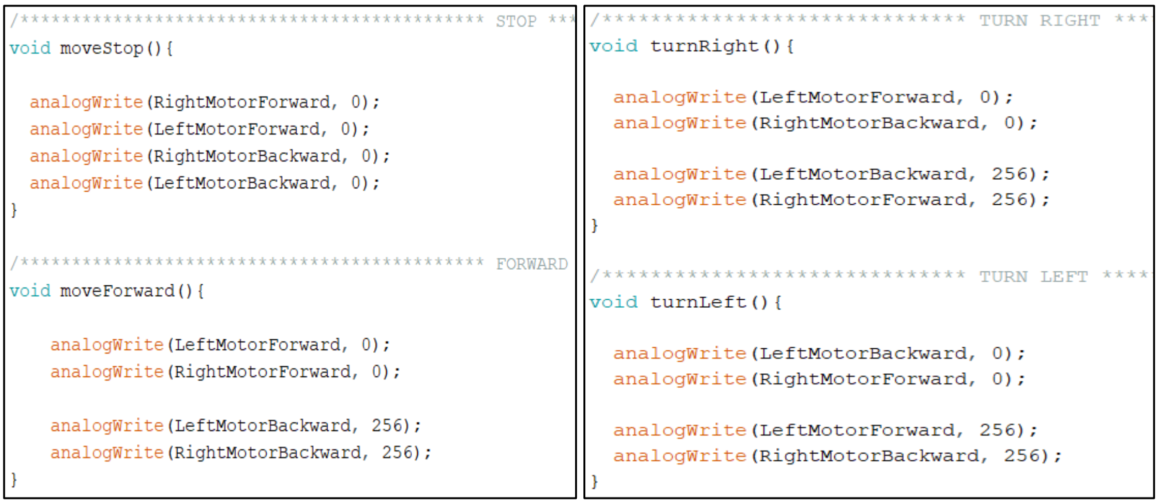


```
void moveStop(){

  digitalWrite(RightMotorForward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);

    digitalWrite(LeftMotorBackward, HIGH);  // 0
    digitalWrite(RightMotorBackward, HIGH); // 4
}
```

```
void turnRight(){

  digitalWrite(LeftMotorForward, HIGH); // 2
  digitalWrite(RightMotorBackward, HIGH); // 4

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);

}

void turnLeft(){

  digitalWrite(LeftMotorBackward, HIGH);  // 0
  digitalWrite(RightMotorForward, HIGH);  // 5

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
```

Figure 5.15 *Part of codes to control the motors using digitalWrite function*

In the robotic application, we use *analogWrite* function to configure the pins in order to control the speed of the motors from range 0 to 1024 as shown in Figure 5.16. As we can see from the codes, we use sink method to program the motor driver, in which it will move forward when the backward motor is putting high, while the forward motor is putting low.



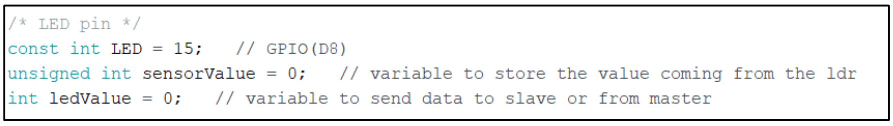
```
/******************************************** STOP ***
void moveStop(){

  analogWrite(RightMotorForward, 0);
  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);
  analogWrite(LeftMotorBackward, 0);
}

/******************************************** FORWARD
void moveForward(){

    analogWrite(LeftMotorForward, 0);
    analogWrite(RightMotorForward, 0);

    analogWrite(LeftMotorBackward, 256);
    analogWrite(RightMotorBackward, 256);
}
```

```
/****************************** TURN RIGHT ***
void turnRight(){

  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);

  analogWrite(LeftMotorBackward, 256);
  analogWrite(RightMotorForward, 256);
}

/****************************** TURN LEFT ****
void turnLeft(){

  analogWrite(LeftMotorBackward, 0);
  analogWrite(RightMotorForward, 0);

  analogWrite(LeftMotorForward, 256);
  analogWrite(RightMotorBackward, 256);
}
```

Figure 5.16 *Part of codes to control the motors using analogWrite function*

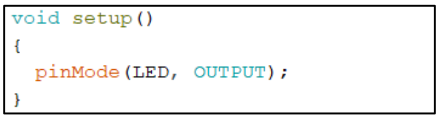
LDR Sensor and LED

To program ESP8266 microchip to carry out LDR sensor and LED functionalities, similar steps as above are done. Firstly, pin number that connects LED to the ESP8266 microchip is defined, and variables are defined to store the value measured by the LDR sensor and the data for slave and master communication. The pin mode needs to be setup as the output for the LED to light up. By default, the LEDs remain turned off.



```
/* LED pin */
const int LED = 15;   // GPIO(D8)
unsigned int sensorValue = 0;   // variable to store the value coming from the ldr
int ledValue = 0;   // variable to send data to slave or from master
```

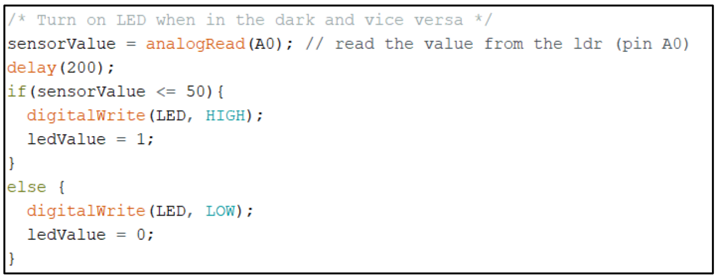
Figure 5.17 *Configuration of pin number and variable for LDR sensor and LED*



```
void setup()
{
  pinMode(LED, OUTPUT);
}
```

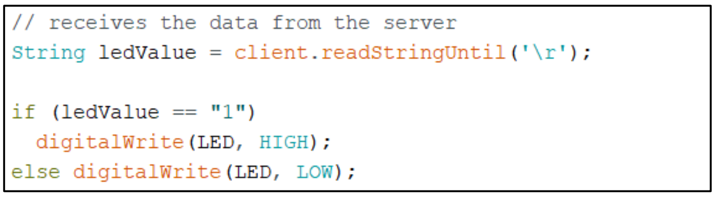
Figure 5.18 *Setup of LED pin mode as output*

For the master robotic car, LDR sensor is functioning to detect the room brightness and a limit is set to consider the brightness as dark or bright. In the robotic application, we set a limit of 50 as a threshold. If the *sensorValue* obtained from the LDR sensor is below 50, the room brightness is considered as dark and the LED is lighted up. A variable called *ledValue* is also stored and sent to the slave robotic car so that the car turns on the LED as well. Whereas for the slave robotic car, it receives data from the master and responses to it automatically. If the *ledValue* received is 1, the car turns on the LED and vice versa.



```
/* Turn on LED when in the dark and vice versa */
sensorValue = analogRead(A0); // read the value from the ldr (pin A0)
delay(200);
if(sensorValue <= 50){
  digitalWrite(LED, HIGH);
  ledValue = 1;
}
else {
  digitalWrite(LED, LOW);
  ledValue = 0;
}
```

Figure 5.19 *Configuration of LED and LDR functions in Master robotic car*

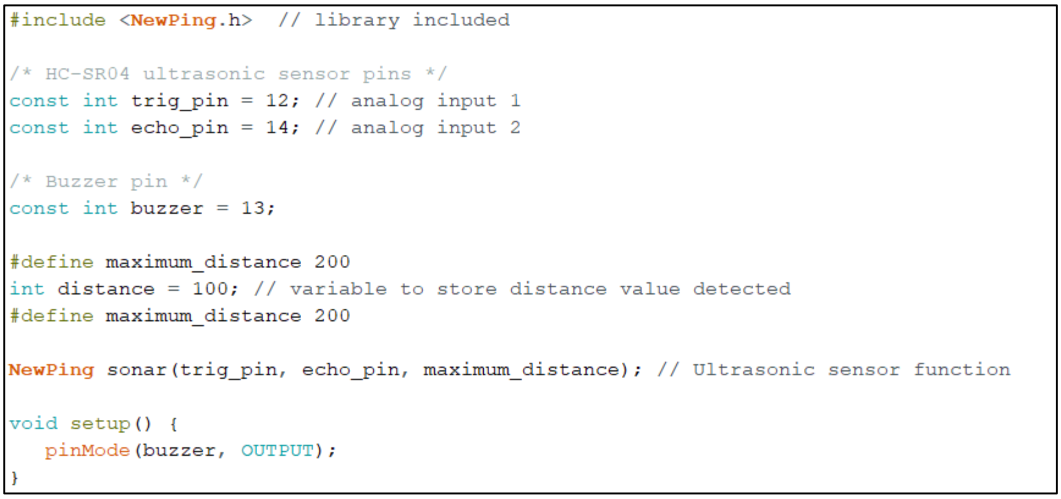


```
// receives the data from the server
String ledValue = client.readStringUntil('\r');

if (ledValue == "1")
  digitalWrite(LED, HIGH);
else digitalWrite(LED, LOW);
```

Figure 5.20 *Configuration of LED functions in Slave robotic car*

HC-SR04 Ultrasonic Sensor and Buzzer

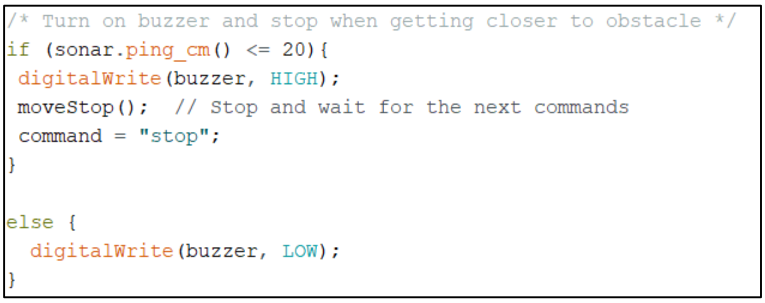
To perform the functionalities of ultrasonic sensor and buzzer, ESP8266 microchip required to be programmed. NewPing library which equipped with the features such as built-in function for ultrasonic sensor is included to ease our programming process. Defining the pins and setting up the pin modes is a must, and to program ultrasonic sensor, few variables are added to store the distance value detected by the sensor. Built-in function from NewPing library, *sonar()* is called to measure the distance between the obstacle and the robotic car. The distance between obstacle and the robotic car is measured in centimetre by calling the function *sonar.ping_cm()*.



```
#include <NewPing.h>  // library included

/* HC-SR04 ultrasonic sensor pins */
const int trig_pin = 12; // analog input 1
const int echo_pin = 14; // analog input 2

/* Buzzer pin */
const int buzzer = 13;

#define maximum_distance 200
int distance = 100; // variable to store distance value detected
#define maximum_distance 200

NewPing sonar(trig_pin, echo_pin, maximum_distance); // Ultrasonic sensor function

void setup() {
   pinMode(buzzer, OUTPUT);
}
```

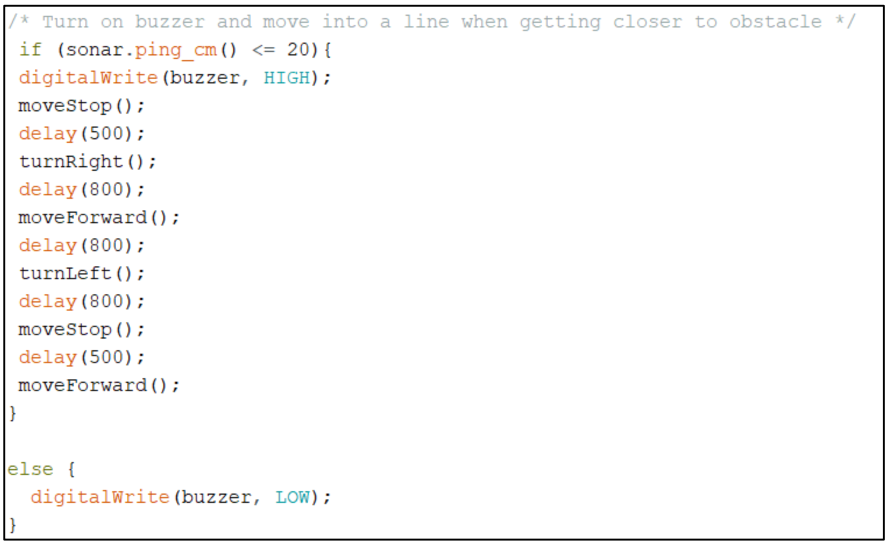
Figure 5.21 *Configuration and setup of HC-SR04 ultrasonic sensor and buzzer*

The function of ultrasonic sensor is different for master robotic car and slave robotic car. For master robotic car, when the robotic car is approaching the obstacle within 20cm, the buzzer is turned on to alert the user and the car is stop automatically. A command is also stored and sent to the slave robotic car to stop the slave robotic car as well. Whereas for the slave robotic car, when the robotic car is approaching the obstacle within 20cm, the buzzer is turned on to alert the user and the car is moved into a formation topology which is to move in a line. In the robotic application, the location and distance between the master car and the slave car is fixed, hence, the algorithm designed is also hardcoded for the slave robotic car.



```
/* Turn on buzzer and stop when getting closer to obstacle */
if (sonar.ping_cm() <= 20){
 digitalWrite(buzzer, HIGH);
 moveStop();  // Stop and wait for the next commands
 command = "stop";
}

else {
  digitalWrite(buzzer, LOW);
}
```

Figure 5.22 *Function of HC-SR04 ultrasonic sensor and buzzer in Master robotic car*

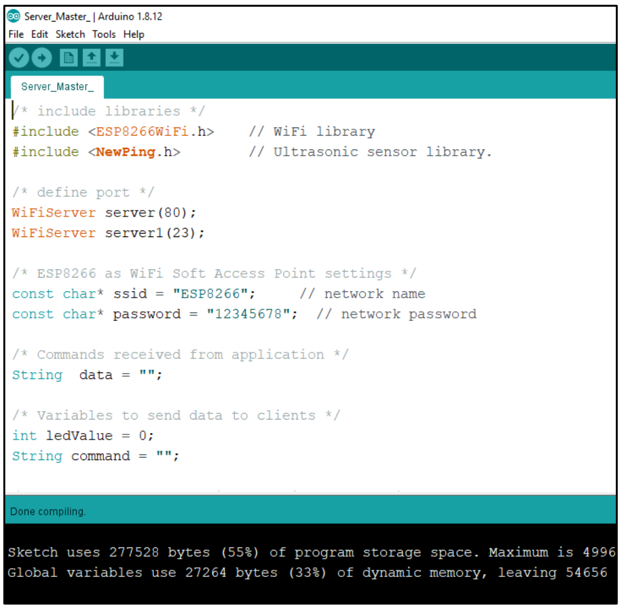


```
/* Turn on buzzer and move into a line when getting closer to obstacle */
 if (sonar.ping_cm() <= 20){
 digitalWrite(buzzer, HIGH);
 moveStop();
 delay(500);
 turnRight();
 delay(800);
 moveForward();
 delay(800);
 turnLeft();
 delay(800);
 moveStop();
 delay(500);
 moveForward();
}

else {
  digitalWrite(buzzer, LOW);
}
```

Figure 5.23 *Function of HC-SR04 ultrasonic sensor and buzzer in Slave robotic car*

5.4 System Operation

After all the configuration and setup is completed, the codes are compiled and uploaded to ESP8266 microchips in order to carry out their functionalities.



```
/* include libraries */
#include <ESP8266WiFi.h>     // WiFi library
#include <NewPing.h>         // Ultrasonic sensor library.

/* define port */
WiFiServer server(80);
WiFiServer server1(23);

/* ESP8266 as WiFi Soft Access Point settings */
const char* ssid = "ESP8266";      // network name
const char* password = "12345678";  // network password

/* Commands received from application */
String  data = "";

/* Variables to send data to clients */
int ledValue = 0;
String command = "";
```

Done compiling.

```
Sketch uses 277528 bytes (55%) of program storage space. Maximum is 4996
Global variables use 27264 bytes (33%) of dynamic memory, leaving 54656
```

***Figure 5.24** Code compilation succeed without error*

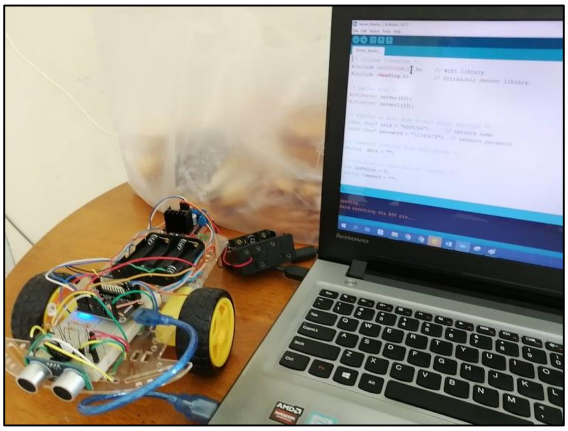


***Figure 5.25** Code uploaded to ESP8266*

5.5 Concluding Remark

The hardware setup and software setting and configuration for the robotic application is written in details. Different components will have different setup, and the setup need to be carried out carefully and correctly to protect the hardware component from burning-out. Network configuration is also done on ESP8266 microchip to allow Wi-Fi connection and server-client communication by mobile application and the subsequent robotic car. After all the configuration is completed, the robotic application has to go through with the system operation to upload all the setup and configuration into ESP8266 microchip before it can perform its functionalities.

Chapter 6

System Evaluation and Discussion

In this chapter, we present about the system testing and performance metrics, testing setup and results, project challenges as well as objectives evaluation.

6.1 System Testing and Performance Metrics

6.1.1 Verification Plan

Table 6.1 *Verification Test for the project*

No.	Test Features	Test Sequence	Expected Output
1.	Robotic car motion	Controls the master robotic car to move in four directions by uploading the code or controlled using mobile application. Controls the subsequent cars by sending commands from master car through Wi-Fi.	Motor runs in correct directions and the robotic car moves in correct directions as commanded. Subsequent cars follow the commands and move in the correct direction as master car.
2.	Obstacle avoidance	Controls the master robotic car to move forward and purposely put an obstacle in front of the master car. Controls the master robotic car to move forward and purposely	All robotic cars stop from moving forward and only the master robotic car turns on the buzzer to alert user. The particular robotic car turns on buzzer to alert user as well as stops from moving

		put an obstacle in front of the subsequent car.	forward and automatically turns left or right to adjust the topology before continues to follow the motion of master car.
3.	Brightness detection	Moves the robotic car which already joint with LDR sensor from a brighter place to a darker place or vice versa.	LED as the headlight of the robotic car lights up when the robotic car is in the dark place and vice versa.
4.	Connectivity through Wi-Fi	Configures the ESP8266 microcontroller on the master car as a soft access point and connect to it using other device.	Mobile application developed and the subsequent robotic cars are able to connect to the Wi-Fi from the microcontroller implemented on the master robotic car.
5.	Connectivity between master car and subsequent cars	Configures the ESP8266 microcontroller on the master car as a soft access point and connect using subsequent cars. Then, sends commands from the master car to the subsequent cars such as turn on the LED.	The subsequent robotic cars are able to connect to the Wi-Fi from the microcontroller implemented on the master robotic car, and also perform the commands sent by master car which is to turn on the LED.
6.	Mobile application control	Press any buttons in the application, for example the forward button and observe the master robotic car’s motion.	Robotic car receives the command from the mobile application correctly, for example, it moves forward when the forward button is pressed from the application.

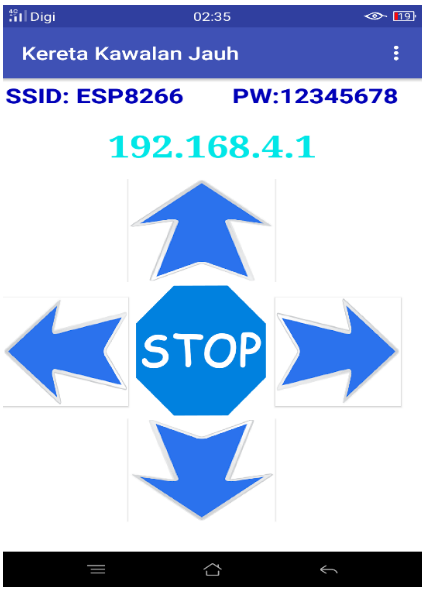
6.1.2 Checklist according to RBT course syllabus

***Table 6.2** Checklist for functionalities according to RBT course syllabus*

Syllabus	Fulfilled?
Detect the light phase	✓ Detected using LDR sensor
Prompt microcontroller to turn on LED light	✓ Automatically turn on the LED when in dark places and turn off the LED when in bright places
Detect the safe distance between the car and the obstacle	✓ Detected using ultrasonic sensor
Prompt microcontroller to stop the car when approaching an obstacle	✓ Stop and reverse automatically to avoid collide with obstacle
Tell the user about the presence of obstacles on the channel	✓ Alert the user about the presence of obstacle using buzzer

6.2 Testing Setup and Result

The complete setup of the robotic application is shown in figures below.



Kereta Kawalan Jauh

SSID: ESP8266 PW:12345678

192.168.4.1

***Figure 6.1** Interface of the mobile application*

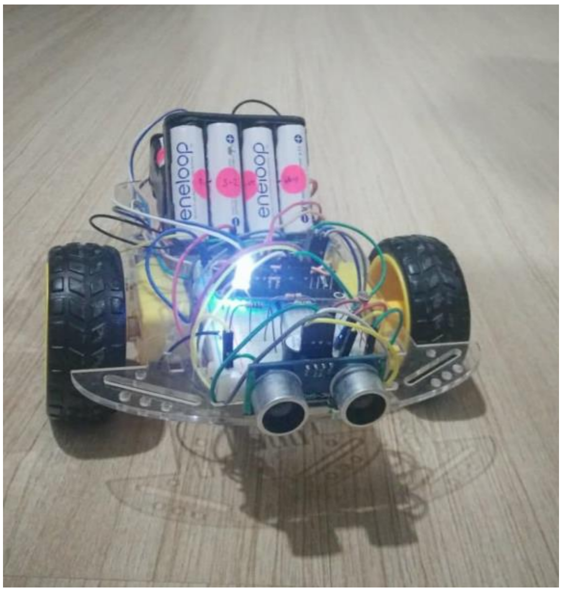


***Figure 6.2** Front view of the robotic car*

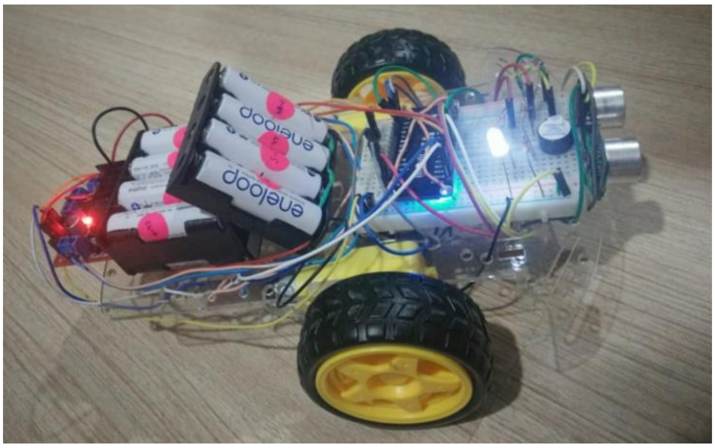


***Figure 6.3** Side view of the robotic car*

Results of brightness detection by the robotic application is shown in Figure 5.27. The LED light up when it is dark (the LDR sensor is covered to represent dark) and vice versa.

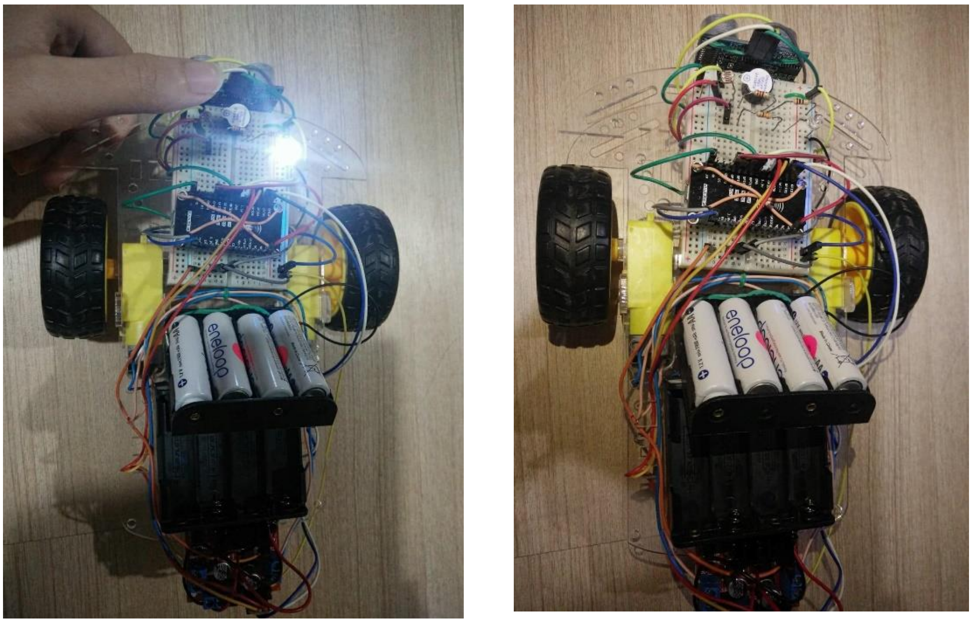


***Figure 6.4** Result of brightness detection*

Example of communication between master robotic car and slave robotic car is shown in Figure 5.28. The slave robotic car follow the command from master robotic car to turn on the LED.

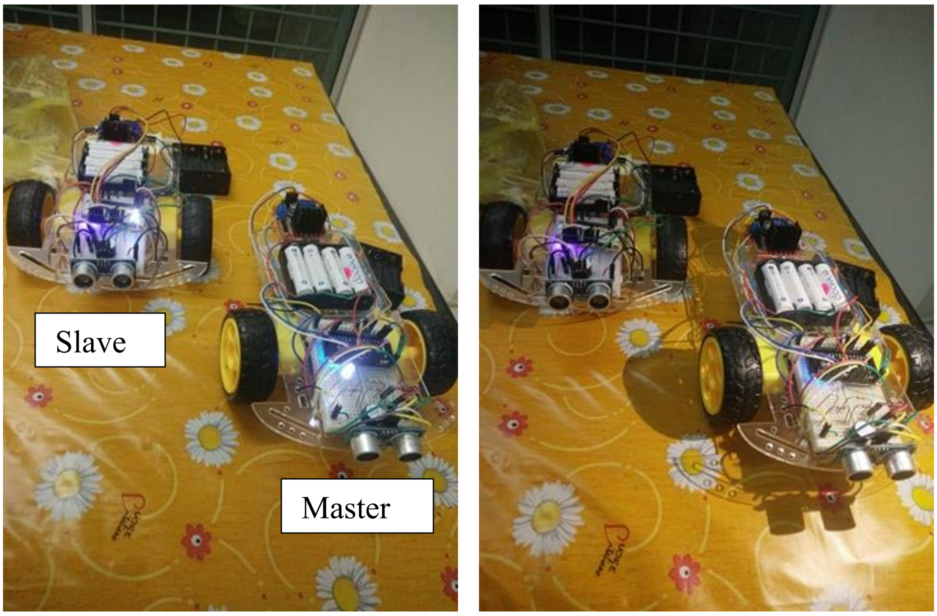


***Figure 6.5** Example of communication between master robotic car and slave robotic car*

Results of the robotic car's motion controlled by mobile application and the obstacle avoidance could not clearly shown using figures, but they are clearly observed during demonstration or testing.

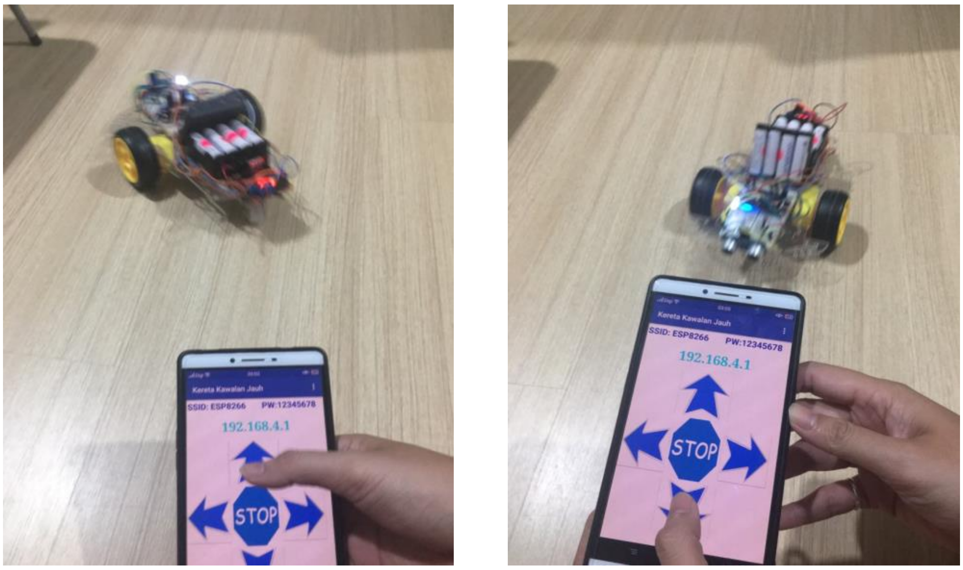


Figure 6.6 *Result of robotic car's motion according to the commands from mobile application such as forward and backward.*

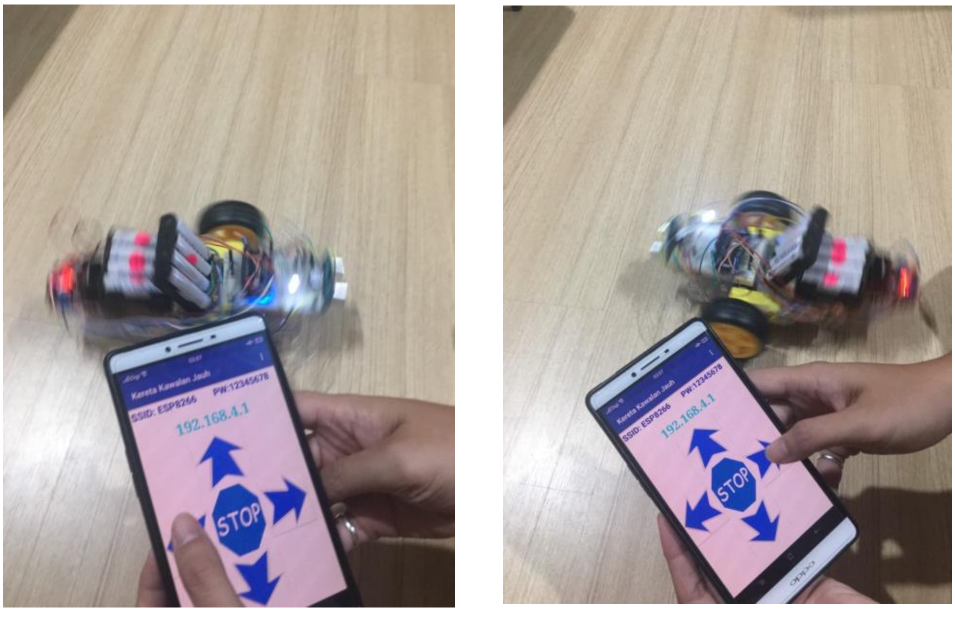


Figure 6.7 *Result of robotic car's motion according to the commands from mobile application such as turn left and turn right.*

The result for every functionalities is good and it is feasible for the use of secondary school RBT course due to its simplicity to assembly, program and understand, affordable and it performs functionalities according to the course syllabus exactly.

6.3 Project Challenges

Some difficulties arise during the project development, one of it is the loosen connection of wires. During the implementation and testing of the robotic application, challenges occur when there is a strong movement on the robotic application, which will loosen the wires connected. The wires loosen problem causes the implementation and testing of the robotic application become intermittent and unstable. We have to always check for the wires before or during the implementation to ensure that all components are connected and functioned. For instance, during testing, the robotic application will stop or switch off suddenly due to the wire has loosen. We need to plug in the wire again and it takes time for the connection of Wi-Fi from the mobile application before we can continue to test the robotic application.

Another challenge faced is the battery life to supply power for the ultrasonic sensor and ESP8266 microchip. Since only four 1.5V batteries need to supply power for both microchip and ultrasonic sensor, the battery life cannot stay longer. The batteries can only last for around two hours and needs to recharge again to ensure the robotic application functions perfectly. However, it is quite obvious to know if the battery is running out, because the ultrasonic sensor requires around 4V to 5V to function well. Hence, when the ultrasonic sensor starts to function improperly, it will cause the buzzer to always sound on. This indicates that the batteries is running out of power and it cannot provide efficient power for the ultrasonic sensor to work well. Nevertheless, if we separate the power supply for ESP8266 microchip and ultrasonic sensor, means that we use four batteries to provide power for ESP8266 microchip and another four batteries for ultrasonic sensor, there will have not enough space for us to place the batteries on the robotic car platform. Therefore, we can only use the four batteries to supply power for both microchip and ultrasonic sensor, and constantly recharge the batteries for the use that longer than two hours.

Besides that, while configuring the communication between the master robotic car and the subsequent car, some challenges occurred. First of all, we planned to configure WiFiMesh as the algorithm for the multiple robot communication. However, the configuration failed due to our lack of understanding in the WiFiMesh algorithm. Thus, we came out with another algorithm for the communication between the master robotic car and the subsequent cars without affecting the original connection between the master robotic car and the mobile application. We tried to configure the robotic applications with server-client communication in which the master car is configured as the server to allow commands sending to the subsequent cars using WiFiServer library available in ESP8266 microchip. Whereas for the subsequent cars, they are configured as the clients that can retrieve data from the server using WiFiClient library. Challenge occurred as we have to figure out the configuration of two servers in one ESP8266 microchip, and at the same time the microchip has to work as a soft access point for internet connection. Once we successfully overcome this challenge, the communication between the master car and the subsequent cars is established. The master car is able to send commands to the server and the client is also able to retrieve and perform the commands. Nonetheless, another challenge occurred in which the client configured in WiFiClient did not support multiple clients and we have to try figuring out the way to make it access for multiple clients. For this algorithm configuration of the robotic application, it used up around three to four weeks for us to complete.

6.4 Objectives Evaluation

***Table 6.3** Checklist for objectives evaluation*

Objectives	Evaluations
1. Build an affordable robotic car using low cost components with suitable functionalities according to RBT course syllabus.	✓ Affordable robotic car with total expenses not exceed MYR100 per unit (Estimated Cost provided in Chapter 3) ✓ Functionalities fulfilled RBT course syllabus (Functionalities checklist provided in Chapter 6)
2. Control the master robotic car and the subsequent robotic car will follow.	✓ Master robotic car controls subsequent robotic car through Wi-Fi and Server-Client communication for motion control and brightness detection
3. Design algorithms that enable the robotic car to adjust their topology in order to line up become one in	✓ Robotic car is able to adjust their topology to line up become one in narrow walkway

narrow walkway and expand when come to wider walkway.	× Not able to expand when come to wider walkway.

6.5 Concluding Remark

A verification test is created with the types of input for testing. The system is then tested and the result is shown in figures. Because of the car is moving, obvious direction of movement cannot be shown using figure. In overall, the functionalities of the robotic application is performing well. The remote control of motion using mobile application and the communication between master car and subsequent car are also working well. The battery life and the smoothness of surface as well as the algorithm configured are the challenges faced in the implementation of this project. The objectives of this project is also evaluated in this chapter together with the checklist for functionalities according to RBT course syllabus.

Chapter 7
Conclusion and Recommendation

In this chapter, we present about the summary of the project and the recommendation to improve the robotic application.

7.1 Conclusion

With the presence of STEM-based learning program, teaching adolescent education robotic turns into an inspiring key for the future in every country. So as to allow each student get the opportunities to learn robotics, the educational robotic kits must be made in the way that the devices are available, reasonable and that can extensively embraced by teachers and students in school. As per the course schedule, this project builds up a robotic car controlled using mobile application through Wi-Fi, and with the functionalities of movement, obstacle-avoidance as well as light detection. In order to increase students' interest and make the learning process fun, additional formation topology is added for multiple robot communication which is also known as swarm robotics. Despite the fact that there are a wide range of robotic applications that are for educational purpose accessible on the market, every one of them exists with certain restrictions, for example, costly and the functions coverage does not fit to the course syllabus. Subsequently, this project creates robotic application which reasonably priced, yet fit with the course syllabus so as to overcome the secondary school budget limitation issue and enhance students' learning.

Agile system development model is being referred in the development of robotic application, wherein every functionalities is initially partitioned into smaller parts and after a continuous iteration of development and testing, each piece of the functionalities is then joined together to form the final product with all the features required. Main component used in the project is ESP8266 microchip as the brain of the application. L298N motor driver is utilized to control the turning speed and direction of motors. For obstacle-avoidance function, HC-SR04 ultrasonic sensor is used to detect the distance between the obstacle and the robotic car, and buzzer is used to trigger sound to alert users. While for light detection function, LDR is used to recognize the room brilliance and trigger signal to turn on or off the LED. Open-source Arduino IDE is used as the software environment together with the ESP8266 libraries introduced. A set of verification test is completed to test each function created and a checklist is given to guarantee the functionalities fit to the course schedule. The difficulties faced in this project are the short battery life and loosen of wire during the testing the robotic car. Besides that, incomprehension of the algorithm needed is also one of the challenges faced during the implementation of the project. The timeline of this project is shown and an expected expense not more than MYR100 is determined for one unit of the proposed robotic application.

The overall system architecture is indicated using the hardware block diagram to succinctly demonstrate the hardware components required in the project. A basic functional modules is drew to give a brief understanding about the functions of the robotic application consists of. To have a clear mind-set on the system flow, a flowchart and pseudo code is created, as well as the GUI design of the mobile application. All the hardware components are associated through the pins using wires dependent on the block diagram and the components' datasheets accurately and cautiously to avoid each component from burning-out. The software setup is also done before the configuration and setup of the functionalities of the robotic application. In the project, ESP8266 microchip is configured as the soft access point to allow Wi-Fi connection from the mobile application and the subsequent robotic car. Server-client communication is also established to allow leader-follower approach between the master robotic car and the subsequent robotic car. The system is finally tested and the outcome demonstrates a success of the robotic application, where every one of the functionalities of the robotic application are performing well. Hence, the project is completed effectively and successfully with objectives accomplished.

7.2 Recommendation

Additionally, there are some improvements can be done for the enhanced robotic application. First and foremost, a mesh algorithm can be designed to replace the server-client communication for multiple robots coordination in the robotic application. A mesh algorithm will be more suitable for swarm robotics and ease the process of algorithm design as well as the implementation of the robotic application rather than server-client communication. With this, the scalability for the master robotic car to connect with more subsequent cars can be enhanced as well. This enhancement can provide higher commercial value to the robotic application and it can be applied to all mobile devices such as multiple drones' coordination in a plantation field. Moreover, the fixed location and distance between the master robotic car and the subsequent robotic car can be enhanced by designing them in a more flexible way without hardcoding their location and distance. The location and distance can be improved by replacing the ultrasonic sensor with tactile sensor or whisker sensor that can sense the obstacles from the side of the cars, so that the cars can line up in narrow path and expand in wider way effectively. Lastly, the speed of the robotic cars can also be designed in a flexible way in order to allow the cars to move faster when the distance between the master robotic car and the subsequent robotic car is larger as well as move slower when the subsequent car is approaching the master car.

Bibliography

A., B. & T., A., 2019. A Survey of Multi-mobile Robot Formation Control. *International Journal of Computer Applications*, 181(48), pp.12–16.

Anon, 2018. French state education: more robotics in schools? *Génération Robots - Blog*. Available at: https://www.generationrobots.com/blog/en/french-state-education-robotics-schools-2/ [Accessed March 25, 2019].

Aqeel, A., 2018. Introduction to Arduino IDE. *The Engineering Projects*. Available at: https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html [Accessed March 25, 2019].

Aqeel, A., 2019. Introduction to HC-SR04 (Ultrasonic Sensor). *The Engineering Projects*. Available at: https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html [Accessed March 25, 2019].

Arvin, F. et al., 2018. Mona: an Affordable Open-Source Mobile Robot for Education and Research. *Journal of Intelligent & Robotic Systems*, 94(3-4), pp.761–775.

Espressif Systems, n.d.. *ESP8266 Overview | Espressif Systems*. Available at: https://www.espressif.com/en/products/hardware/esp8266ex/overview [Accessed March 25, 2019].

Innorobo Community, 2016. *You are being redirected...*Available at: https://www.thedisruptory.com/2016/11/robots-in-education/ [Accessed March 27, 2019].

Jdeed, M. et al., 2017. Spiderino - A low-cost robot for swarm research and educational purposes. *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*.

JOJO, 2018. Light Dependent Resistors (LDR) - Working,Construction,Symbol,Applications. *Electronic Circuits and Diagrams-Electronic Projects and Design*. Available at: http://www.circuitstoday.com/ldr-light-dependent-resistors [Accessed August 8, 2019].

Kids, P.B.E.F., 2016. Why Is STEM Education So Important? *Engineering For Kids*. Available at: https://www.engineeringforkids.com/about/news/2016/february/why-is-stem-education-so-important-/ [Accessed March 23, 2019].

Mondada, F. et al., 2017. Bringing Robotics to Formal Education: The Thymio Open-Source Hardware Robot. *IEEE Robotics & Automation Magazine*, 24(1), pp.77–85.

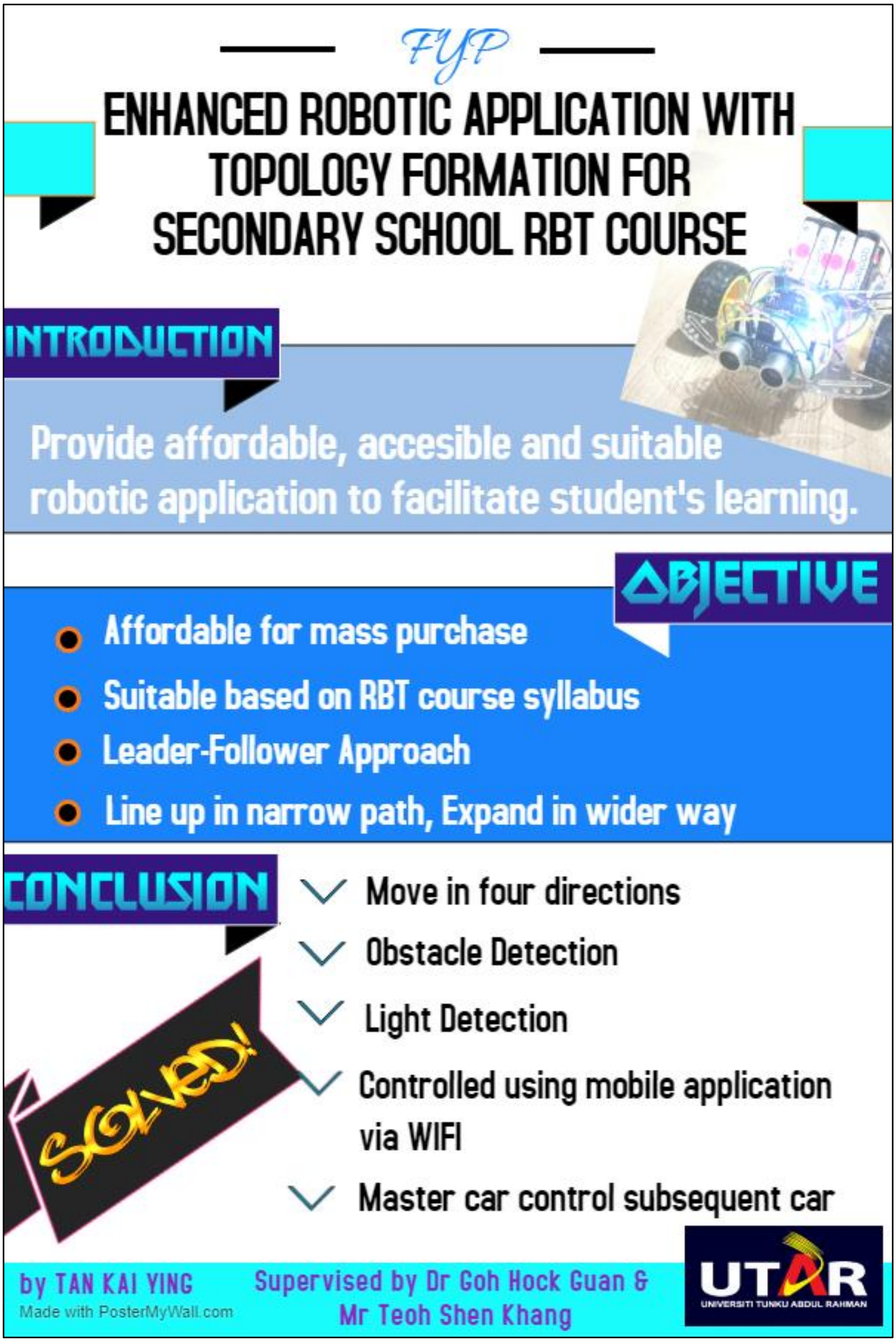
Pelayo, R., 2018. How to Use L298N Motor Driver. *Teach Me Microcontrollers!* Available at: https://www.teachmemicro.com/use-l298n-motor-driver/ [Accessed March 25, 2019].

Rezeck, P.A.F., Azpurua, H. & Chaimowicz, L., 2017. HeRo: An open platform for robotics research and education. *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*.

Rouse, M., 2017. What is swarm robotics? - Definition from WhatIs.com. *SearchEnterpriseAI*. Available at: https://searchenterpriseai.techtarget.com/definition/swarm-robotics [Accessed April 1, 2020].

Vitanza, A. et al., 2019. Robot swarms as an educational tool: The Thymio's way. *International Journal of Advanced Robotic Systems*, 16(1), p.172988141882518.

Appendix A: Poster



FYP

ENHANCED ROBOTIC APPLICATION WITH TOPOLOGY FORMATION FOR SECONDARY SCHOOL RBT COURSE

INTRODUCTION

Provide affordable, accesible and suitable robotic application to facilitate student's learning.

OBJECTIVE

- ● Affordable for mass purchase
- ● Suitable based on RBT course syllabus
- ● Leader-Follower Approach
- ● Line up in narrow path, Expand in wider way

CONCLUSION

SOLVED!

- ✓ Move in four directions
- ✓ Obstacle Detection
- ✓ Light Detection
- ✓ Controlled using mobile application via WIFI
- ✓ Master car control subsequent car

by TAN KAI YING

Made with PosterMyWall.com

Supervised by Dr Goh Hock Guan & Mr Teoh Shen Khang

UTAR UNIVERSITI TUNKU ABDUL RAHMAN

Appendix B: Final Year Project Biweekly Report (Week 1)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 1**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Discussion with supervisor regarding the title refinement for Final Year Project 2.

2. WORK TO BE DONE

Carry out research based on multiple robots communication or coordination.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Student's signature

Appendix C: Final Year Project Biweekly Report (Week 3)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 3**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Purchasing of components needed for additional robotic application.

2. WORK TO BE DONE

Complete the components assembly and Start the configuration for multiple robots communication.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

-

Kying

Supervisor's signature

Student's signature

Appendix D: Final Year Project Biweekly Report (Week 5)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 5**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Additional robotic car was built successfully.

2. WORK TO BE DONE

Self-study mesh algorithm and try for mesh algorithm configuration.

3. PROBLEMS ENCOUNTERED

L298N required 12V to function, instead of 5V.

4. SELF EVALUATION OF THE PROGRESS

Follow exactly the previous components assembly. Only make changes after the car works from previous components assembly.

Supervisor's signature

Student's signature

Appendix E: Final Year Project Biweekly Report (Week 7)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 7**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Self-study another algorithm instead of mesh algorithm.

2. WORK TO BE DONE

Configuration of multiple robot communication using Server-Client methods.

3. PROBLEMS ENCOUNTERED

Do not understand the mesh algorithm.

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Kying

Student's signature

Appendix F: Final Year Project Biweekly Report (Week 9)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 9**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Test the basic functionalities on newly built robotic car.

2. WORK TO BE DONE

Continue self-studying and configuring Server-Client communication.

3. PROBLEMS ENCOUNTERED

L298N motor driver ruined, have to buy a new L298N motor driver.

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Student's signature

Appendix G: Final Year Project Biweekly Report (Week 11)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 11**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Configuration of Server-Client communication using WiFiServer and WiFiClient algorithms.

2. WORK TO BE DONE

Configuration of server-client communication for multiple clients.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Student's signature

Appendix H: Final Year Project Biweekly Report (Week 13)

Trimester, Year: Semester 3, Year 3	**Study week no.: Week 13**
Student Name & ID: Tan Kai Ying 16ACB05033	
Supervisor: Dr Goh Hock Guan	
Project Title: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course	

1. WORK DONE

Configuration of server-client communication for multiple clients.

2. WORK TO BE DONE

Further improve the functionalities and demonstrate the final product to supervisor.

3. PROBLEMS ENCOUNTERED

One of the ESP8266 ruined and cannot connect to the server.

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Student's signature

Appendix I: Source Codes for Master robotic car

```
/* include libraries */
#include <ESP8266WiFi.h>    // WiFi library
#include <NewPing.h>        // Ultrasonic sensor library.

/* define port */
WiFiServer server(80);
WiFiServer server1(23);

/* ESP8266 as WiFi Soft Access Point settings */
const char* ssid = "ESP8266";     // network name
const char* password = "12345678";  // network password

/* Commands received from application */
String data = "";

/* Variables to send data to clients */
int ledValue = 0;
String command = "";

/* L298N motor control pins --> sink method */
const int RightMotorForward = 5;  // GPIO(D1) -> IN1
const int RightMotorBackward = 4; // GPIO(D2) -> IN2
const int LeftMotorBackward = 0;  // GPIO(D3) -> IN3
const int LeftMotorForward = 2;   // GPIO(D4) -> IN4

/* HC-SR04 ultrasonic sensor pins */
const int trig_pin = 12; // analog input 1
const int echo_pin = 14; // analog input 2

/* Buzzer pin */
const int buzzer = 13;

/* LED pin */
const int LED = 15;   // GPIO(D8)

#define maximum_distance 200
int distance = 100;
unsigned int sensorValue = 0;   // variable to store the value
coming from the ldr

NewPing sonar(trig_pin, echo_pin, maximum_distance); //
Ultrasonic sensor function

void setup(){
  /* initialize motor control pins as output */
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  /* initialize buzzer and LED pins as output */
  pinMode(LED, OUTPUT);
  pinMode(buzzer, OUTPUT);
  /* Start Serial port */
  Serial.begin(9600);   // start serial for output testing

  delay(2000);
  Serial.println("Configuring WiFi access point...");
  WiFi.mode(WIFI_AP);           //Configure as Access point
  WiFi.softAP(ssid, password);  //Start HOTspot removing
password will disable security
  IPAddress myIP = WiFi.softAPIP(); //Get IP address
  Serial.print("Host IP Address: ");
  Serial.println(myIP);

  /* start server communication */
  server.begin();
  server1.begin();
}

void loop()
{
  /* Turn on LED when in the dark and vice versa */
  sensorValue = analogRead(A0); // read the value from the ldr
(pin A0)
  delay(200);
  if(sensorValue <= 50){
    digitalWrite(LED, HIGH);
    ledValue = 1;
  }
  else {
    digitalWrite(LED, LOW);
    ledValue = 0;
  }

  /* Turn on buzzer and stop when getting closer to obstacle
*/
  if (sonar.ping_cm() <= 20){
   digitalWrite(buzzer, HIGH);
   moveStop();  // Stop and wait for the next commands
   command = "stop";
  }

  else {
    digitalWrite(buzzer, LOW);
  }

    /**** Master car sends data to subsequent cars *****/

    // Use WiFiClient class to create TCP connections
    WiFiClient client1 = server1.available();
    if (client1) {
    if (client1.connected()) {
      Serial.println(".");
      // sends the commands to the client
      client1.println(ledValue);
      client1.println(command);
    }
  }

    /**** Master car receives command from application ****/

    WiFiClient client = server.available();
    if (!client) { return; }
    while(!client.available()){  delay(1); }
    data = (client.readStringUntil('\r'));
    data.remove(0, 5);
    data.remove(data.length()-9,9);

/** Run function according to incoming data from application
**/

/* If the incoming data is "forward", run the "moveForward"
function */
   if (data == "forward"){
      moveForward();
      command = "forward";
    }

/* If the incoming data is "backward", run the "moveBackward"
function */
   else if (data == "backward"){
      moveBackward();
      command = "backward";
    }

/* If the incoming data is "left", run the "turnLeft" function
*/
   else if (data == "left"){
      turnLeft();
      command = "left";
    }

// If the incoming data is "right", run the "turnRight"
function
   else if (data == "right"){
      turnRight();
      command = "right";
    }

/* If the incoming data is "stop", run the "moveStop" function
*/
   else{
      moveStop();
      command = "stop";
    }
}
/**** STOP *****/
void moveStop(){

  analogWrite(RightMotorForward, 0);
  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);
  analogWrite(LeftMotorBackward, 0);
```

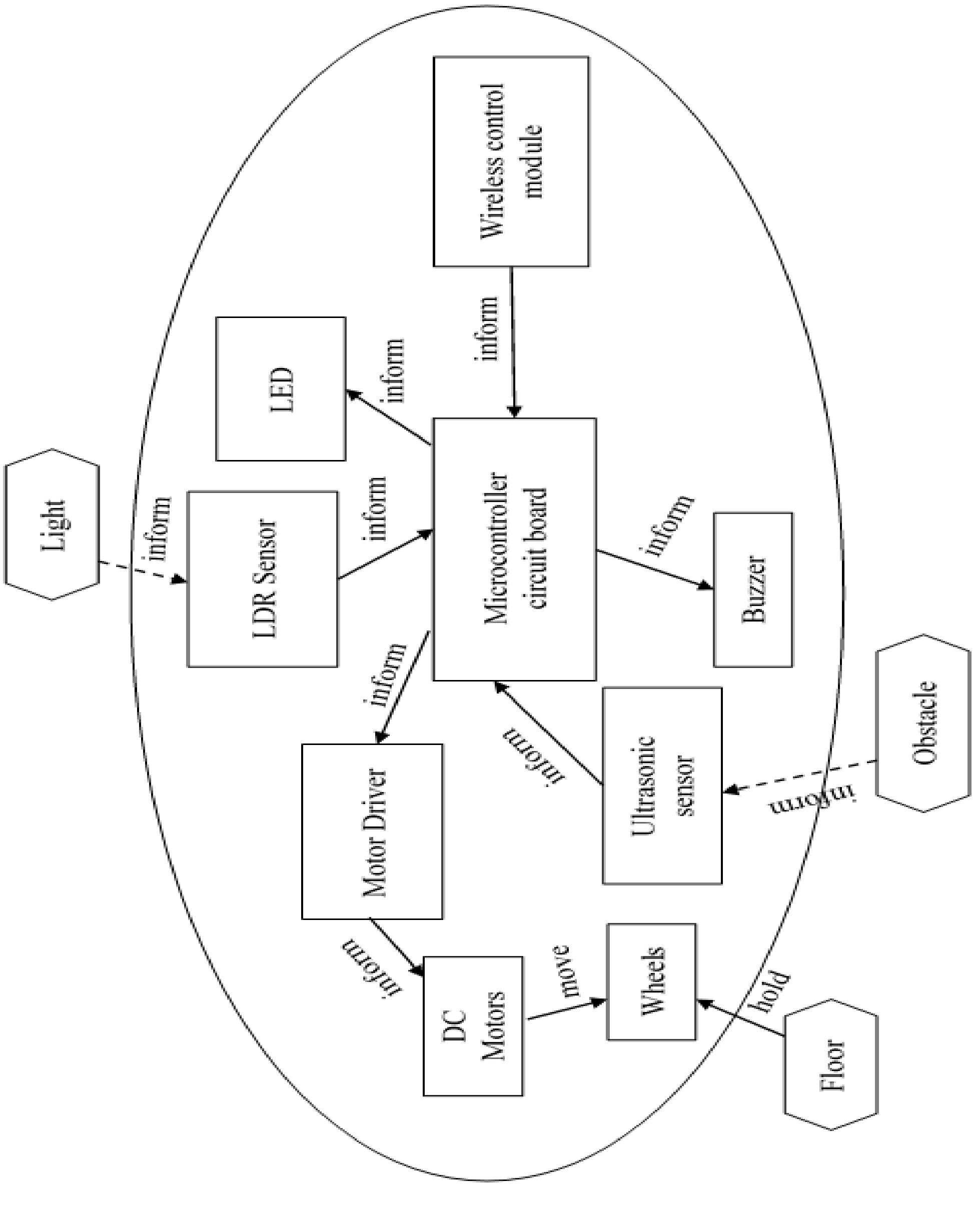
```
}

/**** FORWARD *****/
void moveForward(){

    analogWrite(LeftMotorForward, 0);
    analogWrite(RightMotorForward, 0);

    analogWrite(LeftMotorBackward, 256);
    analogWrite(RightMotorBackward, 256);
}

/**** BACKWARD *****/
void moveBackward(){

  analogWrite(LeftMotorBackward, 0);
  analogWrite(RightMotorBackward, 0);

  analogWrite(LeftMotorForward, 256);
  analogWrite(RightMotorForward, 256);
}

/**** TURN RIGHT *****/
void turnRight(){

  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);

  analogWrite(LeftMotorBackward, 256);
  analogWrite(RightMotorForward, 256);
}

/**** TURN LEFT ****/
void turnLeft(){

  analogWrite(LeftMotorBackward, 0);
  analogWrite(RightMotorForward, 0);

  analogWrite(LeftMotorForward, 256);
  analogWrite(RightMotorBackward, 256);
}
```

Appendix J: Source Codes for Slave robotic car

```
/* include libraries */
#include <ESP8266WiFi.h>   // WiFi library
#include <NewPing.h>        // Ultrasonic sensor library.

/* Connect to Server/Master */
const char* ssid = "ESP8266";     // network name
const char* password = "12345678";  // network password

/* Configure pins */
/* L298N motor control pins --> sink method */
const int RightMotorForward = 5;  // GPIO(D1) -> IN1
const int RightMotorBackward = 4; // GPIO(D2) -> IN2
const int LeftMotorBackward = 0;  // GPIO(D3) -> IN3
const int LeftMotorForward = 2;   // GPIO(D4) -> IN4

/* HC-SR04 ultrasonic sensor pins */
const int trig_pin = 12; // analog input 1
const int echo_pin = 14; // analog input 2

/* Buzzer pin */
const int buzzer = 13;

/* LED pin */
const int LED = 15;   // GPIO(D8)

#define maximum_distance 200
int distance = 100;

// Ultrasonic sensor function
NewPing sonar(trig_pin, echo_pin, maximum_distance);

IPAddress server(192,168,4,1);// the fix IP address of the
server
WiFiClient client;

void setup(void)
{
  /* initialize motor control pins as output */
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  /* initialize buzzer and LED pins as output */
  pinMode(LED, OUTPUT);
  pinMode(buzzer, OUTPUT);

  Serial.begin(9600);
  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
     delay(500);
     Serial.print("*");
  }
  Serial.println("");
  Serial.println("WiFi connection Successful");
}

void loop()
{
  /* Turn on buzzer and move into a line when getting closer
to obstacle */
   if (sonar.ping_cm() <= 20){
   digitalWrite(buzzer, HIGH);
   moveStop();
   delay(500);
   turnRight();
   delay(800);
   moveForward();
   delay(800);
   turnLeft();
   delay(800);
   moveStop();
   delay(500);
   moveForward();
  }

  else {
    digitalWrite(buzzer, LOW);
  }

  /**** Receives data from master car ****/
  client.connect(server, 23);   // Connection to the server
  Serial.println(".");

  // receives the data from the server
  String ledValue = client.readStringUntil('\r');
  if (ledValue == "1")
    digitalWrite(LED, HIGH);
  else digitalWrite(LED, LOW);

  String command = client.readStringUntil('\r');
  if (command == "\nforward")
    moveForward();
  else if (command == "\nbackward")
    moveBackward();
  else if (command == "\nleft")
    turnLeft();
  else if (command == "\nright")
    turnRight();
  else
    moveStop();
}

/**** STOP ******/
void moveStop(){
  analogWrite(RightMotorForward, 0);
  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);
  analogWrite(LeftMotorBackward, 0);
}

/**** FORWARD *****/
void moveForward(){

    analogWrite(LeftMotorForward, 0);
    analogWrite(RightMotorForward, 0);
    analogWrite(LeftMotorBackward, 256);
    analogWrite(RightMotorBackward, 256);
}

/**** BACKWARD ****/
void moveBackward(){

  analogWrite(LeftMotorBackward, 0);
  analogWrite(RightMotorBackward, 0);
  analogWrite(LeftMotorForward, 256);
  analogWrite(RightMotorForward, 256);
}

/**** TURN RIGHT ****/
void turnRight(){

  analogWrite(LeftMotorForward, 0);
  analogWrite(RightMotorBackward, 0);
  analogWrite(LeftMotorBackward, 256);
  analogWrite(RightMotorForward, 256);
}

/**** TURN LEFT *****/
void turnLeft(){

  analogWrite(LeftMotorBackward, 0);
  analogWrite(RightMotorForward, 0);
  analogWrite(LeftMotorForward, 256);
  analogWrite(RightMotorBackward, 256);
}
```

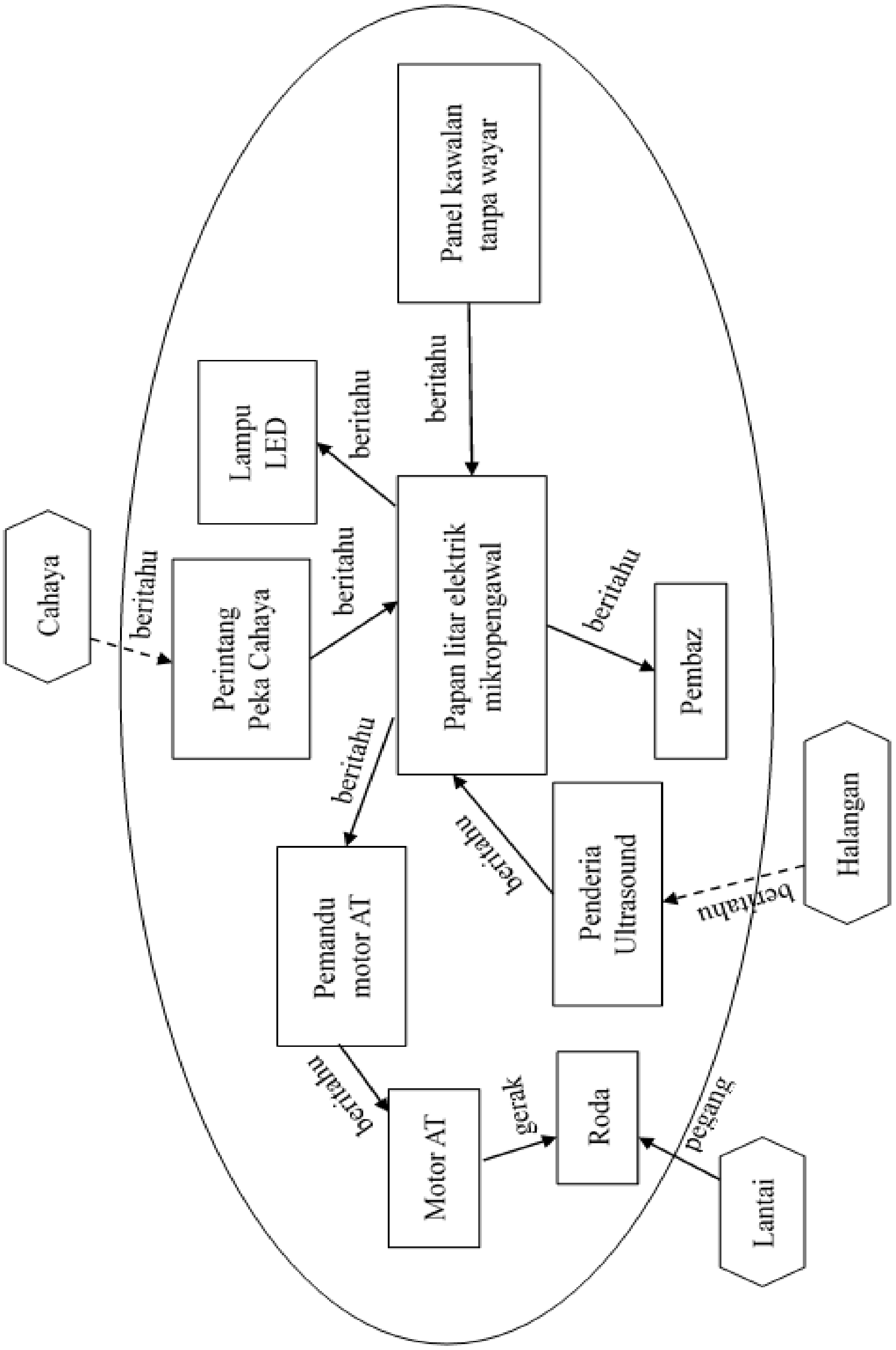
Appendix K: Correction of the TRIZ functional model for the robotic application

From the RBT textbook, the issue found in the functional model is that the model did not show clearly about the relationship between each components and their functionalities for the robotic car when operating, instead, it shows the operation among the robotic car's components during static. According to the documentation in RBT course textbook, the language used is in Malay. Thus, the TRIZ functional model developed as a correction is in Malay for textbook documentation.



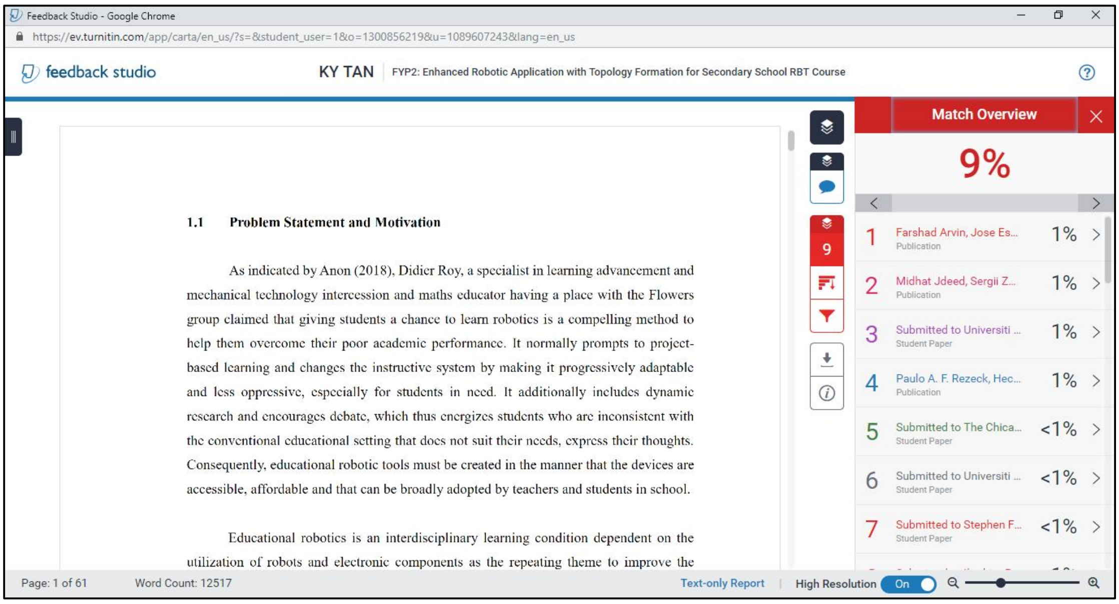
Overview of the links between components and functions of remote controlled car (English version)

Appendix L: Correction of the TRIZ functional model for the robotic application in Malay Version



Overview of the links between components and functions of remote controlled car (Malay version)

Plagiarism Check Result



Feedback Studio - Google Chrome
https://ev.turnitin.com/app/carta/en_us/?s=&student_user=1&o=1300856219&u=1089607243&lang=en_us
feedback studio

KY TAN FYP2: Enhanced Robotic Application with Topology Formation for Secondary School RBT Course

1.1 **Problem Statement and Motivation**

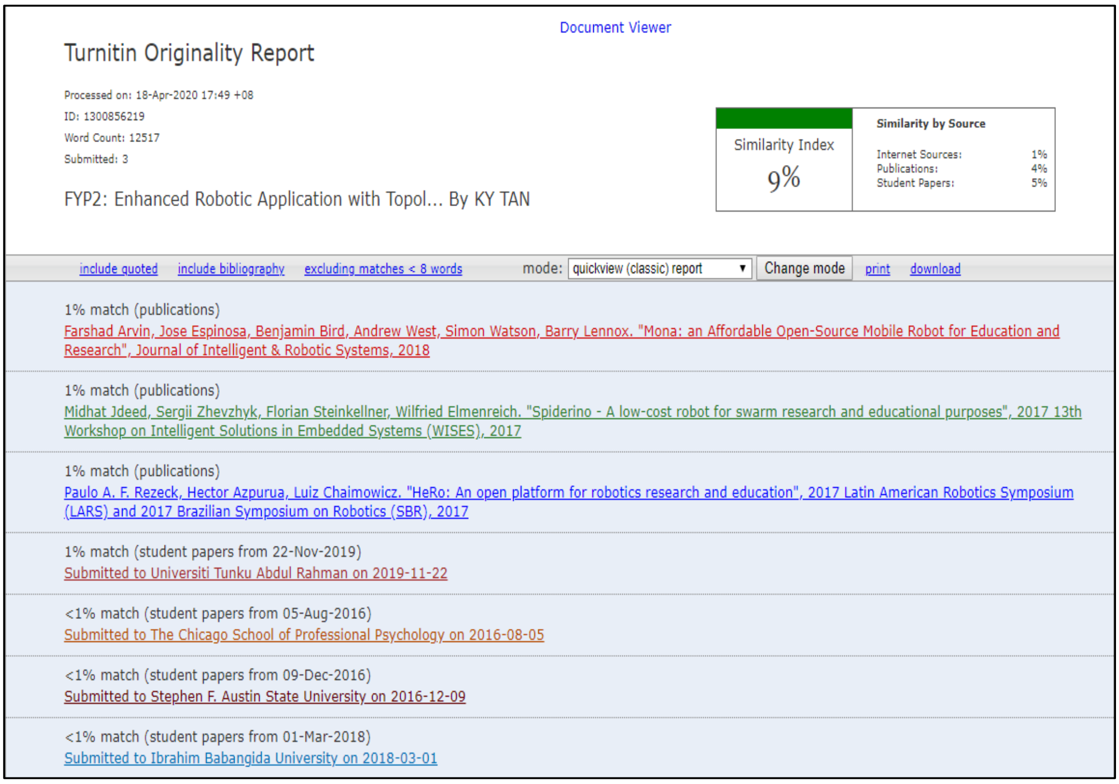
As indicated by Anon (2018), Didier Roy, a specialist in learning advancement and mechanical technology intercession and maths educator having a place with the Flowers group claimed that giving students a chance to learn robotics is a compelling method to help them overcome their poor academic performance. It normally prompts to project-based learning and changes the instructive system by making it progressively adaptable and less oppressive, especially for students in need. It additionally includes dynamic research and encourages debate, which thus energizes students who are inconsistent with the conventional educational setting that does not suit their needs, express their thoughts. Consequently, educational robotic tools must be created in the manner that the devices are accessible, affordable and that can be broadly adopted by teachers and students in school.

Educational robotics is an interdisciplinary learning condition dependent on the utilization of robots and electronic components as the repeating theme to improve the

Match Overview

9%

1	Farshad Arvin, Jose Es... Publication	1%
2	Midhat Jdeed, Sergii Z... Publication	1%
3	Submitted to Universiti ... Student Paper	1%
4	Paulo A. F. Rezeck, Hec... Publication	1%
5	Submitted to The Chica... Student Paper	<1%
6	Submitted to Universiti ... Student Paper	<1%
7	Submitted to Stephen F... Student Paper	<1%



Turnitin Originality Report

Processed on: 18-Apr-2020 17:49 +08
ID: 1300856219
Word Count: 12517
Submitted: 3

FYP2: Enhanced Robotic Application with Topol... By KY TAN

Similarity Index

9%

Similarity by Source

Internet Sources:	1%
Publications:	4%
Student Papers:	5%

1% match (publications)
Farshad Arvin, Jose Espinosa, Benjamin Bird, Andrew West, Simon Watson, Barry Lennox. "Mona: an Affordable Open-Source Mobile Robot for Education and Research", Journal of Intelligent & Robotic Systems, 2018

1% match (publications)
Midhat Jdeed, Sergii Zhevzhyk, Florian Steinkellner, Wilfried Elmenreich. "Spiderino - A low-cost robot for swarm research and educational purposes", 2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), 2017

1% match (publications)
Paulo A. F. Rezeck, Hector Azpurua, Luiz Chaimowicz. "HeRo: An open platform for robotics research and education", 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 2017

1% match (student papers from 22-Nov-2019)
Submitted to Universiti Tunku Abdul Rahman on 2019-11-22

<1% match (student papers from 05-Aug-2016)
Submitted to The Chicago School of Professional Psychology on 2016-08-05

<1% match (student papers from 09-Dec-2016)
Submitted to Stephen F. Austin State University on 2016-12-09

<1% match (student papers from 01-Mar-2018)
Submitted to Ibrahim Babangida University on 2018-03-01

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	TAN KAI YING
ID Number(s)	16ACB05033
Programme / Course	Bachelor of Information Technology (HONS) Computer Engineering
Title of Final Year Project	Enhanced Robotic Application with Topology Formation for Secondary School RBT Course

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 9 % **Similarity by source** Internet Sources: 1 % Publications: 4 % Student Papers: 5 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) **Overall similarity index is 20% and below, and**

(ii) **Matching of individual sources listed must be less than 3% each, and**

(iii) **Matching texts in continuous block must not exceed 8 words**

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Dr. GOH HOCK GUAN	Name: Mr. TEOH SHEN KHANG
Date: 18 April 2020	Date: 18 April 2020

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB05033
Student Name	TAN KAI YING
Supervisor Name	Dr. Goh Hock Guan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. Tan Kai Ying 18/04/2020	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. Dr. Goh Hock Guan, 18/04/2020