Simultaneous Localization and Mapping For Mobile Robot Car

By

Chea Kim Cuan

A PROPOSAL

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

Department of Computer and Communication Technology
(Perak Campus)

January 2018

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: _____________

I ____

(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:

____

_________________________ _________________________

____

Supervisor's name

Date: ____________________ **Date**: ____________________

Simultaneous Localization and Mapping For Mobile Robot Car

By

Chea Kim Cuan

A PROPOSAL

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

Department of Computer and Communication Technology
(Perak Campus)

January 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Simultaneous Localization and Mapping for Mobile Robot Car"** 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 : ____________________________

Date : ____________________________

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my supervisor Mr. Lee Wai Kong who gave me the golden opportunity to engage in Simultaneous Localization and Mapping for Mobile Robot Car project. He gave me a lot of advices and guidance that assisted me to solve the problems along this project. A million of thanks to you. Secondly, my family, their financial support, mentally support and love being through with me along the way to finish this project. I glad to did this project and thank for their unconditional support and love.

Abstract

This project is to develop a Simultaneous Localization And Mapping (SLAM) for mobile robot car controlled by using a “tiny computer” called Raspberry Pi as the main control board. The purpose of this project is to resolve the old line-following method for automatic guided vehicle or mobile robot car that utilized by factory for transportation purpose. Since old line-following method has been encountered some issues that caused the mobile robot car unsuccessful to deliver materials from starting point to destination. The major issue is line such as physical white line along predefined path on floor eliminating gradually due to problem of tear and wear. Thus, mobile robot car unable to detect the line on floor to travel. Therefore, by using new technique called Quick Response code (QR code) able to resolve the problem of line-following method. It is to replace the line along the path on floor with QR code. QR code is the trademark of a type of matrix barcode or two-dimensional barcode that able to store the data and fast readability. The mobile robot car compatible with the Raspberry Pi Camera can captures the image containing QR code and decodes it quickly in order to retrieve the data within it. The data stored within QR code is location point in coordinate form (x, y). Each QR code stored one location point. QR code symbols is placed on floor with same distance with each other. So that mobile robot car has to travel a certain distance to capture QR code and decode, then continue to move until it reach at destination. While decoding it, monitoring system on web also keep track of position by showing the location of mobile robot car. The monitoring system can let user clearly to determine position of mobile robot car either on track or stuck at any point along the pathway. Lastly, the QR code places on the floor not only reduces the problem of tear and wear as compared to line, but also the fast readability of QR code that made it became more reliable. Thus, QR code is a great solution to replace the old line-following method and suitable to implement it on this project.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

QR code	Quick Response code
SLAM	Simultaneous Localization And Mapping
GPS	Global Positioning System
IPS	Indoor Positioning System
DC	Direct Current
PHP	PHP:Hypertext Preprocessor
HTML	Hyper Text Markup Language
RFID	Radio Frequency Identification
IR	Infrared
TOA	Time Of Arrival
BLE	Bluetooth Low Energy
UART	Universal Asynchronous Receiver/Transmitter
SD Card	Secure Digital Card
VGA	Video Graphic Adapter
HDMI	Hight-Definition Multimedia Interface
USB	Universal Serial Bus
NOOBS	New Out Of Box Software
TV	Television
VNC	Virtual Network Computing
SSH	Secure Shell
HTTP	Hypertext Transfer Protocol

CAN	Controller Area Network
GPIO	General-Purpose Input/Output
CSI	Camera Serial Interface
SQL	Structured Query Language
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
IDLE	Integrated DeveLopment Environment
CPU	Central Processing Unit
GPU	Graphics Processing Unit
LAN port	Local-Area Network port
RAM	Random Access Memory
SoC	System on Chip
OLED screen	Organic Light-Emitting Diode screen
PWM	Pulse Width Modulation
bps	bits per second
FYP	Final Year Project

Chapter 1: Introduction

1-1 Problem Statement and Motivation

1-1-1 Problem Statement

Nowadays, most of manufacturing plant and warehouse industries are attempted to improve the operations and efficiency on their work in order to adapt the growing on their business. Mobile robot car is one of the approach to replace reliance on human workers and boost the efficiency. It is used to deliver the materials to some places. However, line-following method applied to mobile robot car for transportation purpose in warehouse has been encountered some issues and caused it not working properly.

Line-following method is one remarkable method which allowed automatic guided vehicle or mobile robot car to transport the materials from starting point to destination. It is detected and followed the line drawn or created guide tapes on floor until it to reach destination. By applying this approach, human power can be reduced in transportation by replacing the mobile robot car to deliver the materials. However, the created guide tapes or line drawn on floor has been encountered some problems over a period of time. One of the problem is created guide tapes or line drawn on floor that eliminated gradually over a period of time due to kinetic friction or tear and wear. Eventually, line drawn on floor likely become dotted line. Since dotted line that contained space in between, it caused mobile robot car difficulty to detect and follow it or even out of track. Mobile robot car also cannot to keep track of its position accurately. Furthermore, due to problem out of track, distance measurement from starting point to destination cannot to calculate precisely.

Therefore, QR code approach is one of the solution to replace the line-following. This is because QR code approach is to scan QR code symbols placed on floor with certain distance between each other to travel instead of following line. Thus, the problem of line-following approach can be resolved. For those developing and evolving factories or warehouses, them constantly adapting to meet ever-increasing productivity and efficiency demands. This is an enhancement solution to apply in transportation instead of line-following method.

1-1-2 Motivation

Most of the factory is faced the problem occurred during the pathway for mobile robot car transport the materials from or to the inventory. Since the old line-following method for mobile robot car to travel that is not functioned well over sometimes. For example, mobile robot car moved out from the track while travelling. The one of the reason is friction lead to the line gradually eliminating, made the mobile robot cannot detected the line to follow up. The consequences of it is resulting the financial loss in factory due to frequently to renew a line track. Thus, I would like to further improvement from the previous method which is mobile robot car using the QR code approach. QR code is to act as landmarks or location points in coordinate form, thus mobile robot car able to scan through and decode it in order to retrieve the data or location point. This also able to keep track the position of mobile robot car. QR code symbols are placed a certain distance with each other along the pathway, thus mobile robot car have to travel a distance only able to scan QR code and continue to move until it reach destination. Since it placed some QR codes instead of line all the way along pathway, thus it is only to replace QR code during maintenance. It had been solved the problems faced by factory such as reduced the financial loss in maintenance the track and getting lesser tear and wear problem emerged.

1-2 Background Information

In this modern era, the topic regarding “Robot” became a famous topic to those scientists which attracting their view to explore or investigate on it. So, what is a “Robot”? Robot is a machine that can be programmable by a computer and capable of carrying out a complex series of actions or tasks automatically after programmed. It can be taken over by either remote or internal control to perform some tasks. Thus, the mobile robot car is an example of robot to perform a task such as localization and navigation in indoor environment. Since the Global Positioning System (GPS) is suitable for outdoor environment, but its localization accuracy is degraded in the indoor environment. Thus, an Indoor Positioning System (IPS) is one of the solution for solving the problems regarded the mobile robot car. An Indoor Positioning System (IPS) is a system to locate objects in certain area using radio waves, magnetic fields and so on.

Therefore, a system is required to develop so that it can keep track of the mobile robot car by using the QR code technology known as Quick Response code which is the trademark for a type of matrix barcode or two-dimensional barcode. The advantages of the QR code are fast readability and larger storage capacity for data or information. However, it is required a scanner or camera to scan through QR code symbol in order to decode it and retrieve the data. Therefore, by using the QR code technology, the old line-following method applied in the factory can be replaced. In the factory, the guide tapes or lines on the floor formed a pathway for the mobile robot car to transport materials from source to destination. However, the guide tapes or lines on the floor is eliminated gradually due to physical contact such as shoes, loads and so on, which it caused to tear and wear problems to happen. Thus, mobile robot car no longer can be navigated based on the line to perform transportation from source to destination.

In order to improve the accuracy, landmarks or correction points represented by QR code is provided to keep track the mobile robot car. The mobile robot car is relied on the encoder that mounted on DC geared motor to calculate the distance travelled between two correction points. The encoder is continued output the pulses signal to omni-wheel main board while travelling. The pulses are obtained to calculate the distance travelled by using algorithm. When approaching the correction point, the Raspberry Pi Camera is captured an image to decode the QR code and retrieve the data. After that, adjusted the position of mobile robot car into an accurate pathway before continuing to move towards next correction point. This adjustment is required to reduce the errors accumulated during mobile robot car travelling. Then, mobile robot car is continued to move towards to next correction point and adjusted its position until it is arrived the destination.

In addition, a monitoring system on web is designed to aid in keep track the position of mobile robot car. Each QR code location is showed on map. Whenever mobile robot car is scanned and decoded the QR code symbols, the particular location on map is lighted up to indicate its current position. In contrast, monitoring system able to know if QR code symbols unsuccessfully scanned and decoded by mobile robot car at that particular location. Thus, monitoring system is useful to know the real-time situation and keep track the position of mobile robot car along the pathway.

1-3 Project Objectives

The objectives of this project can be summarized as below:

1. To develop technique for measuring travelled distance for mobile robot car through dead reckoning. The error of travelled distance is expected to be large when measured with this approach, especially when the robot travels for a long distance.

2. To improve the travelled distance measurement by introducing the correction points at fixed position (use QR code to represent the correction point) to correct the accumulated error in dead reckoning.

3. To develop the online web application for tracking the locations travelled by mobile robot.

4. To develop technique for reducing the errors accumulated along the pathway by introducing adjustment on robot position when it arrived at each correction point.

1-4 Proposed Approach /Study

There are a few proposed approach that used in this project. The QR code as a correction point methodology is one of the proposed approach. This approach is to place some QR codes along the pathway with the same distance in between. Mobile robot car is required to depend on the QR codes along the pathway to travel.

In addition, problem of friction between surface and wheel of mobile robot car lead to apply the position adjustment on each correction point. Whenever mobile robot car stops at correction point to decode the QR code, the situation will occurs which the position of mobile robot car shifted a bit. Therefore, this should be to adjust it before moving to next correction point and error accumulated along the pathway will be decreased gradually.

Besides, the distance travelled methodology is applied to this project. In order for mobile robot stops at each correction point, the distance between two correction points have to compute. So that mobile robot car able to stop by travel the fixed length of distance. This distance can be measured by using the encoder to detect the number of motor revolution.

Lastly, monitoring system on web page approach is required to keep track the location of the mobile robot car. There are some requirements to keep track the mobile robot car. The first requirement is to possess a database system to store the location of QR codes or correction point. When mobile robot car at the specific correction point, it able to insert its location into database. Therefore, web page able to retrieve the data from database and display to the user-friendly web page. This allowed the user to keep track the mobile robot car via monitoring system on web page.

1-5 Achievement

Below are the highlights of what have been achieved in this project:

- Able to control Omni-wheel main board (slave) by Raspberry Pi (master) via UART.
- Able to determine the direction of mobile robot car by controlling its motors. For instances, move to left, right, forward, backward and so on.
- Raspberry Pi Camera was used to capture QR code image successfully.
- Adjustment on each correction point to ensure the position of mobile robot car accurately. This was to reduce the errors accumulated along the pathway.
- Able to upload and store the decoded data from QR code to MySQL database.
- PHPMyAdmin software tool was used to manage and handle the MySQL database over web.
- Able to retrieve the stored data on MySQL database via using PHP.
- Able to display the stored data on MySQL database to webpage designed using HTML and CSS.
- Monitoring system on web page was created to keep track the position and real-time status of mobile robot car.

1-6 Report Organization

This report is organized with few chapters. Chapter 1 is an introduction to the project, basically to describe problem statement, motivation, objectives and project background. In chapter 2 is about the literature review to review and comparison on previous or related project. It is to highlight and compare proposed approach with previous one in order able to improve. Chapter 3 is described in detail how the project is developed and provided all necessary information, so that to allow someone able to rebuild the system. This included top-down system design diagram, details of system flow, implementation steps and so on. Chapter 4 is regarding the design specification, it is discussing about the methodology and tools being used in project and requirement for the system. In addition, analysis and verification plan are written on chapter 5. Lastly, chapter 6 is to conclude the project. It is described on what have been achieved in project, relate to objectives and future improvements or development.

Chapter 2: Literature Review

There are some existing product that similar to this project but the technologies involved are different. The first product called "RFID Positioning Robot" which uses the radio frequency identification (RFID) technology for localization and navigation. The robot is designed to mount three RFID antennas in an equilateral triangle position, and some sensor such as ultrasonic and infrared (IR) for obstacle detection and avoidance during moving. According to L.Chunag (2012), he using the method of Time Of Arrival (TOA) to determine the present location through data collected from the three antennas. By varying each antenna's transmission power level, it can determine an estimated distance from tag in its read range. Therefore, with data collected from three antennas, coarse localization can be performed to triangulate the location using the distance formula. The strength of this product is to use the TOA's method that reduce the time of signal to travel from source to destination because signal might travel on the direct or shortest path. Hence, it can handle the condition from multi-path environment. However, in order to perform the accurate time synchronization between the source and receiver, the micro or nano second for the time measurement is required, thus it will increase the hardware cost to implement for accurate localization. As compared to the using the QR code method to acts as the correction point to determine the position the mobile robot, QR code method will more accurate due to the QR code is placed on a fixed position, thus position is indicated when the mobile robot car scan through it. Besides, QR code is lower cost and easier to get from the internet.

In addition, the next article is Kalman Filter based Indoor Mobile Robot Navigation by Md Anam Mahmud, Md Sayedul Aman, Haowen Jiang, Ahmed Abdelgawad, and Kumar Yelamarthi (2016). Mobile robot is to navigate by using wheeled encoder as sensor. While using wheel encoder the robot is commanded to move certain paths like an octagon, rectangle and travelled distances are calculated. The wheel encoder provides Cartesian coordinate and orientation angle of the robot which can easy to track the robot. However, robot's encoder will accumulates the error along the navigation path. Hence, a Kalman filter is proposed to reduce the error to minimum. At this point, the system of robot also checks if destination is reached. If it reached the destination, the robot stop moving or turned off, otherwise it will continue to move until it reaches its destination. The strength of this robot is to gain more accuracy by minimize the error by using the probability estimation techniques called Kalman filter. The weakness of this robot is the error will accumulates along the travel path.

Furthermore, the next article is Positioning in an Indoor Environment Based on iBeacons by Lida Yang, Quanyu Wang and Guangjie Wang (2016). The iBeacon technology also can apply to robot for indoor positioning which released by Apple Inc. in 2013. iBeacon provides automatic transmission of location-based advertisement information to smartphones which build upon Bluetooth Low Energy (BLE). According to Fujihara, Akihiro, and Takuma Yanagizawa (2015) said, “iBeacon has characteristic of easy-deployed, low power, low cost and solves the problem that the GPS positioning cannot penetrate the thick shield which lead to the positioning deviation”. In order to locate the robot, install multiple iBeacons in the known environment, so that the robot can retrieve the distances between every iBeacon and the terminal through computing. The position of the robot can be obtained through algorithm to compute. The strength of this technology is lower power consumption, lower cost as compared to others technologies such as RFID and Wi-Fi. The weakness is out of the effective range of tens of meters in indoor environments, the accuracy of positioning will be degraded. However, QR code compared with iBeacons, QR code is cheaper and it has a fast readability which allowed robot to scan through it in a quick way, therefore it much more efficient than iBeacons.

Chapter 3 System Design

3-1 System Design Block Diagram

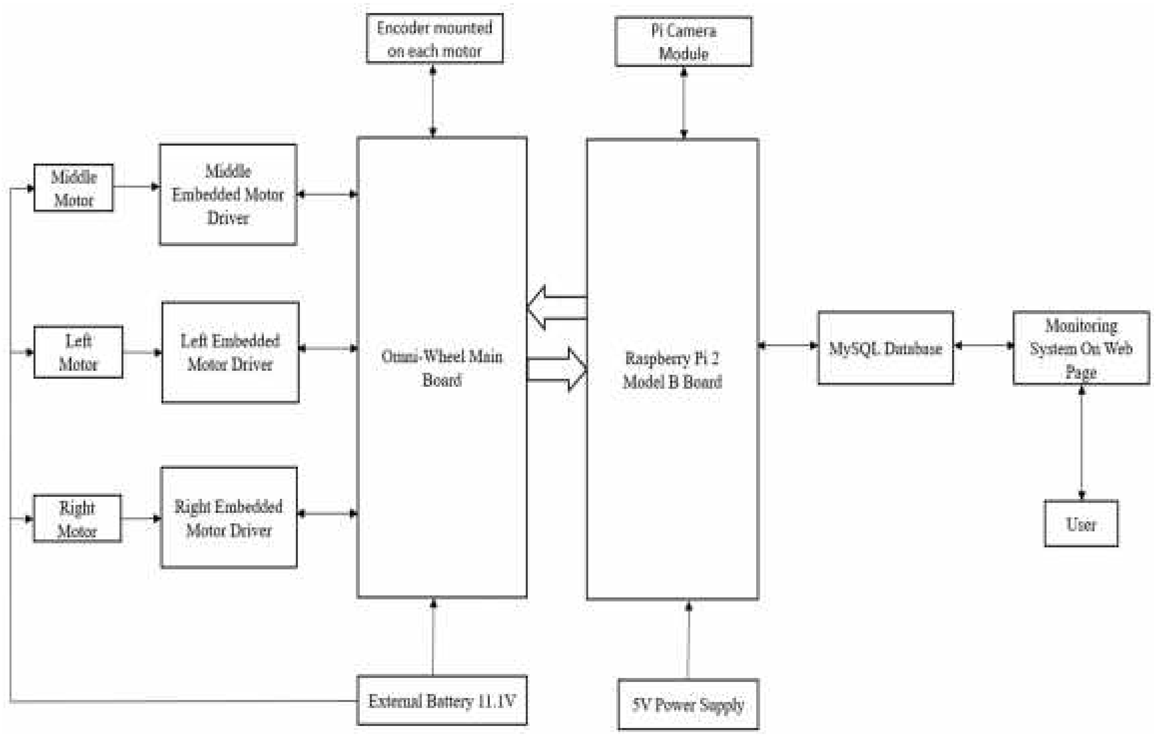


Figure 3-1-1 System Design Block Diagram

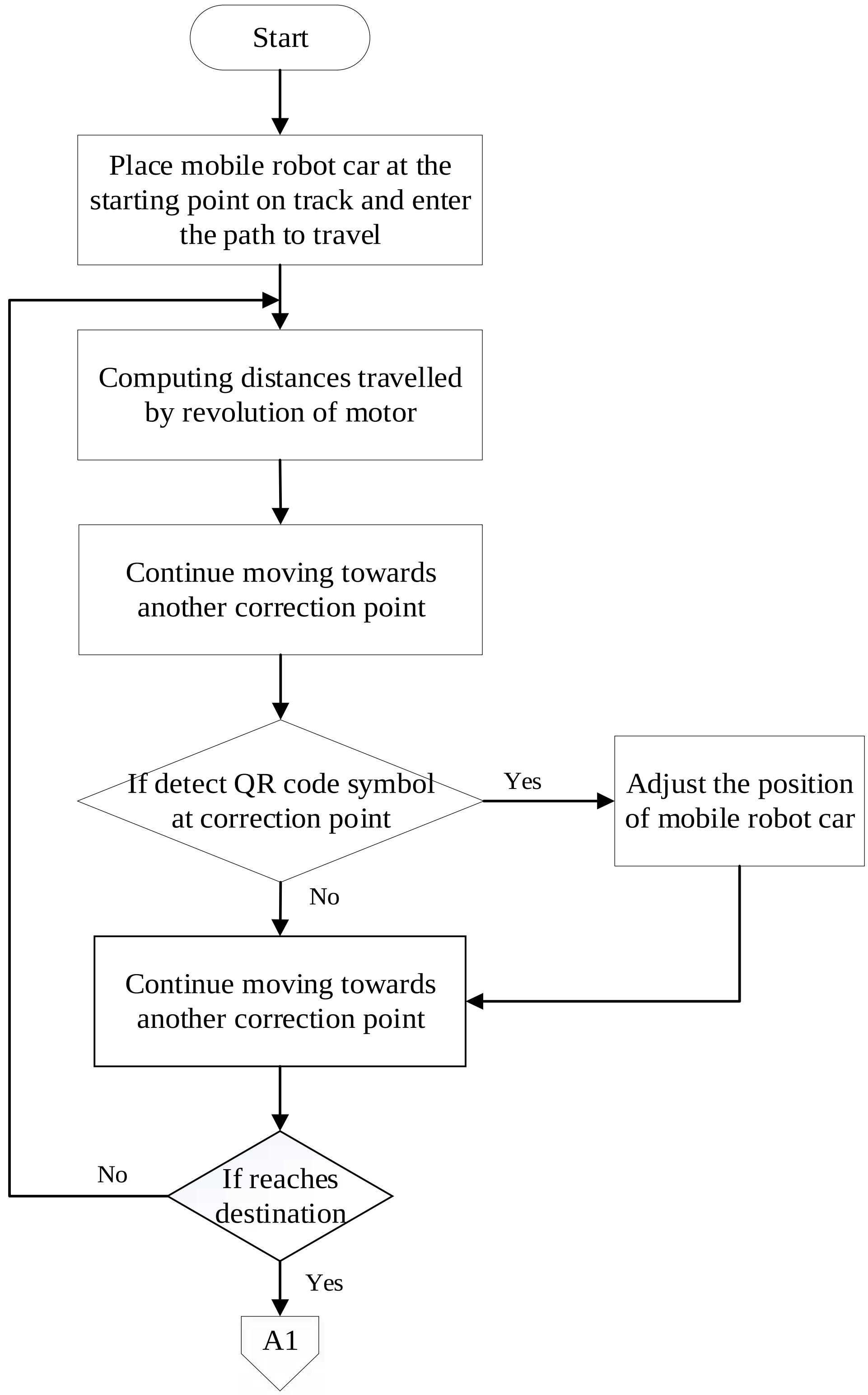
Based on above system design block diagram, there are separated into two parts, left side part is hardware part to control the physical movement of the mobile robot car, right side part is software part to monitor the location of mobile robot car on designed web page.

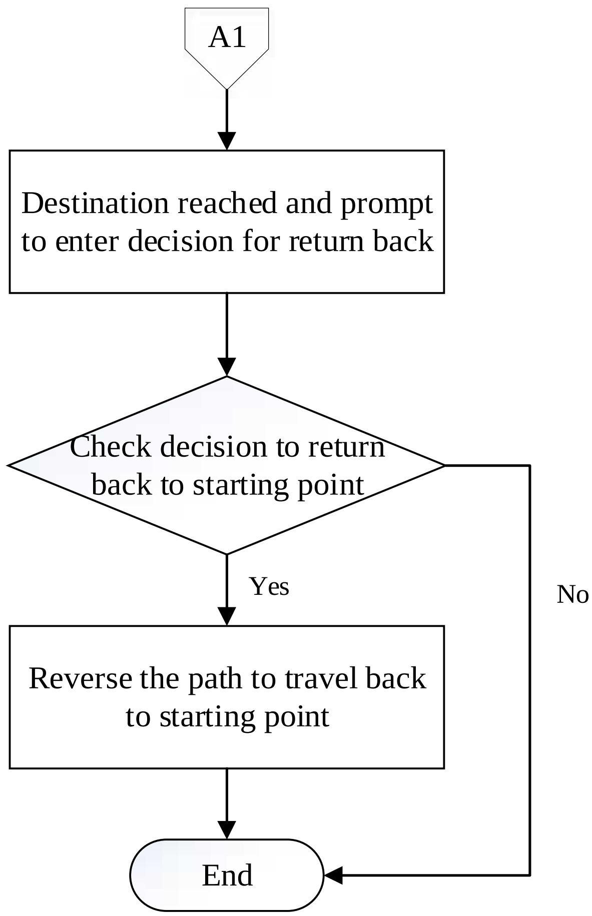
On the hardware part, Omni-wheel main board is acted as a slave for Raspberry Pi. External battery 11.1V as a power supply that is connected to the slave. The slave has three embedded motor driver modules to drive three motors respectively. The embedded motor driver module is to control the direction of movement and speed of motor. Thus, each motor is connected to one embedded motor driver module. Moreover, the encoder mounted on each motor also connected to slave. The encoder is to keep output the pulses to the slave while motor moving. Thus, slave able to further process the pulse signals such as to compute the revolution of the motor and distance travelled by the mobile robot car.

Besides, Raspberry Pi as a master is to control over the slave. It is connected to 5V power supply. In order to become master-slave relationship, a communication is required between both which is UART communication. The communication allowed the master to control slave and retrieve the required data from slave especially the revolution and speed of motor. In addition, master also connected to Pi Camera that used to capture QR code image and decode it to retrieve the data such as location of correction point in coordinate form. The captured image is then used to adjust the position of mobile robot car accurately.

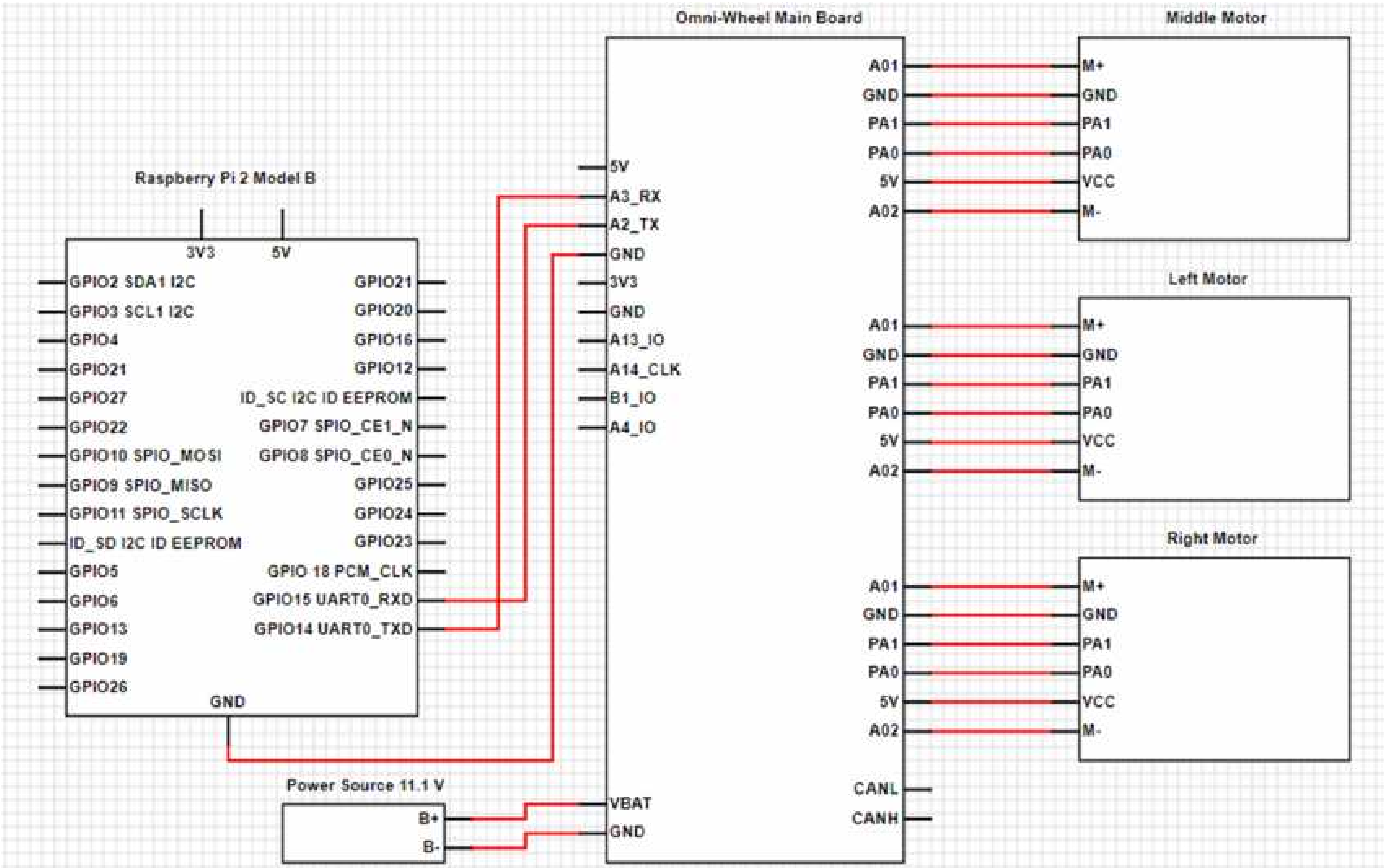
On the software part, Raspberry Pi also as a web server to host or serve the web pages. It also installed the database call MySQL database that used to store the data retrieved from both slave and master. Therefore, the database within the web server is able to link with web page. The data from database is retrieved and showed to the web page that hosting by web server. So that, user able to view data on web page such as location of mobile robot car.

3-2 Flow Chart




A1
Destination reached and prompt to enter decision for return back
Check decision to return back to starting point
Yes
No
Reverse the path to travel back to starting point
End

3-3 Schematic Diagram



3-4 Prototype

The figure below shows the prototype is developed and used it to complete in this project.

Top View of Prototype

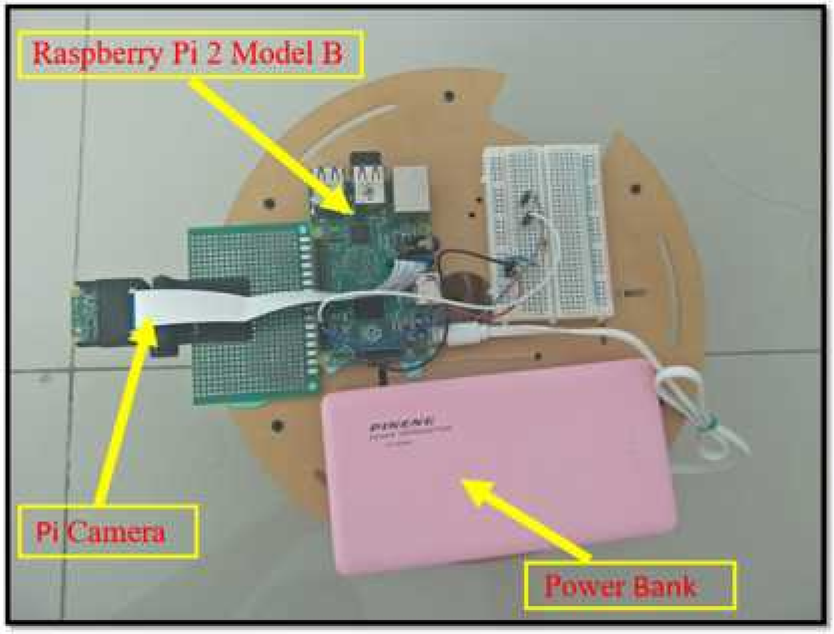


Figure 3-4-1 Top view of prototype

Side View of Prototype

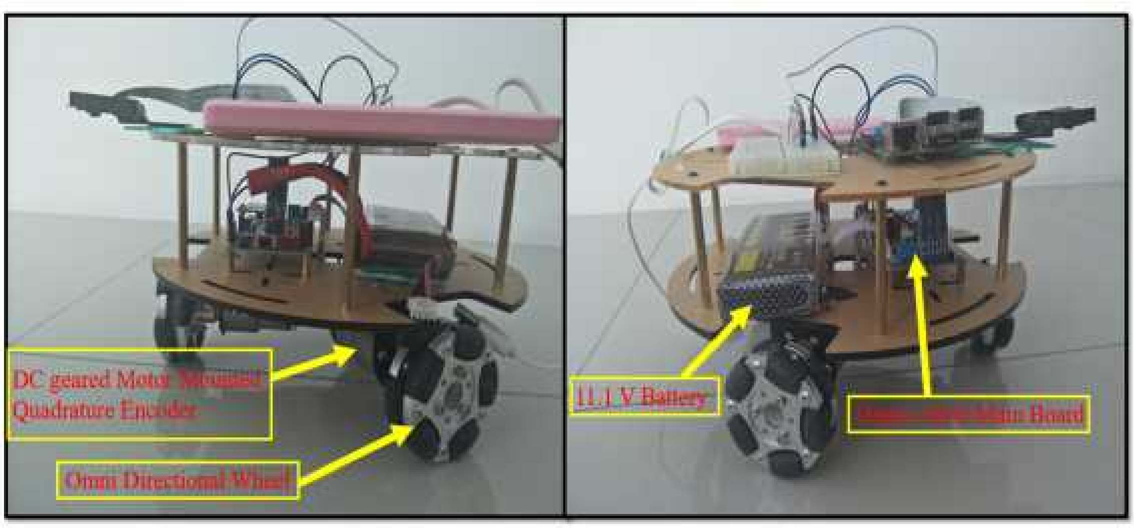


Figure 3-4-2 Side view of prototype

3-5 Hardware Development

3-5-1 Connection of Raspberry Pi 2 Model B, Omni-wheel main board, and DC geared motor mounted quadrature encoder

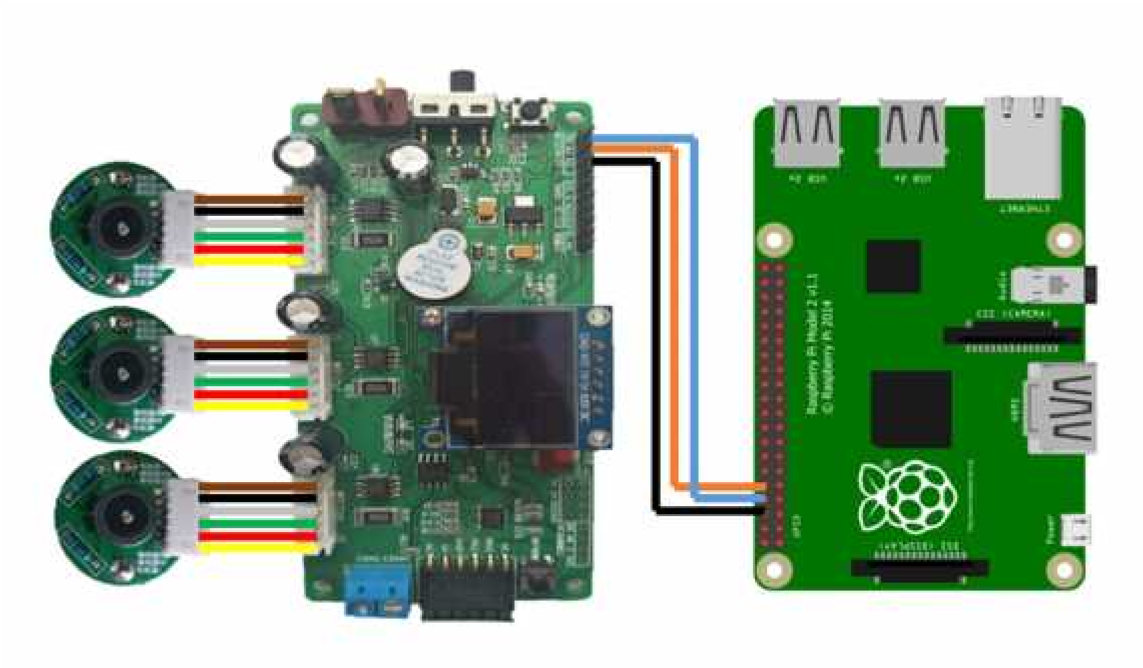


Figure 3-5-1 Connection of Raspberry Pi 2 Model B, Omni-wheel main board, and DC geared motor mounted quadrature encoder

It is embedded the motor driver to Omni-wheel main board and has a well connection, so it greatly reduced the difficulty of utilization. Thus, the user just able to use instruction to control three motors on mobile robot car. There are two communication used to control the mobile robot car which are Controller Area Network (CAN) and Serial Interface communication.

In this project, serial interface communication called Universal Asynchronous Receiver/Transmitter (UART) is adopted and used to control mobile robot car. Serial communication is the process of transmit the bit data sequentially over a communication channel. 8 bits of data is transferred in each time of transmission. Thus, Raspberry Pi is connected to the Omni-wheel main board via UART in order to control the mobile robot car by sending instructions. Table 3-5-1 shows the pins assigned to Omni-wheel main board.

Omni-wheel Main Board Pin (slave)	**Raspberry Pi 2 Model B GPIO Pin (master)**	**Function**
2 (RX)	8 (TXD0)	Transfer data from master to slave
3 (TX)	10 (RXD0)	Transfer data from slave to master
4 (GND)	6 (GND)	-

Table 3-5-1 Pins connection between Raspberry Pi and Omni-wheel main board

The serial communication between master and slave is required Baud rate 115200bps. Figure 3-5-2 shows the python code for serial communication.

```
import serial

ser = serial.Serial('/dev/ttyAMA0', baudrate=115200,
parity=serial.PARITY_NONE, stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS, timeout=10)
if(ser.isOpen() == False):
        ser.open()
```

Figure 3-5-2 Python code for serial communication

The library for serial communication in the Raspberry Pi is serial. In order to use the library, pyserial has to install by using following command.

- sudo pip intall pyserial

The function serial.Serial() is required to input the serial port, baud rate, byte size, timeout and store into variable ser, then open the serial port on Raspberry Pi. The function ser.isOpen() is used to check whether serial port is successfully open, if successful its function returns True, else returns False. If function ser.isOpen() returns False, then used function ser.Open() to reopen the serial port.

After done for the serial connection, slave is required 10 bytes send form master as a complete serial control instruction in order to control the mobile robot car. For serial control instruction, the location of motor and axis direction for mobile robot car must to know, it can be referred from figure 3-5-3.

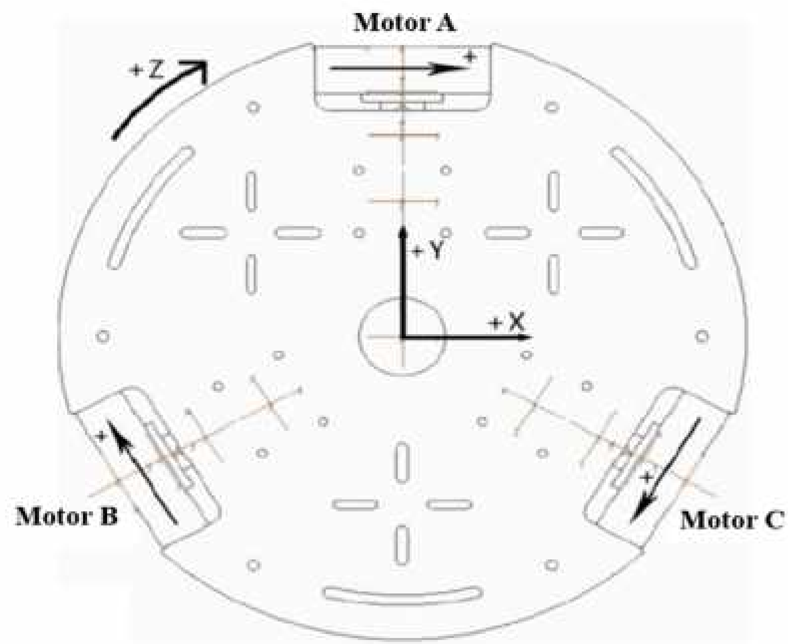


Figure 3-5-3 Location of motor and axis direction for mobile robot car

For the serial control instruction, there are two modes can be used to control the motor which are speed control mode and position control mode. However, speed control mode is only used in this project. Therefore, it must turn button to speed mode on Omni-wheel main board shows on figure 3-5-4.

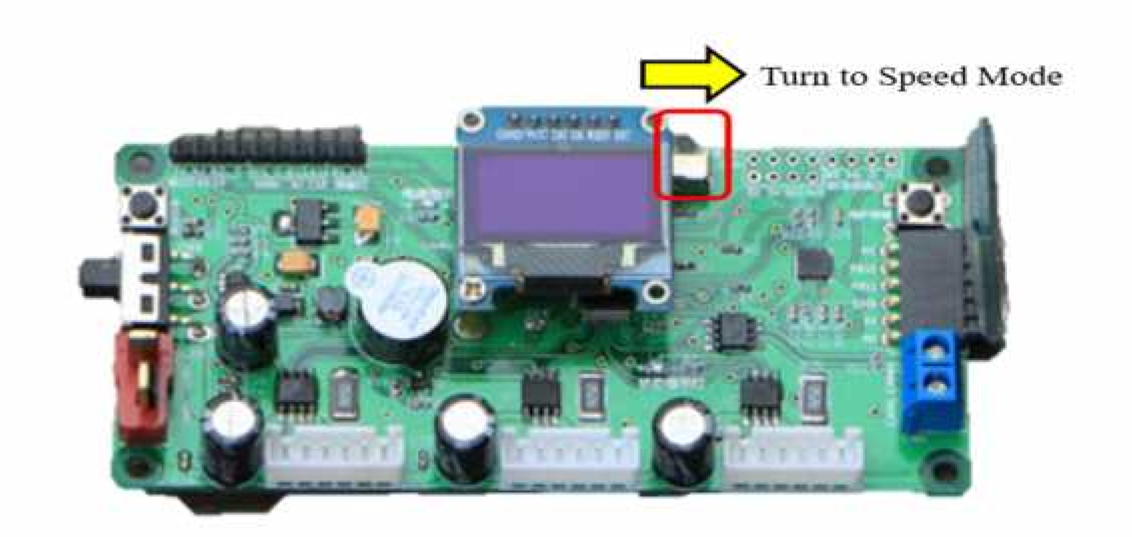


Figure 3-5-4 Speed Mode button on Omni-wheel main board

<u>Serial Control Instruction</u>

Data Fields	Frame Header	Frame Header	tx[0]	tx[1]	tx[2]	tx[3]	tx[4]	tx[5]	tx[6]	tx[7]
Details	0xff	0xfe	Mode	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	Direction control 8 bits

Referred to **Speed Control Mode Instruction**

Table 3-5-2 Serial Control Instruction

Speed Control Mode Instruction

When Model = 1, operate a coordinate control to mobile robot car, it requires to enter the amount of speed in X, Y and Z-axis. tx[0] an tx[1] combined to 16 unsigned number to control speed in X-axis. tx[2] an tx[3] combined to 16 unsigned number to control speed in X-axis. tx[4] an tx[5] combined to 16 unsigned number to control speed in X-axis. tx[7] is direction control, lowest 3 bits is used to control the direction for three axis.

Data Fields	**tx[0]**	**tx[1]**	**tx[2]**	**tx[3]**	**tx[4]**	**tx[5]**	**tx[6]**	**tx[7]**
Details	Mode	X-Axis speed control upper 8 bits	X-Axis speed control lower 8 bits	Y-Axis speed control upper 8 bits	Y-Axis speed control lower 8 bits	Z-Axis speed control upper 8 bits	Z-Axis speed control lower 8 bits	Direction control 8 bits

tx[7]	**0**	**0**	**0**	**0**	**0**	**0/1**	**0/1**	**0/1**
Details	Reserved	Reserved	Reserved	Reserved	Reserved	0 : Positive X-Axis movement 1 : Negative X-Axis movement	0 : Positive Y-Axis movement 1 : Negative Y-Axis movement	0 : Positive Z-Axis movement 1 : Negative Z-Axis movement

Table 3-5-3 Speed Control Mode Instruction for mode = 1

When Model = 2, operate a closed-loop control to each motor, it requires to enter the amount of speed in motor A, B and C. tx[0] an tx[1] combined to 16 unsigned number to control speed in motor A. tx[2] an tx[3] combined to 16 unsigned number to control speed in motor B. tx[4] an tx[5] combined to 16 unsigned number to control speed in motor C. tx[7] is direction control, lowest 3 bits is used to control the direction for three motors.

Data Fields	**tx[0]**	**tx[1]**	**tx[2]**	**tx[3]**	**tx[4]**	**tx[5]**	**tx[6]**	**tx[7]**
Details	Mode	Motor A speed control upper 8 bits	Motor A speed control lower 8 bits	Motor B speed control upper 8 bits	Motor B speed control lower 8 bits	Motor C speed control upper 8 bits	Motor C speed control lower 8 bits	Direction control 8 bits

tx[7]	**0**	**0**	**0**	**0**	**0**	**0/1**	**0/1**	**0/1**
Details	Reserved	Reserved	Reserved	Reserved	Reserved	0 : Motor A forward direction 1 : Motor A backward direction	0 : Motor B forward direction 1 : Motor B backward direction	0 : Motor C forward direction 1 : Motor C backward direction

Table 3-5-4 Speed Control Mode Instruction for mode = 2

Figure 3-5-6 shows the python code to send 10 bytes of data to control mobile robot car.

```python
import serial
import time

forward = bytearray([255,254,1,0,0,0,32,0,0,0])
left = bytearray([255,254,1,0,32,0,0,0,0,4])
right = bytearray([255,254,1,0,32,0,0,0,0,0])
backward = bytearray([255,254,1,0,0,0,32,0,0,2])
resetMotor = bytearray([255,254,1,0,0,0,0,0,0,0])

ser = serial.Serial('/dev/ttyAMA0', baudrate=115200,
parity=serial.PARITY_NONE, stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS, timeout=10)
if(ser.isOpen() == False):
        ser.open()

ser.write(forward)
time.sleep(2)

ser.close()
```

Figure 3-5-6 Python code for sending serial data

Since the serial port is opened, the serial control instruction able to send to slave to take over control for the mobile robot car. In python code above, bytearray is used to store 10 bytes data as the serial control instruction, then function ser.write() is to send 10 bytes for serial control instruction to slave. Below table 3-5-5 shows the serial control instruction to control the direction of mobile robot car.

No	**Serial Control Instruction**	**Result**
1	bytearray([255,254,1,0,0,0,32,0,0,0])	Move forward
2	bytearray([255,254,1,0,32,0,0,0,0,4])	Move left
3	bytearray([255,254,1,0,32,0,0,0,0,0])	Move right
4	bytearray([255,254,1,0,0,0,32,0,0,2])	Move backward
5	bytearray([255,254,1,0,0,0,0,0,0,0])	Stop

Table 3-5-5 Serial Control Instruction to control mobile robot car

3-5-2 Connection between Raspberry Pi 2 Model B and Pi Camera

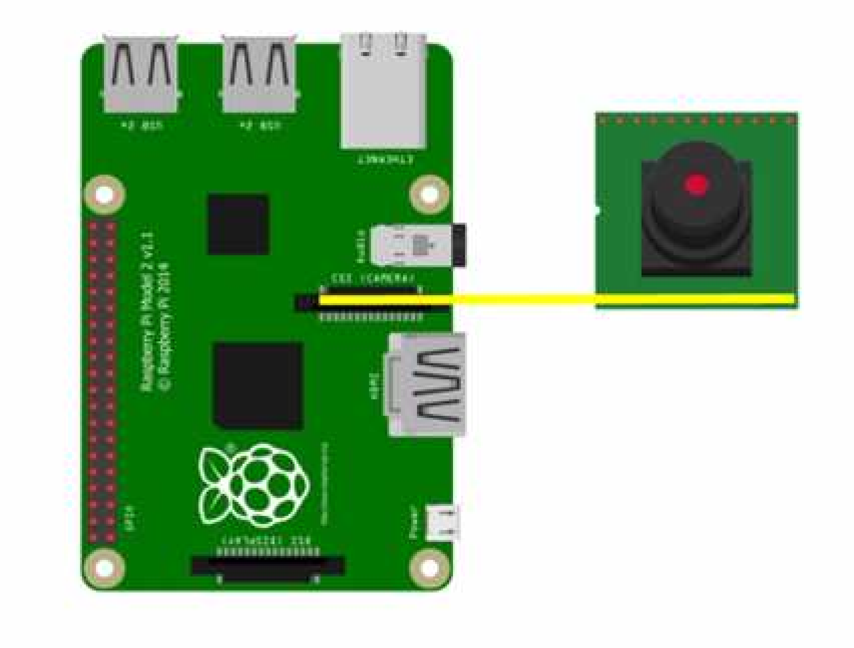


Figure 3-5-7 Connection of Raspberry Pi 2 Model B and Pi Camera

Pi Camera is connected to Raspberry Pi CSI port via flex cable. In this project, the Pi Camera is used to capture the QR code image and the decode it to retreive the data. The Pi Camera is fixed at a certain height on mobile robot car due to camera focusing issue. Thus, the camera able to capture the clear image. Figure 3-5-8 shows the python code to capture and decode the QR code image.

```
import time
import picamera
import qrtools
qr = qrtools.QR()

camera = picamera.PiCamera()
camera.start_preview()
time.sleep(2)
camera.capture('image.png')
camera.stop_preview()
qr.decode('image.png')
if qr.data:
    print qr.data
else:
    print "No QR data"
```

Figure 3-5-8 Python code for capture and decode QR code image

From above figure, library for picamera and qrtools are required to import into python. The below commands is to install both library.

- sudo apt-get install python-picamera
- sudo pip install qrtools

The camera is an instance of PiCamera class, then camera.capture() function is to take a QR code image. After that, qr.decode() function is used to decode the captured QR code image and obtain data from qr.data.

3-6 Software Development

3-6-1 MySQL Database

In this project, PHPMyAdmin is used to manage the MySQL database due to convenience and easier to manage. In PHPMyAdmin, a database is created called Map_database and a Map table is created within it. This table contained two fields that are id and location. Only the value of that two fields have to insert into table. Figure shows the database on PHPMyAdmin.

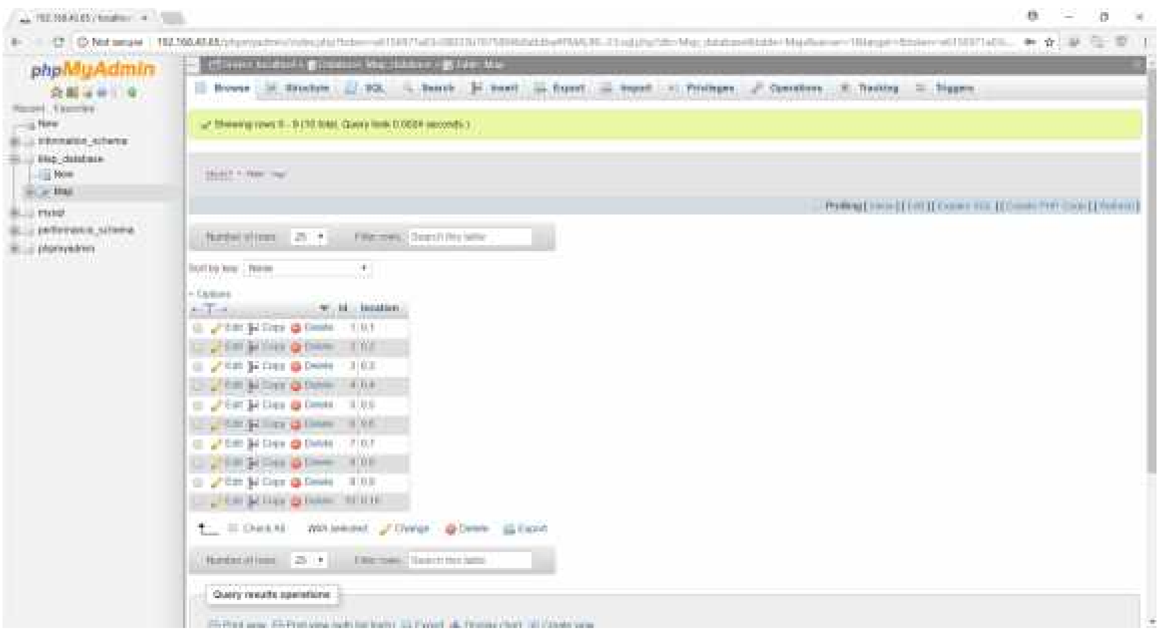


Figure 3-6-1 Interface of PHPMyAdmin

Before inserting the data into database using python code, mysqldb library has to install by following command.

- sudo apt-get install python-mysqldb

```python
import MySQLdb

db = MySQLdb.connect("localhost","root","cheakcsql123","Map_database")
cursor = db.cursor()
time.sleep(1)

def insert_data(db,cursor,id,location):
        sql=("""INSERT INTO Map(id,location) VALUES (%s,%s)""",(id,location))
        try:
                print "Writing to database..."
                # Execute the SQL command
                cursor.execute(*sql)
                db.commit()
                print "Write Complete"
        except:
                # Rollback in case there is any error
                db.rollback()
                print "Failed writing to database"
        return


db.close()
```

Figure 3-6-2 Python code for connection and execution on database

From the python code above, in order to insert data into database, first step is to connect to the database. The function of MySQLdb.connect() is required input with local host, database name, username and password to connect database. For the insert_data function is to pass the data and execute the Structured Query Language (SQL) command to insert data into database.

3-6-2 Monitoring system on web page

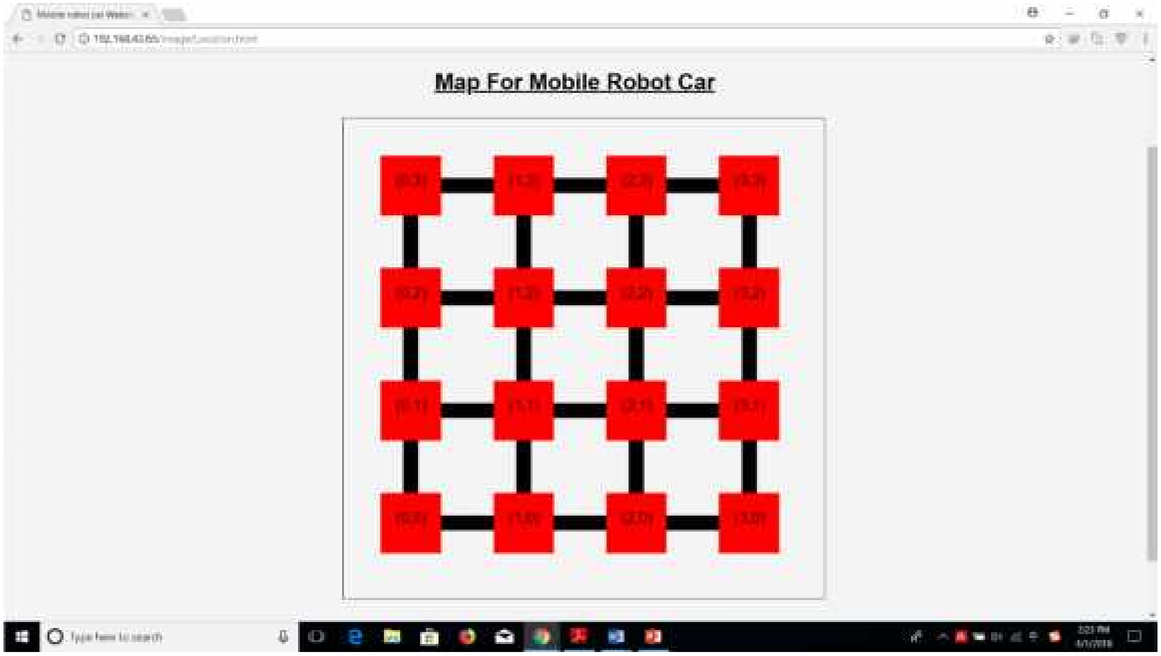


Figure 3-6-3 Interface of monitoring system on web page

Figure above shows the interface of the monitoring system on web page. The code for this web page is attached at the Appendices. The development of web page design involved Java Script, HTML and CSS. Web page is used the PHP code access to MySQL database and AJAX to reload the content. In the web page, it will send HttpRequest to the server every 0.5 seconds. The request will look through and execute the PHP code to query the database, then return back the data into a JSON format. Next, use JSON.stringify() to convert data into string. The data in string because it is used to compare with the location on map, if same the particular location on map will change to green colour to indicate the current position of mobile robot car, else remain red colour.

3-6-3 Overall System Interconnection

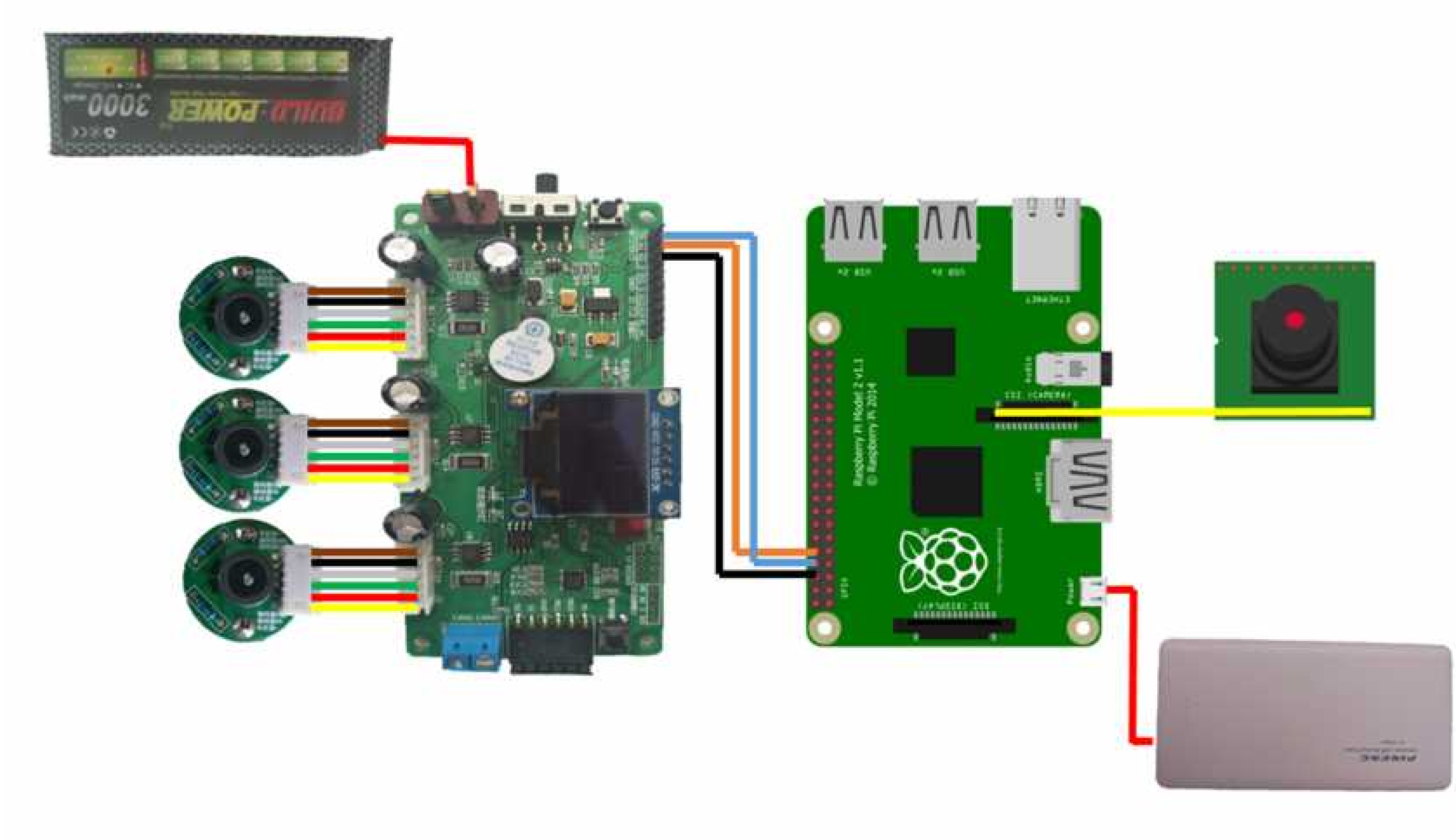


Figure 3-6-4 Overall system Interconnection

3-7 Software Installation

3-7-1 Set up Raspberry Pi 2 Model B

Requirement

Hardware:

- SD card (recommended with type of class 10 and minimum 8GB storage size)
- A monitor with a VGA cable (or if needed, an HDMI to VGA adaptor)
- A USB keyboard and mouse
- Power supply with 5V and 2A
- Ethernet cable (optional)

Software:

- NOOBS (easy installer of Raspbian)

Steps to set up Raspberry Pi 2 Model B

1. Download NOOBS zip file (easy operating system installer that contains Raspian) from https://www.raspberrypi.org/downloads/noobs/.
2. Unzip the downloaded NOOBS zip file and extract it.
3. Copy the extracted NOOBS folder into SD card.
4. Insert the SD card into SD card slot on Raspberry Pi.
5. Plug in keyboard and mouse into USB ports and connect to monitor or TV via HDMI port on Raspberry pi.
6. Plug in power source for Raspberry Pi.

7. While booting up, it will pop out a window for operating system selection. Choose Raspbian and click “install”.

Figure 3-7-1 Interface for Raspbian Installation

8. Installation is required to take sometimes to complete.
9. After finished installation, connect the Raspberry Pi to internet via Ethernet cable or WIFI.
10. Open terminal on desktop environment and type below commands:
    - sudo apt-get update
    - sudo apt-get upgrade

These two commands is to make sure the software on SD card up to date.

11. Close the terminal and navigate to “Preferences”, then select “Raspberry Pi Configuration” and click “Interfaces” to enable VNC, SSH and Camera. Click “OK”. Others features remain “Disabled” due to no use.

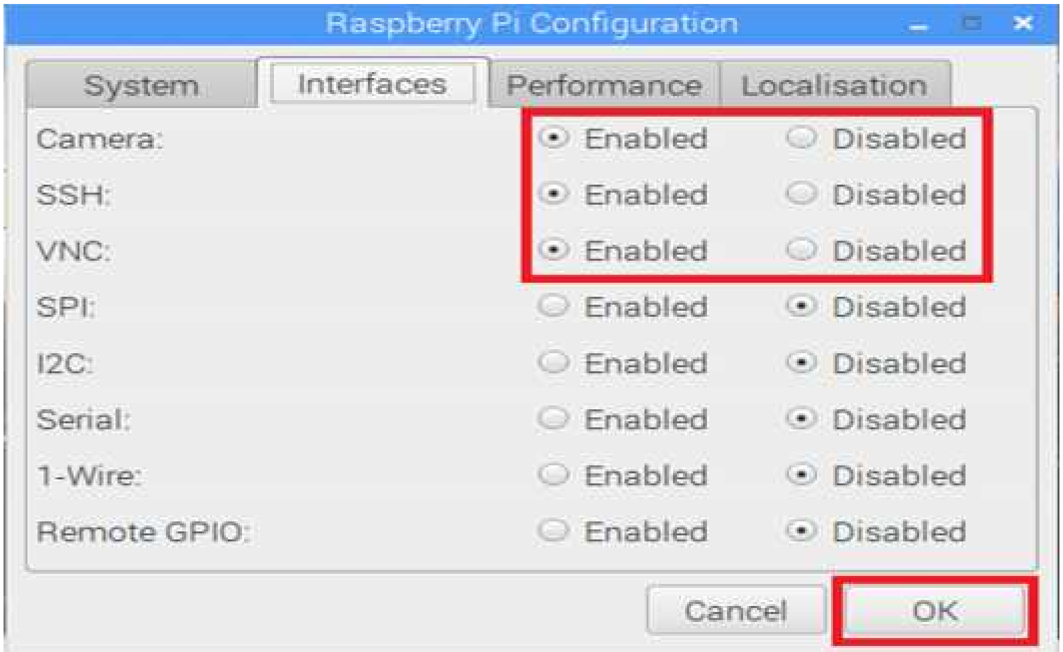


Figure 3-7-2 Interface of Raspberry Pi Configuration

12. Apply remote access method to Raspberry Pi to convenience the work later. Download VNC Viewer from https://www.realvnc.com/en/connect/download/viewer/ and install on PC or laptop.
13. Get the IP address of Raspberry Pi by typing “ifconfig” on terminal.

14. Enter IP address of Raspberry Pi on VNC Viewer to make connection. It will pop out to fill in username and password. The default username is "pi" and password is "raspberry". Then, click "OK".

    Note: The default password is generated on the first time, it can be changes after that.

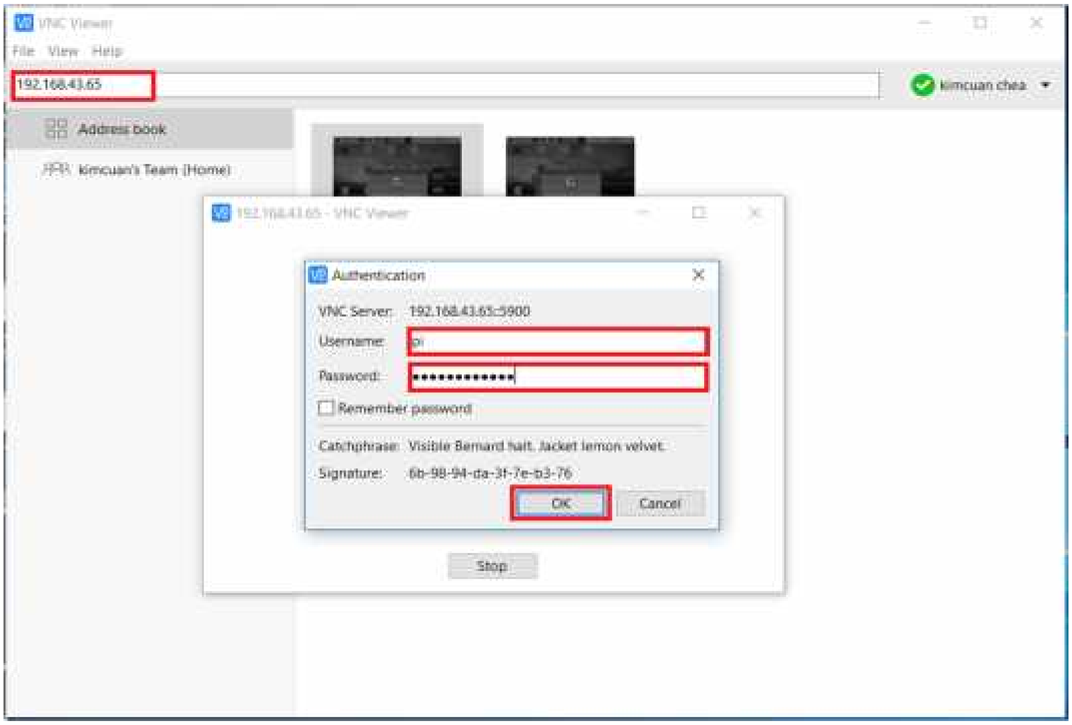


Figure 3-7-3 Interface of VNC viewer

15. HDMI cable, mouse, and keyboard can remove form the Raspberry Pi since PC or laptop is take over the control of Raspberry Pi remotely.

3-7-2 Set up USB WiFi Adapter with Raspberry Pi 2 Model B

Requirement material:

- USB WiFi Adapter (TP-Link 150Mbps Wireless N Nano USB Adapter, model TL-WN725N is used in this project.)

Since the Raspberry Pi 2 Model B no built-in WiFi, thus only two ways to connect the internet. First is to connect the internet via the Ethernet port using Ethernet cable. Second way is to use a USB WiFi adapter via USB port to connect the internet. In this project, USB WiFI adapter is selected to be used due to more flexibility and wireless.

Steps to set up USB WiFi adapter with Raspberry Pi 2 Model B

1. Plug in USB WiFi Adapter into any USB ports on Raspberry Pi.
2. Power up the Raspberry Pi and use VNC Viewer to remote access it.
3. Click terminal icon in taskbar on desktop to open.
4. Type below command to open the network interfaces file on Raspberry Pi.
   - sudo nano /etc/network/interfaces
5. Edit the file by adding few line shown on below figure 3-7-4.

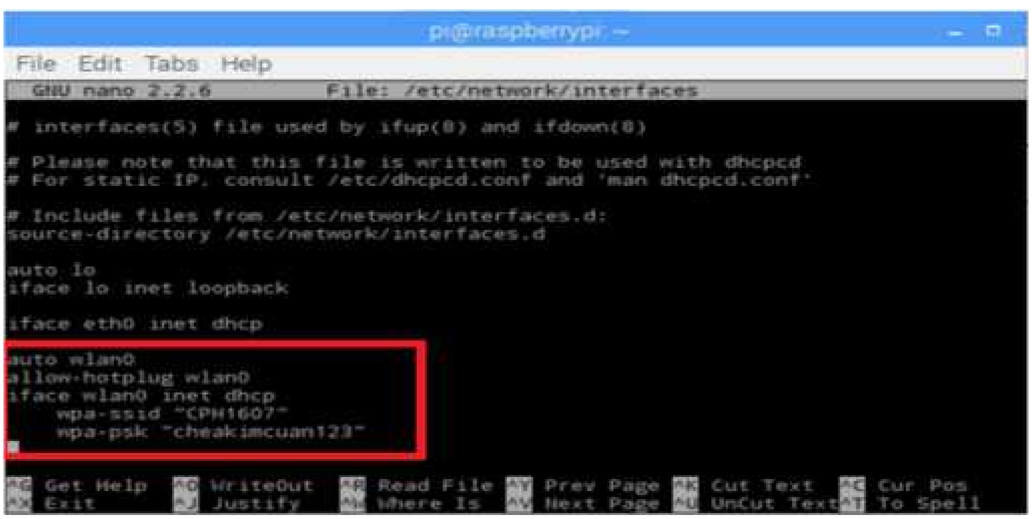


Figure 3-7-4 Raspberry Pi network interface file

6. Press Ctrl-x to exit and ‘Y’ to save the file.
7. Reboot the Raspberry Pi, type the below command
   - sudo reboot

8. Test the WiFi connection on Raspberry Pi. Open terminal and type.
   - sudo ping –c 5 www.google.com

   The above command is to send 5 packets to google and it will return the connection statistics. If successful ping, result will show as below figure 3-7-5, else it will pop the message “ping: unknown host www.google.com”.

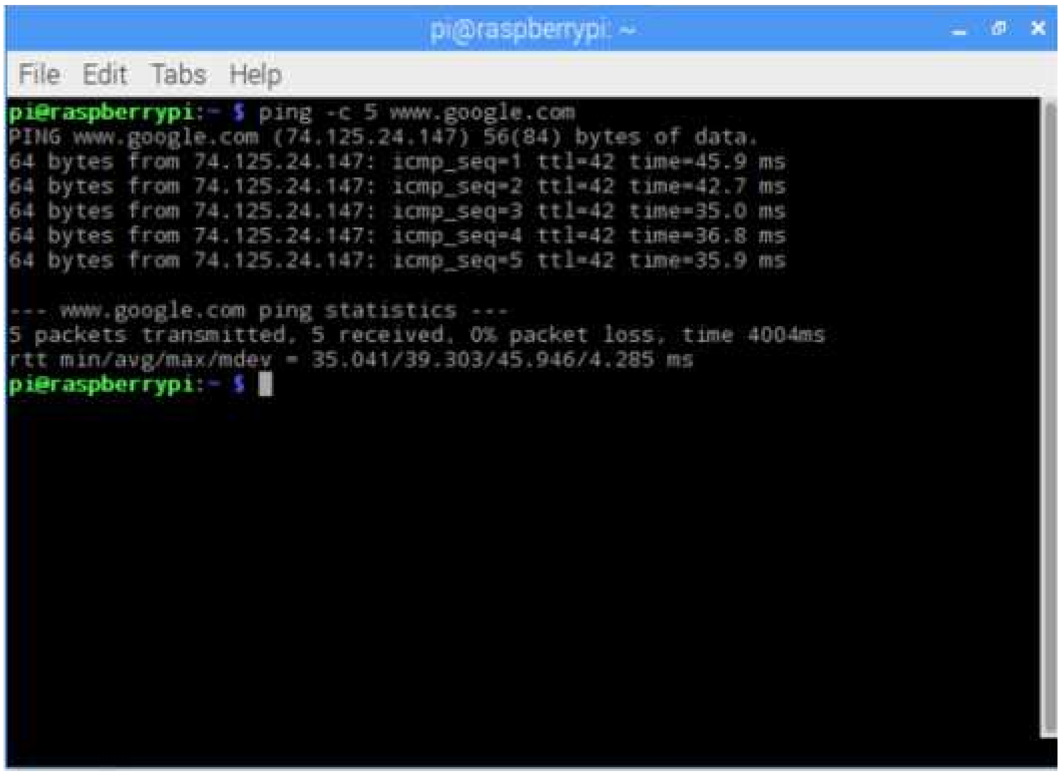


Figure 3-7-5 Ping statistics on terminal

9. Find the IP address on Raspberry Pi within same network and type.
   - sudo ifconfig

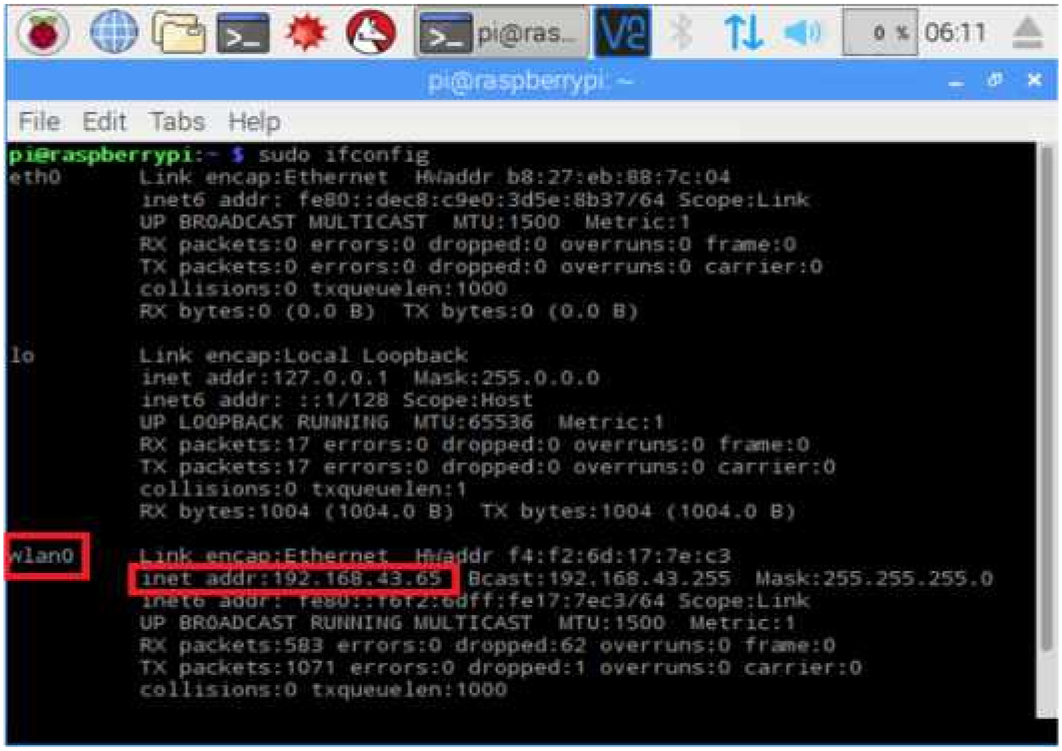


Figure 3-7-6 Network status on terminal

From the figure above, the IP address is located under “wlan0” and find the “inet addr” (IPv4).

3-7-3 Installation a web server on Raspberry Pi (Apache, PHP5, MySQL, PHPMyAdmin)

Installation steps of Apache HTTP server on Raspberry Pi

1. Before installing the Apache HTTP server, make sure the Raspberry Pi is up-to-date. Type the below sudo commands on terminal.
    - sudo apt-get update
    - sudo apt-get upgrade
2. Once the Raspberry Pi is up-to-date, then install the Apache HTTP server.
    - sudo apt-get install apache2 apache2-doc apache2-utils
3. Restart the Raspberry Pi after done for installation.
    - sudo reboot
4. Obtain the IP address of Raspberry Pi before testing the Apache HTTP server.
    - sudo ifconfig

    Go to “wlan0” and look for “inet addr”, it is an IPv4 address for Raspberry Pi.

5. Apache contains a default HTML page (index.html) located at directory /var/www/html/, it is used to test whether server is working properly. Open a web browser on Raspberry Pi to access that page and type URL address.
    - http://localhost

    OR type IP address of Raspberry Pi.

    - http://192.168.43.65
6. The HTML page is loaded on browser and same as figure 3-7-7. It means Apache HTTP server is working properly, else it is not working.

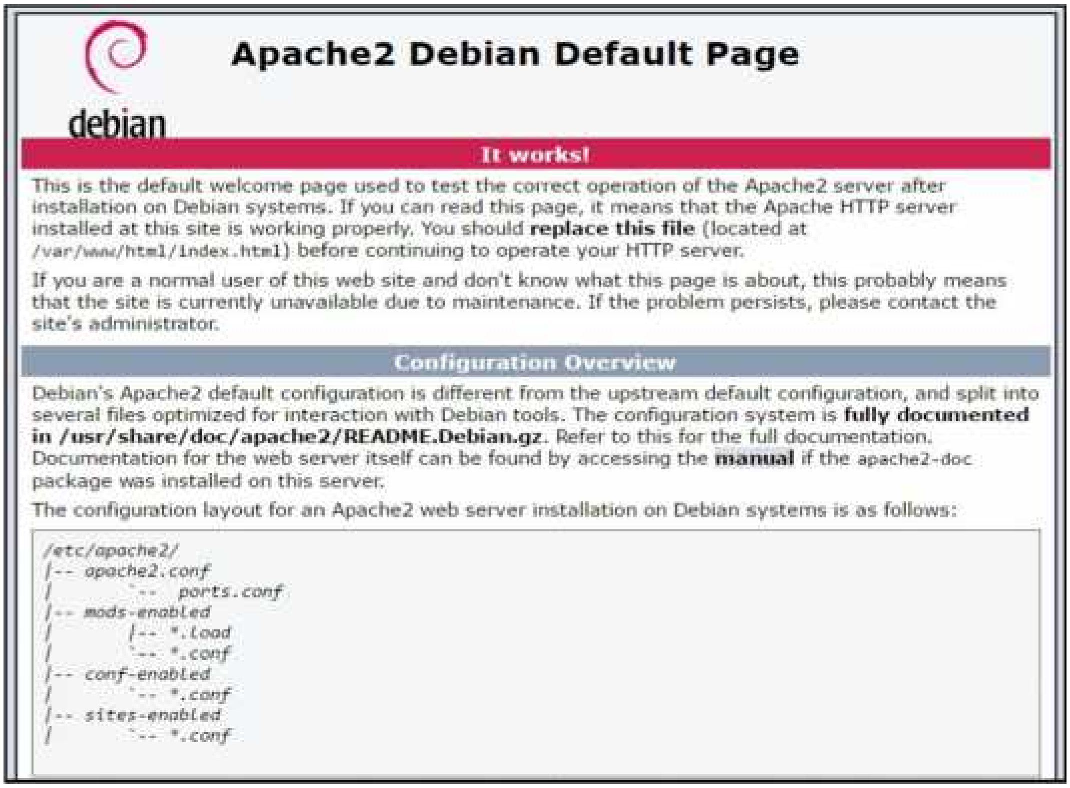


Figure 3-7-7 Apache2 Debian default page shows on browser

<u>Installation steps for PHP5</u>

1. Install PHP5 by typing the following command on terminal.
    - sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache
2. Check whether PHP5 is working properly after installation.
    i. Navigate to following directory.
        - cd /var/www/html/
    ii. Remove the index.html file.
        - sudo rm index.html
    iii. Create a file called index.php.
        - sudo nano index.php
    iv. Add some PHP content in the file.
        - <?php echo “PHP5 is working properly”; ?>
    v. Press Ctrl-x to exit and enter ‘Y’ to save the file.
    vi. Open the web browser and type the IP address of Raspberry Pi. The page is loaded and display similar to below figure if PHP5 installation is successful.

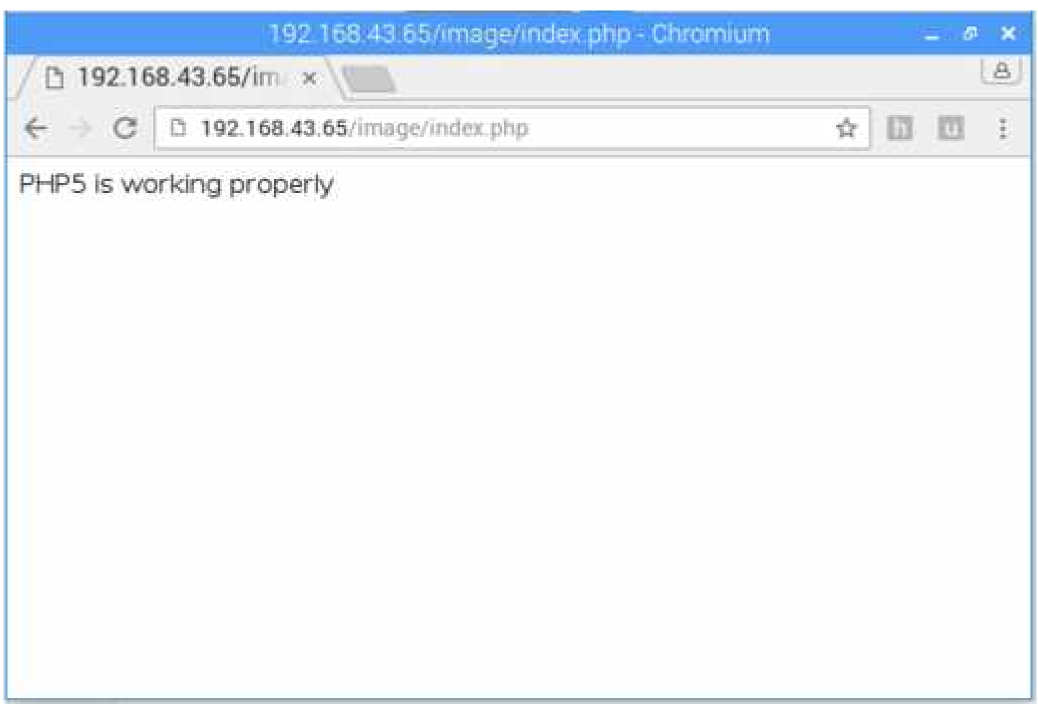


Figure 3-7-8 Content of index.php shows on browser

<u>Installation steps for MySQL database on Raspberry Pi</u>

1. Install the MySQL database, python and PHP binding with database by typing following command on terminal.
   - sudo apt-get install mysql-server php5-mysql python-mysqldb

   It will pop out the configuration box and prompt to enter a password.

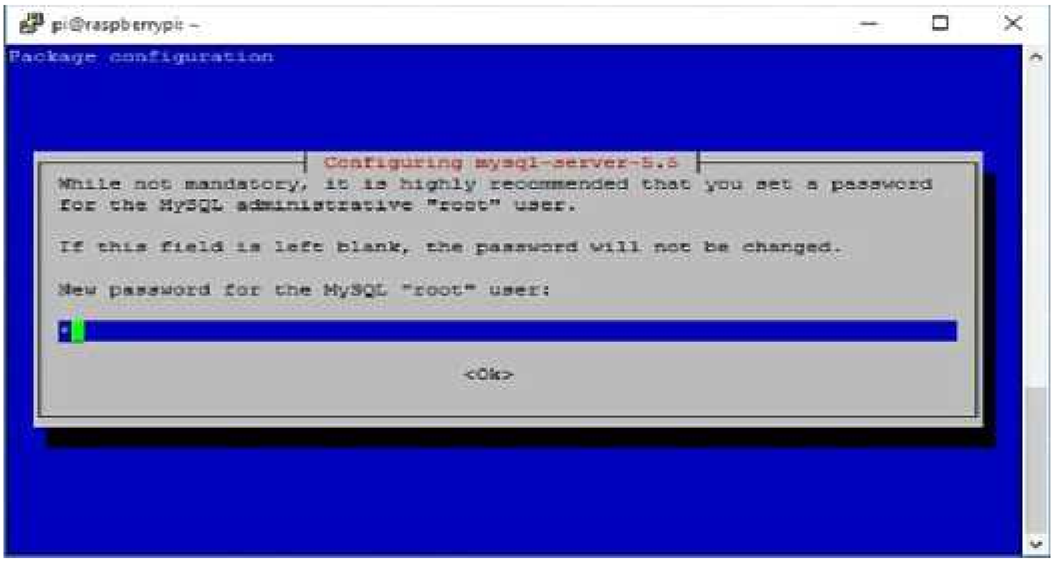


Figure 3-7-9 Interface of MySQL database configuration

2. Enter a password for root – “cheakcsql123” and hit “OK”.

<u>Installation steps for PHPMyAdmin on Raspberry Pi</u>

1. Install the PHPMyAdmin package by typing following command on terminal.
   - sudo apt-get install phpmyadmin
2. It will pop out a configuration box for phpmyadmin shown on below figure, choose 'apache2' and hit 'Ok' to proceed to next step.

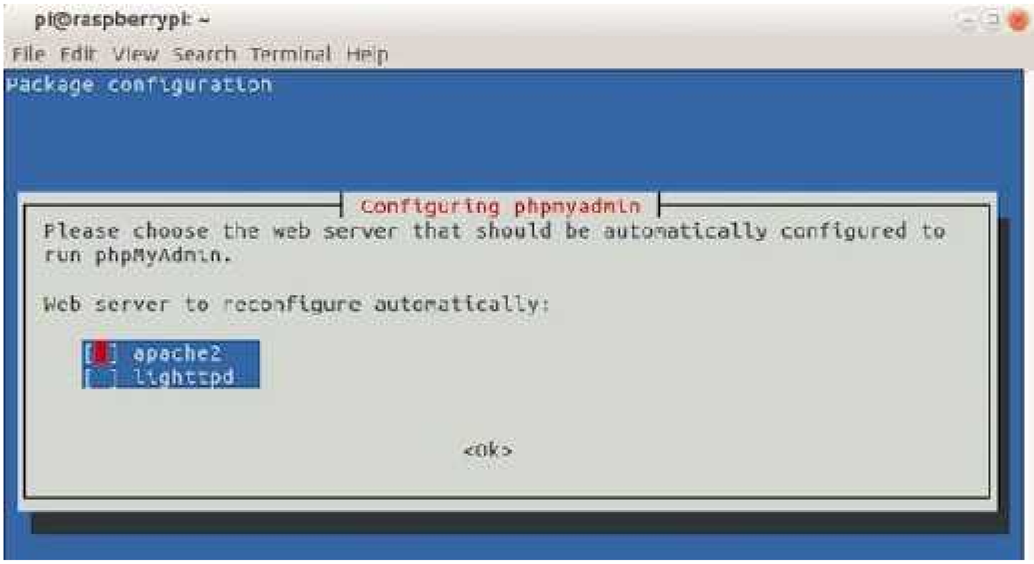


Figure 3-7-10 Interface of PHPMyAdmin configuration

3. Select "Yes" and hit enter to configure the PHPMyAdmin to connect MySQL.

Figure 3-7-11 Interface of PHPMyAdmin configuration

4. Then, setup configuration tool will prompt to enter a password for a root that is needed for PHPMyAdmin.

5. Enter a password – "cheakcphp123" and hit enter.

   Note: The password is better different from the root password of MySQL.

6. Configure the Apache HTTP server for use with the PHPMyAdmin.

   i. Open the terminal and type following commands to access the apache configuration file.
      - sudo nano /etc/apache2/apache2.conf

   ii. Add the following line at the bottom of file.
      - Include /etc/phpmyadmin/apache.conf

   iii. Press Ctrl-x to exit and enter 'Y' to save the file.

   iv. Restart the Apache HTTP service by entering following command.
      - sudo /etc/init.d/apache2 restart

7. Check whether PHPMyAdmin is able to access via web browser. Open web browser and type following URL address.
   - http://192.168.43.65/phpmyadmin

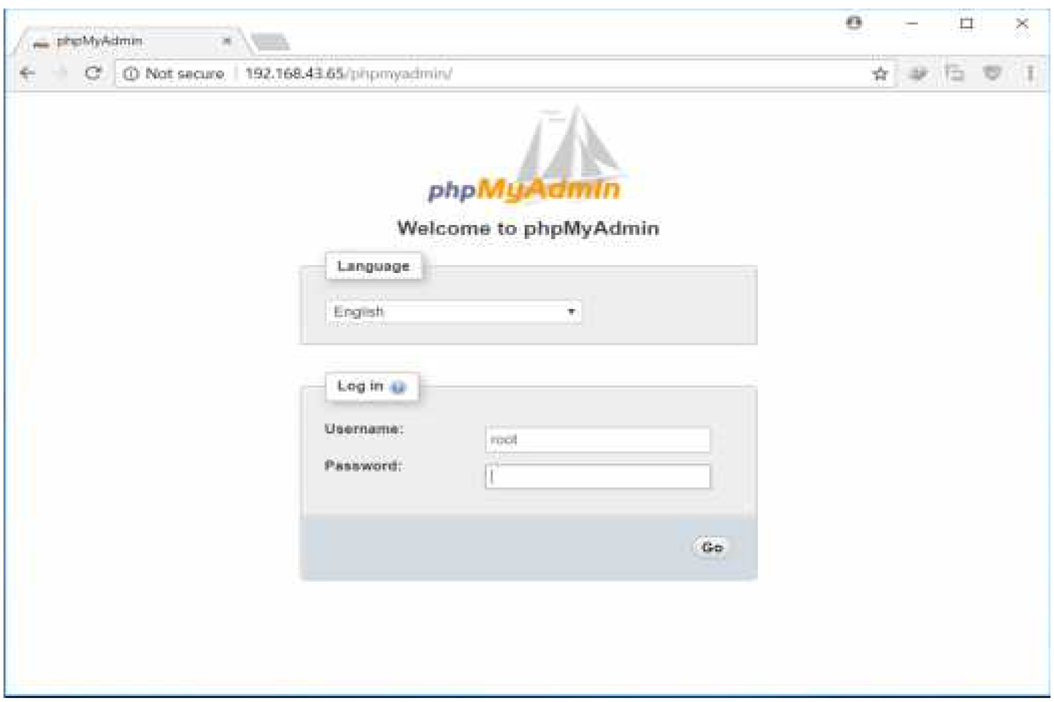


Figure 3-7-12 Interface of PHPMyAdmin on browser

The HTML page as the above figure shows if it is successful to access via web browser.

3-7-4 Installation and Configuration for Raspberry Pi Camera

Required material

- Raspberry Pi Camera Module V2.1
- Ribbon cable

Set up and configuration steps for Raspberry Pi Camera

1. Insert the Raspberry Pi Camera attached with ribbon cable to CSI port on Raspberry Pi as shown on below figure 3-7-13.

Figure 3-7-13 Connection of Raspberry Pi 2 Model B and Pi Camera

2. Open the terminal and type following command to access the Raspberry Pi software configuration tool.
   - sudo raspi-config

3. Navigate to “Interfacing Options” and hit “Select”.

Figure 3-7-14 Interface of Raspberry Pi software configuration tool.

4. Select “P1 Camera” and hit “Select” to choose the enable or disable options for camera.

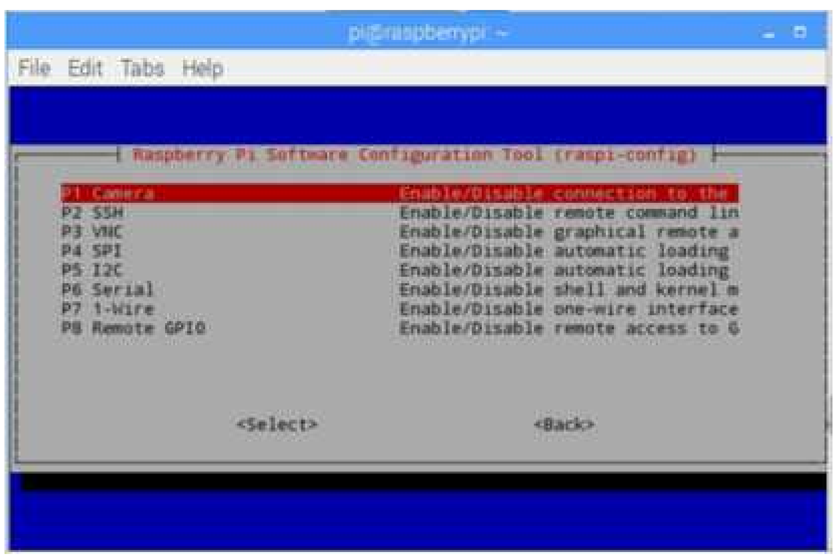


Figure 3-7-15 Interface of Interfacing Option

5. Click “Yes” to enable the Raspberry Pi camera interface.

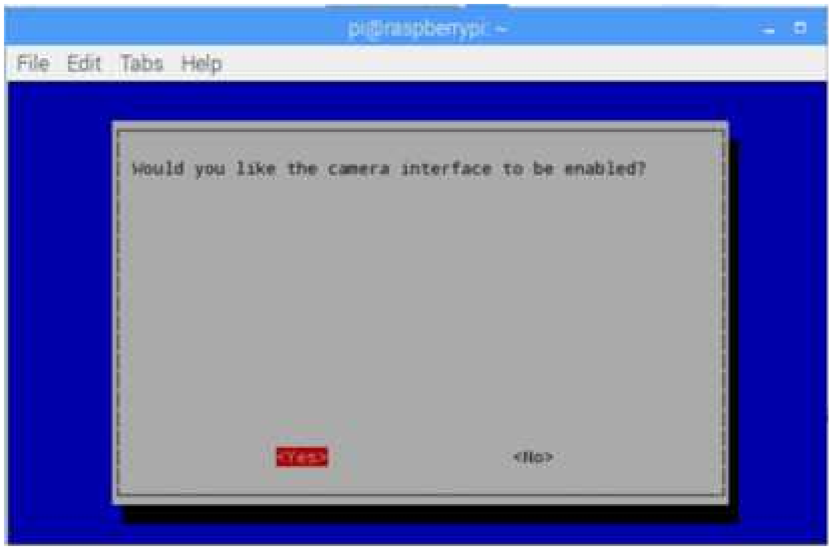


Figure 3-7-16 Interface for Pi Camera enable option

6. Check whether the camera is working properly. Capture an image with the camera module by typing below command.
    - sudo raspistill –o image.jpg
7. The captured image is saved on desktop and able to open it if the camera module is captured successfully.

3-8 Steps to check module and run program on Raspberry Pi 2 Model B

3-8-1 Steps to check module on python IDLE 2.7.9

1. Open up the written python program by double click on it. It will directly open with python IDLE 2.7.9. Figure below shows the interface of python IDLE 2.7.9 after successful open.

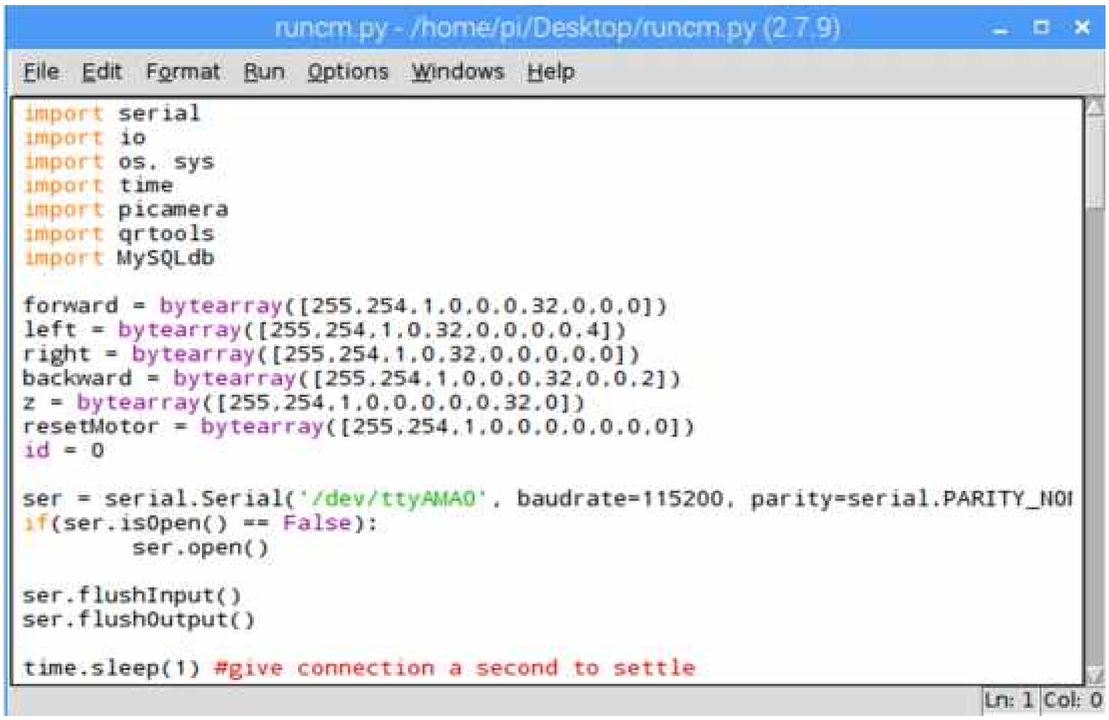


Figure 3-8-1 Interface of Python IDLE 2.7.9

2. Go to Run tab, then click on the "Check Module" or using shortcut key "Alt + X". This "Check Module" is to check the syntax of the program. It have to save the module on python IDLE before clicking it, else python IDLE will prompt the user to save it. If prompt then click 'OK' to save the module. Figure 3-8-2 shows the "Check Module" and pop up box for save on python IDLE.

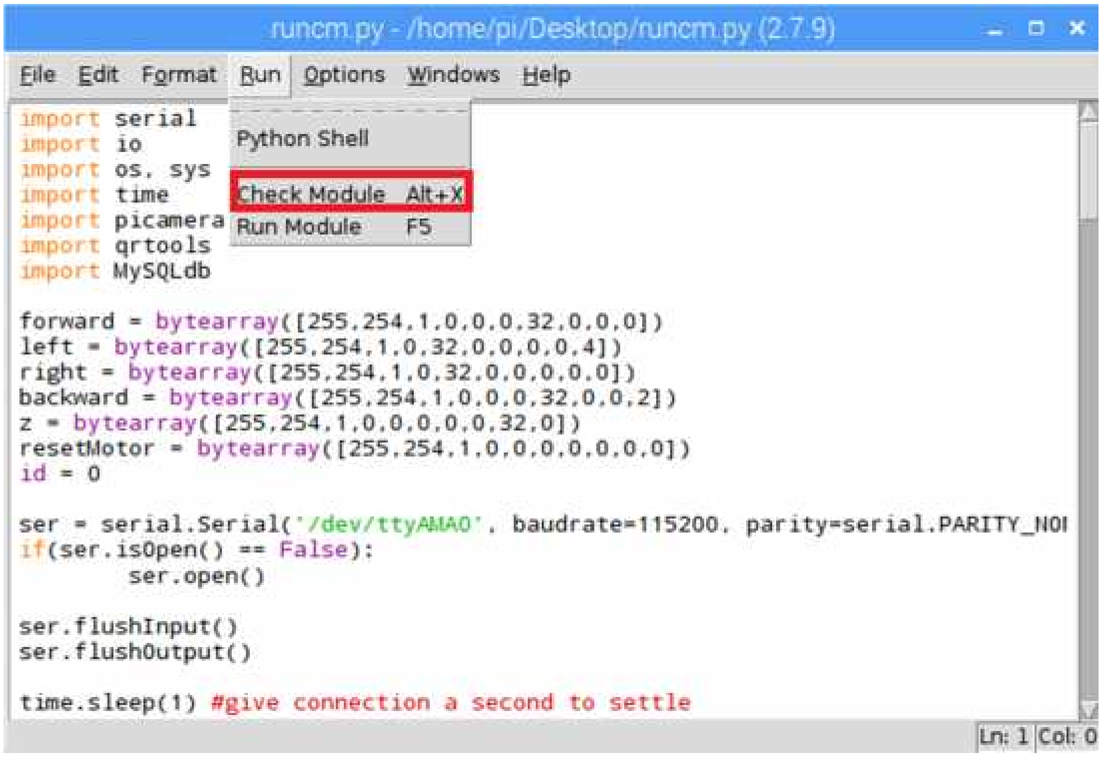


Figure 3-8-2 “Check Module” on Python IDLE 2.7.9

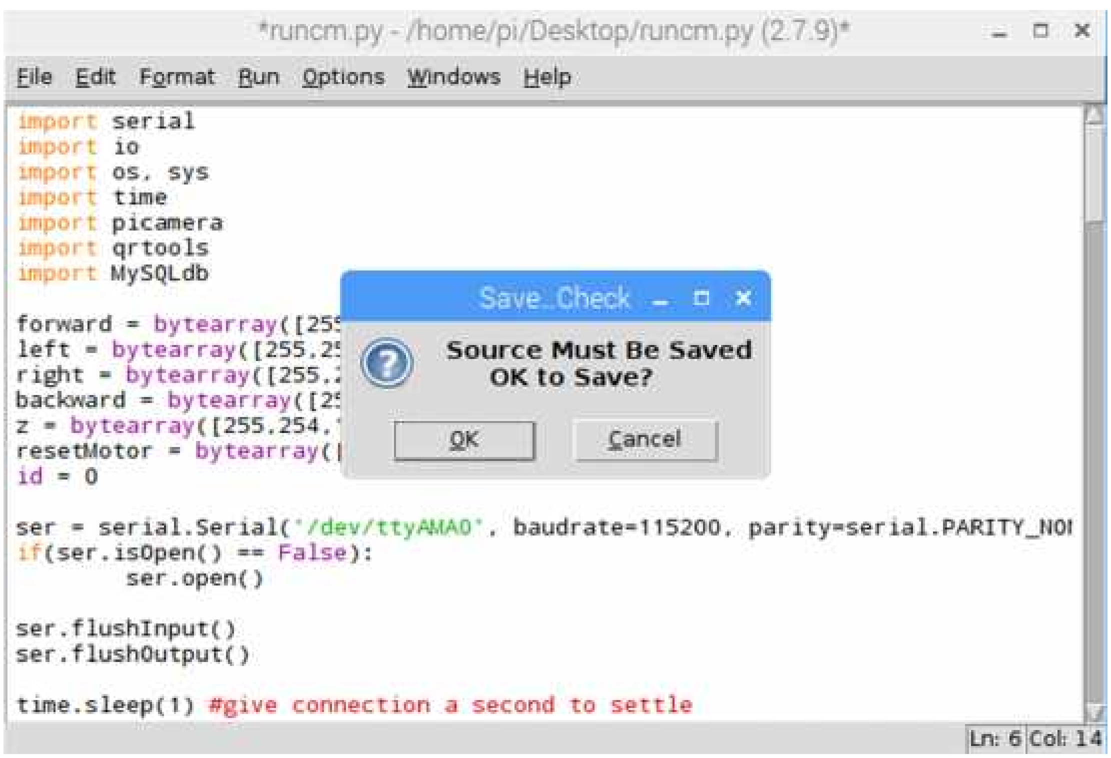


Figure 3-8-3 Save message box on Python IDLE 2.7.9

3. If program no syntax error, then can run the program. In contrast, user have to resolve the syntax error before run the program, Below figure 3-8-4 shows program checking without syntax error.

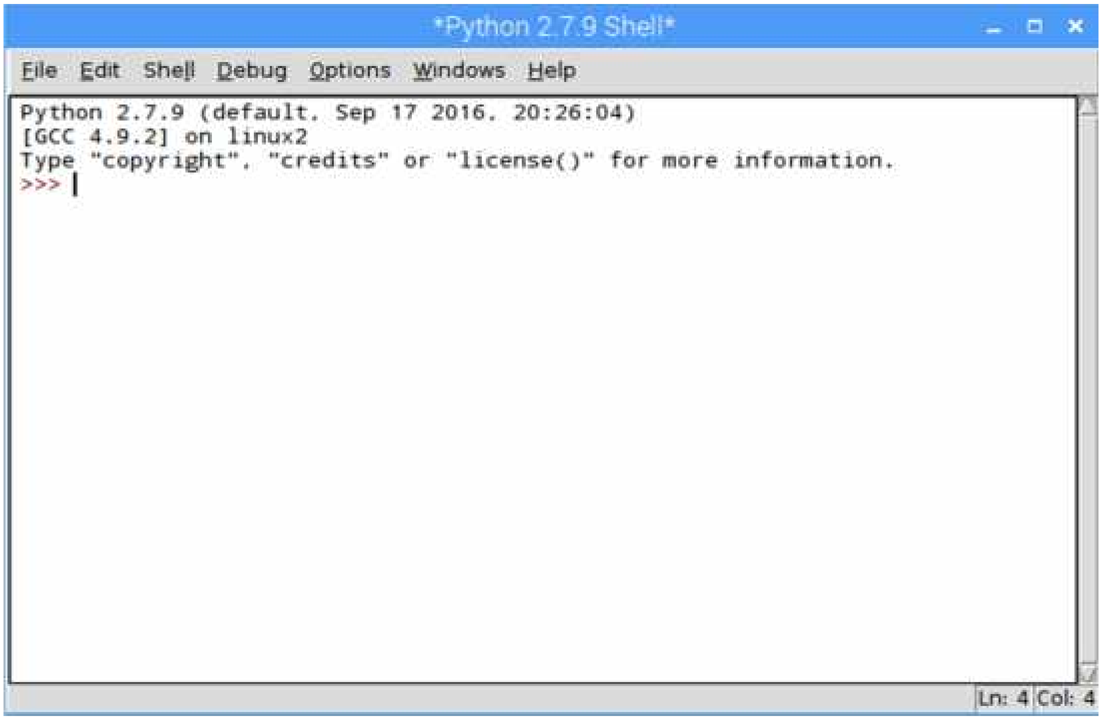


Figure 3-8-4 Shell window on Python IDLE 2.7.9

3-8-2 Steps to run the program on Raspberry Pi 2 Model B

There are two ways to run the python program on Raspberry Pi.

1. Run python program via command prompt.
2. Run python program on python IDLE.

Run python program via command prompt

1. Open the terminal, navigate to the location where you saved the file. This can be done using cd command.
2. Make the python program in an executable file by entering following command.
   - chmod 755 your-file-name.py
3. Then, type below command to run the program.
   - ./your-file-name.py

Run python program on python IDLE

1. Double click on python program. It will directly open with python IDLE.
2. Go to "Run" tab, and click "Run Module" or shortcut key "F5". This Run Module also will do the checking for syntax error like "Check Module", then execute the program on Shell window if there is no error found. From the Shell window, it will show out the output from the program with the use of print command until the end of program. Below figure 3-8-3 shows the "Run Module" in python IDLE.

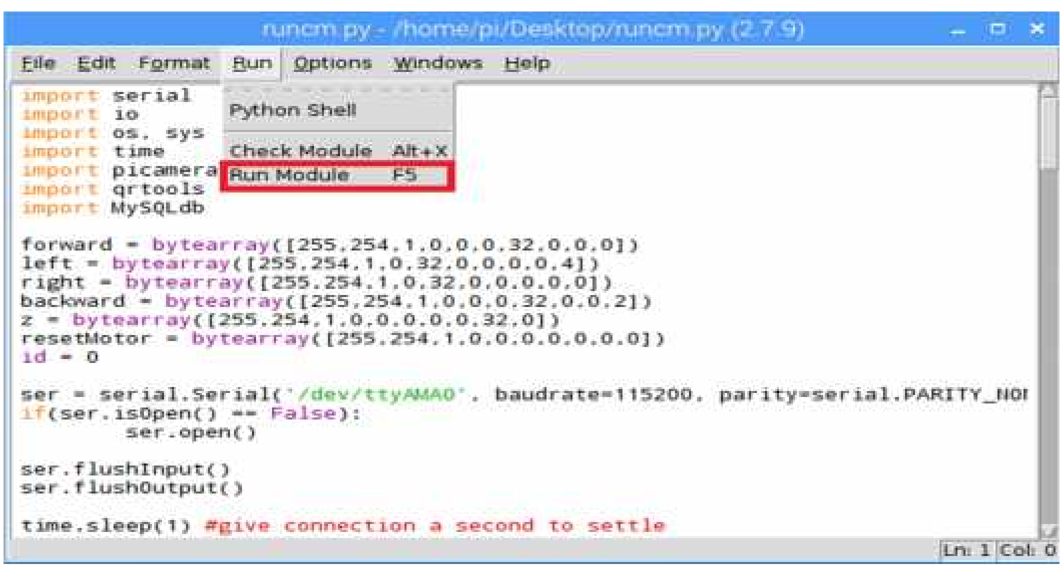


Figure 3-8-5 "Run Module" on Python IDLE 2.7.9

3-9 Steps to operate mobile robot car

1. Omni directional wheel is much easier to slip as compared to normal tire. The dust on tire or floor that reduced the friction. Therefore, each time for testing or demo, recommended to clear the dust on tire and floor. For cleaning dust on tire, used the half wet clothes to wipe it.
2. Plug in power source to the Omni-wheel main board and power up the Raspberry Pi.
3. Place the mobile robot car into preliminary starting correction point as below figure 3-9-1.

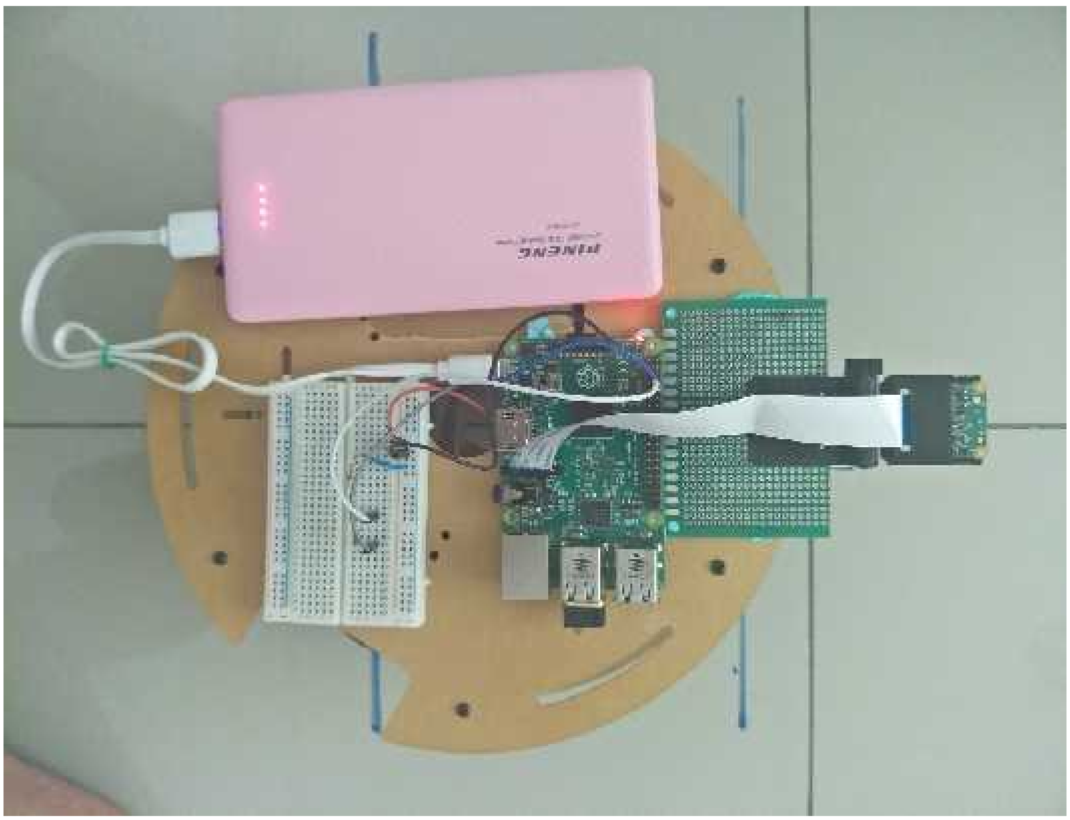


Figure 3-9-1 Mobile robot car placed on starting point

4. Open VNC viewer on laptop. Type username : pi and password : raspberry520. Click “OK”, it will remote login to Raspberry Pi.

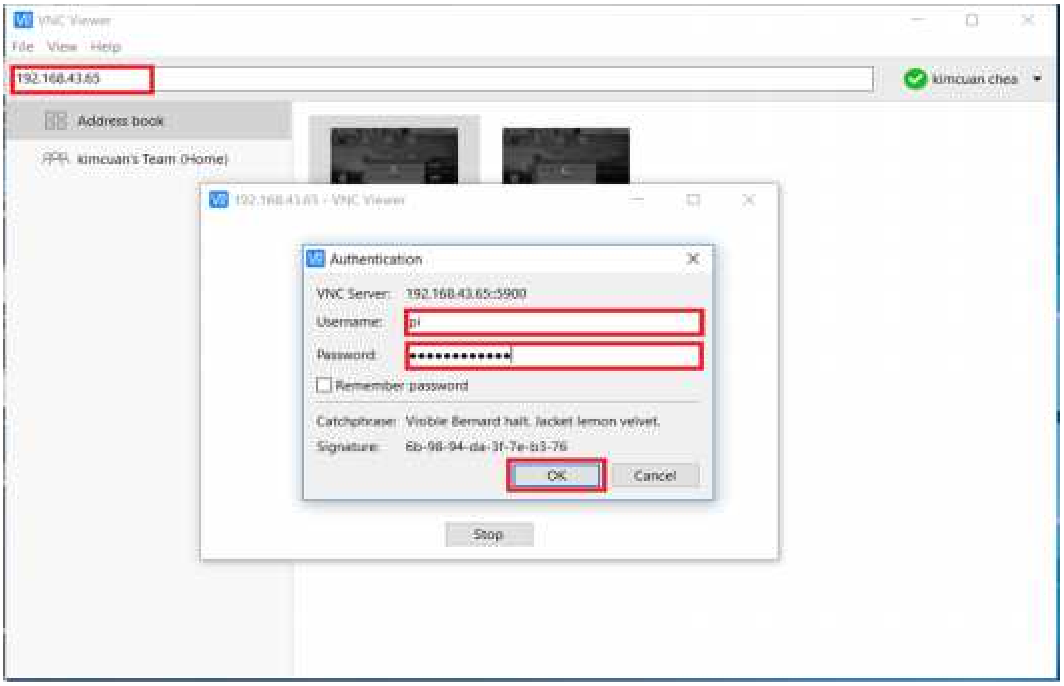


Figure 3-9-2 Interface of Authentication on VNC viewer

5. Open the browser on laptop and type http://192.168.65.43/image/home.html. It automatically link to a web page as shown on below figure 3-9-3.

Figure 3-9-3 Interface of Monitoring system on browser

6. Go to Location tab and click, a monitoring web page shows as below figure 3-9-4.

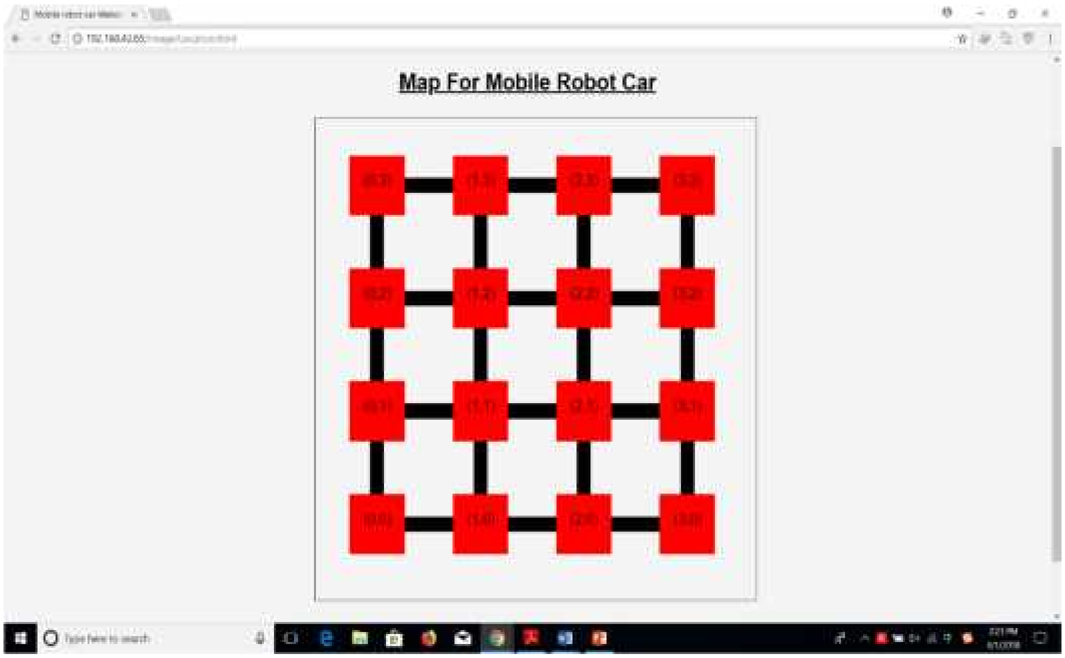


Figure 3-9-4 Interface of map on browser

7. Back to VNC viewer, open terminal.
8. Go to desktop by typing below command.
   - cd Desktop/
9. Type following command to run the python program.
   - python FYP.py

10. From terminal window, prompt to key in the path for mobile robot to travel. The selection for path to travel can be viewed on web page before entering the path. The figure 3-9-5 shows the commands on terminal.

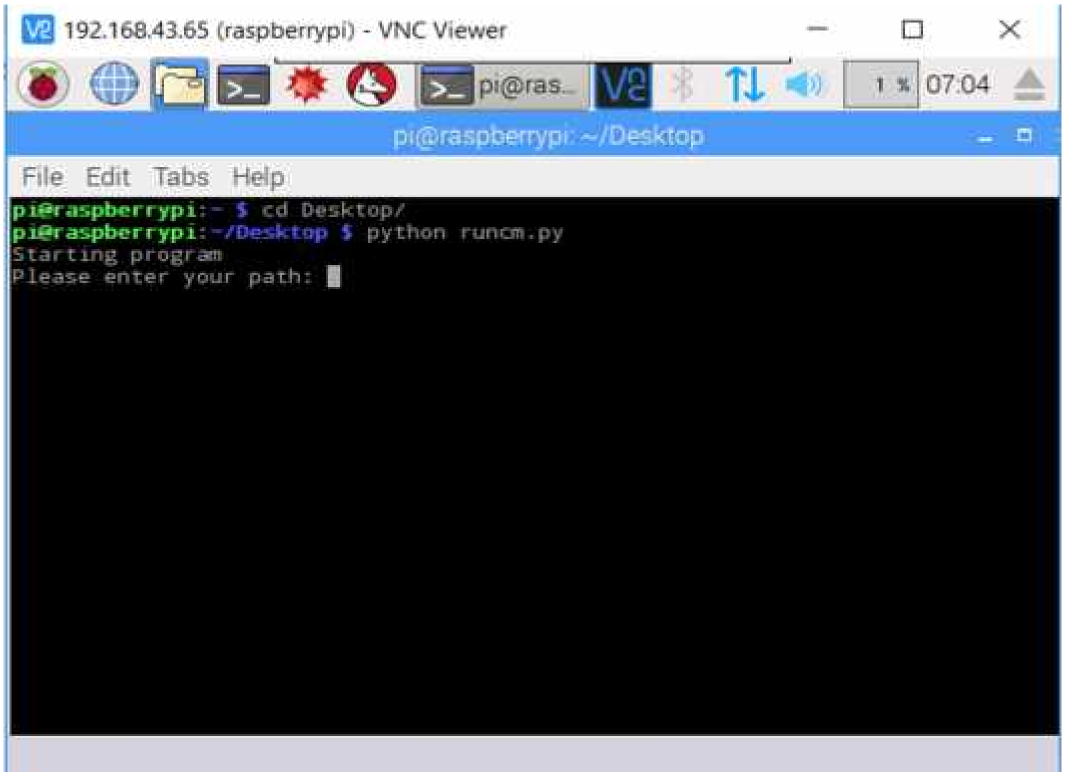


Figure 3-9-5 Commands on terminal

11. Once entered the path, observe the web page to keep track the mobile robot car.

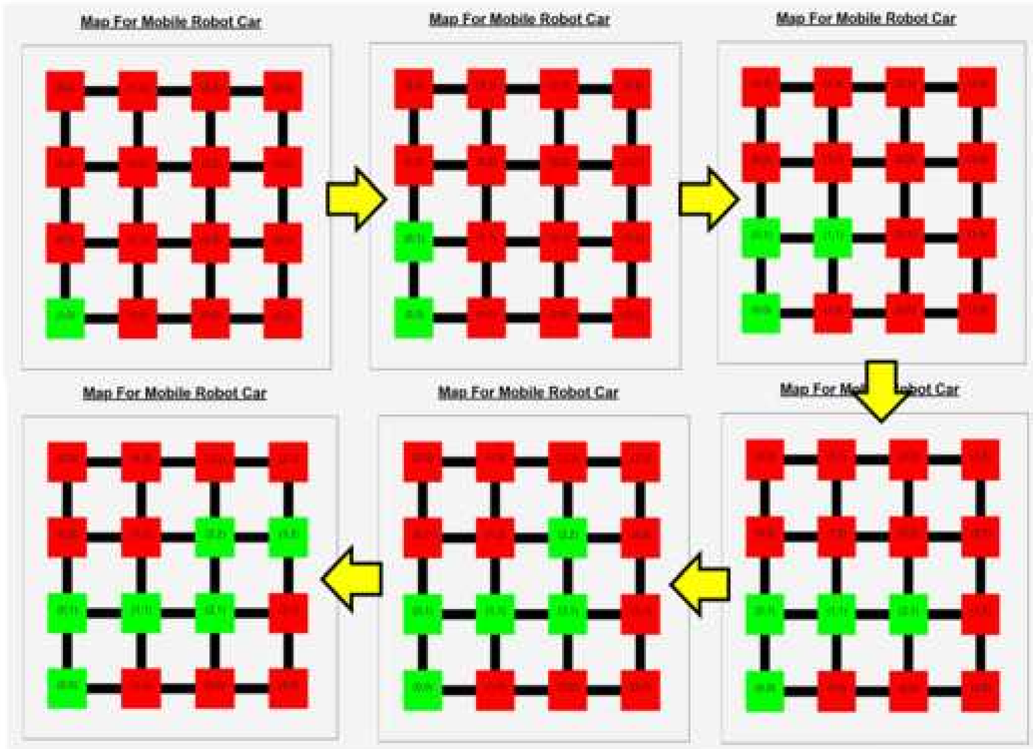


Figure 3-9-6 Trace mobile robot car on browser

12. Once destination reach, terminal window will prompt to key in ‘Y’ or ‘N’ to reset the path. ‘Y’ is for mobile robot car travels back to starting correction point, whereas ‘N’ is for mobile robot car remains at the destination.
13. If ‘Y’ then all data within database will clear and mobile robot car returns back to start location, else stop at destination and program terminate.

Chapter 4 Design Specification

4-1 Methodology

4-1-1 QR code acts as correction point methodology

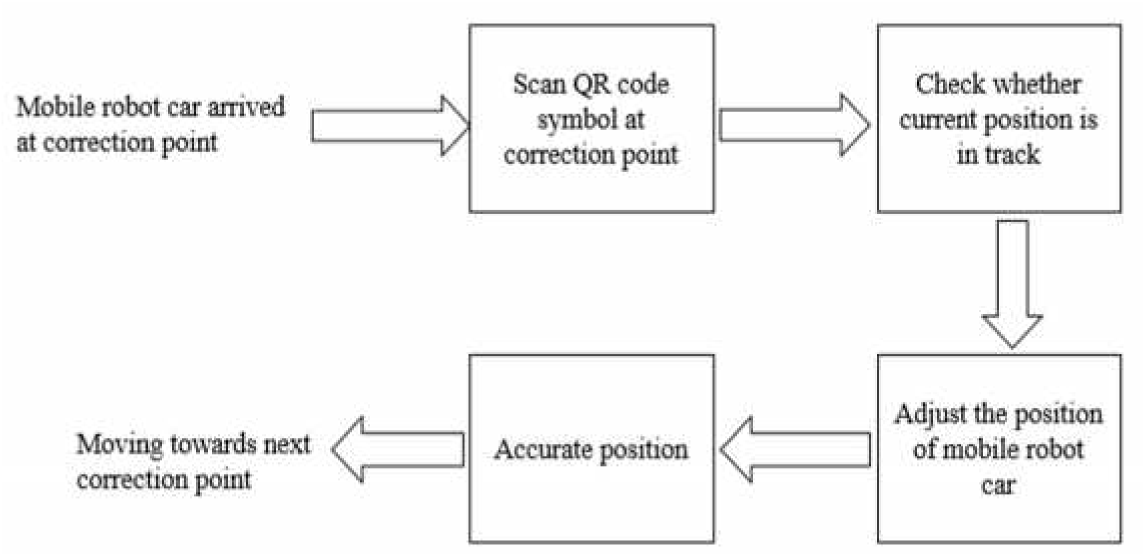


Figure 4-1-1 System flow of QR code methodology

QR code acts as the correction point is a method to keep track the position of the mobile robot car. QR code is the trademark for a type of matrix barcode or two-dimensional barcode which allowed the scanner such as webcam or camera to scan through it. Due to fast readability of it, mobile robot car that installed raspberry pi camera is able quick to capture and decode the information or data stored. Thus, in this project, the mobile robot car will moved towards to the correction point that contained QR code symbol and scan through it in order to adjust and keep track of the position of mobile robot car. After adjusted the position, mobile robot is continued moving towards to the next correction point until it reaches the destination.

Adjustment Algorithm

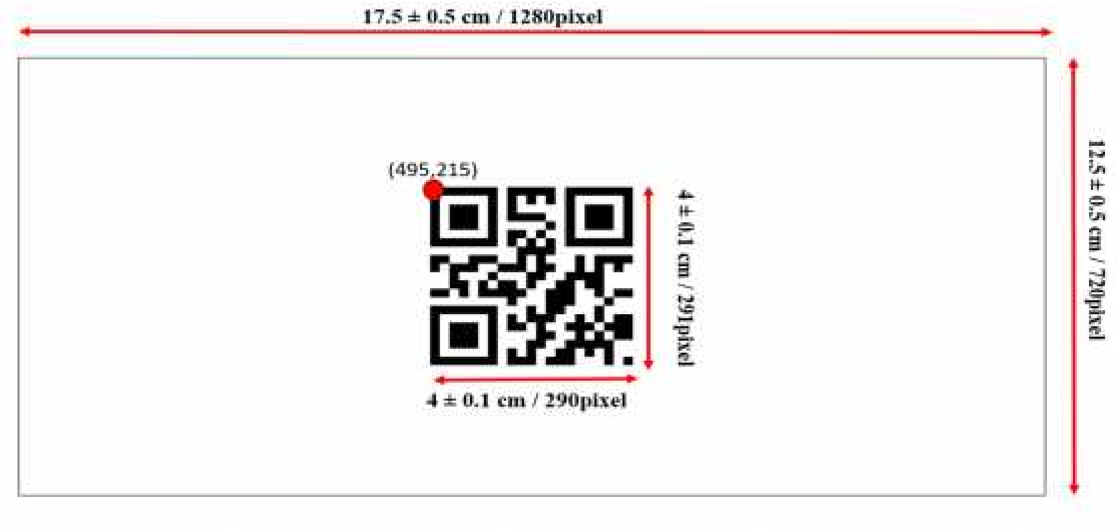


Figure 4-1-2 Image size for QR code within camera view

The above figure is showed QR code is located at the centre of camera. The red point that labelled on top left of QR code that is a predefined point. Thus, each time adjustment, the mobile robot car will move to predefined point, then only adjust the z-axis to ensure the QR code is perpendicular to the camera view.

Adjustment on X and Y axis algorithm

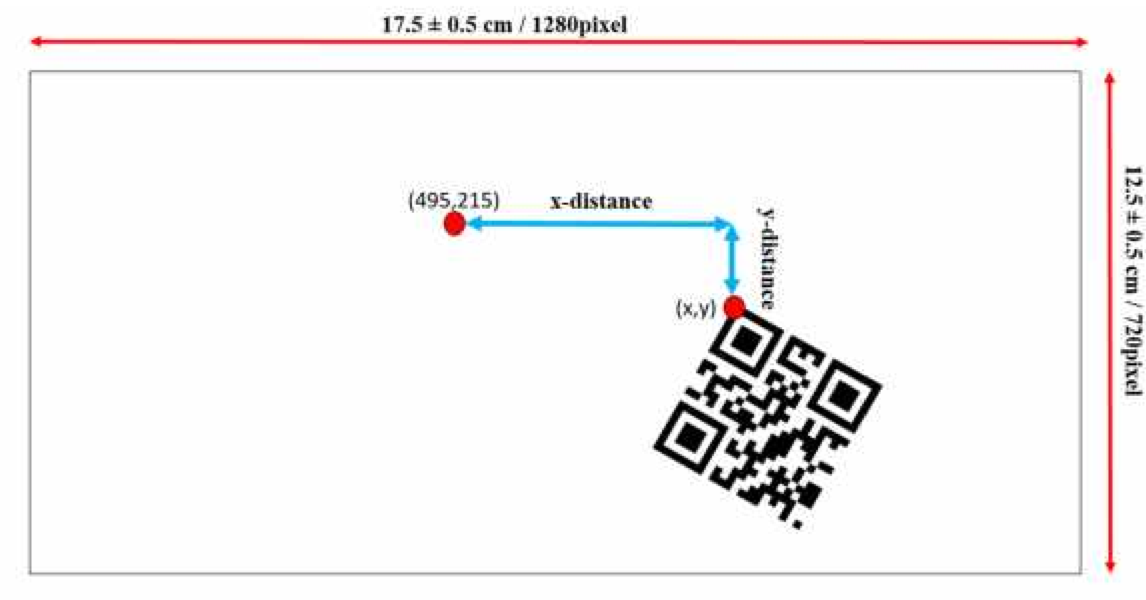


Figure 4-1-3 Top left point of QR code different from predefined point

X - Distance Formula:

x-distance per pixel = $\frac{17.5}{1280}$ = 0.01367

If QR code top left point, x >= 495:

difference pixel in x-axis = (1280 – 495) – (1280 – x)

else QR code top left point, x < 495:

difference pixel in x-axis = 495 – x

x-distance for adjustment = different pixel in x-axis * x-distance per pixel

Y - Distance Formula:

y-distance per pixel = $\frac{12.5}{720}$ = 0.01736

If QR code top left point, y >= 215:

difference pixel in y-axis = (720 – 215) – (720 – y)

else QR code top left point, y < 215:

difference pixel in y-axis = 215 – y

y-distance for adjustment = different pixel in y-axis * y-distance per pixel

Adjustment on Z Axis algorithm

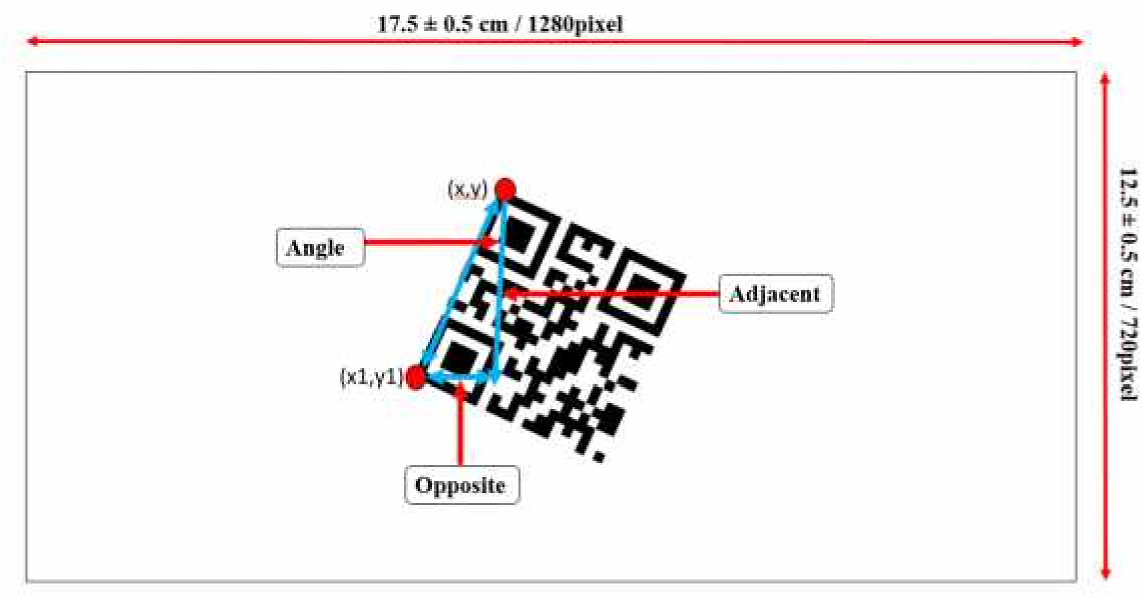


Figure 4-1-4 QR code shifted at certain angle

Z - Angle Formula:

Let top left point = x, y

Let bottom left point = x1, y1

Opposite = x – x1

Adjacent = y – y1

Angle in degree = $\tan^{-1} \frac{\text{Opposite}}{\text{Adjacent}}$

If Opposite >= 0:

move in clockwise direction

else Opposite < 215:

move in anti-clockwise direction

4-1-2 Prototyping Model

The prototype model is a software development model and developed based on the currently known requirements. Prototyping is an idea to define the requirements of the system without the assist of the manual process or existing system. By interacting with prototype frequently and evaluating on it, the client is better to understand and comprehend the requirement of the desired system. Besides, development of the prototype is made the error to be detected much earlier and also getting the feedback lead to a better solution to improve from the current system. Below figure 4-1-5 shows the system flow of prototyping model.

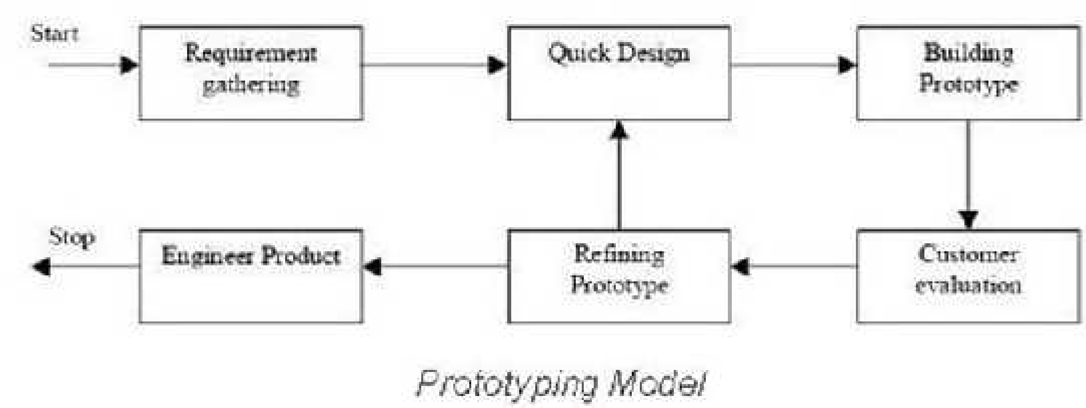


Figure 4-1-5 System flow of prototyping model

Explanation for each phase in prototyping model:

- **Requirement Gathering:** First stage before a project is started, all the requirement must be gathered and performed the analysis on the gathered requirement.
- **Quick Design:** Once the requirement is acquired, a preliminary design for the system is created. The system is not much in detail, but it included the important parts of the system which provides an idea about the system works to user.
- **Building Prototype:** The prototype is built by getting the information from quick design and further to modify it. It represented as a foundation design for the system.

- **Customer Evaluation:** The prototype is presented to the user and get the feedback after evaluated. This part considered as a middle stage of development process.
- **Refining Prototype:** Prototype is refined according to the requirements from user after evaluated it. Once the prototype that fulfilled the user's requirements and satisfied by user, a finalized system is created based on the final prototype.
- **Engineer Product:** The thoroughly evaluation and testing must be done on final system. It also to prevent system failures and minimize the downtime with the help of routine maintenance.

In this project, the first step is to gather all requirement of the project and analyse on it. The hardware components and software tools used in this project were listed down. The datasheet of all hardware components has been studied. Then, a quick design for the project was developed by drawing out the block diagram, system requirement and flowchart of the system. This was provided an idea on how the system works. After that, a prototype for simultaneous localization and mapping of mobile robot car was built based on modified from the quick design. Next, the prototype was evaluated by user to get some feedbacks and suggestions on improvement. By getting the feedbacks and suggestions from user, the prototype of mobile robot car was carried to the refining stage, which was to redesign the quick design of the system and produce the prototype of mobile robot car that fulfilled the user requirements. If the user satisfied with the last refined prototype of mobile robot car, then the finalized of prototype was created. Lastly, the mobile robot car has to evaluate and testing thoroughly to prevent failures and minimize downtime.

4-1-3 Distances Travelled Methodology

Since the mobile robot car is travelled from one correction point to another correction point, there required to measure the distance travelled between it. Mobile robot car have an encoder that mounted on DC geared motor that to sense the rotation of wheel. The encoder will generated 390 pulses per revolution. Omni-wheel main board is received the pulses from encoder and used counter to accumulate it. From that, the number of revolutions for mobile robot car to travel within two correction points is computed by dividing the total number of pulses counted with 390 pulses per revolution. Then, the circumferences of wheel is required to compute in order to calculate the distance travelled. Hence, distance travelled between two correction points is calculated by multiplying the number of revolutions and circumferences of wheel. The formula to calculate the distance travelled is shown on below.

Distance Travelled Formula:

$$\text{Number of revolutions} = \frac{\text{Total number of pulses counted}}{\text{390 pulses per revolution}}$$

$$\text{Circumferences of wheel} = 2 * \pi * \text{radius of wheel}$$

$$\text{Distance travelled } = \text{number of revolution} * \text{circumferences of wheel}$$

4-1-4 Monitoring System On Web Page Methodology

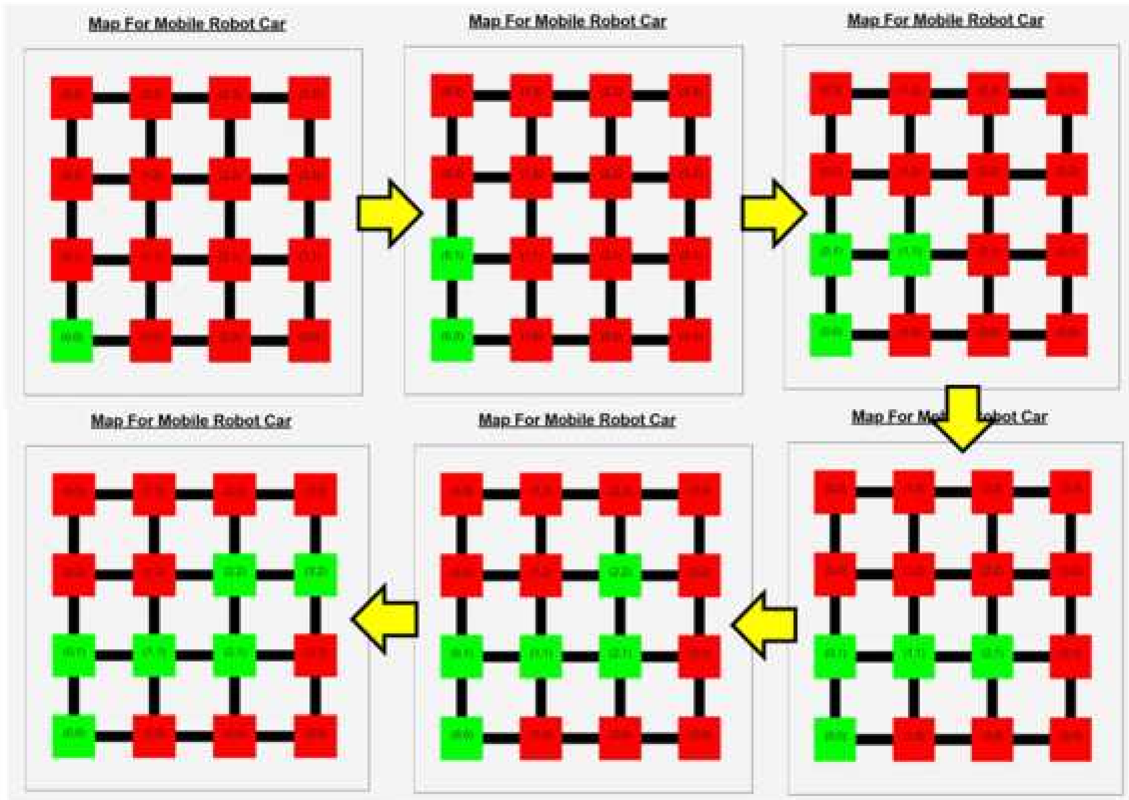


Figure 4-1-6 Tracing process for mobile robot car on web page

Monitoring system on web page approach that is used to keep track the location of the mobile robot car. It allowed the user to check whether destination is reachable by mobile robot car. However, there are some requirements in order to achieve the tracking purpose.

The first requirement is to possess a database system to store the location of correction point. When mobile robot car reach at the specific correction point, its location has to insert into MySQL database. Then, for the second requirement is to design a web page that able to retrieve the data from database and display. The PHP language is used to retrieve the data from database and pass to web page. Therefore, whenever web page get the data or location, it will turn the red colour box on web page into green colour to indicate the current position of mobile robot car. This allowed the user to keep track the mobile robot car via monitoring system on user-friendly web page.

4-2 Tools

4-2-1 Hardware Tools

Raspberry Pi 2 Model B board

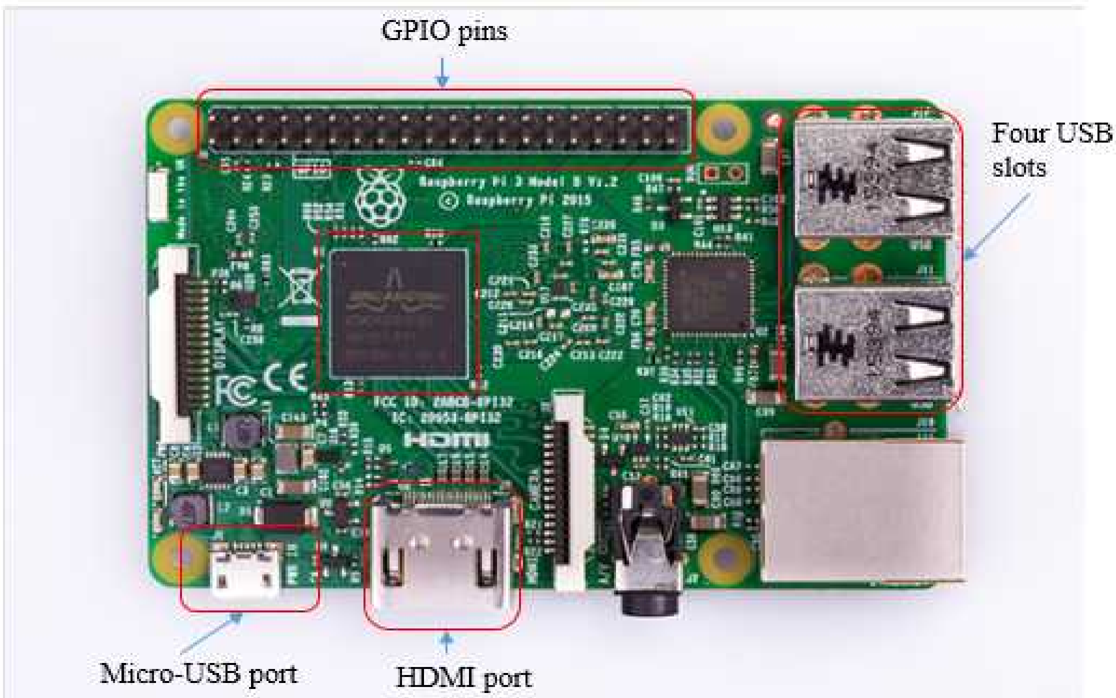


Figure 4-2-1 Raspberry Pi 2 Model B board

Raspberry Pi 3 Model B is equivalent to a mini computer. It has an ARM compatible processing unit (CPU), on-chip graphic processing unit (GPU), on board memory range from 256MB to 1 GR RAM. It is required an operating system which is stored into Micro Secure Digital (SD) card to boot up. The board also has MicroSD card slot, Micro-USB port for power supply, HDMI port for display purpose, audio output jack, LAN port, four USB ports and forty pin extended GPIO pins. In this project, Raspberry Pi 2 Model B is selected as master board which controlled the slave (Omni-wheel main board) through UART and also set up Apache, MySQL server on it. Table 4-2-1 shows specification of Raspberry Pi 2 Model B.

SoC	Broadcom BCM2837
CPU	900MHz 32-bit quad-core ARM Cortex-A7
GPU	Dual Core VideoCore IV
Board Memory (RAM)	1GB LPDDR2
USB 2.0 port	4
Video output	HDMI(rev 1.3 & 1.4)
Ethernet	10/100 Base Ethernet socket
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
GPIO Connector	40-pin 2.54mm (100 mll) expansion header
Memory Card Slot	Micro SDIO
Operation Power	Micro USB socket 5V, 2A
Dimension	85 x 56 x 17mm
Weight	45g

Table 4-2-1 Specification of Raspberry Pi 2 Model B

Raspberry Pi Camera Module V2

Figure 4-2-2 Raspberry Pi Camera Module V2

The Raspberry Pi Camera Module V2 has a Sony IMX219 8-megapixel sensor which is enhanced image quality as compared to 5-megapixel of original Camera Module. The camera works with models of Raspberry Pi 1, 2 and 3. It also supports 1080p30, 720p60 and VGA90 video modes. Besides, it attaches via a 15 cm ribbon cable to the CSI port on the Raspberry Pi and it can be accessed through third-party libraries built especially Picamera Python library. Therefore, this camera module is used to capture the image for further processing on Raspberry Pi 2. Table 4-2-2 shows the specification of Pi Camera Module V2.1.

Still Resolution	8 Megapixels
Video Modes	1080p30, 720p60 and 640 x 480p60/90
Linux Integration	V4L2 driver available
C programming API	OpenMAX IL
Sensor	Sony IMX219
Sensor Resolution	3280 x 2464 pixels
Focal length	3.04mm
Size	15 x 24 x 9 mm
Weight	3g

Table 4-2-2 Specification of Pi Camera Module V2

Omni-Wheel Main Board

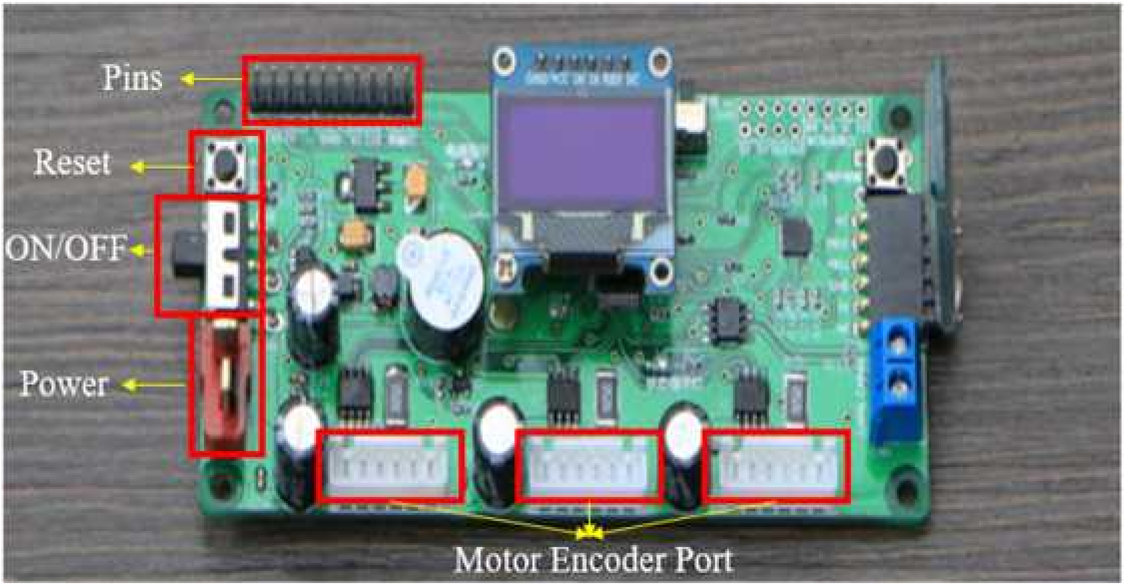


Figure 4-2-3 Omni-wheel main board

Omni-wheel main board is acted as a microcontroller to control the peripheral devices. It equipped with OLED screen and CAN (Controller Area Network). Through CAN, it able to send instruction to CAN port for control or receive data. Furthermore, Omni-wheel main board provided support for serial communication, SWD, and I2C port. 5v and 3.3v output pins on Omni-wheel main board can supplier power, but 5v output is recommended less than 800mA load and 3v output less than 200mA. It also supported STLINK, JLINK for debugging. Besides, Omni-wheel main board also equipped with a buzzer, it will alarm when battery voltage lower than 11.1v. Therefore, in this project, Omni-wheel main board is used to control the DC geared motor mounted quadrature encoder by connecting to motor encoder port. Lastly, it also acted as a slave that controlled by Raspberry Pi 2 Model B via UART.

Quadrature Encoder on DC Geared Motor

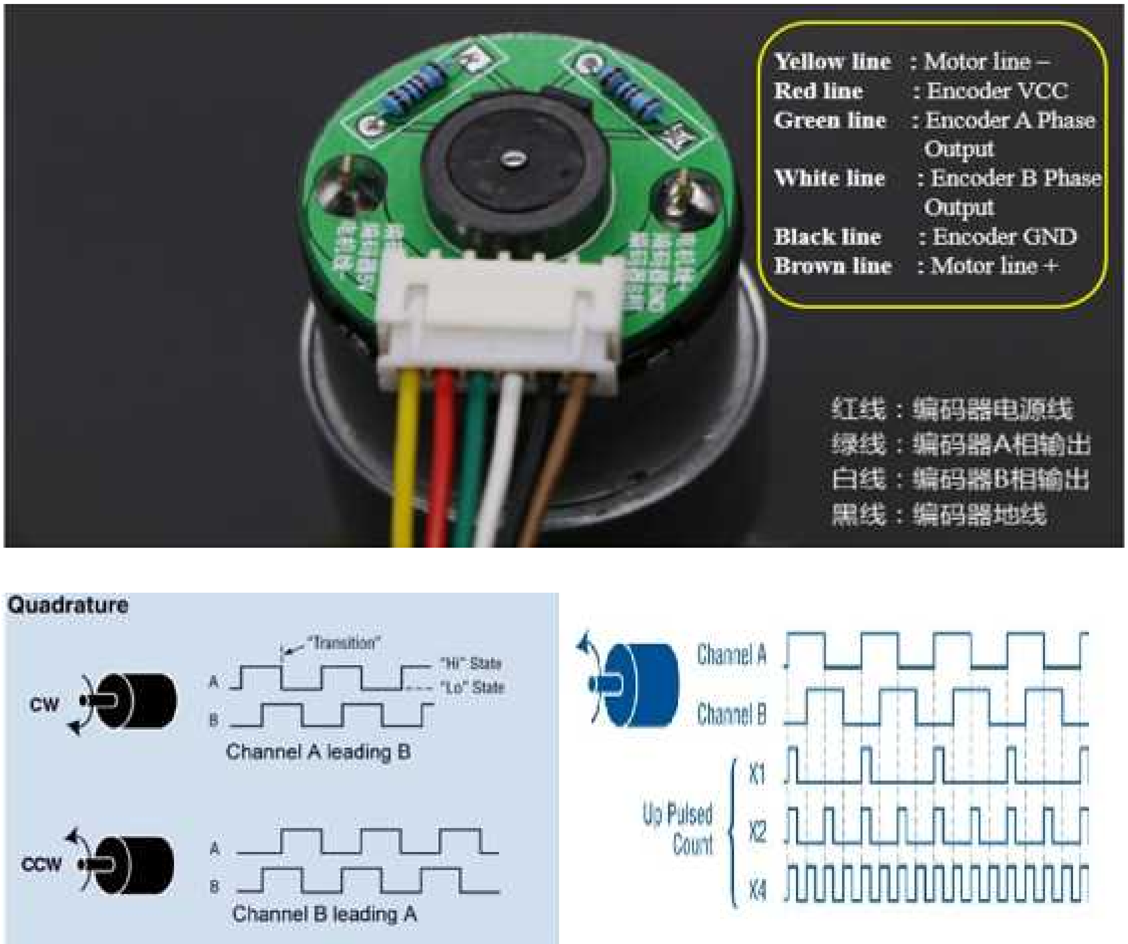


Figure 4-2-4 Quadrature encoder on DC geared motor and how to work

The quadrature encoder contained two channels output usually denoted Channel A and Channel B. The pulses in Channel B are coded ninety degree shifted as compared to Channel A. Omni-wheel main board can be determined the direction of movement by using phase relationship between Channel A and B. If the pulses in Channel A leading Channel B which rotates on clockwise. In contrast, it rotates on counter clockwise. Furthermore, in general approach, it only detects raising or falling edges of Channel A (or Channel B) to calculate the number of pulses per revolution. However, it still lack of accuracy. Therefore, in order to enhance the encoder accuracy, quadruple frequency approach is used. It detects the both Channel A and B's raising and falling edges, so that it calculates extra 4 times for each pulses. The encoder will output 390 pulses per revolution to Omni-wheel main board.

Omni Directional Wheel

Figure 4-2-5 Omni directional wheel

Diameter	60 mm
Axial width	25 mm
Wheel Material	Aluminium Alloy
Roller Material	Rubber
Roller diameter	13 mm
Net Weight	62 g
Load Capacity	3.5 kg

Table 4-2-3 Specification of Omni directional wheel

Omni directional wheels are used in this project. Its wheel is made up by two plate aluminium alloy omni-wheel that sticked together. This unique design made the mobile robot car able to roll or move in two directions, horizontal and vertical direction. Besides, these wheels became more powerful as compared to general wheel that controlled only in one direction. It also able to attach to the motor as shows on the figure 4-2-5 above.

DC Geared Motor Mounted Quadrature Encoder

Figure 4-2-6 DC gear motor mounted quadrature encoder

DC geared motor mounted quadrature encoder is used in this project to control the movement of mobile robot car. Shaft on motor is used to attach with the wheel, thus shaft caused the wheel to move whenever motor is operated. In addition, quadrature encoder is used to detect the revolution of the wheel, therefore able to compute the distance travelled by mobile robot car. Table 4-2-4 shows the specification of DC geared motor.

Rated Voltage	12 V
Rated Power	4.32 W
Rated Current	360 mA
Original Speed	10000 rpm
No-load speed	330±10 rpm
Reduction Ratio	1 : 30
Output Shaft	Diameter 6 mm

Table 4-2-4 Specification of DC geared motor

Camera Holder

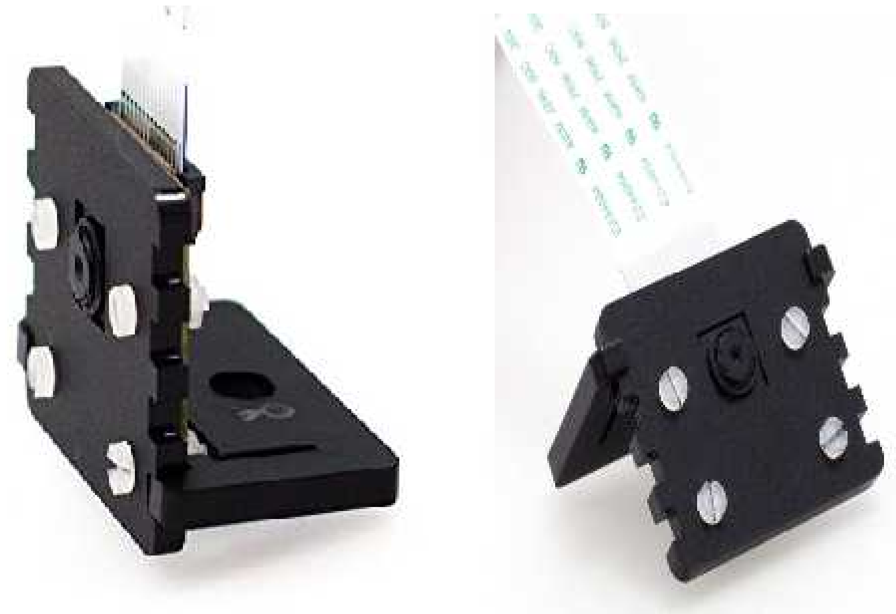


Figure 4-2-7 Camera holder

The Raspberry Pi Camera is used in this project to capture image. Therefore, camera holder is fixed the position of Raspberry Pi Camera. This is to ensure image captured clearly instead of blur by reducing the shake action while mobile robot car travelling. Camera holder also able to adjust the angle of Raspberry Pi Camera in order to widen or narrow down the view of image.

USB to TTL (UART Module) Converter

Figure 4-2-8 USB to TTL (UART Module) Converter

USB to TTL converter provides a serial port connection between host computer and microcontroller, therefore it can be used for debugging on host computer. Microcontroller contained UART port can connect to the converter in order to transmit the bytes or bits data to host computer for further analysis and debugging. Before it can transmit data to host computer, CH340G driver must install on host computer first, so that host computer can recognize it.

150Mbps Wireless N Nano USB Adapter (TL-WN725N)

Figure 4-2-9 150Mbps Wireless N Nano USB Adapter (TL-WN725N)

In Raspberry Pi 2 Model B, there is no built-in WiFi inside. Therefore, External Wireless N Nano USB Adapter is selected to use in this project to connect to internet. This adapter is smaller in size and possessed a high-performance in wireless speed. Table 4-2-5 shows the specifications of Wireless N Nano USB Adapter.

Interface	USB 2.0
Antenna	Internal antenna
Wireless Standards	IEEE 802.11b, IEEE 802.11g, IEEE 802.11n
Frequency	2.4000 to 2.4835GHz
Wireless Security	Supports 64/128 WEP, WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES), supports IEEE 802.1X
Weight	2.1g
Dimensions	18.6 x 15 x 7.1mm

Table 4-2-5 Specification of 150Mbps Wireless N Nano USB Adapter (TL-WN725N)

Pineng Power bank (PN 958) 10000mAh

Figure 4-2-10 Pineng Power bank 10000mAh

Power bank is used in this project is to act as power supply for Raspberry Pi. It outputs 5V with 2.1A that is sufficient to operate the Raspberry Pi connected via Micro USB. It is a Li-polymer battery and has 10000mAh for battery capacity. Table 4-2-6 shows the specification of Pineng Power Bank.

Capacity	10000 mAh
Input	5V 2A (Micro USB)
Output	5V 2.1A and 5V 1A
Battery Type	Lithium Polymer
Working Temperature	-10 to 45 degrees Celsius
Indicator Light for Battery Usage	Yes
Dimension	150.5 x 77 x 11.8 mm
Net Weight	224 g

Table 4-2-6 Specification of pineng power bank

Rechargeable Lithium Battery

Figure 4-2-11 Rechargeable Lithium Battery

The rechargeable lithium battery is used as power source to supply the DC geared motor. Since the rated voltage for the DC geared motor is 12V, thus it can output 11.1V and sufficient to operate the motor. It has 3000mAh for the battery capacity, discharge rate is 25C and rechargeable capability.

Acrylic Chassis

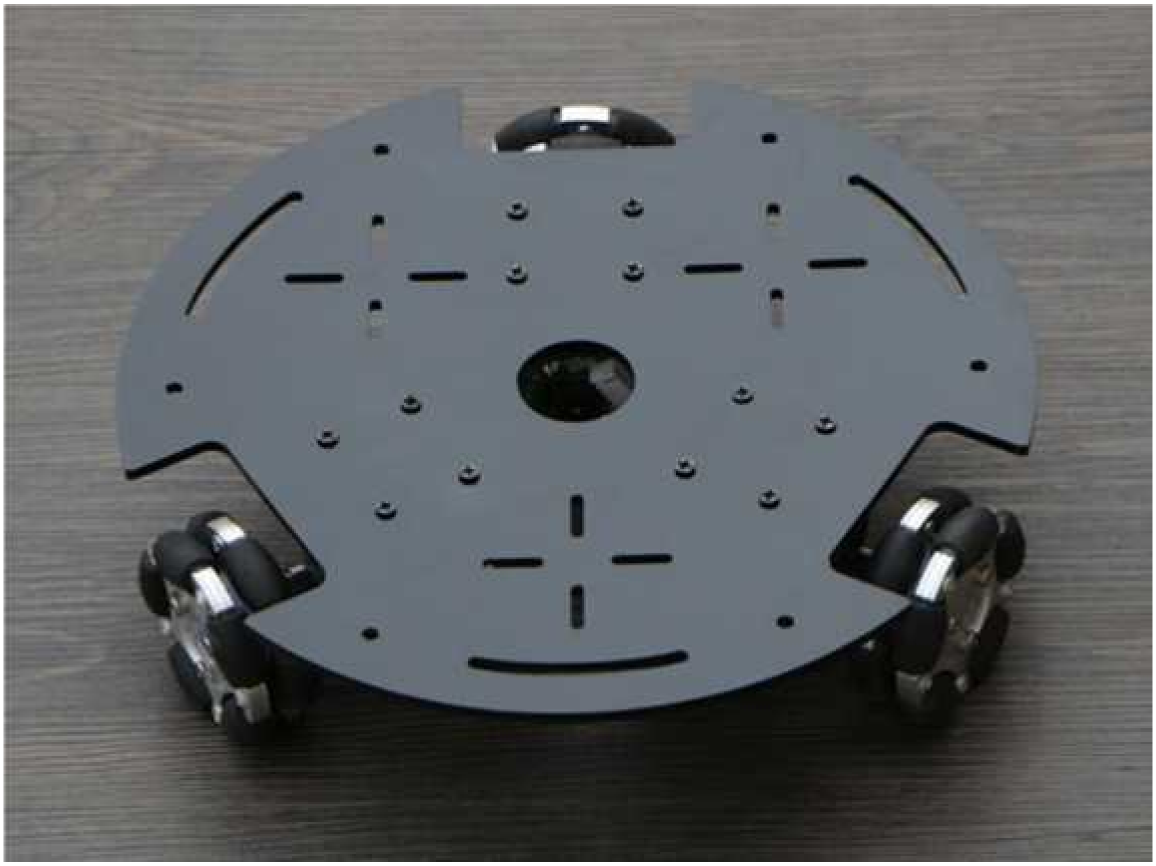


Figure 4-2-12 Acrylic Chassis

The acrylic chassis is adopted to house the hardware components. This chassis is designed to possess holes, it is allowed to lock the position of hardware components by using screw. Besides, it also possessed the extendibility with second or third level of acrylic chassis. In this project, two level of acrylic chassis is used to house the hardware components and fixed the position of the wheels.

DC Geared Motor Bracket

Figure 4-2-13 DC geared motor bracket

In this project, DC geared motor bracket is used to hold the DC geared motor and fix the position on acrylic chassis through using screws and nuts. Thus, it enables the movement of mobile robot car.

4-2-2 Software tools

Python IDLE 2.7.9

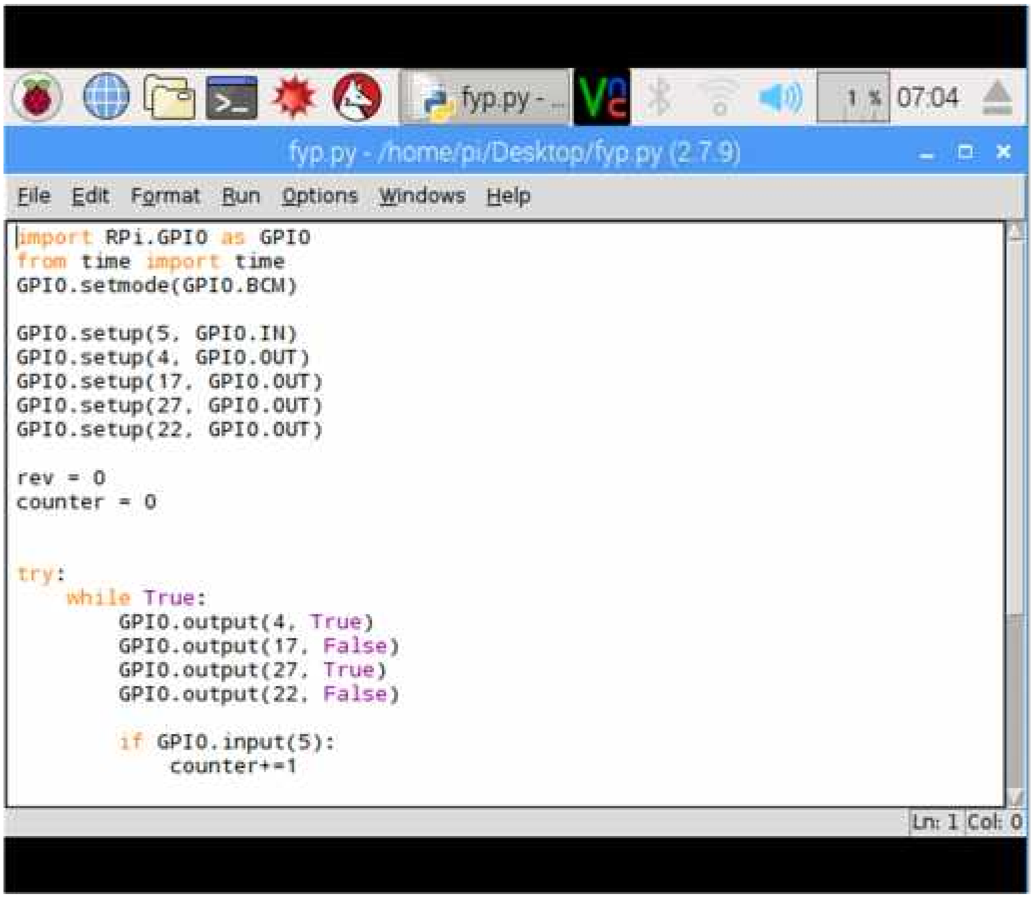


Figure 4-2-14 Interface of Python IDLE 2.7.9

Python IDLE 2.7.9 is a software where allowed to write the code by using python programming languages. It is supported by Raspberry Pi 2 Model B. Thus, it is selected to program into mobile robot car.

MiniBalance V3.5

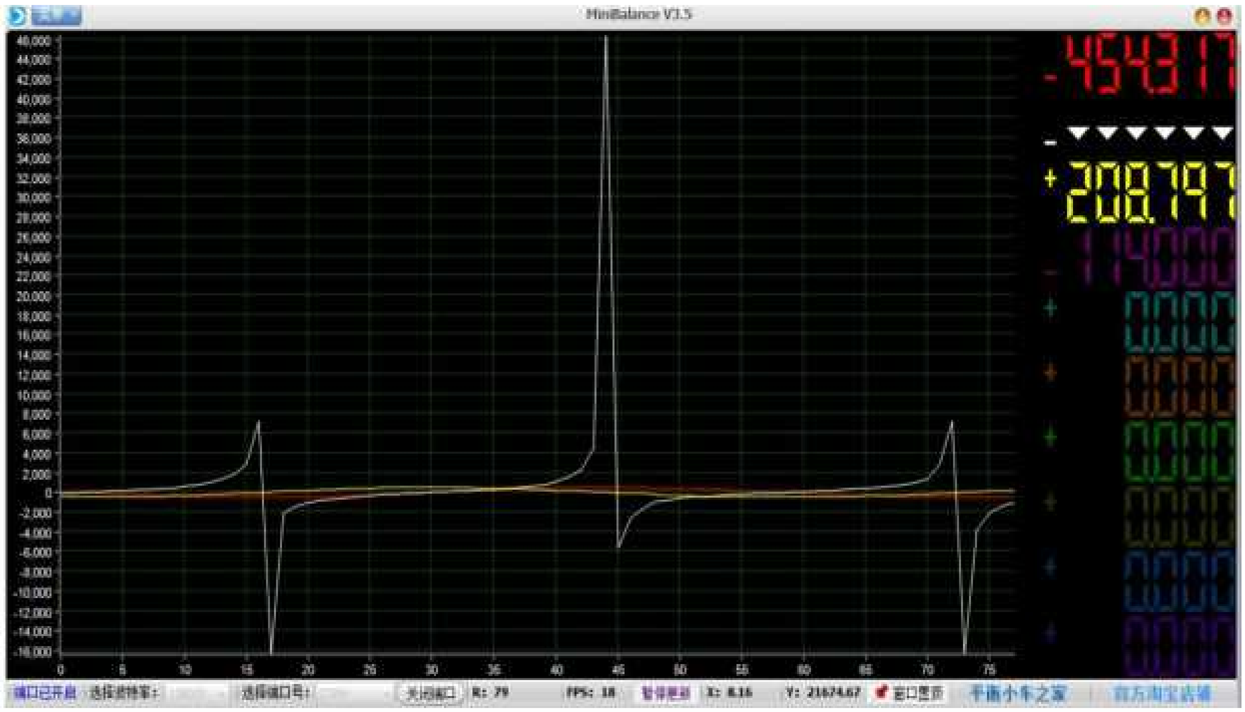


Figure 4-2-15 Interface of MiniBalance V3.5

MiniBalance is a window form application that for data analysis and debugging purpose. It is free installation and right click as administrator to run it. It has to set the baud rate to 128000 for ready to receive data transferred from microcontroller. In speed mode, MiniBalance is displayed target and real-time speed for three motors. Whereas in position mode, it is to display target and real-time position for three motors. Therefore, by setting the value of PWM, it able to observe the real-time speed for each motor on MiniBalance window. Once speed of motor retrieved, the distance travelled by mobile robot car can computed also.

MySQL database

In this project, a database system is required to support mobile robot car monitoring system on web. Therefore, MySQL is selected as a database system because is an open source database that ease to use, reliable and high performance. It is required to run on a server. Besides, MySQL also used the Structured Query Language (SQL) to create the table, perform query and so on. Through the SQL, data from the Raspberry Pi is able to store into database and retrieve it whenever data is needed.

PHP

PHP is an acronym for "Hypertext Preprocessor". It is a severer-side scripting language and making for dynamic and interactive web pages. PHP scripts are only executed on the server. PHP also is an open source and selected it to use in this project. It is used to interact with database. For example, PHP script query the MySQL database to retrieve the location point and post it to HTML for displaying purpose. The PHP not only to read the data, but also allowed to delete, modify data in database on server.

PHPMyAdmin

For convenience purpose, PHPMyAdmin is selected to use in this project. It is a free software tool written in PHP to handle the administration of MySQL over Web. It has user interface that allowed to perform all operation like SQL such as create, delete, add and so on to MySQL database. So that, it is making the management for database became easier as compared by writing SQL statement.

Apache HTTP Server

Apache is an HTTP server that to serve the web pages. Since in this project, HTML and dynamic web pages using PHP scripting language are designed for the mobile robot car monitoring system, thus it required a server that to serve both web page that is Apache. This Apache HTTP server has been installed on Raspberry Pi.

4-3 Requirement

The table 4-3-1 shows the components are required for this project.

Components	**Number of components**
Battery (11.1V)	1
Power Bank (5V, 2A)	1
DC Geared Motor Mounted Quadrature Encoder	3
External USB WIFI Adapter	1
Omni Directional Wheel (60mm)	3
Raspberry Pi 2 Model B	1
Omni-Wheel Main Board	1
Pi Camera Module	1
Camera Holder	1
B3 Charger	1
Micro USB cable	1
Acrylic Chassis with holes (2 Layers)	2
M3*10 Copper Pillar	4
M3*60 Copper Pillar	6
Steel Motor Bracket	3
Screw For Copper Pillar	20
Screw For Bracket	30
Nut For Bracket	30

Table 4-3-1 Requirement for required components in this project

Requirement for Raspberry Pi 2 Model B

Operating Voltage	5V and 2A
Operating Temperature	-40 to 85 degrees Celsius
Operating System	Raspbian
Minimum SD Card Memory Storage	8GB recommended (4GB is not advisable as OS required around 3 to 4GB)
Communication Protocol	UART
Baud Rate	115200bps

Table 4-3-2 Requirement for Raspberry Pi 2 Model B

Requirement for External USB WIFI Adapter

Module	TP Link TL-WN725N 150Mbps Wireless N Nano USB Adapter
Connection	Connect via USB port
Driver	TL-WN725V V2 (Linux OS)

Table 4-3-3 Requirement for External USB WIFI Adapter

Requirement for Omni-Wheel Main Board

Operating Voltage	11.5 to 12V
5V Pin	Output 5V but not over 800mA load
3.3V Pin	Output 3.3V but not over 200mA load
Communication Protocol	UART
Baud Rate	115200bps

Table 4-3-4 Requirement for Omni-wheel main board

Requirement for DC Geared Motor

Rated Voltage	12V
Rated Current	360mA
Speed Control	PWM (10KHz)

Table 4-3-5 Requirement for Dc geared motor

Requirement for Quadrature Encoder

Operating Voltage	5V

Table 4-3-6 Requirement for quadrature encoder

Requirement for Rechargeable Lithium Battery

Voltage	11.1 – 12V

Table 4-3-7 Requirement for rechargeable lithium battery

Chapter 5 Implementation And Testing

5.1 Experiment

5.1.1 Experiment On Before And After Adjustment On Each Correction Point

Some of the experiments were conducted to ensure the result of this project. The target of this project is to allow mobile robot car to travel from starting correction point until destination. However, the amount of times that able the mobile robot car to travel until destination which has to concern about. Therefore, the first experiment that carried out was before and after introducing the adjustment on each correction point, the distance between each correction point was 60 cm. In the experiment, a fixed path was selected to allow the mobile robot car to travel, it required to travel from starting point until destination and return back. After that recorded down the amount of times that mobile robot car able to travel with and without position adjustment on each correction point. Below figure shows the selected path used in experiment.

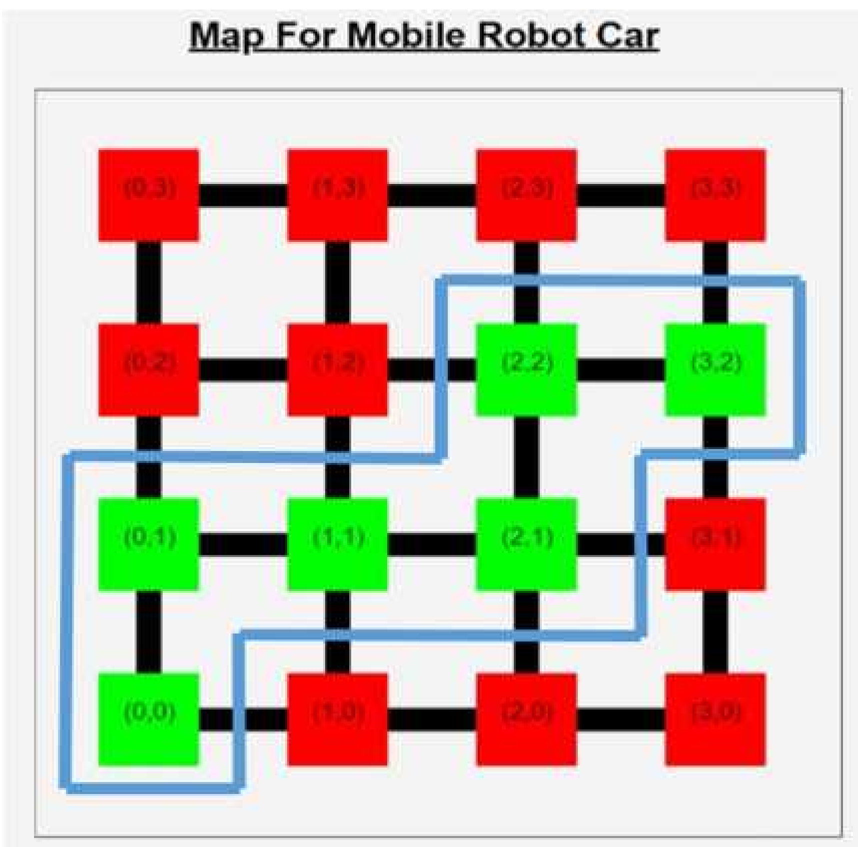


Figure 5-1-1 Selected path used in experiment

Since the mobile robot car stops at each correction point, this caused the position of mobile robot car shifted due to friction problem. The shifted position was called an error. This error was accumulated along the pathway. Thus, by introducing the position adjustment for mobile robot car on each correction point before it move to next. The accumulated error was reduced and improved the performance of mobile robot car especially the amount of times for mobile robot car to travel. The table below shows before and after the adjustment on mobile robot car.

	Amount of times to travel	**Accuracy**
Before adjustment	1	-
After adjustment	More than 10 times	10 times more accurate as compared to without adjustment

Table 5-1-1 Result of before and after adjustment on each correction point

After the position adjustment for mobile robot car on each correction, the amount of times to travel was increased sharply which was mobile robot car able to travel more than 10 times without out of track as compare to the without adjustment. Besides, the error accumulated along the pathway also reduced due to adjustment. From the experiment, the position adjustment on each correction point that was a key to improve the performance of mobile robot car particularly in amount of times to travel.

5.1.2 Experiment On Speed Of Mobile Robot Car

In this experiment, it was to test with various speed for mobile robot that suitable to adapt the position adjustment algorithm applied. Therefore, the track was used back as same as previous experiment, but just changed the speed of mobile robot car to test it. Table 5-1-2 shows the speed and result of adaptability for position adjustment.

Speed	20 cm/s	25 cm/s	30 cm/s	35 cm/s	40 cm/s
Adaptability of position adjustment	Yes	Yes	Yes	Yes	No

Table 5-1-2 Speed and result of adaptability of position adjustment

From the experiment, it can be concluded the maximum speed of mobile robot car used to adapt the position adjustment was 35 cm/s. In another way to say that was the speed below 35 cm/s that suitable for mobile robot car to adapt position adjustment. This is because the speed of mobile robot car and surface friction have the indirectly relationship. The surface friction is a force that affects the motion of mobile robot car, whereas the speed that affects the speed of spinning the wheel on mobile robot car. Due to the wheel of mobile robot car touched on the surface, the speed of mobile robot car against the surface friction will affects the motion of mobile robot car. Thus, whenever the speed of mobile robot car increased, the position of mobile robot car is shifted more and caused the error accumulated while travelling, eventually the position adjustment failed to apply. In contrast, if decreased the speed of mobile robot, the error accumulated become lesser, the position adjustment able to apply.

Chapter 6 Conclusion

In conclusion, the problems existed on mobile robot car that used the old-line following method are encountered by factory and caused to reduce the productivity and efficiency. These problems such are the guided line track eliminated gradually, low accuracy of tracking method and high error rate accumulated while travelling. Thus, it will lead to a poor performance of the mobile robot car especially in movement response. This also a reason to develop this project to further improvement from that by introducing new method which is QR code as correction point method. This project is separated into Final Year Project 1 (FYP1) and Final Year Project (FYP2) to complete.

In FYP1, all the hardware components that required for the project was bought and being tested with some basics coding to ensure the hardware components are functioning and working properly. Some of the objectives have been achieved which were developed a four wheel mobile robot car, applied straight algorithm to mobile robot car, and computed the distance travelled between two correction points. However, four wheel mobile robot car was found out that difficulty to control the direction of movement such as turn left, right and certain angle. Therefore, Omni-wheel mobile robot car was selected to replace the four wheel mobile robot car. This is because Omni-wheel mobile robot car is much easier to control the direction as compared four wheel mobile robot car. The Omni-wheel mobile robot car made to save cost due to three wheels and motors to be operated. So, Omni-wheel mobile robot will be focused on FYP2 in order to complete the project.

In FYP2, the project is continued with the new Omni-wheel mobile robot car. Therefore, the first objective which three wheeled omni-directional mobile robot car is capable to keep track of the distance travelled from one correction point to another by using quadrature encoder mounted on DC geared motor is achieved. However, the error of distance measured with this approach is became larger when robot travels for long distance. In addition, the mobile robot car is able to travel from starting point to destination through capturing QR code placed on fixed position along the pathway. The camera attached on mobile robot car is able to capture QR code and decode it to retrieve its location point, at the same time allowed the mobile robot car to decide the direction to move. However, if a lot of dusk sticks on QR code symbols will make camera to capture an unclear image and lead to decode failure.

Furthermore, mobile robot car is able to perform the adjustment on each correction point while travelling to ensure its position accurately. This is because each time for mobile robot car will stops at correction point, thus its position will shift a bit depends on magnitude of friction. The larger the magnitude of friction, the more accurate for position of mobile robot car. Moreover, online database MySQL is created to store the location point of QR code in coordinate form. Therefore, whenever the mobile robot car is decode the QR code at correction point, it will automatically to insert its location into database. It able to keep track the travelled path of mobile robot car. For monitoring purpose, a user-friendly web page is designed for user. The web page is to show the path travelled of mobile robot car by retrieving data from database. This user interface allowed the user convenience to keep track on web page instead of on database.

6-1 Future Work

For further development, there are some improvement from this project can be done. The first improvement is to replace the Pi Camera with QR code scanner. The Pi Camera is required mobile robot car to stop, then took some time to capture the image and decode the QR code image with an algorithm which resulting in slow speed of mobile robot car. In contrast, for dedicated QR code scanner has a faster decoding speed as compared to Pi Camera. Thus, the mobile robot car no needs to stop and continue to move. This will improve the performance by increasing the speed of mobile robot car. Furthermore, obstacle detection can be apply in mobile robot car. The ultrasonic sensor can be installed on the mobile robot car for detection. Therefore, whenever it detects the surrounded object with a certain distance, the mobile robot car will stop. This is to avoid the collision between object and mobile robot car.

Bibliography

Raspberry Pi. (2018) Raspberry Pi – Teach, Learn, and Make with Raspberry Pi. Available at: https://www.raspberrypi.org/.

W3schools.com. (2018) PHP : MySQL Database. Available at: https://www.w3schools.com/php/php_mysql_intro.asp.

Pythonhosted.org. (2018) Short introduction – pySerial 3.0 documentation. Available at: https://pythonhosted.org/pyserial/shortintro.html#opening-serial-ports.

Raspberrypi.org. (2018) Python picamera – Raspberry Pi Documentation. Available at: https://www.raspberrypi.org/documentation/usage/camera/python/README.md.

Raspberrypi.org. (2018) Setting up an Apache Web Server on a Raspberry Pi – Raspberry Pi Documentation. Available at: https://raspberrypi.org/documentation/remote-access/web-server/apache.md.

PiHome – Smart Heating Control. (2018) Install Apache, PHP, and MySQL on a Raspberry Pi (LAMP). Available at: http://www.pihome.eu/2017/10/11/apache-php-mysql-raspberry-pi-lamp/.

W3schools.com. (2018) HTML Tutorial. Available at: https://www.w3schools.com/html/

Raspberrypi.org. (2018) The Raspberry Pi UARTs – Raspberry Pi Documentation. Available at: https://www.raspberrypi.org/documentation/configuration/uart.md

Tp-link.com. (2018) TL-WN725N 150Mbps Wireless N Nano USB Adapter. Available at:

https://www.tp-link.com/us/products/details/cat-5520_TL-WN725N.html.

M. Faisal, R. hedjar, M. Alsulaiman, K. Al-M. and H. Mathkour. (2014) Robot localization using extended kalman filter with infrared sensor .IEEE/ASE 11th International Conference on Computer System and Applications, pp. 356-360.

L. Chunag, (2012) Object localization strategy for a mobile robot using RFID, Master's Thesis, Department of Computing Science, Umeå University, Umeå, Sweden.

Shi, Zh. et al. (2015) Research on Indoor Positioning Technique Based on iBeacon Base Station (in Chinese), presented in Business and Operations.

Level, I., Tutorial, A., Dates, 2., Tests, I., Us, C., Policy, P., Use, T. and Us, A. (2017). *What is Prototype model- advantages, disadvantages and when to use it?*. [online] Istqbexamcertification.com. Available at: http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/.

Simultaneous Localization and Mapping for Mobile Robot Car

ORIGINALITY REPORT

3%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Northampton Student Paper	1%
2	Mark Edward Soper. "Expanding Your Raspberry Pi", Springer Nature, 2017 Publication	1%
3	www.justanswer.com Internet Source	1%
4	Brian Olszewski, Steven Fenton, Brian Tworek, Jiao Liang, Kumar Yelamarthi. "RFID positioning robot: An indoor navigation system", IEEE International Conference on Electro-Information Technology , EIT 2013, 2013 Publication	1%
5	www.pcforce.co.nz Internet Source	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	On		

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)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

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

Name: __________________________

Date: ___________________________

____
Signature of Co-Supervisor

Name: __________________________

Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (PERAK CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	
Student Name	
Supervisor Name	

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. ________________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ________________________ (Signature of Supervisor) Date: