RESTAURANT WAITER BOT

Nyi Nyi Myo Zin Htet
Zin Zar Zar Lwin

Assumption University
Vincent Mary School of Engineering
Thailand
March 2021

Restaurant Waiter Bot

by

Nyi Nyi Myo Zin Htet
Zin Zar Zar Lwin

A report submitted in partial fulfillment of the requirements for
the degree of Bachelor of Engineering in
Computer Engineering
and
Electrical and Electronic Engineering

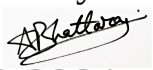
Project Advisor:
Dr. Amulya Bhattarai

Co- Project Advisor:
Mrs. Sakchhi Paudel

Examination Committee:
Dr. Jerapong Rojanarowan, Dr. Wisuwat Plodpradista,
Assoc. Prof. Dr. Jiradech Kongthon, Mr. Sunchanan Charanyananda,
Dr. Amulya Bhattarai, Mr. Ehsan Ali

Assumption University
Vincent Mary School of Engineering
Thailand
March 2021

Approved by Project Advisor:

Name: Dr.Amulya Bhattarai
Signature: 
Date: 10.03.2021

Plagiarism verified by:

Name: Mr.Ehsan Ali
Signature:
Date:

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

ABSTRACT

This project intends to design the replacement of manpower at the restaurant. In the restaurant field, there are many situations where human beings are unable to accomplish a certain task on time in the real life. By using autonomous technology, we are going to optimize the mobile bot that can be able to use in the restaurant. The main objective of the restaurant bot is the capability of serving food by using the python program. Moreover, to deliver the designated place according to the input code, the double line sensors method is used to track the place where we want to restaurant serving auto machine that would like to go to the table. Additionally, the use of double line sensors is the ability of stability speed and less friction. The robot is driven by DC Motors to control the movement of the wheels. The Raspberry Pi 3 Microcontroller will be used to perform the whole processes. Motor driver controller control the speed of the motors to travel along the line smoothly.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Acknowledgement

We would like to special thanks to our advisor Dr. Amulya Bhattarai who gave us advice from the selection of project topic till its accomplished project, for providing throughout the research and testing program and we must be more relevant my project finish on time.

Moreover, we would like to thanks to Mrs. Sakchhi Paudel for helping our project and giving feedback for project report.

Our sincere thanks to Mr.Eshan Ali for giving your valuable time to guide our report once a week and help us the useful suggestions, comments for the project report to be more perfect.

We also thank to our parents for being financially support and encouragement in every aspect.

Sincerely, this completion of our project would not have been possible without the support from all our teachers and friends.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Contents

Chapter 1 : Introduction 7
1.1 Overview 7
1.2 The Bot 7
1.3 Application of the Restaurant Bot 7
1.4 Objective 8
Chapter 2 : Background Study 9
2.1 Analysis on waiter robot use at MK restaurants 9
2.1 Data Acquisition 10
2.2 Data Processing 10
2.3 Actuation 10
2.3 Line Follower 11
2.4 Turning Process. 11
2.4.1 Go Forward. 12
2.4.2 Turn Left 12
2.4.3 Turn Right 13
2.5 Avoiding Obstacle Feature 13
2.6 Distance Calculation between Ultrasonic Sensor and Object 14
2.7 DC Motor - Gear Motor 15
Chapter 3 : Hardware and software development 16
3.1 Project Setup 16
3.1.1 Ultrasonic Sensor (HC-SR04) 16
3.1.2 Raspberry Pi 3 Model B+ Overview 17
3.1.3 Motor Driver (Dual H- Bridge L298N) 18
3.1.4 Power Supply (11.4 V, 2600mAh, High Discharge Battery) 18
3.1.5 Car Chassis 19
3.1.6 Infrared Sensor 19
3.1.7 Wheel 20
3.1.8 Python Code 20
3.2 Ordering System Design 22
3.3 Overview system diagram 23
3.4 System schematic diagram 24
3.4.1 Detailed Connection 24
3.5 Flowchart of ultrasonic sensor detecting object 25
3.6 Infrared sensor flowchart 26

Chapter 4 : Progress and Results 29
4.1 Ultrasonic sensor coding progress 29
4.2 Line sensor progress 30
4.3 Motor driving test 31
4.3.1 Motor Code Testing 31
4.3 Waiter Bot testing 33
4.4 Designing and Actual Structure 36
4.4.1 Assembly of the Hardware 36
Chapter 5 : Problem and conclusion 41
5.1 Issues and Problems 41
5.2 Conclusion 41
Bibliography 42

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Chapter 1 : Introduction

1.1 Overview

This module project was to combine with the Artificial Intelligence technology. Serving system which tends to enhance the productivity level in the restaurant industry. Robots can work in 24 hours continuously without feeling tired unlike being human that confined to certain time. But there are no bot that are able to function perfectly and are still making error. The bot is absolutely integrating to make less errors in real-world and enable to work more efficiently than human's ability.

Waiter operates the order to go to the desired table by following the marked line which determines with the line sensors and records the order data, and then continuously returns to the service counter after the machine has accomplished its tasks. The Raspberry Pi 3 is to implement the motor driver controller to stabilize the induced voltage to the motor. In addition, the motor driver adjusts the rotation angle by controlling the speed of motor and it proceed the operating data with PWM (Pulse Width Modulation).

This project tries to implement a Raspberry Pi3 controller on autonomous waiter bot to see whether the robot performs efficiently. This waiter bot set up the four infrared sensors that perform a line-tracking module, where they will track the route from black tape. This is the source where the Pi3 microcontroller was implemented, the bot would be able to follow the black tape effectively and moving along the line with less friction.

1.2 The Bot

Basically, the bot is classified into three sub-systems. They are optimization that install sensors which operate the sense of environment condition, processor which works as the memory tasks , and motor controller system which generates as the movement of the legs. Likewise, robots that can perform automated tasks in any field with no human guidance. Certainly, the ability of this performance task could be faster than the human tasks. Moreover, bot can track the location place by obeying the program tasks. In addition, bot can ensure to implement the system without human intervention.

1.3 Application of the Restaurant Bot

The automated robot system has attracted the attention of the food and beverage (F&B) industry, in part due to the lack of human resources; failure to handle this risk situations always. Therefore, the design concept is basically the needs of food and beverage industry to reduce the rate of manpower but not purpose to replace it. Humans are essential for personal relationship purpose, to fulfill the customers' satisfaction and to make their perfect dining. By Analyzing the variety of research on smart autonomous robots, it still has an unsolved issue for such robots to work in a real-world environment of a restaurant.

Restaurant bots propose to minimize human labor because it can work faster than human in a real world. The bot is not necessary to pay monthly fees except checking its efficiency regularly and fixing the program errors. Moreover, restaurant bot doesn't need to have employee's rest time though it needs battery charging during the specific time. After that,

Unlike the human, it can perform the completed work with high efficiency. Moreover, it will not stop or slow as the design program until the task is accomplished.

Restaurant bot was controlled by the Raspberry Pi 3 for the whole processes. The main process was to receive the table number command from the customers and start to leave from the kitchen to the customer table by following the marked line on the floor. Moreover, it returned to the kitchen (original place) while it was waiting for the next command order.

We are eager to believe regarding the implementation of robot to aid the moving and carrying of heavy loads, restaurant owners can alter their issues into solving other problems.

1.4 Objective

Project Objectives of our project focus on:

- The waiter machine is capability of receiving the order information from the kitchen and a destination which is selected by the program. And then it can self -navigate the marked path toward its destination (customer table). After completing the non-repetitive tasks, it will always return to the specified place.
- To reach out the designated table, the machine is following the marked line on the floor and going through the corresponding table. Thus, we should consider on different approaches as well as perception, mapping, and localization. For detecting the path, the infrared sensors are used by decoding/encoding the algorithm.
- Perhaps, we must predict in advance any possibility that are the ways to safeguard themselves how it stimulates the problems on some occasion. For instance, when the object is blocking in front of the waiter machine, it would respond the blocked object and calculate the distance of the object from the machine and have to optimize how it will be stopped or continuously moved. The Ultrasonic sensor will use to detect any barrier in front of the waiter machine. It must read the distance of the obstacle and stop until the object disappear.
- It should be executed by the Raspberry Pi3 program and the effective design stage would be innovated. It should be delivered to the desired table within the specific period.
- The waiter machine is imperative to focus on facing the less errors concerning with the restaurant order problems such that it must be able to use it proficiently in restaurant workplace.

Chapter 2 : Background Study

2.1 Analysis on waiter robot use at MK restaurants

There have been significant improvements in technologies over the past few decades. The technologies have been improved significantly that they can be seen in homes as vacuum cleaning robots and as transformation of goods and equipment in many industries such as car production industry, hospital, and storage facilities.

There has been a lot of development over the year but one of the interesting projects done within the Thailand was the project “Restaurant Service Robots Development in Thailand” by Akkharaphong Eksiri and Tetsuya Kimura. The project was done in collaboration between Bangkok University and famous restaurant MK Restaurants Group Company Limited. Two type of robots were developed the first one for taking orders from customers and the second one for delivering food to the table. This experiment was conducted for 6 months in 5 different branches of MK restaurant chains in Bangkok area from 2009 to 2012.

The robot was designed according to requirement by MK restaurants whereas our project is done for general purpose. The way the robot operate is according to as follow at first the robot is stationed at the assigned station. During operation, a professionally trained robot operator will turn on the robot. Then the operator or staff member will put the food in food container which is the chest of the robot. The operator or staff member will enter the table in the robot, which then robot automatically determined the path to move along. When arriving at the destination, the robot will greet the customers and open the container box which food is contained. Then the staff member can move the food to the table.

We can find some similarities and differences between the MK robot and our project. For our project, the robot will be in the kitchen where the cook can directly put the food on the robot and instead of having the controller inside the robot itself but instead, we used a python interface to control the robots. When input is entered the robot will automatically determine the path delivered the food and will return to the kitchen. We did not include the greeting system or humanoid interface since it will cost more, and we want the greeting and interaction to be done by the humans because the aim of our project is not to replace humans but to reduce the workload for the waiter and the staff members who are working in the restaurant.

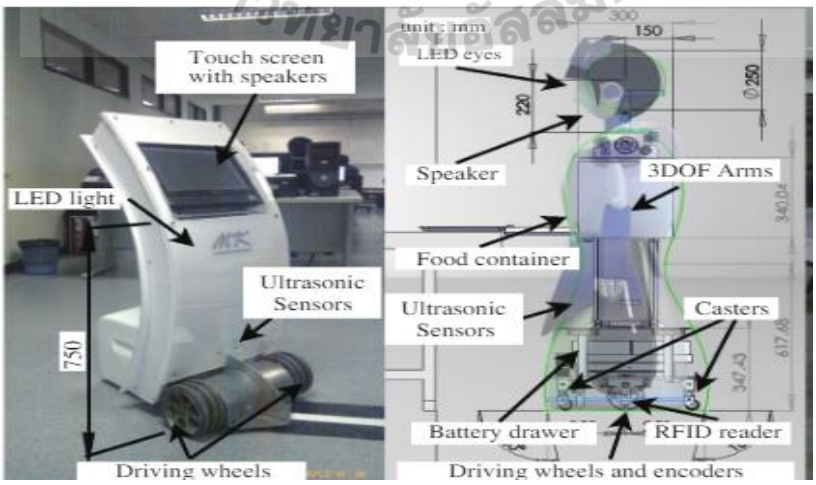


Figure 2.1 (a) Order Robot (b) Server Robot

Food ordering system for MK robot use GAMBAS2 programming for user interface for customer to order food on their order robot. For server robot they also use GAMBAS2 programming to decide the table number.

For MK robot system a master DPS microcontroller is connected to slave ones via a data bus (100kHz 12C-Bus). Robots' wheel and joints and drive wheels are connected by using PID controller in slave microcontroller. The master controller obtains a command from host computer (Mini PC) via serial communication (RS-232). A user commands the robot with a touch screen tablet through device through host computer via Wi-Fi router. Whereas in our project we use Raspberry Pi 3 as our microcontroller and, Python IDE and python programming language as programming language to control our robot.

Safety Factors are also considered, to prevent from crashing, spilling and other dangerous factors we use ultrasonic sensor to determine whether there is any obstacle near the robots, and it is carefully programmed to stop in certain distance from the obstacles. The power of the robot is minimized so that it can stop as soon as possible if any obstacle is detected..

2.1 Data Acquisition

As the data acquisition section, it would require the source data from the environment condition and transfer the specified data to the control system which part is called data processing.

2.2 Data Processing

Based on data processing, utilizing the data from the sensors by encoding the signal language and transmit to the microcontroller. After that, microcontroller will implement the program to function the actuation.

2.3 Actuation

In actuation part, the torque of the motor is start rotating so that the wheels will work depending on the command of the rotating direction from the microcontroller.

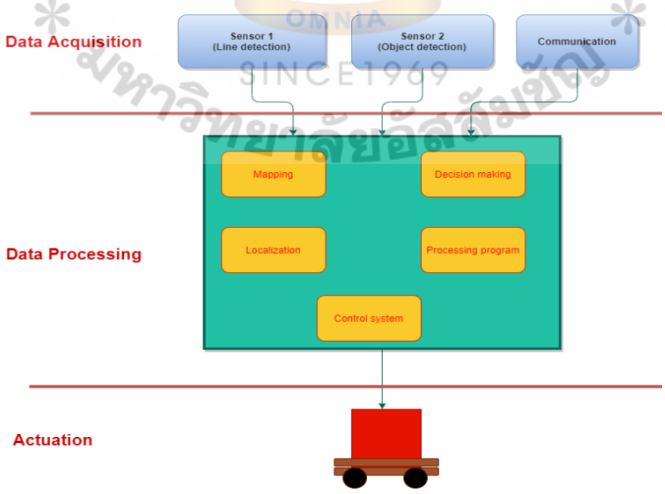


Figure 2.2 Sub-system of the Bot

2.3 Line Follower

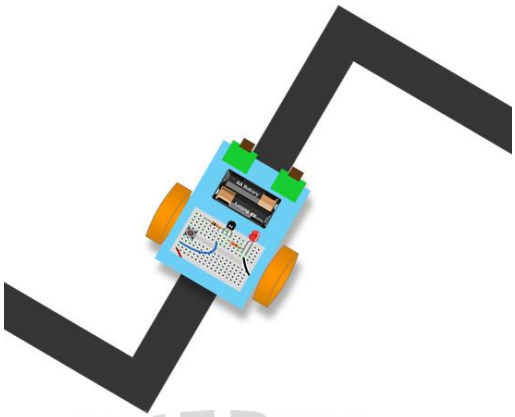


Figure 2.3 Line Follower

The bot uses 4 motors to rotate 4 wheels. And the 4 infrared sensors are placed at the bottom of the bot. The features of infrared sensor are for navigating the black tracking tape. If the sensors make sure to read the black color, the sensor output is given the data to the Pi 3 Microcontroller.

As we consider that black color is the capable of absorbing the radiation. White color or any color cannot reflect the radiation back except black color. The output signal from the sensors is an analog which rely on the light how it responds back, and this analog is determined to produce 0s and 1s, which are fed to the Raspberry Pi.

2.4 Turning Process.

For rotating the bot, we use 4 infrared sensors to determine the direction in which it goes through the right table. Therefore, the direction of the bot can be distinguished by depending on the analog output (HIGH & LOW) from two sensors.

Figure 2.4 Set up IR sensor under the front of bot

2.4.1 Go Forward.

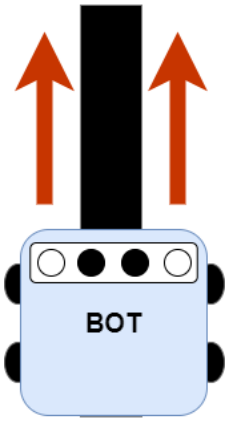


Figure 2.5 Bot Move Straight

When the two middle sensors are HIGH, we are going to move constantly in straight, at the same time, the two remaining sensors feedback are LOW.

In fact, the middle sensor will always be on the black tracking line. As the line is black, it will reflect the emitted radiation back and the response of the sensor will be low and the response of the remaining two sensors will be high as they will be on the bright surface.

2.4.2 Turn Left

Figure 2.6 Bot move left

To turn the left direction according to marked line, the left most sensor and left sensor response will be changed from LOW to HIGH as the sensor will now face the black surface. And Bot will also move left when the left most sensor, left sensor and right sensor are HIGH. Perhaps, the wheels' control changes the same process for all the turns and the bot moves continuously.

2.4.3 Turn Right

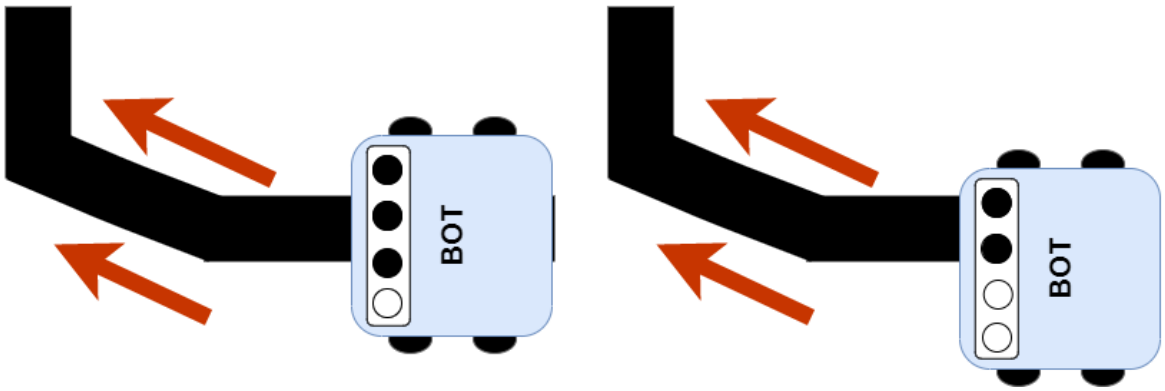


Figure 2.7 Bot turn right

When a right direction is turned on the line, the left most sensor will change LOW to HIGH while the left sensor is still HIGH such that sensors will be read the black line and the reaming right most sensor response will be LOW. If his data is achieved, the control of the wheels is changed. Like the front Left and Right wheel is rotating in clockwise direction whereas the back of Right and Left wheel is rotated in anticlockwise direction. Then the same process repeats again.

2.5 Avoiding Obstacle Feature

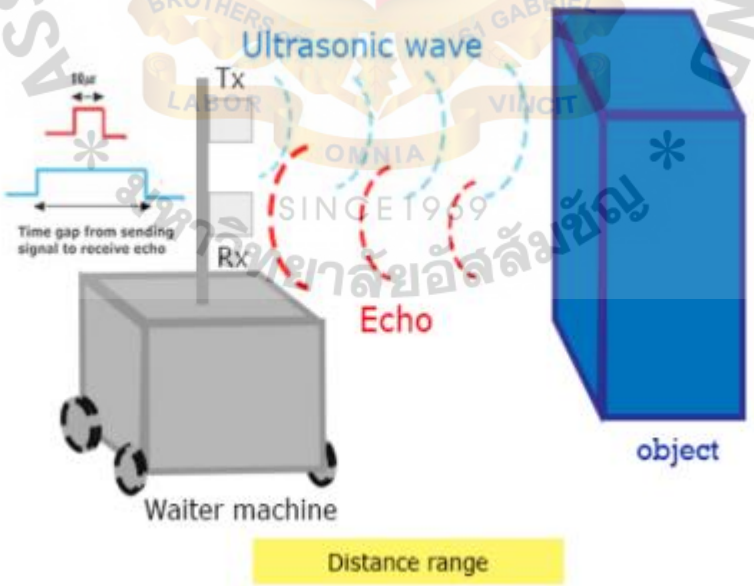


Figure 2.8 The Process of Transmitting Ultrasonic Wave and Reflecting Echo

This distance adjustable sensor works for emitting the ultrasonic wave from an transmitter toward a sensing object; the object database which results the location of the object receives as the reflected waves to the receiver. The measured distance sensor describes the pr

object location which is depending on the time required from whether the ultrasonic wave is delivered till they are received(echo) by using the speed of sound.

The sensor can read any shape of object. It does not depend on object color. If it is transparent, sensor can be detected. If the two objects have the same shape, they can be both detected with the same settings.

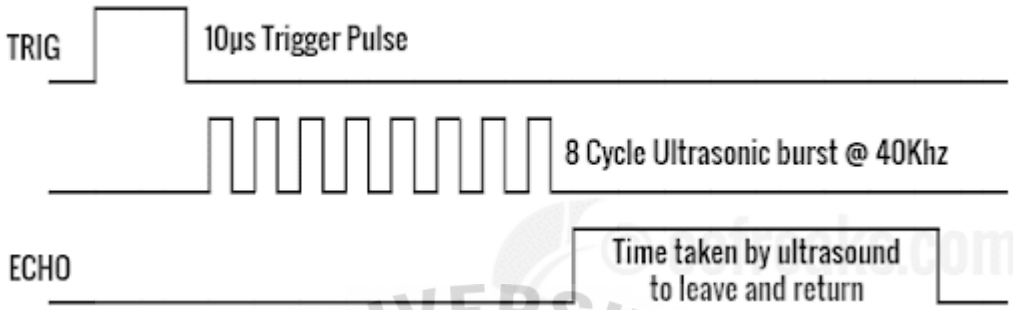


Figure 2.9 Representation of Ultrasonic burst, Trigger pulse and Echo pulse

2.6 Distance Calculation between Ultrasonic Sensor and Object

There is a formula that calculate the distance to the object's location. In the air, the speed of sound is 343 meter per second at the temperature 20◦C. To observe the speed of sound, it relies on two parameter such as temperature and relative humidity.

To obtain the actual distance from the transmitter (Tx) to the object's position, the following equation calculate the distance which is dividing by 2.

$$Distance = \frac{Time\ Taken \times speed\ of\ sound}{2}$$

ASSUMPTION UNIVERSITY OF THAILAND
OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

2.7 DC Motor - Gear Motor

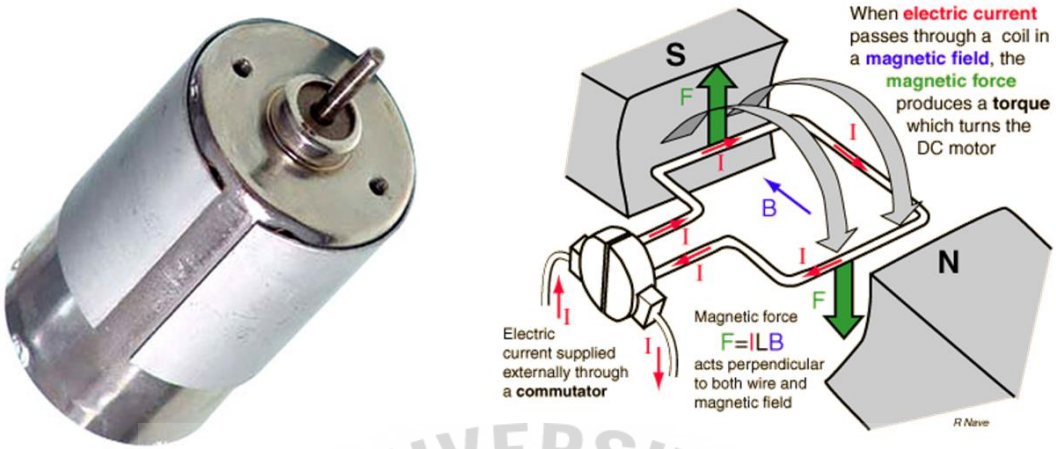


Figure 2.10 Basic Feature of DC Motor

The fundamental concept of DC motor is transformed electrical energy into mechanical energy. The shaft connects to the wheel which executes for rotating. Inside the stator, two permanent magnets are constructed which form north and south pole running through the center of the motor. Those tube magnets of opposite polarities into the motor stator to form a strong magnetic field through the rotor. The center of copper rod is called shaft to transfer mechanical energy attached to it. This shaft contains two fixed magnets on both sides which operates both a repulsive and attractive force, and torque loaf are produced. The coil winding are looping at the arm of the rotor which flow the electrical current from the power supply as well as the current pass through the coil. Moreover, each coil is positioned by 120 degree. The electromagnetic field occurs so that the polarity of magnetic field create rotation.

Magnetic loss occurs which is power loss owing to the current flowing. This term is called eddy current which swirls around inside these are generated by induced electromagnetic force. Eddy currents affect the efficiency of motor to reduce the eddy current. Although insulated disks are installed, eddy current will still flow. Perhaps the amount of eddy current become smaller, the thinner the insulated disks, the smaller the eddy currents value.

Gear motor is the combination of motor and gearbox which purpose reducing the speed of controllable motor when the torque output is increasing. To consider the type of motor for project, Gear motor are suitable for our project when we estimate the output power.

$$Pout = Torque(Nm) \times \omega$$

And In addition, the output torque must be appropriate with its torque rate in steady state. The lower the torque rating, service life of gearbox is more tolerant. The higher the torque ratings, the gear motor would not use no longer life (short service life).

Chapter 3 : Hardware and software development

3.1 Project Setup

Hardware Component

- Raspberry Pi 3 Model B+ Microcontroller
- Dual H Bridge L298N Motor Driver
- Ultrasonic Sensor (HC-SR04)
- Infrared Sensor
- 11.1V,2600mAh Lipo Battery
- 5V, Adapter
- 4 x DC Motor
- 4 x Wheels
- 1 x 470Ω Resistor
- 1 x 330Ω Resistor
- 1 x USB wire
- 2 x Car Chassis
- Screw Nut Copper Pillar
- NodeMCU esp8266

Software Component

- Python Code
- Draw.io
- Fritzing
- Blynk

3.1.1 Ultrasonic Sensor (HC-SR04)

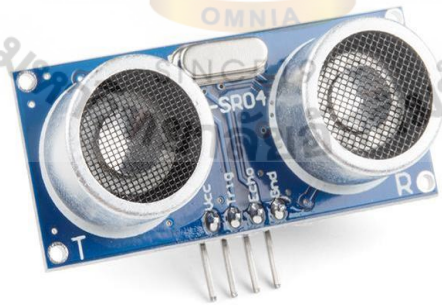


Figure 3.1 Ultrasonic Sensor (HCSR04)

To track the object's proximity, HCSR04 ultrasonic sensor is used which can transmit ultrasonic sound wave at 40kHz and the duty cycle is 10 microseconds for one period of trigger. As the process of the HCSR04 sensor is shown in above figure, when the object is existing in front of the machine, the sensor triggers the object distance by estimating the time lapse. Moreover, the acoustic 8 pulses are initiated by the ultrasonic sensor and the time lapses are started. After the echo signal are reflected from object, one period of time is

done. The result signal of the ultrasonic sensor is a high pulse which happens the time duration difference between transmitted 8 pulses and echo signal as shown in below figure.

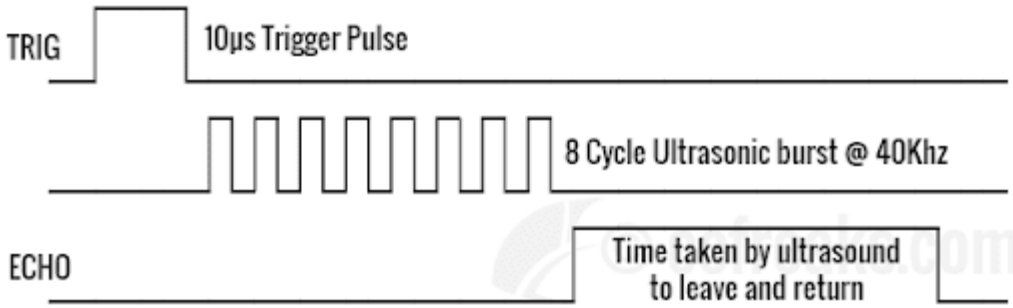


Figure 3.2 Representation of Ultrasonic burst, Trigger pulse and Echo pulse

3.1.2 Raspberry Pi 3 Model B+ Overview

Figure 3.3 Raspberry Pi 3 Model B+

Raspberry Pi is a small computer that is about a credit-card sized that can be plugged into a small monitor and standard keyboard. It can act as a budget desktop, media center, retro game console, or router and there are also thousands of different projects, where people used to build tablets, game console, laptop, phones and much more. Raspberry Pi has two models, model A and model B. The main difference between model A and model B is that model A does not have ethernet while B has ethernet port. Raspberry Pi includes a memory, CPU (central processing unit), GPU (Graphic processing unit), Ethernet port, GPIO pins, XBee socket, Power source connector and URAT.

3.1.3 Motor Driver (Dual H- Bridge L298N)

Figure 3.4 Dual H- Bridge L298N

We used the motor driver to adjust the speed of the motor and control the rotation direction. Dual H- Bridge L298N motor driver that we used is high efficiency to control the two DC motor simultaneously. To control the speed of motor, the various range of input voltages are applied to use pulse width modulation (PWM).

PWM enable to stimulate the average input voltage by delivering the HIGH- LOW pulse (duty cycle) at the very rate. The average input voltage is generally depending on the duty cycle, which is the percentage of HIGH pulse time that means the system is operated with average input voltage.

Motor driver can be used as the switching device. Therefore, motor driver is set up between Raspberry Pi and motors. It gets the input signal from the microcontroller and optimize the corresponding output for 4 motors.

3.1.4 Power Supply (11.4 V, 2600mAh, High Discharge Battery)

Figure 3.5 11.4V ,2600mAh, Lipo Battery

11.4V, 2600mAh batteries will be used as the power source of the motor driver.11.4V power supply is sufficient to power up the motor driver (LN298N) and it does not need to set up voltage regulator. The voltage battery can determine how fast your motor speed is. It means the voltage of the battery is directly proportional to the RPM of the motor. The more voltage that the motors are supplied, the speed of motor (Torque) is spinning rapidly.

However, Internal resistance of battery will change over due to the uprising temperature. We must check the battery temperature frequently because if the batteries are becoming hot, it makes the battery less efficient.

3.1.5 Car Chassis

Figure 3.6 Structure of Two layers of Car Chassis

The hardware components and Battery supply was placed on the car chassis. On the car chassis, the components are held tightly with the screw. The overall apparatus are shown in below figure.

3.1.6 Infrared Sensor

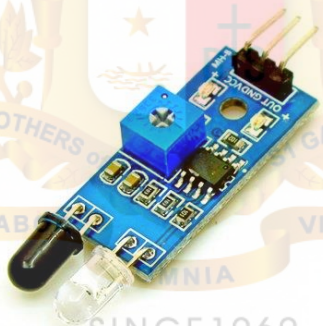


Figure 3.7 Infrared Sensor

The above figure can be explained about the principle of IR sensor. IR sensor consists of an IR LED and an IR Photodiode which we called as Photo-Coupler or Opto - Coupler if we combine them together. Furthermore, IR transmitter emits the radiation which could reach to the object and IR receiver could get the radiation which bounce back from the object. We used the infrared sensor for the detection of the line.

3.1.7 Wheel

Figure 3.8 four sets of wheels

We joined the torque of the DC motor and wheel for rotating and moving. We set up one wheel for each DC motor while rotating smoothly with less friction. As its dimension, its diameter is about 7 cm and its width is approximately about 2.7 cm.

3.1.8 Python Code

Python is one of the high-level programming languages among computer programming. Its advantage is to assist the programmer while writing clearer when it designs the program because its syntax. Moreover, it supports multiple programming paradigms, object-oriented and functional programming. Python library is also usable for machine learning.

3.1.9 NodeMCU esp8266

Figure 3.9 NodeMCUesp

This component that we installed is to connect to the wifi network and can connect directly with other device to put the input data in database. However, It requires to interact with internet. The voltage supplied is the range of 3V to 3.6 V.It can be possible to supply above 600 mA .But we already had the regulated 5V voltage source , we directly connected to this component with the Raspberry Pi3.

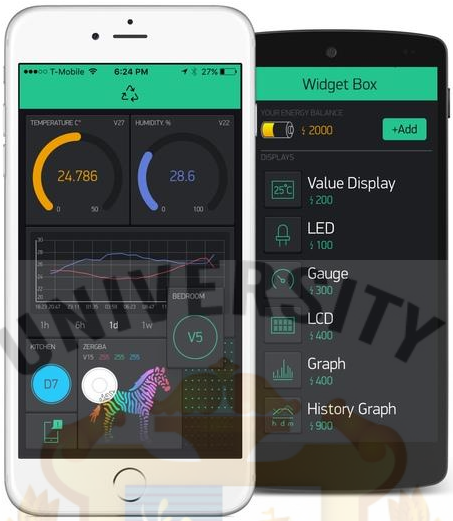


Figure 3.10 Blynk application

Blynk is the digital dashboard with iOS and Android Application which utilize to control the Raspberry Pi 3 by using the internet. It functions not only displaying the sensor data but also storing the data with the free cloud. It is the capably of controlling the hardware components easily so that we were implementing on the Blynk with the Raspberry Pi 3.

ASSUMPTION UNIVERSITY OF THAILAND
LABORE ET FIDE VINCIT
OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

3.2 Ordering System Design

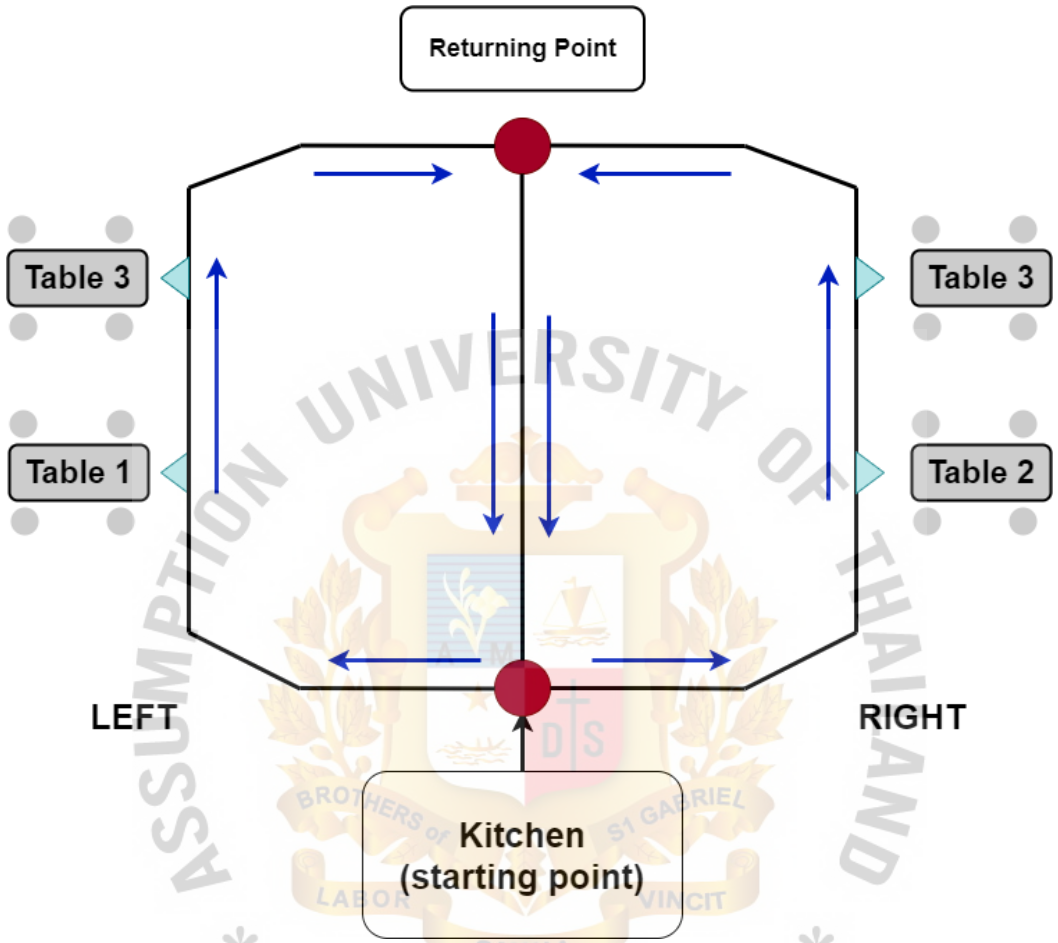


Figure 3.11 Blueprint of Restaurant

The above diagram demonstrates the processing of the machine how it generates in the restaurant. We must do the machine that goes through the table depending on the code arrangement accordingly.

Table Localization

- **Order Input from table (1) and table (3) -** As our plan, if the table (1) or (3) is calling for ordering the food, the bot is waiting for serving the food. The kitchen is determined as the starting point such that the bot starts moving from the kitchen. Then, the bot will require to turn left direction which follows the marked line and going through the table (1) or table (3). After the task is completed, it goes to the returning point and returns to the front of the kitchen (starting point) as shown in above figure 14. The process is repeated for the table (1) and (3).
- **Order Input from table (2) and (4)** – When the bot receives the input database for table (2) and (4), it starts going from the kitchen. It is obvious to turn right direction as it follows the tracking black tape. When the work has done, the bot will move to

the returning point. As the last destination, the bot is coming back to stop in front of the kitchen and waiting for the next order command.

For the repetitive process, the bot was moving with constant speed and navigating the line marked on the floor. In some condition, if the obstacle or person is blocking on the line, the ultrasonic sensor informs to the microcontroller and the machine will deliberately stop on the way of line because this will cause many possible dangerous conditions. In another ways, a person who is standing on the line will get harmful or the plate on the robot might spill.

3.3 Overview system diagram

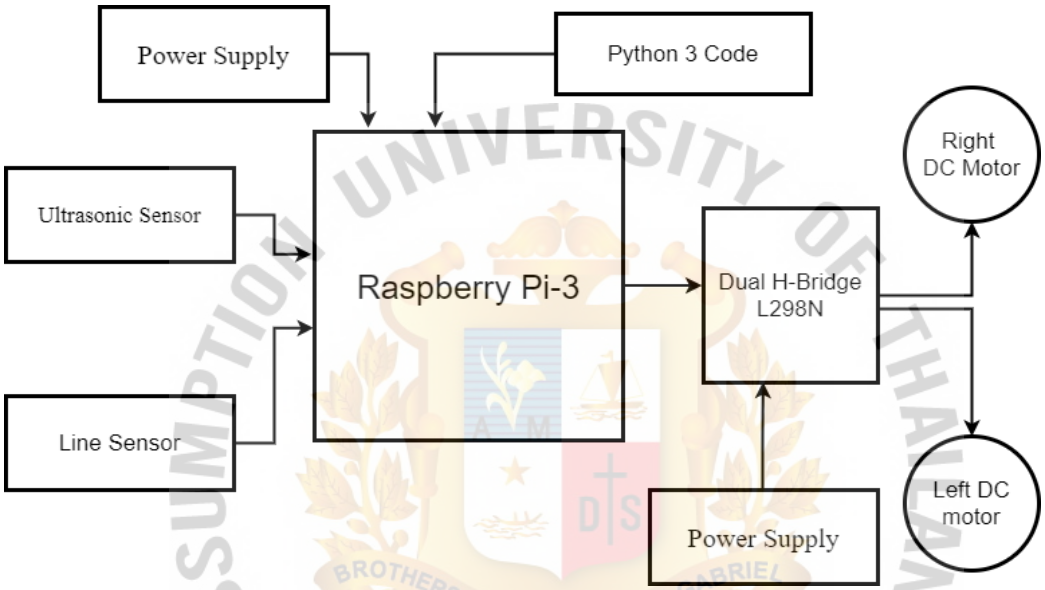


Figure 3.12 Overview System Diagram of Bot

Once we give power to the Raspberry Pi3, it will start running the python3 code and start the program. It will accept input signal from IR sensor to detect the line. Moreover, it will get input from ultrasonic sensor in case if there is an obstacle.

LiPo battery (11.4V) is supplied to L298N motor driver. The motor driver simultaneously receives the data from the Raspberry Pi3 Microcontroller. Pi3 will control the speed of motors by using L298N motor driver. The speed of left and right motor that we give from the program determine the motor will rotate clockwise or anticlockwise direction.

3.4 System schematic diagram

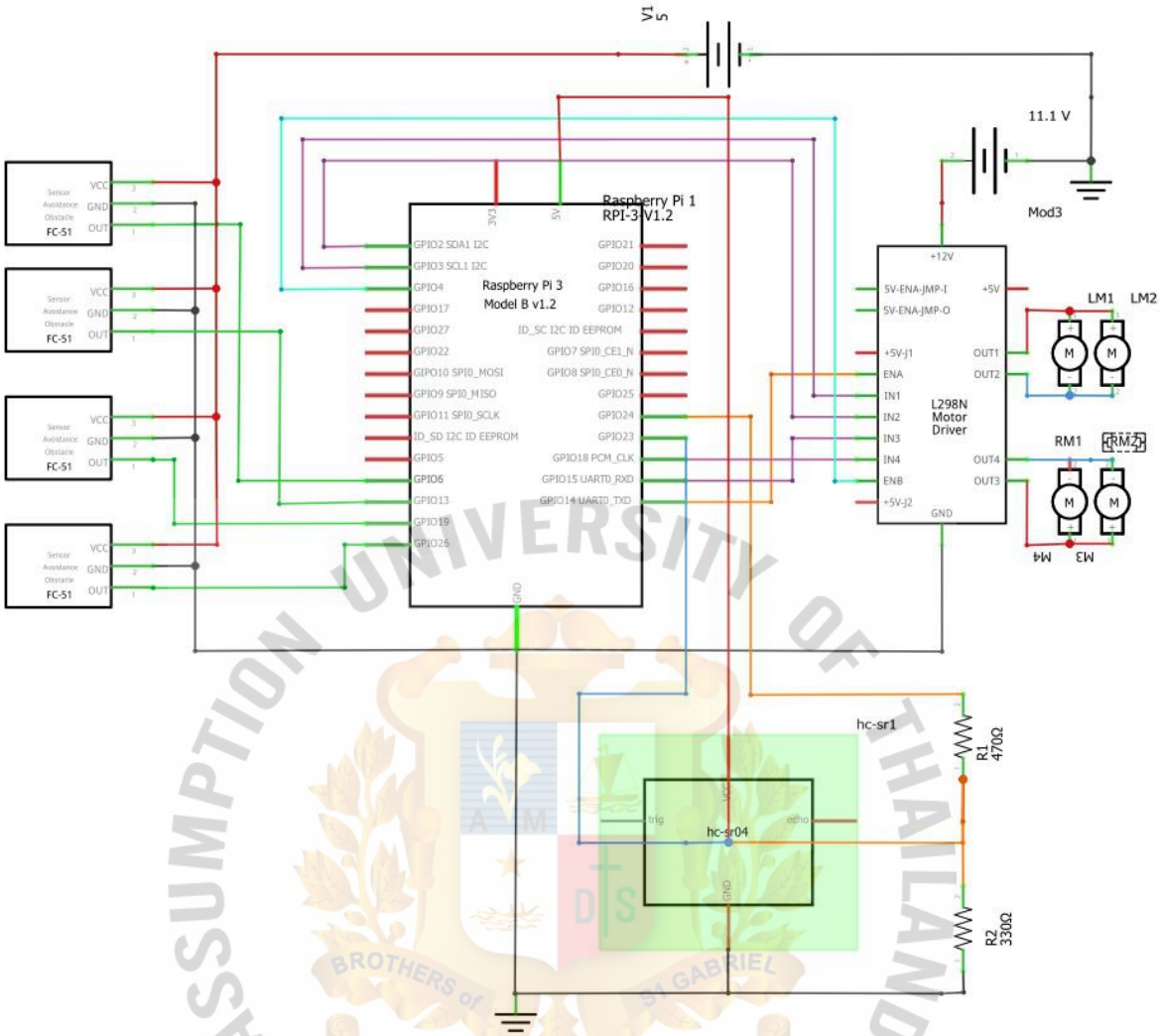


Figure 3.13 Schematic of Waiter Bot System

3.4.1 Detailed Connection.

1. We connected to each wire of two motors by using soldering iron. And we made it together for the left and right side of the wheel.
2. OUT 1 pin and OUT 2 pins of L298N were connected to the left two motors and the other OUT 3 and OUT 4 pins was pinned to the right two motors.
3. The four IR sensors was given to power supply (Vcc = 5V) and GND pin was grounded together on the breadboard.
4. The Output pins from the four IR sensor join with the Pi3 microcontroller as the input that is GPIO Pin number (6,13,19,26).
5. The ultrasonic sensor was set up on the breadboard and trigger pin was connected to the GPIO (22) and Echo pin connected to the Pi 3 pin number (23).
6. We used power bank to power up the Pi 3 microcontroller and 5 V is sufficient to supply it.
7. To observe the data from the microcontroller, four pins of GPIO (18,15,2,3) was connected to the pin number IN (1,2,3,4) of motor driver.

8. Lipo battery was supplied by 11.4V to the Motor Driver (L298N). The batteries ground wire and motor driver ground's pin were linked together to the ground.
9. Owing to the detection of black line on the floor, we used the black tape and portrayed it on the floor as our blueprint figure (18).
10. We designed the whole process by the Python code. We started running the program how it worked.

3.5 Flowchart of ultrasonic sensor detecting object

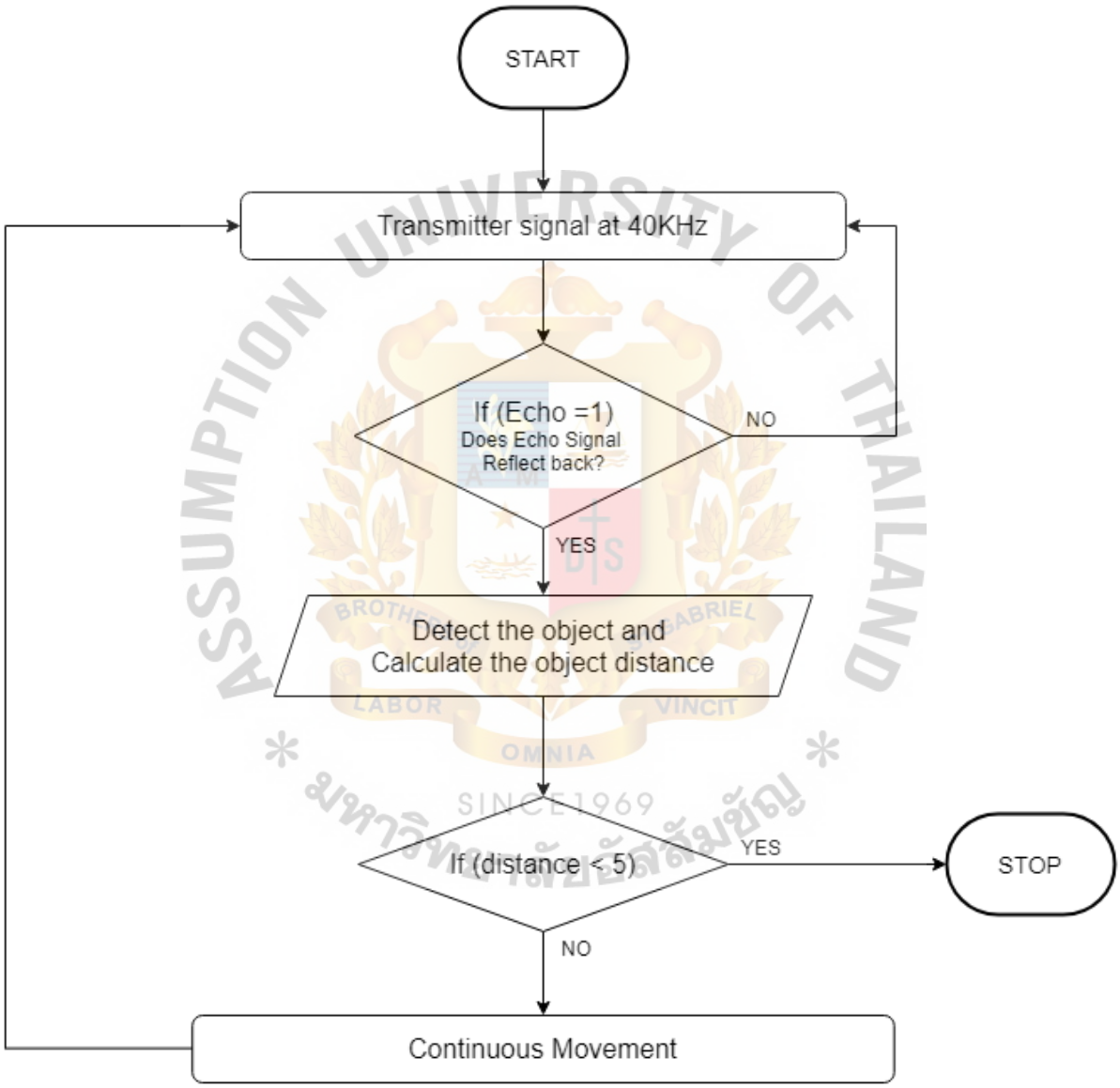


Figure 3.14 Ultrasonic Sensor Flowchart

3.6 Infrared sensor flowchart

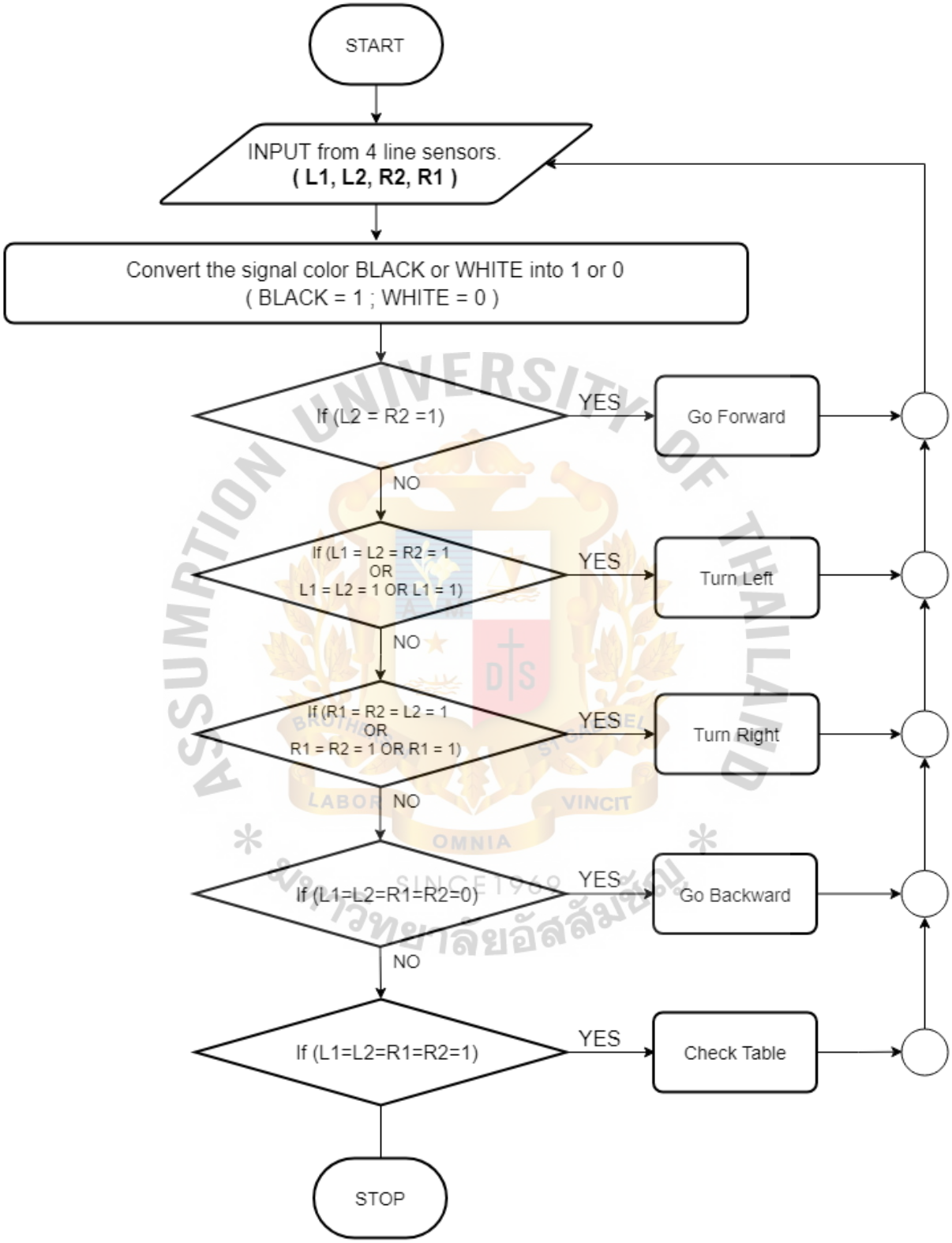


Figure 3.15 Infrared Sensor Flowchart

3.7 System Flowchart

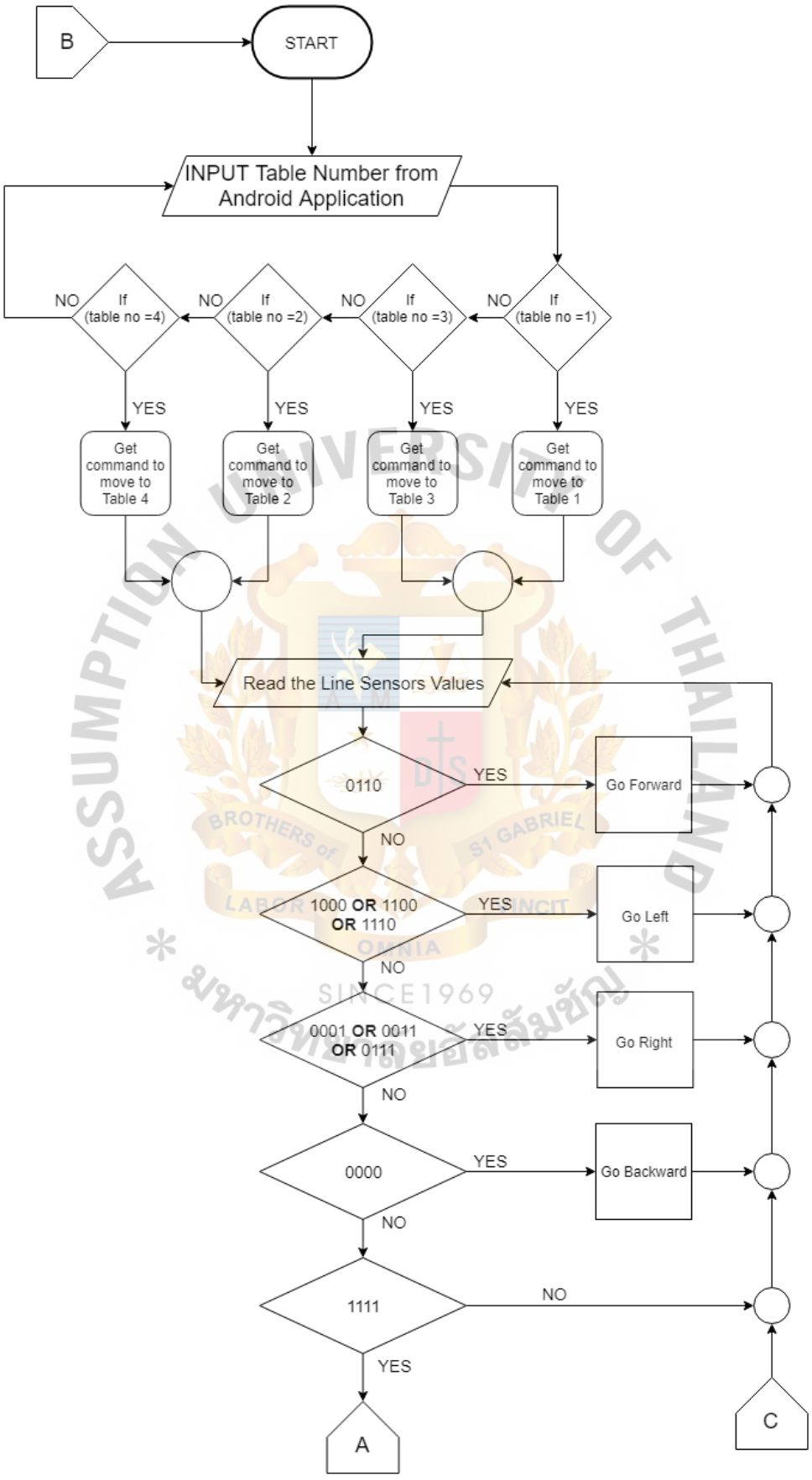


Figure 3.16 System Flowchart (a)

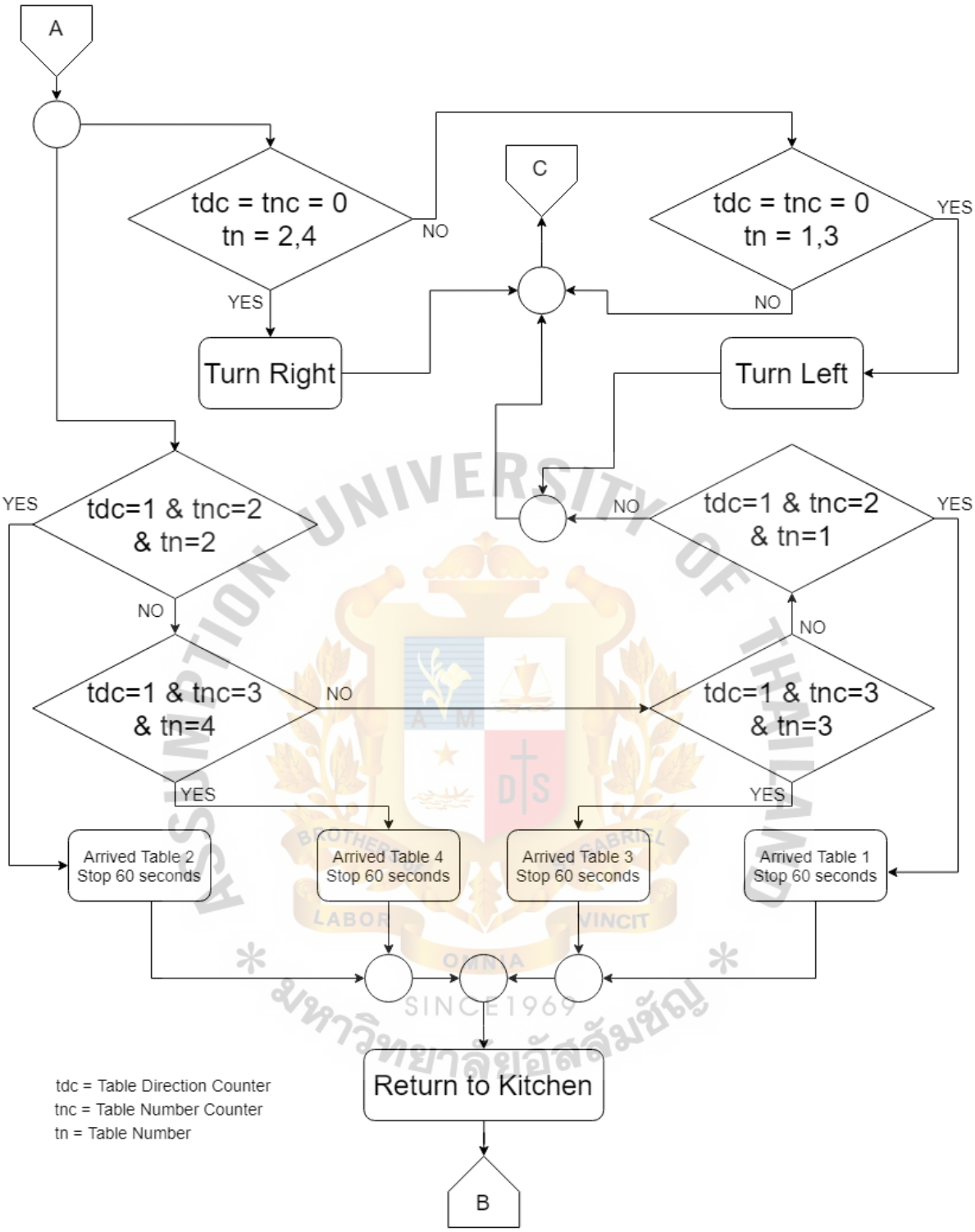


Figure 3.17 System Flowchart (b)

Chapter 4 : Progress and Results

When the program started running, we got the input from Line Sensor and Ultrasonic Sensor. The bot was moving according to the condition of the line sensor. The ultrasonic sensor read the object location, and it measured the distance of 20 cm far away from the bot, the bot would stop until the object was still existing in front of the object. Otherwise, if the object was not blocking on the line route, the bot would continuously move forward and stopped at the destination table.

4.1 Ultrasonic sensor coding progress

```
def distance():
    # set Trigger to HIGH
    GPIO.output(TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()
```

We observed the GPIO (General-Purpose Input/ Output) from Raspberry pi and then got the Echo and Trigger values from the ultrasonic sensor.

```
# save StartTime
while GPIO.input(ECHO) == 0:
    StartTime = time.time()

# save time of arrival
while GPIO.input(ECHO) == 1:
    StopTime = time.time()

# time difference between start and arrival
TimeElapsed = StopTime - StartTime
# multiply with the sonic speed (34300 cm/s)
# and divide by 2, because there and back
distance = (TimeElapsed * 34300) / 2
```

When the echo received the value from the trigger was 0, it means there was no more obstacle within 20 cm away from the bot.

When the echo receives the value which is 1, it means the object is existing. Moreover, the bot functions to stop the current place. The sensor measures the distance between object and bot with the above formula. To calculate the distance, we find the time difference between **Stop Time** and **Start Time**; multiply with the speed of sound per second and is divided by 2.

We can get the distance value from the sensor and we can use in our project. We got the distance values as shown in below table.

Distance Testing	Time difference between start time and arrival time	Distance values (cm)
Measured Distance:	2.5 ms	42.3 cm
Measured Distance:	3.3 ms	55.7 cm
Measured Distance:	2.0 ms	34.7 cm
Measured Distance:	4.8 ms	81.9 cm
Measured Distance:	4.2 ms	71.8 cm
Measured Distance:	2.0 ms	34.7 cm

4.2 Line sensor progress

```
#from left to right ,4 sensors are connected
lf1,lf2,lf3,lf4,lf=0,0,0,0,0
l1 = 6
l2 = 13
r2 = 19
r1 = 26

#Initialise  for line sensor GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(l1,GPIO.IN)
GPIO.setup(l2,GPIO.IN)
GPIO.setup(r1,GPIO.IN)
GPIO.setup(r2,GPIO.IN)
```

We decide the pin numbers (for l1 and l2 = {6,13}; for r1 and r2 = {19,26}) connected with the Raspberry Pi Microcontroller and give the initial state (lf1,lf2,lf3,lf4,lf) as the value is 0 to all the 4 sensors.

```
def read_sensors():
    global lf1,lf2,lf3,lf4
    lf1 = GPIO.input(l1)
    lf2 = GPIO.input(l2)
    lf3 = GPIO.input(r2)
    lf4 = GPIO.input(r1)
```

To track the data sensor from the infrared sensor, lf1 and lf2 was noted as the left sensor 1 and 2. On the other hand, lf3 and lf4 was declared to right sensor 1 and 2.

We setup the corresponding GPIO pin from Pi3 microcontroller and tested the line sensor (IR sensor) if it is detectable or not. IR sensor read the intensity of the blackline, so that we need to adjust the sensor until we get the correct values. After we got the value we can use in our project. The test result is shown in below table.

Designated direction	L1	L2	R1	R2
Stop	1	1	1	1
Go straight	0	1	1	0
Turn Left	1	1	0	0
Turn Right	0	0	1	1
Turn Left	1	1	1	0
Turn Right	0	1	1	1
Go Backward	0	0	0	0

4.3 Motor driving test

Motion	Left Wheel 1	Left Wheel 2	Right Wheel 1	Right Wheel 2
Go Forward	Counterclockwise	Counterclockwise	Counterclockwise	Counterclockwise
Turn Right	Counterclockwise	Counterclockwise	Clockwise	Clockwise
Turn Left	Clockwise	Clockwise	Counterclockwise	Counterclockwise
Backward	Clockwise	Clockwise	Clockwise	Clockwise

4.3.1 Motor Code Testing

```
#L298N port define
in1 = 3
in2 = 2
in3 = 15
in4 = 18
ena = 14
enb = 4
```

The above code showed that L298N port was indicated with pin number from Pi3 as we connected the L298N Motor driver with the Raspberry Pi pin GPIO. (3,2,15,18,14,4). The input signal was obtained from the in1, in2, in3, in4.For the enable A pin and enable B pin are joined with GPIO pin (14 ,4).

```
p1 = GPIO.PWM(ena,1000)
p2 = GPIO.PWM(enb,1000)

p1.start(0)
p2.start(0)
```

P1,P2 was the input of the Motor driver which sent the signal with Pulse width Modulation. It can adjust the speed of motor by changing the duty cycle. P1 is the speed of right motor and p2 is the speed of left motor.

Go forward function

```
def go_forward(speed):
    print("forward")
    p1.ChangeDutyCycle(speed)
    p2.ChangeDutyCycle(speed)
    GPIO.output(in1,GPIO.HIGH)
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.HIGH)
    GPIO.output(in4,GPIO.LOW)
```

In1, in2, in3, in4 are the wires of four motors. HIGH and LOW statement from the code was described as the rate of motor speed. If in1, in3 are HIGH and in2, in4 are LOW, the bot was moving forward.

Go backward function

```
def go_back(speed):
    print("backward")
    p1.ChangeDutyCycle(speed)
    p2.ChangeDutyCycle(speed)
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH)
```

In1, in2 was LOW while the rest of motor was getting HIGH, the bot was functioned to go back with the constant speed according to the duty cycle.

Turn Left

```
#Robot car turn left
def turn_left(speed):
    print("left")
    p1.ChangeDutyCycle(speed)
    p2.ChangeDutyCycle(speed)
    GPIO.output(in1,GPIO.HIGH)
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.HIGH)
```

The bot was going to turn left when the in2,in3,was LOW and in1,in4 was HIGH.

Turn Right

```
#Robot turn right
def turn_right(speed):
    print("right")
    p1.ChangeDutyCycle(speed)     #right motor speed
    p2.ChangeDutyCycle(speed)     #left motor speed
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.HIGH)
    GPIO.output(in3,GPIO.HIGH)
    GPIO.output(in4,GPIO.LOW)
```

If in2, in 3 was HIGH and in1, in 4 was becoming LOW, the bot was moving to turn right.

Stop Function

```
#Robot stop move
def stop():
#     print("stop")
    p1.ChangeDutyCycle(0)
    p2.ChangeDutyCycle(0)
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.LOW)
```

If in1, in2, in3, in4 was becoming LOW, the bot was going to function stop.

4.3 Waiter Bot testing

```
def tracking():
    tn = int(input("Table No:"))
    print("Going to Table:", tn)
    tnc = 0
    tdc = 0
```

We inserted the table input number among the table number (1,2,3,4). Depending on the table number, we must write the route direction which are RIGHT or LEFT. And the variable name ‘tnc’ is referred to table number counter and ‘tdc’ is referred to the table direction counter.

```
while True:
    dist = distance()
    if(dist < 5):
        stop()
        print("stop")
        time.sleep(0.5)
    else:
        read_sensors()
        lf=str(lf1)+str(lf2)+str(lf3)+str(lf4)
        print (lf)
```

The object was locating at the 5 cm far away or less than 5 cm apart from the waiter bot, it sudden to stop at the current place. The four IR sensors was continuously reading the black line at that moment.

```
if(lf == '1111' and tdc == 0 and tnc == 0 and (tn == 1 or tn == 3)):
    tdc += 1
    turn_left(38)
    time.sleep(1)
if(lf == '1111' and tdc == 1 and (tn == 1 or tn == 3)):
    tnc += 1
    time.sleep(1)
```

All IR sensor which got the value was 1111, it expresses as the bot decide turning left or right. At initial stage, we let the tdc and tnc to 0 while bot was deciding the turn according to table number. After tdc had counted to 1, according to above code, the bot would turn left if the table number is 1 or 3.

```
if(lf == '1111' and tdc == 1 and (tn == 1 or tn == 3)):
    tnc += 1
    time.sleep(1)
if(lf == '1111' and tnc == 2 and tn == 1):
    stop()
    time.sleep(0.5)
    print("Arrived Table 1!")
    break
if(lf == '1111' and tnc == 3 and tn == 3):
    stop()
    time.sleep(0.2)
    print("Arrived Table 3!")
    break
```

When tnc was incremented to 1 while tdc was still 1, it showed that waiter bot was received the data to go either table 1 or 3. Therefore, waiter bot would stop at table 1 when the tnc was counted to 2. The bot would stop at table 3 when tnc was counted to 3 but tdc was still 1.

```
if(lf == '1111' and tdc == 0 and tnc == 0 and (tn == 2 or tn == 4)):
    tdc += 1
    turn_right(32)
    time.sleep(0.5)
if(lf == '1111' and tdc == 1 and (tn == 2 or tn == 4)):
    tnc += 1
    time.sleep(0.5)
```

All IR sensors which got the value was 1111, it expressed as the bot decide turning left or right. At initial stage, we let the tdc and tnc to 0 while bot was deciding the turn according to table number.. After tdc had counted to 1, according to above code, the bot would turn right if the table number is 2 or 4.

```
if(lf == '1111' and tnc == 2 and tn == 2):
    stop()
    print("Arrived Table 2!")
    break
if(lf == '1111' and tnc == 3 and tn == 4):
    stop()
    print("Arrived Table 4!")
    break
```

When tnc was incremented to 1 while tdc was still 1, it showed that waiter bot was received the data to go either table 2 or 4. Therefore, waiter bot would stop at table 2 when the tnc was counted to 2. The bot would stop at table 4 when tnc was counted to 3 but tdc was still 1.

```
if(lf=='0000'):
    go_back(25)
    continue
```

If the infrared sensor read ‘0000’, then the bot moves backward.

```
if(lf=='0110'):
    go_forward(35)
    continue

if(lf=='0111' or lf=='0001'):
    turn_right(35)
    continue

if(lf=='1000' or lf=='1110'):
    turn_left(35)
    continue
```

If the infrared sensor read ‘0110’, the car moves forward. If the infrared sensor read ‘0111’ or ‘0001’, the bot turns right. If the infrared sensor read, either ‘1000’ or ‘1110’, the bot turns left.

```
if(lf=='0011'):
    turn_right(33)
    continue

if(lf=='1100'):
    turn_left(33)
    continue

else:
    stop()
```

If the infrared sensor read ‘0011’, the bot turns right. Moreover, if the infrared sensor read ‘1100’, the bot turns left.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
มหาวิทยาลัยอัสสัมชัญ

4.4 Designing and Actual Structure

4.4.1 Assembly of the Hardware

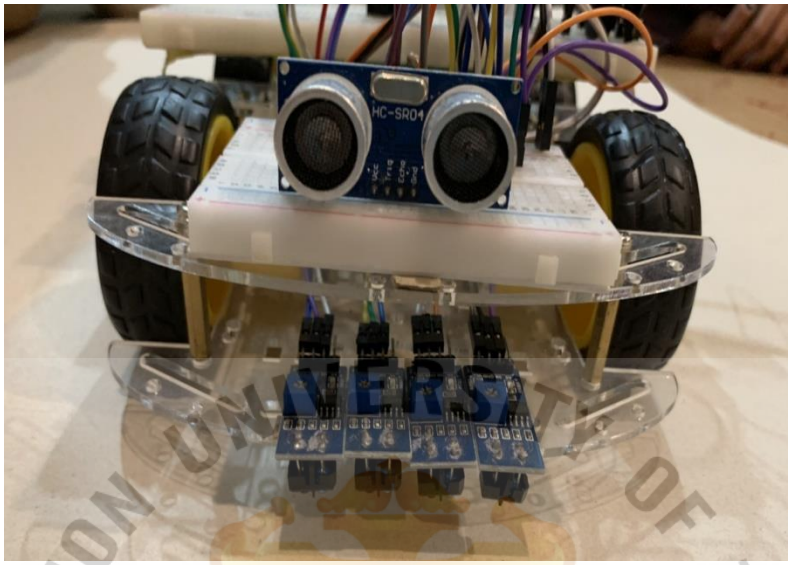


Figure 4.2 Ultrasonic sensor was place at the breadboard and connected with microcontroller.

Figure 4.3 IR sensors and Ultrasonic sensor was connected to the Raspberry Pi 3

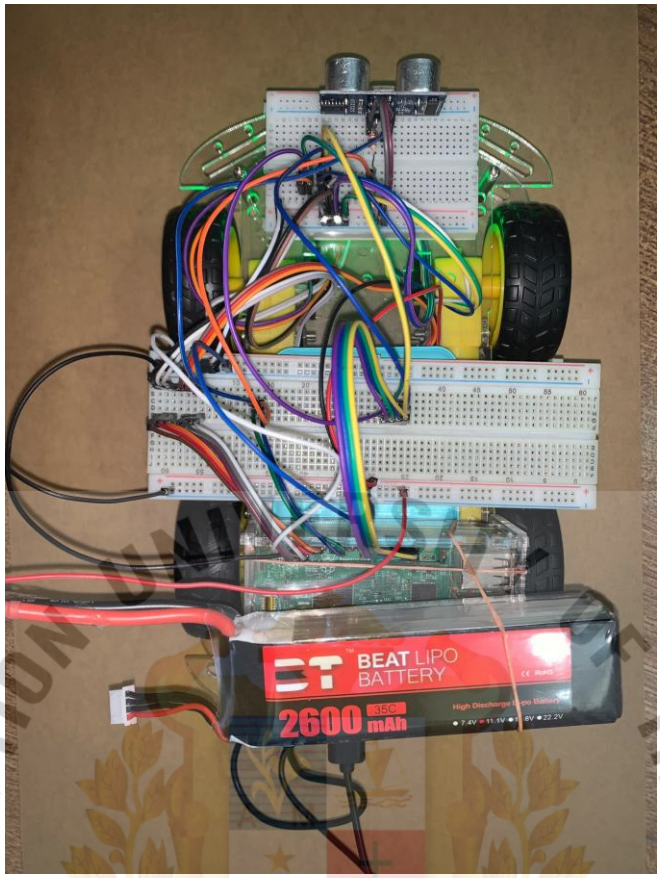


Figure 4.4 Top view of the Waiter Bot

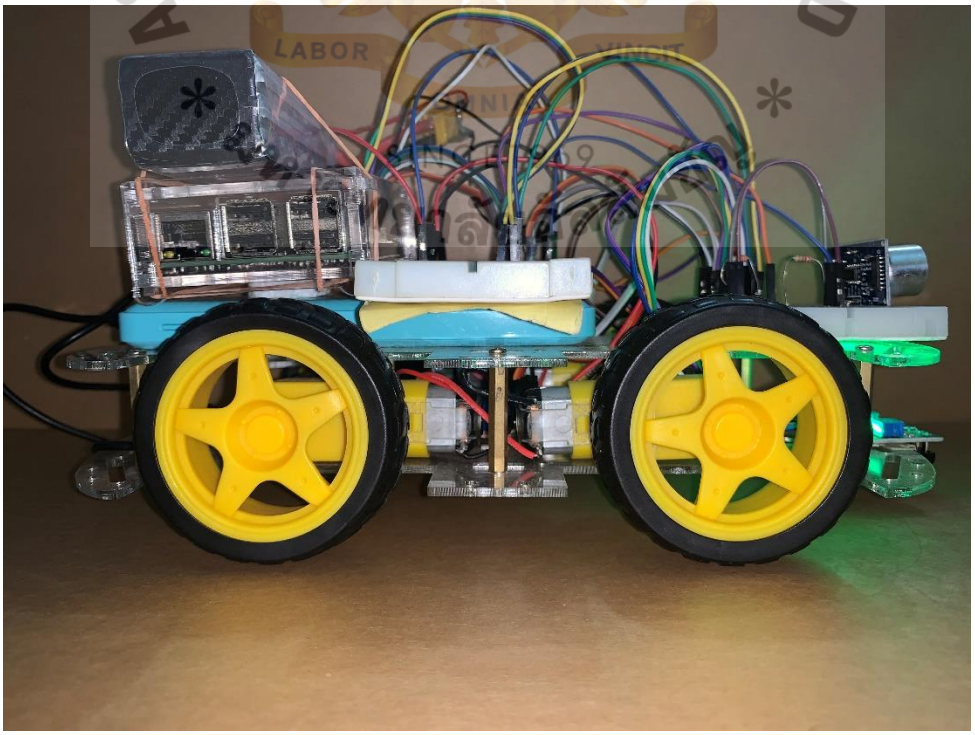


Figure 4.5 Side View Structure of the Waiter Bot

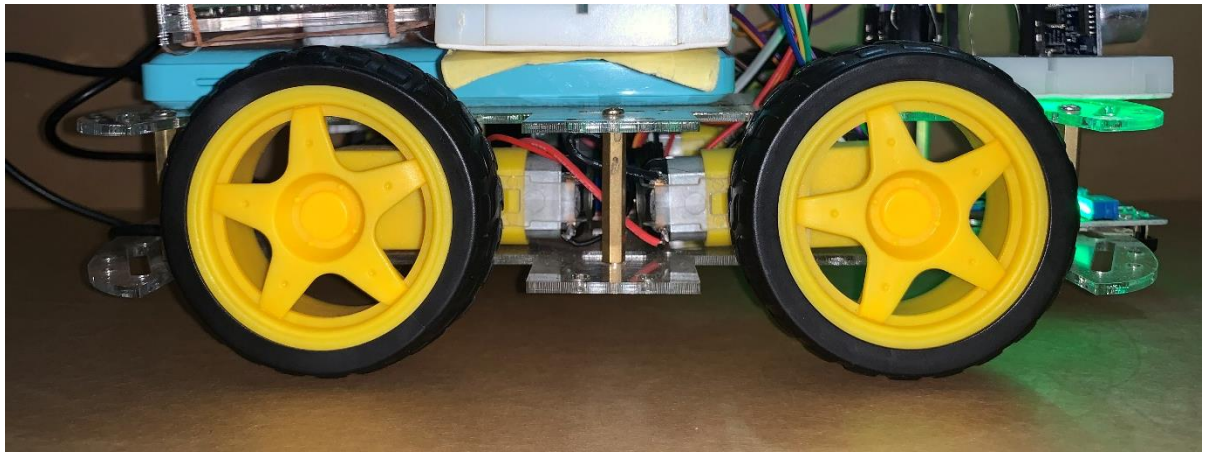


Figure 4.6 Soldering the Wire of Two motors

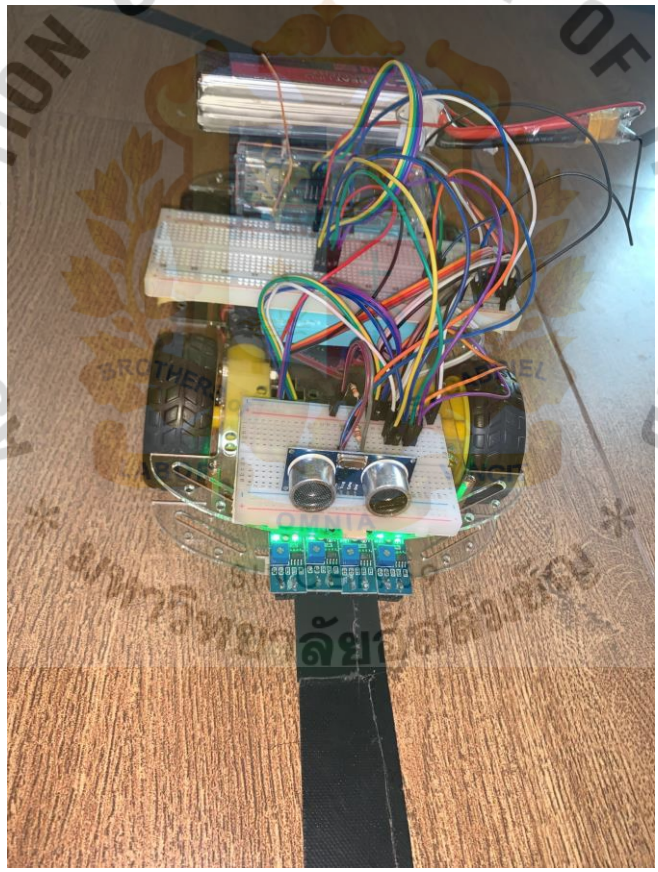


Figure 4.7 Going forward by following the black line

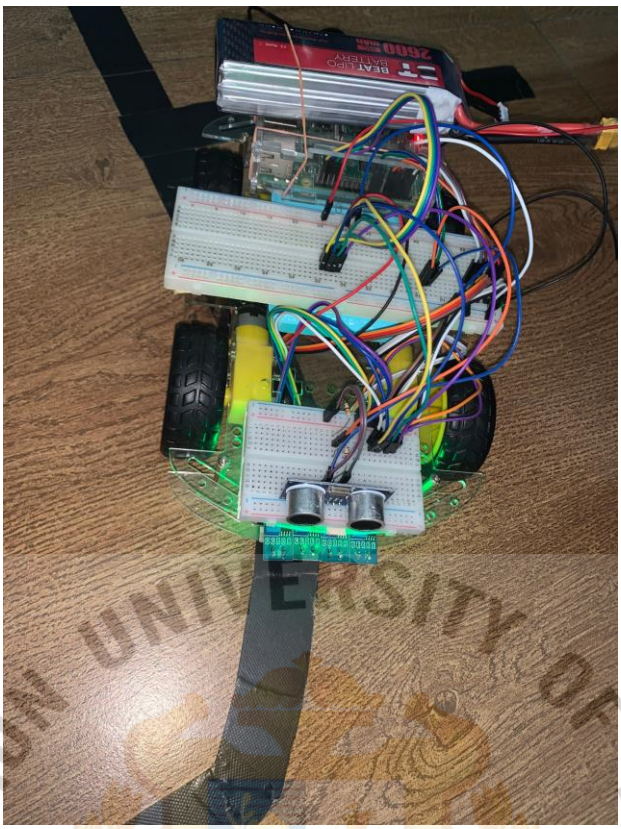


Figure 4.8 Turning left on the black line

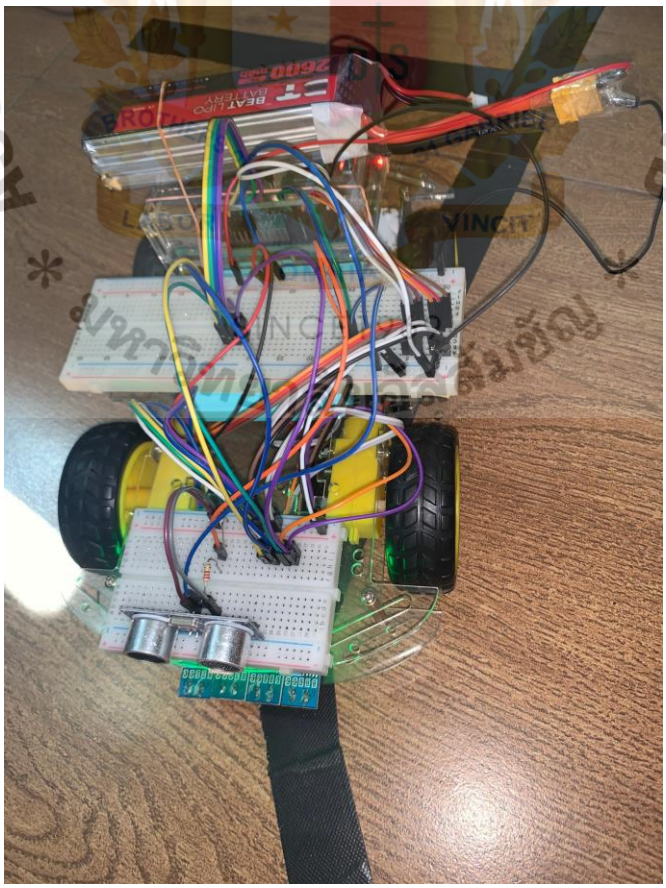


Figure 4.9 Turning Right on the black line

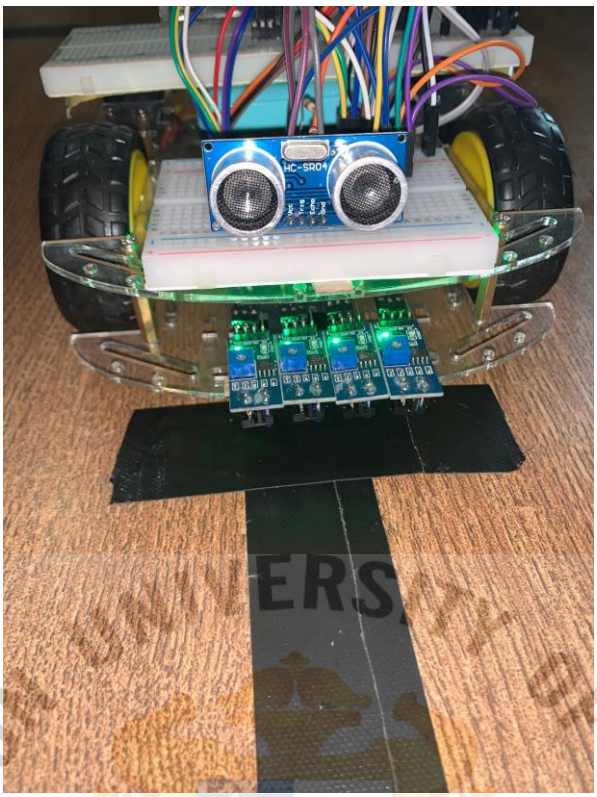


Figure 4.10 Stopping at the Table 1

ASSUMPTION UNIVERSITY THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Chapter 5 : Problem and conclusion

5.1 Issues and Problems

During this project, we had faced many problems such as

- While the localization system was designed on the floor, the marked line from the floor was necessary to be smooth such that constant maintenance to the floor from any physical damage is required to be less friction.

- There is one more major flaw, if the restaurant wants to change the table or refurbish the layouts or decorations the navigation system or the map layout programmed inside the robot needs to be updated or in worst case scenario they will be needed to reprogram entirely by the technician or the developers.

- The waiter bot lacks the ability to return to the path if any external disturbance is affected upon the robot, thus the robot does not response well to any physical disturbances.

5.2 Conclusion

In conclusion, we faced many problems testing senor and testing the scenarios. Moreover, we need to test run a lot of time to make it less errors. Some of the features that we planned was not actually work so that we need to change some of the plans we need to fix and changes. We could run the waiter bot and serve at the table. Furthermore, we could make the bot come back to the kitchen after serving at the table.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of S1 GABRIEL
LABOR VINCIT
OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Bibliography

A Cheong, E Foo, MWS Lau, J Chen. (2015). *Development of a Robotics Waiter System for the Food and Beverage Industry.* Retrieved from Research Gate: https://www.researchgate.net/publication/324240091_Development_of_a_Robotics_Waiter_System_for_the_food_and_beverage_industry

Ghaleb, M. (2018). *Design Of and Obstacle-avoiding Robot car based on Audrino Microcontroller.* North China: researchgate.

Jinsoo Hwang, Kwang-Woo Lee, Dohyung Kim, Insin Kim. (2020). Robotic Restaurant Marketing Strategies in the Era of the Fourth Industrial Revolution: Focusing on Perceived Innovativeness. *Sustainability*, 2.

Kimura, A. E. (2015). *Restaurant Service Robots Development in Thailandand Their Real Environment Evaluation.* Akkharaphong Eksiri.

Mohd. Abdullah Omair, Al Shahriar Hossain Rakib, Md. Amir Khan, Rubaiyat Talat Mahmud. (2015). An autonomous robot for waiter service in restaurants. 10.

Oke Alice O, Afolabi Adeolu. (2014). DEVELOPMENT OF A ROBOTIC ARM FOR DANGEROUS OBJECT DISPOSAL. *6th International Conference on CSIT*, (p. 5). Ogbomoso.

Panda, K. G., Agrawal, D., Nashimiyimana, A., & Hossain, A. (2016). Effects of Environment on Accuracy Ultrasonic Senor operates in Millimeter Range. *Preceptives in Science*, 2-3.

Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Fangkai Yang,Peter Stone. (2017). BWIBots: A platform for bridging the gap between AI and human-robot interaction research. 6.

Prof.Dr. Usman Ali Shah, Faraz Ali, Sana Sohail, Haris Khan. (2016). *Intelligent Robotic Waiter with Menu ordering System.* Karachi: 1st International Electrical Engineering Congress (IEEC 2016).

Python Developer's Guide. (n.d.). Retrieved from Python: https://www.python.org/

Schneidar, B. (n.d.). *A Guide to Understandinf Lipo Batteries*. Retrieved from Roger's Hobby Center: https://rogershobbycenter.com/lipoguide

SHEIKH FARHAN JIBRAIL, RAKESH MAHARANA. (2013). *PID CONTROL OF LINE FOLLOWERS.* Rourkela: Department of Electronics & Communication Engineering.

T. Verstraten, G. Mathijssen, R. Furnémont, B. Vanderborght. (2015). *Modeling and dimensioning of geared DC motors for energy efficiency.*