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:
Mr. Amulya Bhattarai

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

Assumption University
Vincent Mary School of Engineering
Thailand
March 2021

Approved by Project Advisor:

Name: Mr.Amulya Bhattarai
Signature:
Date:

Plagiarism verified by:

Name: Mr.Ehsan Ali
Signature:
Date:

ABSTRUCT

This project intends to design the replacement of manpower of waiter at the restaurant to solve the major problems 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 the autonomous technology, we are going to optimize the mobile bot that can be able to use the restaurant. The main objective of the restaurant bot is capability of taking order and serving food with the python program. Moreover, to reach out 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 table. Additionally, use of double line sensors are 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 and implement the motor driver controller to control the speed of the motors steering the robot to travel along the line smoothly.

This project aims to implement the python programming language and control the movement of the robot by proper tuning of the control parameters and thus achieve better performance. The process is that the machine operates throughout the line sensors to reach out the desired table, and after completing the task order, the machine terminates back to the service counter by following the lines.

Acknowledgement

I would like to special thanks to my advisor Mr. Amulya Bhattarai who gave me advice and provided throughout the research and testing program and I must be more relevant my project finish on time.

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

1.1 **What is the Bot?**

Basically , the bot is classified into three subsystem for optimization which are sensors which operate as the eyes ,processor which acts as the memory tasks or brain of the human , and motor controller system that optimizes as the movement of the legs . Likewise, robots that can perform automated tasks in any field with no human command/guidance. The ability of this performance task could be faster than the human tasks .Bot can navigate the location place by obeying the program tasks. And in addition, bot can ensure to implement the system for months without human intervention.

With the similar theory, autonomous bot can be provided in military to reduce the number of casualties which occur during military actions has been already been prioritized. The military also uses robots for locating and destroying mines on land and in water, spying on enemies and entering enemy bases for gathering information.

1.2 **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 based on the need to reduce the manpower but not to replace it. Humans are still needed for personal relationship purpose, to engage the customers and to make their dining a pleasure. For all the numerous researches on intelligent autonomous robots, it remains a difficult problem for such robots to work in a real-world environment of a restaurant. A more viable approach is the integration of the autonomous robots within a smart environment.

Restaurant bot 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, it can perform the repetitive work with high accuracy and will not stop or slow until the task is accomplished.

Restaurant bot was controlled by the Raspberry Pi 3 for the whole processes. The main tasks was to go through the order table depending on the input order number from the customer, and then follow the black marked line on the floor , after that , go back to the destinated place (original place) to wait 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.

1.3 **Overview**

This module project was to combine with the Artificial Intelligence technology and novelty serving system which tend to enhance the productivity level in the restaurant industry. Robots can work 24 hours continuously without feeling tired unlike human that confined to certain time. But there is no bot able to functions perfectly and are still making error. A better controller is required to allow the robot performs efficiently and make less error.

Waiter(machine) 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 continuously returns back to the service counter after the machine has accomplished its task/operation. The Raspberry Pi 3 is to implement the drive microcontroller to stabilize the induced voltage to the motor. And in addition, the drive microcontroller adjusts the rotation angle by controlling the speed and it proceed with 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 two infrared sensors that perform a line-tracking module, where they will follow the track made from black tape. This is the area where the Pi3 microcontroller was implemented, the bot would be able to follow the black tape effectively and moving along the track smoothly.

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 is selected by the program. And then it can self -navigate the marked path toward its destination (customer table).After completing the non- repitative tasks, it will always return back to the specified place.
- To reach out the destinated 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 in 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/ design how it will be stopped or continuously running .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 problem such that it must be able to use it proficiently in restaurant workplace.

1.5 Problem Satetments

1.6 Background Study

1.6.1 AIC (automatic Intelligence Car)

As the initial consideration to our project is referenced/emulated to the AIC (Automatic intelligence car) which are heading toward the self- driving car with highly efficiency and control operation autonomously .The AIC (Automatic Intelligence Car) Robot is likewise proceeding the whole system automatically which have the features of AI (Artificial Intelligence). By the definition ,it has some properties that has an ability to sense its surrounding environment and decide the navigation path and plus able to respond without any human input like in some occasion such as the obstacle avoiding property. With the sensor, it can smoothly lead in right position. As we, human drivers have the fully responsibility for driving the car to the destination safely. Similarly, our bot's main responsibility is to fetch the order from the kitchen to the patron's table.

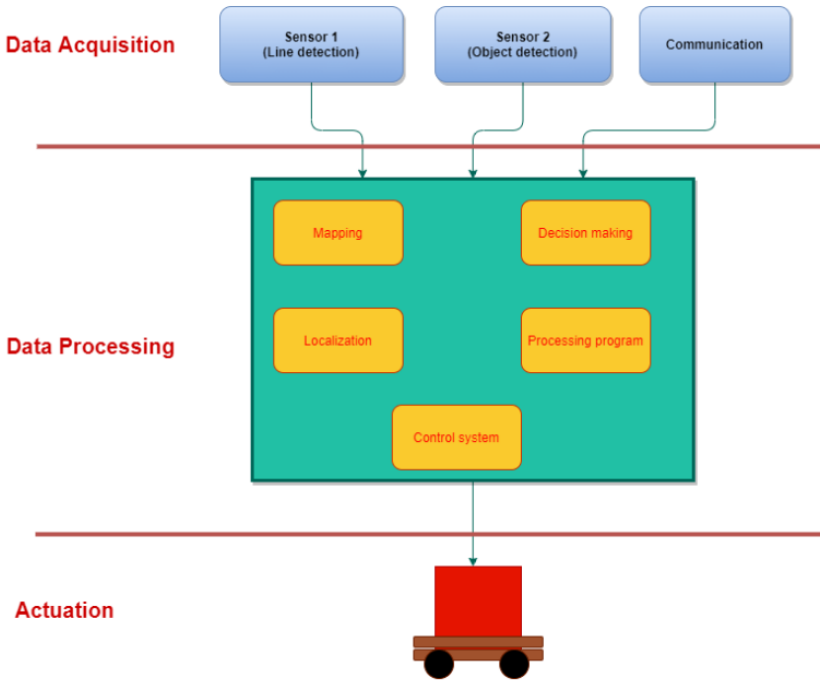
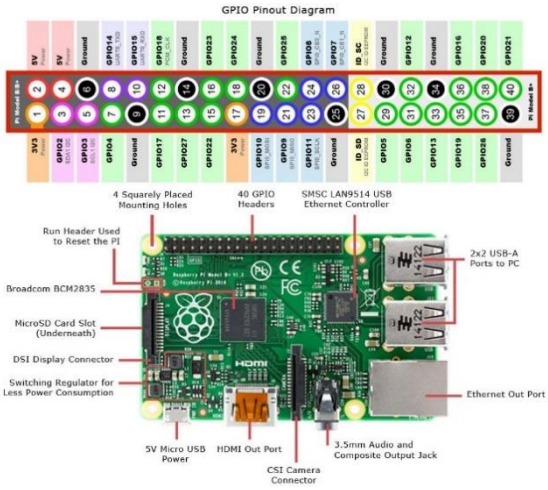


Figure 1.1 Sub-system of our bot

` Our bot's system is divided into three sub-system. They are data acquisition, data processing and actuation. As the data acquisition section, we can call it as the input that collect the data from the environment condition and transfer the specified data to the control system which part is called data processing. In data processing, utilizing the data from the sensors by decoding language and enter to the microcontroller , and then when microcontroller will implement the program ,as the actuation part , the torque of the motor is start rotating so that the wheels are moving forward smoothly.

1.6.2 Raspberry Pi 3 Model B+



1.6.3 Line Follower

1.6.4 Rotation Process

1.6.5 Object Avoiding Features

1.6.6 DC Motor - Gear 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 loop 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 estimates the output power.

$$P_{out} = Torque(Nm) \times \omega$$

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 2: HARDWARE DEVELOPMENT

2.1 SYSTEM DESIGN

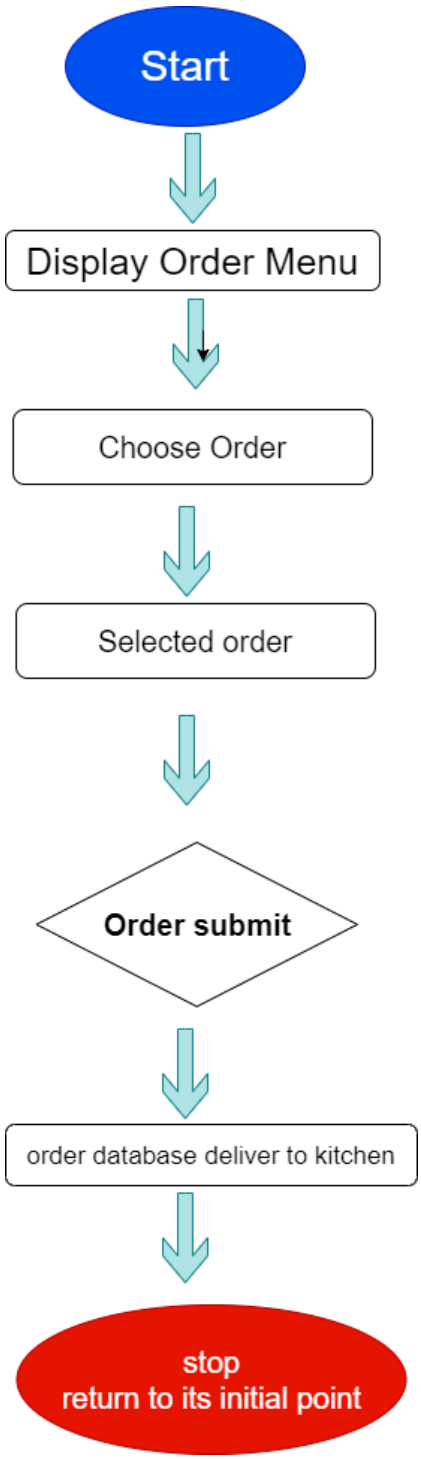


Figure 2.1: Order process diagram

The above diagram is 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. As our plan, for instance, if the table 1 is calling for ordering the food, the machine receives the message with input code and goes to the table with constant

speed and detects the line marked on the floor. When the machine memorized the order from the table 1, remotely returns to the kitchen and wait for serving the food. 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.

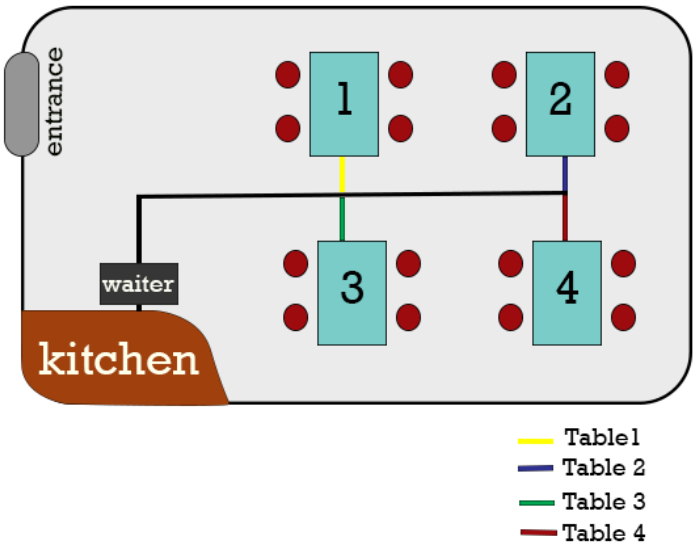


Figure 2.2 Blueprint of restaurant

3.2 Progress

When the program starts, we will get input from Line Sensor and Ultrasonic Sensor. The car will be move according to the condition of the line sensor. If we detected object 20 cm far in front the car, the car will be stop and continue moving after the object is moved.

3.2.1 Ultrasonic Sensor progress

```python
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()

    # 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


    return distance
```

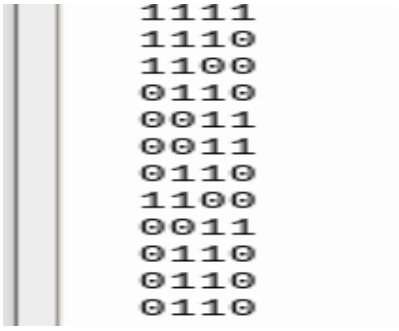
We get the GPIO input and output from Raspberry pi and get Echo and Trigger vales from the ultrasonic sensor. After we got the values from the sensor, we put in the formula found on the web. We can get the distance value from the sensor and we can use in our project. We got the distance values as shown in below picture.

```
Measured Distance = 42.3 cm
Measured Distance = 55.7 cm
Measured Distance = 55.4 cm
Measured Distance = 34.7 cm
Measured Distance = 81.0 cm
Measured Distance = 28.9 cm
Measured Distance = 71.8 cm
Measured Distance = 35.2 cm
Measured Distance = 34.7 cm
Measured Distance = 33.3 cm
```

3.2.2 Line Sensor progress

```
from __future__ import division
import time
# from goto import with_goto
import RPi.GPIO as GPIO
lf1,lf2,lf3,lf4=0,0,0,0

#line follow module port define
l1 = 6
l2 = 13
r2 = 19
r1 = 26

#sensors init
lf1,lf2,lf3,lf4,lf=0,0,0,0,0

#Initialise 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)
# Read tracking senbsors's data
def read_sensors():
    global lf1,lf2,lf3,lf4,lf
    lf1 = GPIO.input(l1)
    lf2 = GPIO.input(l2)
    lf3 = GPIO.input(r2)
    lf4 = GPIO.input(r1)

def destroy():
    GPIO.cleanup()

def main():
    global lf1,lf2,lf3,lf4,lf
    y=0
    while True:
        read_sensors()
        lf=str(lf1)+str(lf2)+str(lf3)+str(lf4)
        print (lf)
        time.sleep(0.5)
```

We decide the pin numbers and give initial value 0 to all the 4 sensors. We setup the GPIO and test the line sensor (IR sensor) if it is detectable or not. IR sensor red 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 as below.



3.2.3 **Waiter Bot testing**

```
def tracking():
    tn = int(input("Table No:"))
    y = 0
    x = 0
    while True:
        dist = distance()
        if(dist < 30):
            stop()
            print("stop")
            time.sleep(0.5)
        else:
            read_sensors()
            lf=str(lf1)+str(lf2)+str(lf3)+str(lf4)
            print (lf)
            if(lf=='0000'):
                #set_speed(high_speed,high_speed)
                go_back(25)
                continue
            if(lf == '1111' and x == 0 and y == 0 and (tn == 1 or tn == 3)):
                x += 1
                turn_left(40)
                time.sleep(1.2)
            if(lf == '1111' and x == 1 and (tn == 1 or tn == 3)):
                y += 1
                time.sleep(1)
            if(lf == '1111' and y == 2 and tn == 1):
                stop()
                time.sleep(0.5)
                print("Arrived Table 1!")
                break
            if(lf == '1111' and y == 3 and tn == 3):
                stop()
                time.sleep(0.2)
                print("Arrived Table 3!")
                break

            if(lf=='0110'):
                go_forward(30)
                continue
```

```
            if(lf=='0111' or lf=='0001'):
                turn_right(30)
                continue

            if(lf=='1000' or lf=='1110'):
                turn_left(30)
                continue

            if(lf=='0011'):
        #           #set_speed(low_speed,low_speed)
                turn_right(35)
                continue
        #
            if(lf=='1100'):
        #           set_speed(mid_speed,mid_speed)
                turn_left(35)
                continue
        #
            else:
                stop()
```

After we tested all the sensors and equipment, we started our project. Firstly, we got the input as table number that we want to stop at. The car will be move according to the line on the floor. If we get the value 0000 it will be moving backward slowly. Moreover, if we get value 0110 it will be moving forward, 1100 will be moving left, 0011 will be moving right. We put the condition 0111 and 0001 to move slightly right to keep moving in line without having problem. We put the condition similarly to 1000 and 1110 to move slightly left.

For the stopping points,

CHAPTER 3: CONCLUSION

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.

References