Journal of Engineering and Technological Advances (JETA)
https://jeta.segi.edu.my/index.php/segi
2022 Vol.7 No. 2

DEVELOPMENT OF AN AUTONOMOUS ASSISTIVE ROBOT FOR HEALTHCARE APPLICATION

Ahmed Alhasani, Anis Fariza Md. Pazil*, Vinukumar Luckose

Centre of Advanced Electrical & Electronic System, Faculty of Engineering, Built Environment and Information Technology, SEGi University, 47810 Petaling Jaya, Selangor, Malaysia

* Corresponding Author: anis@segi.edu.my TEL: (+60)-6145 1777

Abstract: This project mainly focuses on an autonomous assistive robot for healthcare application that is developed to dispense medications to the individual patients in the patient care facility. This autonomous line-follower robot is able to detect and follow the line drawn on the floor while driving through the facility with the ability to stop when it faces obstacles. The source of energy supply to the whole robot is fed from the power supply through the ESP32 microcontroller that is programmed using C++ with infrared sensors connected to the board for sensing the drawn line. The ultrasonic sensors are used to detect any obstacles on their way and put the robot at a halt condition with an alarm indicating the obstacle occurrence. The robot will stop at the junction to dispense the medicine for 10 seconds before proceeds to the next station. After the medicine has been dispensed, a confirmation message will be sent out from CallMeBot API in Telegram to acknowledge the medical staff on the successful delivery of the medicine to each patient. The result presented the experimental data on the path line application and the obstacle detection to demonstrate the efficiency of the application.

Keywords: Assistive robot, line-follower, microcontroller, healthcare, sensors, COVID-19

---

1. Introduction

The Coronavirus disease (COVID-19) that was detected back in December 2019 in Wuhan, China caused a major outbreak across the globe. According to the World Health Organization's (WHO) Situational Report of 127 which was released on November 24, 2021, there were 256,072,650 of confirmed cases and 5,132,202 casualties globally till then (WHO, 2021). The death rate is much higher in older people than in younger people, and male patients are more likely to face fatality than female patients in the same age group. Due to these severe situations globally, the healthcare industry particularly the hospitals was badly impacted as they could not cater to the number of massive COVID-19 patients at one time. The number of doctors and nurses was not able to provide their service at the best level as they had to work long hours due to the insufficient manpower to monitor the COVID-19 patients who have been hospitalized. Besides that, COVID-19 which spread through air, close contact or by touching the objects with virus on their surface has made the monitoring work for the healthcare provider much more challenging whereby they need to be fully equipped with personal protective equipment (PPE) to avoid them from being infected (Manikandan et. al, 2021).

The application of assistive robots in healthcare facilities is seen as an opportunity to minimize the infection issues in handling and monitoring COVID-19 patients. By application, robotics is an emerging research area devoted to the design, construction, maintenance, manufacture, and deployment of robots (Pathak et al, 2017). An assistive robot in general is a device that can sense, process, and perform tasks in the daily lives of persons with disability conditions or assist older adults. Besides that, autonomous assistive robots are also commonly utilized in food deliveries, quantity analysers, product delivery, and agricultural harvesters. In robotics, obstacle avoidance is the task of attaining a control goal while adhering to constraints on the location of non-intersection or non-collision points. It is perceived as a crucial study field that also serves as a foundation for a construction robot's success (Barua et al, 2020).

The use of a line following robot can be incorporated into automation in hospitals. This robot is a type of robot that follows the user's white or black path instructions (Pakdaman et al, 2010). A line-following robot can serve as a temporary nurse, assisting hospital employees in the event of an emergency. This type of robot can also serve as a delivery robot in operating rooms where surgeons may require additional supplies in the event of an emergency. It provides advantages such as faster service, being available 24 hours a day, being more reliable, and being more efficient. While in smart hospital, the patients are managed by four different types of robots namely mobile robots, surgical robots, care robots, and exoskeletons (Kim et al, 2022).

By giving this conventional line follower robot the ability to detect obstacles, it can be made smart and intelligent. This increases the line follower robot's performance because obstacles are the frequent objects which exist in any work area, and if the line follower is unable to detect any obstacles in its path, it will collide with them and be critically harmed. Adding obstacle avoidance features to a typical line follower robot avoids the robot from being damaged. As such, this intelligent robot can be used in hospitals to manage health care, reducing human effort in monitoring patients and delivering items or medicines (Suryawan et al, 2019)

2. Design Methodology

The general block diagram of the overall system design is shown in **Figure 1**. The robot comprises several components such as a power supply (power bank) to supply the energy source to the whole robot through HW-131 Power Module 3.3V/5V Power Supply (later known as MB102) with ESP32 microcontroller that gets programmed using Arduino IDE (C++ programming language). Meanwhile, the IR sensors is connected to the ESP32 board for sensing the drawn line while the ultrasonic sensor is used to detect any obstacles on its way and sense the robot to stop. Here, an alarm is implemented to inform the medical personnel if there are any obstacles that block the way. The medicines that are delivered near the patient's bed will collect the medicines on the robot compartment by themselves. After 10 seconds (10s), the ESP32 microcontroller is programmed to send out the confirmation message to medical staff acknowledging the task completion on medicine delivery through CallMeBot API in Telegram. The 10s is chosen to ensure the patient have ample time to collect the medicine when the robot stop next to their bed.

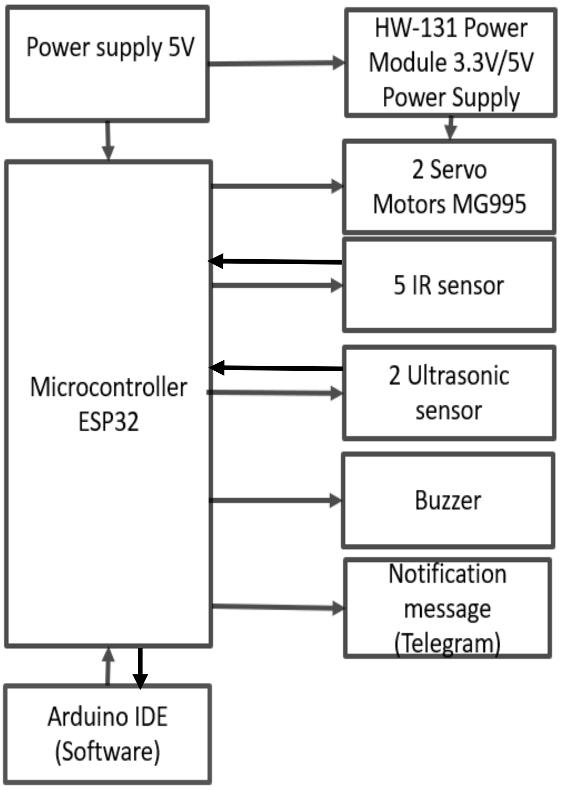


Figure 1. Block diagram of the proposed system design

Figure 2 shows the operation of the line follower robot with obstacle detection. Initially, when the ESP32 is turned on, the robot will first try to sense for the black line to follow via IR sensors. Nevertheless, if the black line is not detected, the robot remains idle until the robot senses the black line once again and then starts to move via the servo motor. When the robot detects an obstacle on its way to the patient's bed, the robot will stop moving, halt both servo motors and subsequently the alarm will turn on. After the obstacle is removed and cleared, the alarm will turn off and the robot continues its route to the patient's beds for delivering the medicines.

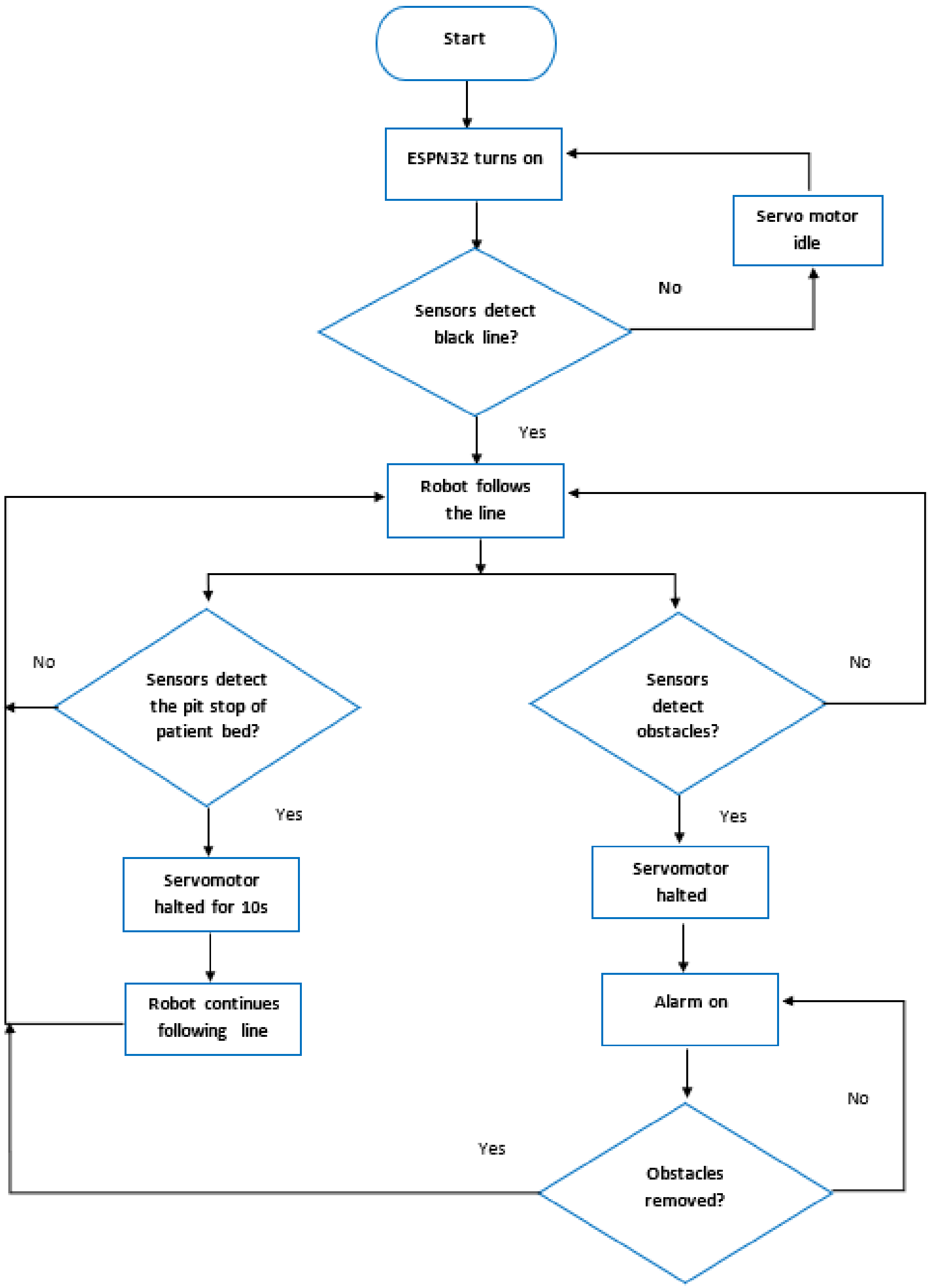


Figure 2. Line following design flow chart

Figure 3 describes the process of medicines delivery to patients in several steps and how the robot operates to achieve the aims of the study. Firstly, the robot will be stationed at the medicine store which marked the starting point for loading the medicine onto the robot by the medical staff. The robot will then follow the line till it finds a crossed line that marks a pit stop near the patient's bed for 10s for medicine collection on top of the robot by the patient. After delivering the required medicines to all patients in the loop, the robot will be heading to the medicine store to reload the medicines as and when needed.

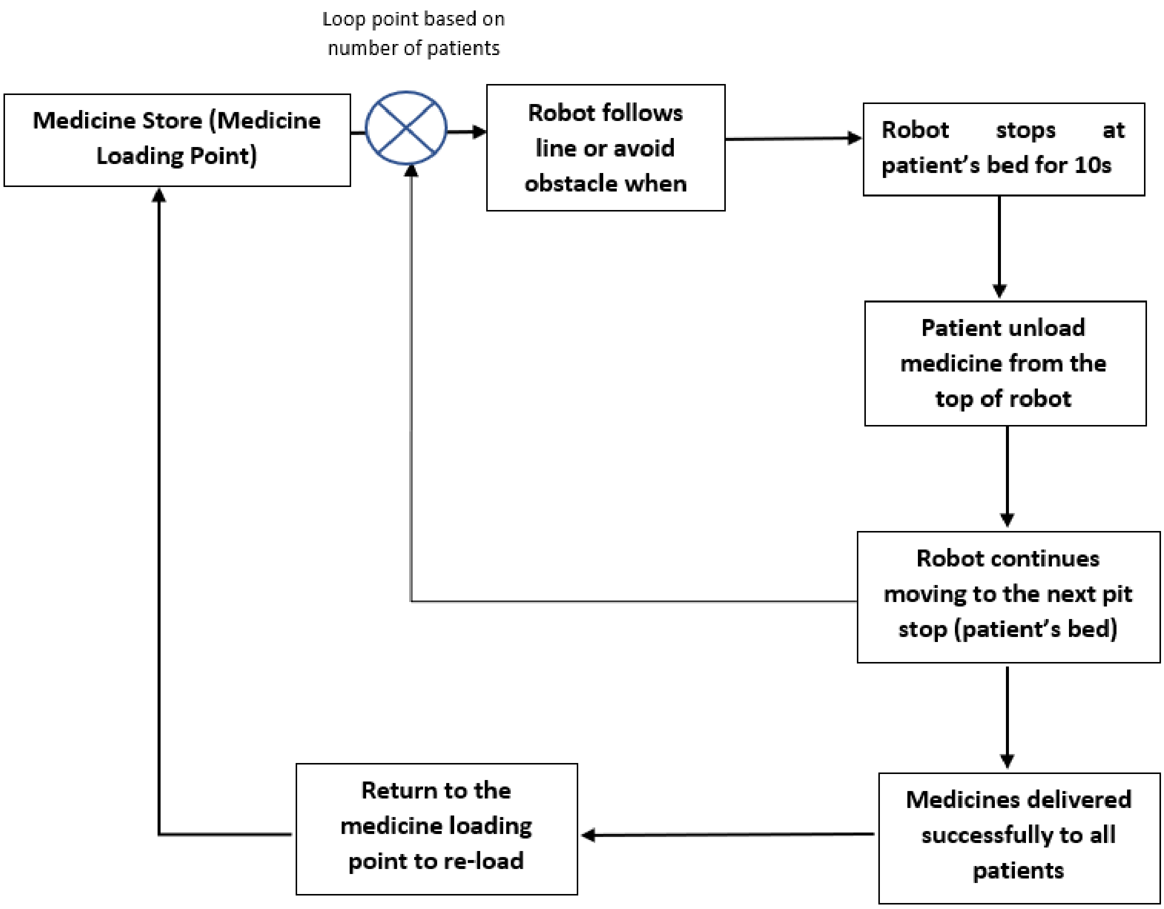


Figure 3. Block diagram of medicine delivery to patients

Figure 4 shows the overall circuit diagram of the line of the following robot. Based on the figure, the components of IR 1, IR 2, IR 3, IR 4, IR 5, Ultrasonic 1, Ultrasonic 2, servo motor 1, servo motor 2, buzzer and MB102 power supply module are connected to the ESP32 board which function as a microcontroller. The MB102 power supply module is capable of supplying 6V; nevertheless, the output voltage is set up to 5V as a threshold value to meet the supply platform of the whole project. From this figure, all components are connected in parallel to the power supply and to the ground with different input signal pin from ESP32. The connection in the circuit diagram below is made in parallel to ensure the voltage drive through each component is always 5V.

The power to all components is supplied by MB102 with each one is connected to a different input signal pin from ESP32. Here, IR 1 is connected to pin 19, IR 2 is connected to pin 23, IR 3 is connected to pin 18, IR 4 is connected to pin 4, IR 5 is attached to pin 13, right servo motor is attached to pin 21, left servo motor is connected to pin 22 and buzzer is connected to pin 12. Each ultrasonic sensor has two signal pins (*echopin* input and *trigpin* as output). For ultrasonic sensor 1, *trigpin* is connected to pin 2 and *echopin* is wired to pin 35 from ESP32. For ultrasonic sensor 2, *trigpin* is connected to pin 15 and *echopin* is wired to pin 5 from ESP32 board.

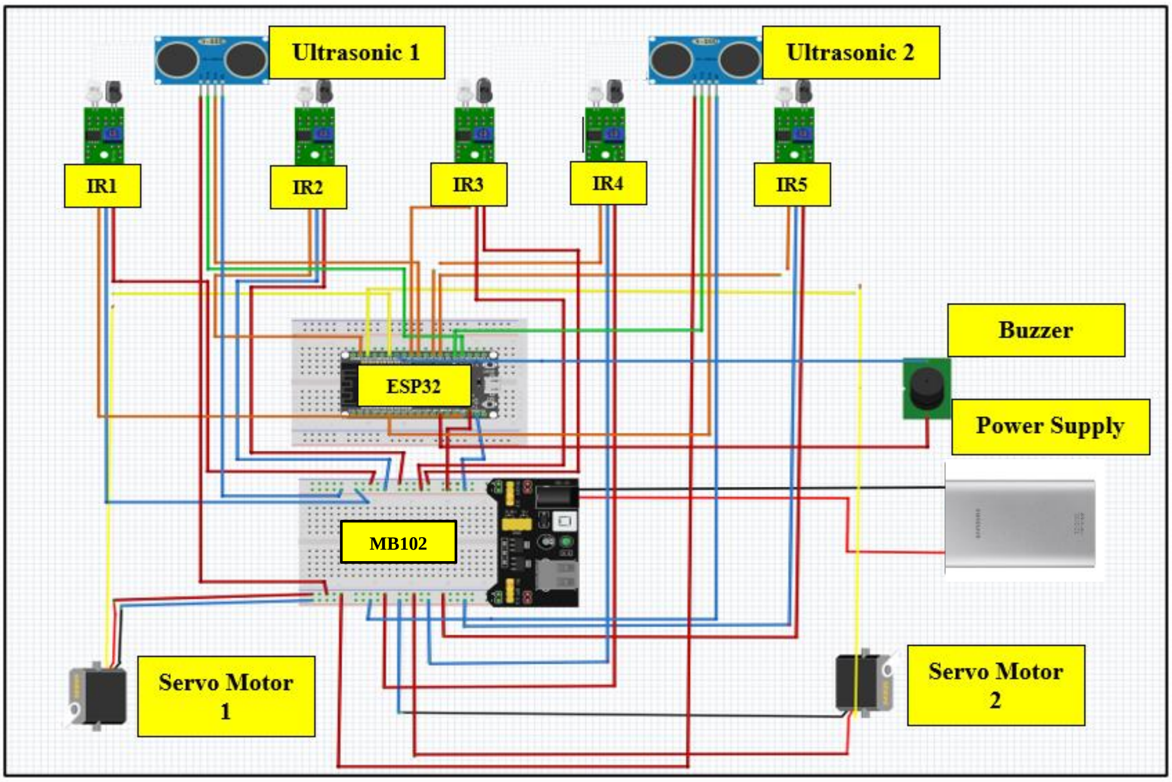


Figure 4. Overall circuit diagram of the line following robot

3. Results and discussion

Figure 5 shows the prototype structure of the line follower robot showing the integration of IR sensors and ultrasonic sensors with two back wheels and one front wheel into the body. The robot body was made up of the corrugated board of a light material with length of the prototype is 30 cm, the width is 30 cm while the height is 30 cm.

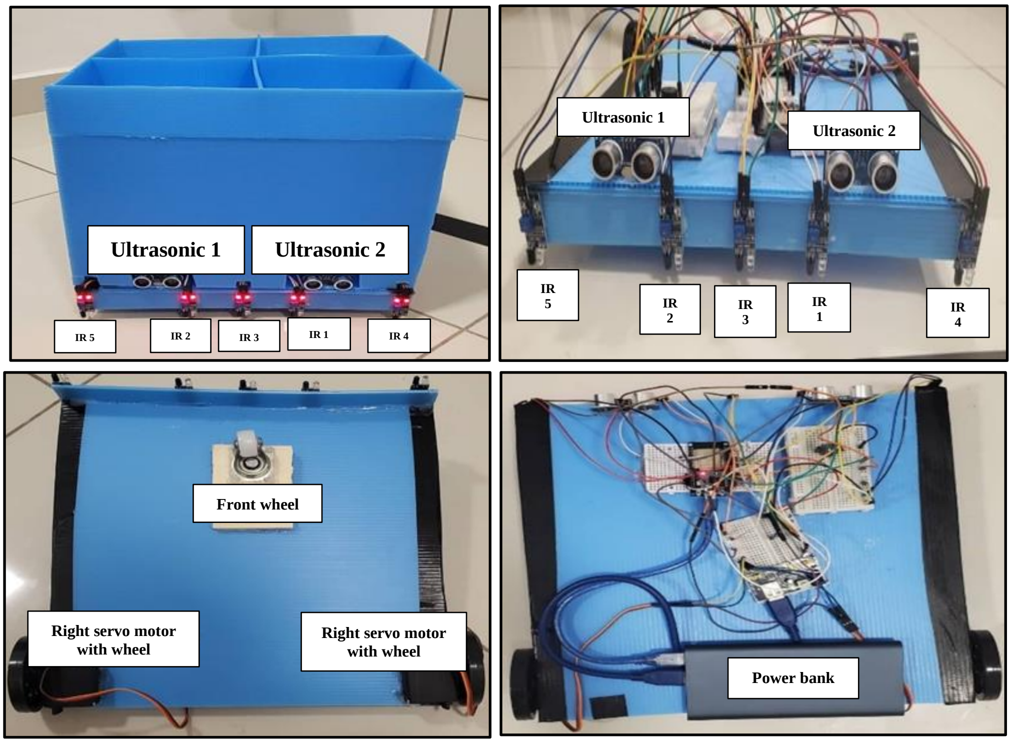


Figure 5. Line follower body structure implementation with IR sensors, ultrasonic sensors, and servo motors with wheels

Figure 6 shows the overall body structure of the line follower demonstrating the top view, side view and back view. The medicine placement sections in **Figure 6a** have been made into 4 compartments to cater to 4 patients in the demonstration while **Figure 6b** shows the position of the wheels from the side. Figure 6b demonstrates the body of the robot which can be opened up for calibration purpose.

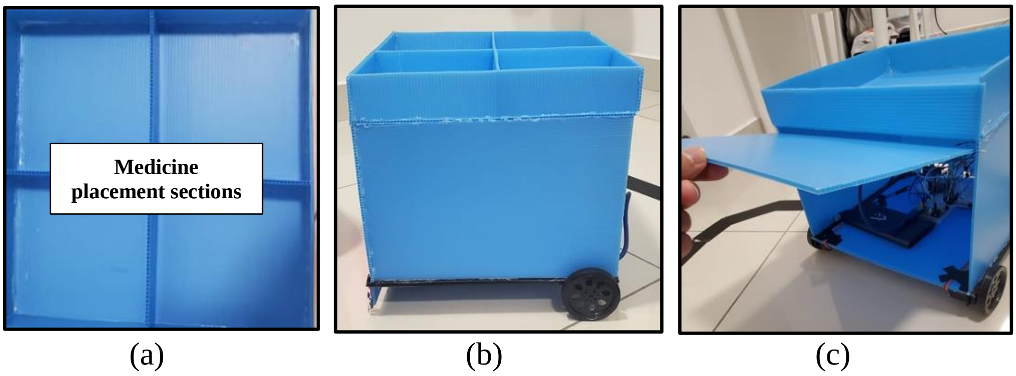


Figure 6. The overall body structure of line follower prototype (a) top view (b) side view (c) back view

Figure 7 shows the path of the line follower robot that will be undertaken for the medicine delivery at the patient's bed. In this path prototype, there will be four beds of patient for the pit stop before heading to the next bed. The robot will first start at the 'point of start' as shown below to start the delivery process after the medicines have been loaded onto the robot. The corner in this path design has been created to be 120° degrees which allowed the prototype to follow the line smoothly due to the huge size of the prototype.

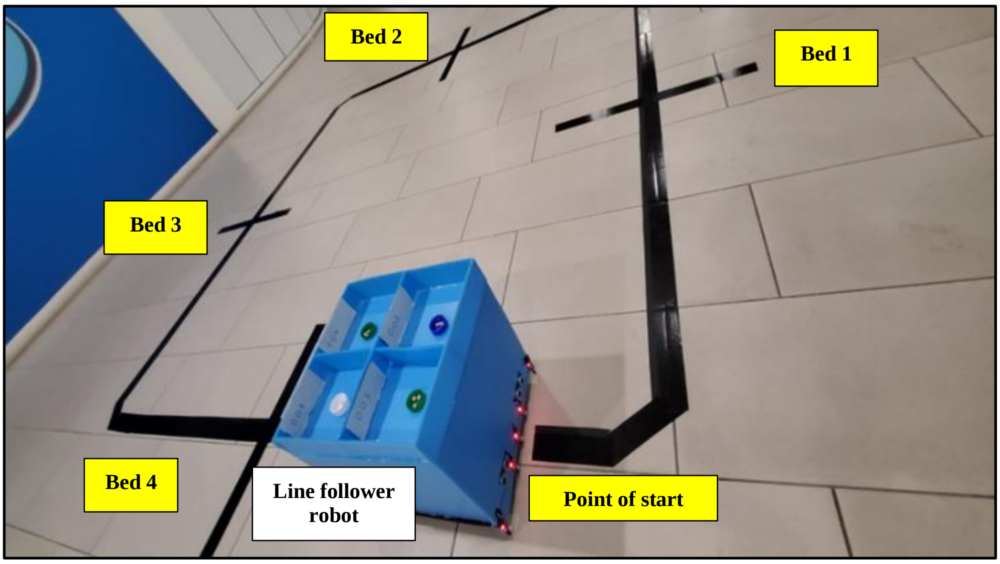
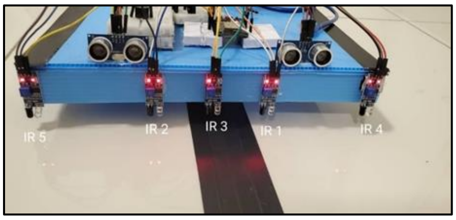
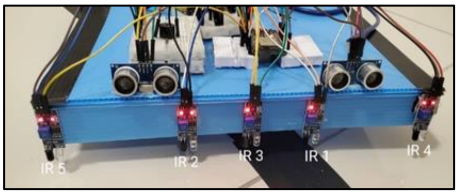
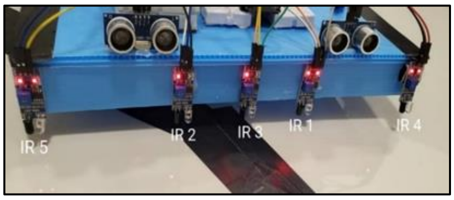
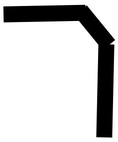
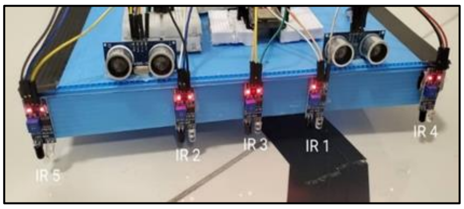
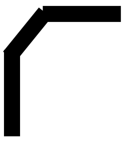
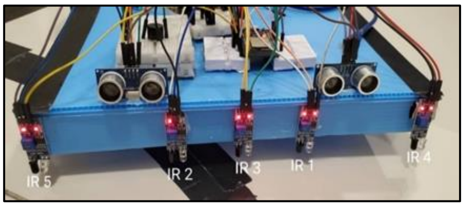
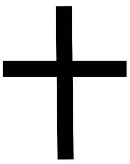
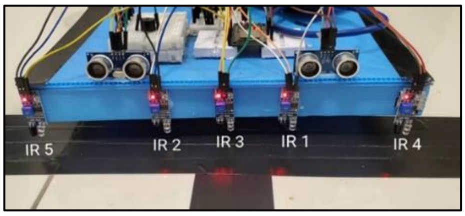
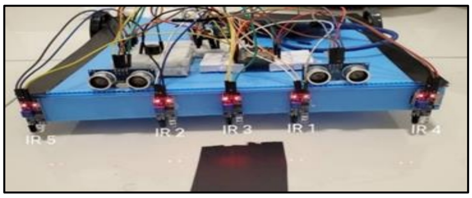


Figure 7. The path of the line follower robot that will be undertaken for the medicine delivered at the patient's bed

To demonstrate the working principles of the line follower robot based on the line structure, **Table 1** below shows the summary of how all five IR sensors detect a particular line structure and subsequently reflect the motor movement in the path.

Table 1. Summary of line detection with demonstration figure

Line structure	**Line follower robot detection figure**	**IR 1**	**IR 2**	**IR 3**	**IR 4**	**IR 5**	**Action**
		Off	Off	On	Off	Off	Move forward
		On	Off	On	Off	Off	Move forward
		Off	On	On	Of	Off	Move forward
		On	Off	Off	Off	Off	Turn left
		Off	On	Off	Off	Off	Turn right
		On	On	On	On	On	Stop 10 seconds to deliver medicine, then moves forward

		Off	Off	Off	Off	Off	Stop until the medical staff push the prototype to the starting point

Meanwhile, **Figure 8** shows the experiments of obstacle detection with the object placed in front of the robot at random distances and locations. The reading of the distance is recorded and tabulated for comparison.

(a) (b) (c)

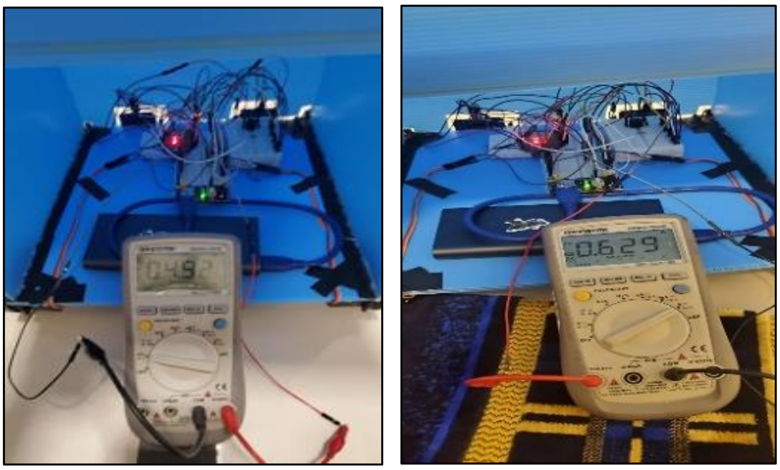
Figure 8. The obstacle detection demonstration (a) for huge object (b) small object on the right path (c) small object on front path

Table 2 shows the summary of the ultrasonic 1 and ultrasonic 2 sensors detection based on the distance from the object of obstacles and subsequently reflects the action taken by the line following robot which agrees well with the analysis of decision making done by Shubha (2019) and Surya et al (2018).

Table 2. Summary of detection obstacle

Distance	Ultrasonic 1	Ultrasonic 2	Buzzer	Action of prototype
>20cm	Off	Off	Off	Move
<20cm	On	Off	On	Stop with alarm on
<20cm	Off	On	On	Stop with alarm on
<20cm	On	On	On	Stop with alarm on

From the testing and measurement, the voltage from the power supply is 5V as input and the output voltage in the circuit is 4.92V as shown in **Figure 9a.** The maximum current fed from the power supply is less than 700mA as input to the circuit and the output current in the circuit was measured at 0.629A as described in **Figure 9b.** Hence, it can be concluded that this line follower robot fits the requirement addressed to function effectively.



(a) (b)

Figure 9. Testing and measurement of (a) output voltage (b) output current

Meanwhile, the testing and measurement of the robot speed of servo motor MG995 have been captured by using a tachometer. The result is represented as 36 rpm as shown in **Figure 10**. The speed presented is suitable to be implemented as it is considered a safe speed for operation around the patient’s ward. If any obstacles suddenly obstruct the path of the line, the robot has the capacity to stop smoothly without affecting the medicines carried on top of it. However, if the speed of the line follower robot needs to be increased, a larger supply voltage is required to support these two servo motors supported by a study done by Chaudhari (2019).

Figure 10. Servo motor speed measurement using the tachometer.

This project is built based on the C++ programming that served the microcontroller (ESP32) which controls all the equipment to operate as instructed. Based on **Figure 11**, the first part of the code is made to define the libraries, Wi-Fi network, variables, and pins that are used in this project for each component.

```
#include <ESP32Servo.h>
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "DSRSB2-1303 2.4Ghz";
const char* password =  "DSR112233";

const int trigPin1 = 2;
const int echoPin1 = 35;
const int trigPin2 = 15;
const int echoPin2 = 5;
Servo myservoR;
Servo myservoL;
long duration1;
int distance1;
long duration2;
int distance2;
int buzzer = 12;

int linePinM = 18;
int haslineM = LOW;  // LOW MEANS NO LINE

int linePinR = 19;
int haslineR = LOW;  // LOW MEANS NO LINE

int linePinL = 23;
int haslineL = LOW;  // LOW MEANS NO LINE

int linePinRS = 4;
int haslineRS = LOW;  // LOW MEANS NO LINE

int linePinLS = 13;
int haslineLS = LOW;  // LOW MEANS NO LINE
```

Figure 11. Initial part of the code

Figure 12 shows the second part of the code is void setup where the code run one time as soon as the program starts and it includes connecting ESP32 board to the Wi-Fi network, defines pins as input and others as output, defines pins for the two servo motors and subsequently initiates the serial communication.

```
void setup() {
WiFi.begin(ssid, password); //Connecting to the wifi

  while (WiFi.status() != WL_CONNECTED) { //if it's still not connected to wifi show the following message

    Serial.println("Connecting to WiFi..");
  }

  Serial.println("Connected to the WiFi network"); // when connected print this message

pinMode(led, OUTPUT);
pinMode(trigPin1, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin1, INPUT); // Sets the echoPin as an Input
pinMode(trigPin2, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin2, INPUT); // Sets the echoPin as an Input
  pinMode(linePinM, INPUT);
  pinMode(linePinR, INPUT);
  pinMode(linePinL, INPUT);
  pinMode(linePinRS, INPUT);
  pinMode(linePinLS, INPUT);
  myservoR.attach(21);
  myservoL.attach(22);
  Serial.begin(115200);
}
```

Figure 12. Void setup part of the code

The third part of the code is the void loop, and it is the longest part of the code due to the loop function when ESP32 is turned on. Thus, all the principles of line following, and obstacles detection are programmed in this section to demonstrate the variations in IRs line detection by using *if* statement. In this section, every principle is repeated twice as each of the ultrasonic 1 (for distance 1) and ultrasonic 2 (for distance 2) could not be combined in one *if* statement.

```
 else if (haslineL == HIGH && haslineM == HIGH && haslineR == HIGH && haslineL3 == HIGH && haslineR3 == HIGH && distance1 > 20) {


  myservoR.write(90);
  myservoL.write(90);
   digitalWrite(busser, LOW);
   delay(5000);
  myservoR.write(120);
  myservoL.write(60);
   digitalWrite(busser, LOW);
    HTTPClient http;

 http.begin("https://api.callmebot.com/text.php?user=@Alhasani772&text=patient+has+taken+the+medicine");
    unsigned int currentMillis = millis();
     String myString = String(currentMillis);

 http.addHeader("Content-Type", "text/plain");

int  Model = 33;
float UPdate = 1.2;

 int httpResponseCode = http.POST("{\"Device_Model\" : \""+String(Model)+"\",\"IP\": \""+WiFi.localIP().toString()+"\",\"Update\": \"" +Stri



   delay(1000);



 }
 else if (haslineL == HIGH && haslineM == HIGH && haslineR == HIGH && haslineL3 == HIGH && haslineR3 == HIGH && distance2 > 20) {


  myservoR.write(90);
  myservoL.write(90);
   digitalWrite(busser, LOW);
   delay(5000);
  myservoR.write(100);
  myservoL.write(80);
   digitalWrite(busser, LOW);
HTTPClient http;
```

Figure 13. Coding to stop at each patient's bed with notification alert

Based on **Figure 13**, there are two else *if* statements which have been programmed to stop at each patient's bed and subsequently send a notification message to the medical staff. Two *else-*

if statements are used in this part of the coding since Ultrasonic 1 (for distance 1) and Ultrasonic 2 (for distance 2) are implemented.

As Telegram was chosen as the acknowledgment platform for task accomplishment notification via CallMeBot, the link created from CallMeBot was embedded into the programming. **Figure 14** presents the notification messages from CallMeBot to indicate the successful delivery of the medicine. In future, this notification message can be modified based on the patient bed's number to make it more efficient for monitoring purpose.

Figure 14. CallMeBot API in Telegram

Conclusion

In this project, an autonomous robot that follows the line and detects obstacles is developed and demonstrated. This line follower robot prototype is composed of major components such as ESP32 Microcontroller, ultrasonic sensor, IR sensor, and buzzer using the platform Telegram via CallMeBot channel notification acknowledgement of medicine delivery. Based on the testing and measurement conducted, this line follower robot prototype is suitable and can be implemented for assistance in the healthcare provider especially in medicine delivering in a contagious environment. In future, this prototype can be further improved in terms of sharp turns for better efficiency and less restriction in path following by adding more motors to the platform for better curvature. Besides that, the medicine compartment can be integrated with suitable sensors to detect all medicines have been taken by the patient and can be taken by the required patient only, else an alarm will be on. This feature will help to ensure the patients will not accidentally leave behind any single medicine in that compartment or accidentally take the medicine of the other patients.

Acknowledgement

The authors would like to thank the Faculty of Engineering, Built Environment, and Information Technology (FoEBEIT), SEGi University for supporting this research work.

References

Barua, V., Arif, M., Uddin, S., Das, M., Shafiul, M., Nahar, N., & Pathak, A. (2020). An Ultrasonic Line Follower Robot to Detect Obstacles and Edges for Industrial and Rescue Operations. International Journal of Computer Applications, 176(31), 31.

Chaudhari, J., Desai, A., & Gavarskar, S. (2019). Line Following Robot Using Arduino for Hospitals. 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 330-332.

Kim, S.-J., Park, J., Na, Y., & Won, H. (2022). Autonomous Driving Robot Wheelchairs for Smart Hospitals. IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Yeosu, Korea, Republic of, 2022, pp. 1-2, doi: 10.1109/ICCE-Asia57006.2022.9954833.

Manikandan, P., Ramesh, G., Likith, G., Sreekanth, D., & Durga Prasad, G. (2021). Smart Nursing Robot for COVID-19 Patients. International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 839-842.

Pakdaman, M., Sanaatiyan, M. M., & Ghahroudi, M. R. (2010). A line follower robot from design to implementation: Technical issues and problems. 2010 the 2nd International Conference on Computer and Automation Engineering (ICCAE).

Pathak, A., Khan Pathan, R., Tutul, A., Tousi, N., Rubaba, A., & Bithi, N. (2017). Line Follower Robot for Industrial Manufacturing Process. International Journal of Engineering Inventions, 6(10), 10–17.

Shubha, P., & Meenakshi M. (2019). Design and Implementation of Healthcare Assistive Robot. 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).

Surya, M.S.V.S., Reddy, K.B., Kalyan, K.P., Murugan, S.S (2018). Smart and Intelligent Line Follower Robot with Obstacle Detection. International Journal of Research and Scientific Innovation (IJRSI), Volume V, Issue IV, ISSN 2321–2705.

Suryawan, S.S., Musamwar, S. M., Kolhe, S.R., Tengane, S.A., Hanumate, S.S., Sahare, P.H. (2019). Line Follower & Obstacles Avoider Robot. International Research Journal of Engineering and Technology (IRJET) , 460-468.

World Health Organiszation (WHO) (2021). World Health Organization Situation Report. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports