Turkish Journal of Engineering

https://dergipark.org.tr/en/pub/tuje

e-ISSN 2587-1366

Design and implementation of a cost effective vacuum cleaner robot

Anıl Eren[1], Hatice Doğan *[1]

[1]Dokuz Eylul U*niversity, Engineering Faculty, Electrical and Electronics Engineering Department, Izmir, Turkey*

Keywords
Vacuum Cleaner Robot
Robot Navigation
Robot Control
Robot Sensing System

ABSTRACT

Robot vacuum cleaners are among the first service robots to enter daily life. However, robot vacuum cleaners are currently inaccessible to many due to their high cost. In order for these robots to become widespread, they must be cheap and functional. In this study, the design and production of a low-cost, high-performance vacuum cleaner robot that can be controlled by smartphones is presented. This autonomous robot moves around the obstacles, vacuums the dust from the floor, performs basic navigation and also it is manually controllable. Designing of a vacuum cleaner robot is divided into three parts which are the mechanical, electrical and software design. In the mechanical design; the robot chassis and cleaning system have been designed and realized. In the electrical design; the electrical components such as batteries, motors, drivers, sensors and microcontroller have been selected and the power system has been designed by considering electrical requirements. In the software design; an easy to use Android application for remote control has been created, autonomous algorithms such as random walk and snake algorithms have been designed. Remote controlling of the robot has been provided by using Bluetooth connection between the Raspberry Pi and Android smartphone.

1. INTRODUCTION

Robot vacuum cleaners are one of the best examples of robotic technology that can be used in homes. The vacuum cleaner robot is a device designed to vacuum all dirt on flat floors without human intervention. In recent years, there have been many advances in vacuum cleaner robot technology to reduce the cost of robots and increase their capabilities. Nowadays, they have gained features such as being autonomous, remote-controlled, lightweight, small space footprint and more.

The vacuum cleaner robot collects external data from sensors mounted on it and uses this data to make various movements according to its navigation algorithm. While navigating and cleaning, the robot avoids obstacles, walls or stairs which are detected by the sensors (Asafa et al. 2018). Simpler vacuum cleaner robots on the market just bump into obstacles, walls and after that randomly change their position. The more complex ones use infrared, ultrasonic, optical and laser sensors to detect and avoid obstacles, walls and stairs. Their navigation models are more accurate due to their ability to create and navigate maps using a camera or laser scanner.

In the literature different types of cleaning robots have been designed (Ulrich et al. 1997; Oh et al. 2002; Dai and Chen 2007; Gao et al. 2007; Liu et al. 2011; Prabakaran et al. 2017; Adithya et al. 2019). The first vacuum cleaner robot called "Trilobite" was developed by James Dizon and manufactured in 1996 by Swedish household company Electrolux. Trilobite is complex and expensive. It creates a map of the room while cleaning and recharges automatically when its battery is low. It uses ultrasonic sensors to avoid obstacles and walls (Maruri et al. 2006). British technology company Dyson designed a vacuum cleaner robot called "DC06" in 2001; but did not go into the mass production because it was too expensive. iRobot company, founded by three robotists from the Massachusetts Institute of Technology, launched the robotic vacuum cleaner robot "Roomba, floor vacuuming robot" in 2002. This robot changes its direction when it counters wall or an object and it could detect dirty places on the floor. Also, the robot could avoid the stairs in order to prevent falling down. The main difference between the vacuum cleaner robots and conventional ones is the autonomy, which means the robots do the cleaning job by moving around without any human intervention. Also, vacuum cleaner

* **Corresponding Author**

(anilerenn@gmail.com) ORCID ID 0000-0001-7340-8590
*(hatice.dogan@deu.edu.tr) ORCID ID 0000-0003-0420-592X

Cite this article

Eren A & Dogan H (2022). Design and implementation of a cost effective vacuum cleaner robot. Turkish Journal of Engineering, 6(2), 166-177

Research Article / DOI: 10.31127/tuje.830282

Received: 23/11/2020; Accepted: 27/04/2021

robots are less noisy and very small in dimension compared to the conventional ones.

The main advantages and disadvantages of using a vacuum cleaner robot can be seen in Table 1.

Table 1. Advantages and disadvantages of vacuum cleaner robots

Advantages	Disadvantages
Save time and energy	Some models can be costly
Lesser noise than standard cleaners	Risk of getting stuck
Scheduling and tracking capabilities	Risk of flipped over
Cleaning by itself	Frequent dust bin changing required
Different cleaning modes	Long cleaning times
Can be remote controlled	Battery life
Lightweight and compact	Ineffective on stairs and thick surfaces
Cordless design	
Automatic Recharge & Docking	
Under furniture cleaning	
Good for people with limited mobility	
Some models have intelligent mapping	
Some models can detect dirt	

In summary, the vacuum cleaner robots do much more than what an ordinary vacuum cleaner offers without requiring human intervention. For widespread use of these robots, they must be functional, cost-effective, and controllable with common devices such as smartphones.

The aim of this study is to design and implement a low-cost, high-performance robot vacuum cleaner that can be controlled by android phones with inexpensive and easily accessible mechanical and electronic components. Most of the low-cost vacuum cleaner robots on the market use only basic switches as sensors for detecting obstacles. The designed robot uses ultrasonic and infrared sensors that enable the robot to detect obstacles before they hit them. In addition, it has a manual control feature and two different autonomous algorithms that are not found in many low-cost vacuum cleaner robots.

2. METHOD

Vacuum cleaner design is divided into three parts: mechanical, electrical and software design.

2.1. Mechanical Design

Mechanical design of the vacuum cleaner robot can be divided into 4 categories. In these categories; how the shape of the robot affects the navigation and cleaning, which driving mechanism used on the robot is suitable, what main materials that can be used as robot chassis and cleaning system design are explained.

2.1.1. The shape of robot body

There are different shapes of vacuum cleaner robots in the market, such as circular, D-shaped and square. If different models of vacuum cleaner robots in the market are examined, it can be appeared that there are few square-shaped vacuum cleaner robots than other circular and D-shaped robots. When different-shaped robots are compared functionally, square vacuum cleaner robots' corners cause them to get stuck in the gaps between obstacles. If these robots get stuck, human intervention is required, which makes the autonomous feature ineffective. Fewer corners the robot has, the lower probability of stuck (Stubbs 2019). D-shaped robots have less stuck probability than the square-shaped robot and the circular-shaped have the least stuck probability than the others. The main purpose of the D-shaped design is to enable the robotic cleaner to clean wall edges and corners more effectively. Circular-shaped robots are less effective about cleaning these wall edges and corners (Stubbs 2019). In order to eliminate this weakness, spinning side cleaning brushes are included. The size of brushes is another factor for the amount of space that the robot can clean in one go. The shape of a vacuum cleaner robot is also an important factor that how much damages it can cause when it collides with other objects. Circular vacuum cleaners have a lower risk to cause damage on the surface of any object they collide due to their soft round edges (Radu 2015). Aesthetics is another factor for vacuum cleaner robots that their shapes affect the sales in the market. Today's buyers largely prefer smooth rounded curves that are more aesthetic in robots (Stubbs 2019).

The shape of the vacuum cleaner robot is selected as circular because of lower risk of getting stuck which ensures the autonomy. Since the corner cleaning performance of circular robots is low, a brush is attached to it to increase performance. One brush is selected instead of two brushes to increase the operating time of the robot since two brushes would have more power consumption than the one brush.

2.1.2. Driving mechanism

Driving mechanism determines how a mobile robot moves from one place to another. There are different types of driving mechanisms that are used on mobile robots such as differential drive, skid steering, synchronous drive, Ackermann steering etc.

Differential drive is the most popular and widely used driving mechanism for mobile robots because it is the simplest and easiest to implement (Dudek and Jenkin 2010; Roh et al. 2013). Also it provides arbitrary motion and in-place rotation (zero radius), however its main disadvantage is the difficulty to maintain straight line motion due to independent motors. To ensure the straight line motion, identical motors should be used and the center of mass should be in the middle of the robot. As the motors deform after a while due to the weight of the robot, their identicality will deteriorate. In order to guarantee straight line motion, open or closed loop control system must be designed. If the system is open-loop, calibration will be needed to make the robot straight line motion.

As shown in Fig. 1, a differential drive robot has two individually driven wheels mounted in one line which are opposite to each other. Also a castor wheel can be added to balance the robot. Each wheel is controlled by a different motor to make different movements such as forward, reverse, rotating left or right. The speeds of individual motors determine the direction of the motion.

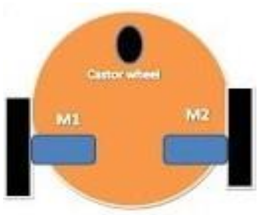


Figure 1. Differential drive robot (Dave 2020)

There are three different cases for the motion of the robot according to the individual motors' speeds (LaValle 2006). In case 1; if the motors have the same speed and rotating at the same direction, the amount of torque produced on both wheels are the same resulting the robot moves forward or backward. In case 2; if the motors have the same speed and rotating at opposite directions, torques are generated on both wheels which are equal in magnitude and opposite in direction resulting the robot rotates around itself left or right. In this case, the center of rotation is equidistant between the motors. In case 3, if the motors rotating at the same direction but have different speeds, the robot starts turning towards to left or right depending the motor that have higher speed. For example, if the left motor has higher speed, then the robot turning towards to right or vice-versa. The motion of the robot depending the individual motors' speed and direction is given in the Fig. 2.

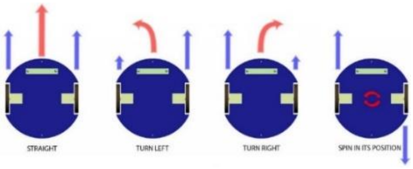


Figure 2. Differential drive robot motions (Ottoviani 2020)

2.1.3. Material and body size

Material is important factor for vacuum cleaner robot chassis that can carry the whole robot components without any deformation.

Base and ceiling of the robot chassis are made of plexiglass because of its durability. PVC foam is used for dust bin of the cleaning system. The diameter of circular base and ceil components of the robot chassis is selected as 33 cm in order to have enough space for electrical components and cleaning system. The ceiling part of the robot has been produced by using the same dimensions as the body plexiglass and it is supported by small plexiglass columns.

2.1.4. Cleaning system design

Vacuum cleaner robots have a cleaning system consist of vacuum motor, dust bin and brush. The vacuum motor cleans the dirt from the floor where the robot moves. Dirt taken from the floor goes to the dust container placed inside the robot. The dust container can be removed easily after it is full. The brush sweeps the floor and pushes the dirt into the area affected by the vacuum motor so the dirt can be vacuumed easily. Since vacuum cleaner robot is circular, the brush also helps the robot sweep dirt near walls or obstacles. In this way, the brush increases the cleaning performance of vacuum cleaner robots. As a vacuum motor, a DC blower can be used.

Figure 3. DC Blower (Blower Documentation, 2020)

While the circular side of the DC blower in Fig. 3 vacuums (sucks) the air, the rectangular side blows to the outside. The material used for dust bin is selected as PVC foam sheet. The dust bin is designed to cover the circular side of the blower completely to increase the air pressure that affects the cleaning performance. It is placed on the middle of the circular body. A plug-in plastic is added to the end of the dust bin space to reduce the distance between the floor and the robot, which improves cleaning performance. The specifications of the selected 12V 1A DC blower are:

- Speed: 3100 RPM
- Air Flow: 36.2 CFM
- Air Pressure (in Water): 1.35 Pa

2.2. Electronic Hardware Design

Electronic hardware design consists of electronic components of the vacuum cleaner robot. It can be divided into 4 categories; motors and motor drivers, sensors, microcontroller, power and charging system.

2.2.1 Motors and motor drivers

Different types of electric motors such as DC (Direct Current) motors, AC (Alternating Current) motors, servo motors and stepper motors can be used in robotic applications. For vacuum cleaner robots, DC motors can be used for the movement of the robot, vacuuming, and brushes.

Motors have two important variables that must be considered depending on the application. These variables are the maximum speed and the torque. The speed and the torque of the motor are inversely proportional. Generally, DC motors have high speeds and low torque values. DC motors used on the vacuum cleaner robots must have enough torque to carry the weight of the robot and provide movement.

Motor drivers are used to control the motors in terms of speed, rotation direction, angle etc. DC motor drivers have H-bridge ICs which consist of MOSFETs or BJTs acting as switches. Simple H-Bridge circuit with a DC motor is given in Fig. 4. In this figure, S1, S2, S3 and S4 are the switches. Different states of these switches determine the direction of the DC motor either clockwise or anti-clockwise. They can also be used to brake the motor. In Fig. 4, assuming that the direction of rotation of the DC motor is clockwise when the current flows from the left to the right, the operations of the motor for some different states of the switches are given in Table 2. Switch state "1" meaning the switch is closed (ON) and switch state "0" meaning the switch is opened (OFF).

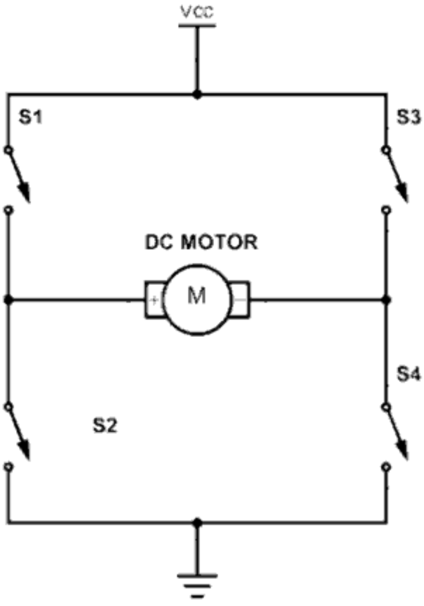


Figure 4. Simple H-Bridge

DC motors attached to wheels have been chosen in gear and micro size to provide sufficient torque for movement and take up small space on the robot body. High torque is required for the robot to move and carry its weight continuously. The voltage for geared DC motors has been selected as 6 V because if the required torque is provided with low voltage, the power consumption will be low. Motor speed has been chosen as 60 RPM that is enough for a vacuum cleaner robot. Also 3-6 V DC motor with gearbox is used for the rotation of the brush.

Table 2. H-bridge motor operations

S1	S2	S3	S4	Operation
1	0	0	1	Clockwise Direction
0	1	1	0	Anti-Clockwise Direction
0	1	0	1	Braking Mode

The motor driver module "L298N" is used to control DC motors. This motor driver has dual H-bridge inside and it can control speed and direction at the same time individually with PWM.

2.2.2 Sensors

Sensors are the most important and essential electrical components of vacuum cleaner robots because robots acquire information about their environment through sensors. They are indispensable for accurate navigation.

The most commonly used sensors are the infrared and ultrasonic sensors for the detection of walls, obstacles or cliffs that are close to robot. Both of the ultrasonic and infrared sensors have the transmitter and receiver parts. The transmitter emits a signal and if there is an object in the way of the signal, the signal reflected back by hitting the object. This signal is detected by the receiver. While the distance is measured according to the intensity of the reflected signal in the infrared sensor, it is measured according to the reflection time in the ultrasonic sensor.

Ultrasonic and infrared sensors are used together on the vacuum cleaner robot. Since the performance of the infrared sensors depends on the lighting conditions and their field of view is very low, ultrasonic sensors have been chosen as the main distance measurement sensors. The infrared sensors are the auxiliary sensors for the distance measurement and they are used for the detection of low density objects. Vacuum cleaner robots in the market have lots of infrared sensors because field of view of one infrared sensor is very low.

The design includes 3 ultrasonic sensors and 2 digital infrared sensors placed 60 degrees apart in front of the vacuum cleaner robot. Ultrasonic sensors are placed at most front-left, most front-right and at the front. Infrared sensors are placed between these ultrasonic sensors. In addition, an ultrasonic sensor is placed under the front of the robot as a cliff sensor to prevent falling from the stairs. This sensor let the vacuum cleaner robot know when it's near a "cliff," such as stairs or a balcony. Placement of the sensors is given in Fig. 5.

The ultrasonic sensor model selected is HC-SR04P with operating voltage between 3V and 5V. The detection range of this ultrasonic sensor is 2 cm to 400 cm. The KY-032 model is used for the infrared sensor. The detection range of this infrared sensor is maximum 10 cm and is sufficient for vacuum cleaner robots. Sensing distance can be adjusted by the potentiometer on the sensor board.

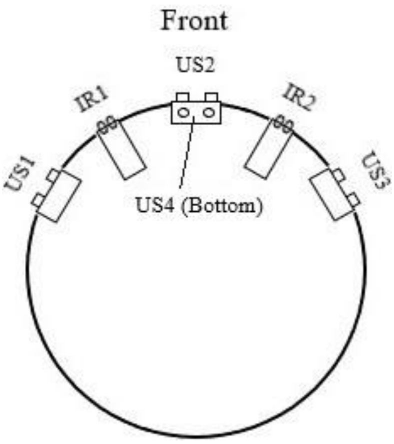


Figure 5. Sensor placement (robot top view)

2.2.3 Microcontroller

The microcontroller is the brain of the vacuum cleaner robot. It controls the robot by sending signals to the motor drivers. These control signals are generated based on the sensor data and the navigation algorithm. There are different types of microcontrollers such as Arduino, Raspberry Pi, BeagleBone. Raspberry Pi and BeagleBone are small computers running Linux OS inside. These microcontrollers can be programmed by using C, C ++ or Python languages.

To choose the microcontroller model; first, the total pin numbers of the sensors and motor drivers are considered. In the design, 4 HC-SR04P ultrasonic sensors, 2 KY-032 infrared sensors and 2 L298N motor drivers are selected. If the supply (Vcc) and ground (GND) pins are not considered, totaly 17 GPIO pins are needed. The Raspberry Pi 3B+ provides this total number of GPIO pins. It is also inexpensive compared to other microcontrollers.

All pins of the Raspberry Pi are digital. It has Wi-Fi and Bluetooth modules that will be useful to control the vacuum cleaner robot remotely. The CPU power and the capacity of RAM are sufficient for the vacuum cleaner robot.

2.2.4 Power and charging system design

Batteries provide energy to keep vacuum cleaner robots running for a specified time. This time should be high enough for the robot to clean a certain area before the battery runs out. These batteries must meet the minimum requirements such as operating voltage and current to supply the electrical components used. The weight of the batteries is also another important factor to consider. Batteries should be as light as possible to reduce the weight of the robot. If the weight is increased, the power consumption will increase and the batteries will drain faster.

Li-ion (lithium ion) batteries are chosen as the main energy source of the robot. Li-ion batteries have high energy, high capacity and low maintenance characteristics. They can be recharged in a short time and are lightweight. However, Li-Ion batteries need protection circuit against over charge and over discharge (Elias et. al 2003; Chen et. al 2019).

Since one Li-on battery is not enough for the robot, 4 li-on batteries that connected series (4S1P) are used. If the nominal voltage of one battery is used (INR18650-30Q Li-on Battery Datasheet, 2014), the maximum voltage can be calculated as

$$Maximum\ Voltage = 3.61 * 4 = 14.44\ V \qquad (1)$$

Since the batteries are connected series, the current and capacity are not changed, only the total voltage is increased.

For the 4S1P battery system, the 4S1P BMS (Battery Management System) charging and protection circuit is chosen to prevent the batteries from overcharging and deep discharging because overcharging and deep discharging may damage li-on batteries. The BMS circuit also provides balanced charging and discharging for 4 batteries, which means that 4 batteries are simultaneously charged and discharged.

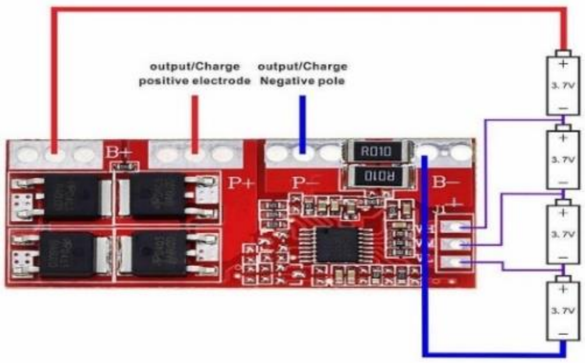


Figure 6. YH11047A4S1P (BMS protection board, 2020)

In the BMS protection board given in the Fig. 6, the anode and cathode of every battery is connected to the board to provide balanced charge and discharge. The batteries can be charged by connecting the charger to the output of the board. Since 4 li-on batteries have total voltage 14.44 V, 15 V charging adapter can be used to charge the batteries.

The total voltage of the batteries connected series cannot be used directly. This voltage is high for the other components such as motors, motor drivers and Raspberry Pi so that this voltage must be reduced to appropriate levels. For this purpose, DC – DC step-down voltage converters can be used.

2.3 Component Placement and Electrical System Layout

The component placement of the prototype vacuum cleaner robot is given in Fig. 7 and the layout of the electrical system is given in Fig. 9. The components in the figure marked with letters are:

A: Raspberry Pi 3B+ Microcontroller
B: Vacuum Motor (12 V 1 A DC Blower) And Dust Bin
C: 18650 4S1P Li-on Battery
D: YH11047A 4S1P BMS Protection Board
E: LM2596-ADJ DC-DC Step Down Voltage Regulator 3A
F: 6 V 60 RPM Micro DC Reducer Motor
G: L298N Dual Channel Motor Driver
H: HC-SR04P Ultrasonic Sensor
I: KY-032 Digital Infrared Sensor
J: 3-6 V DC Geared Motor For Brush

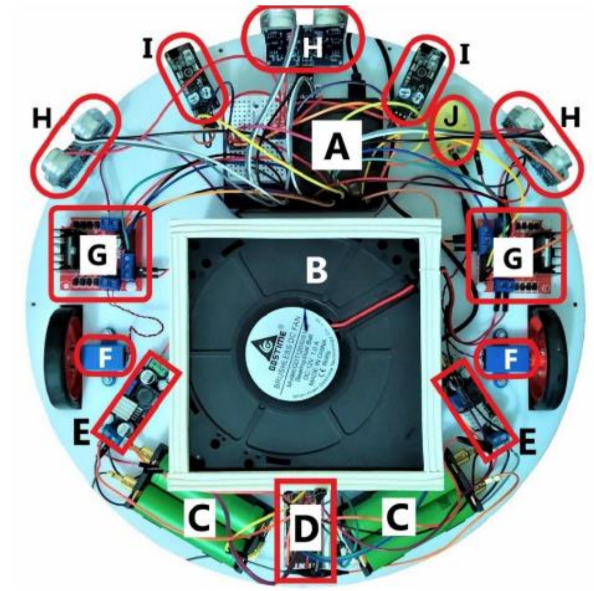


Figure 7. Robot top view and component placement

2.4 Software Design

The robot needs remote control software to be controlled by a smartphone and navigation software to prevent it from hitting the obstacles it encounters and to clean a room efficiently.

2.4.1 Remote control software

In order to control the robot by using Android smartphone, an Android application is needed. With this remote control application, the movement of the robot, the speed of the brush and the vacuum motor can be controlled, and the navigation algorithm can be selected. This Android application have been created by using MIT App Inventor.

MIT App Inventor is a programming environment that is used for building functional applications for Android and iOS. It has a GUI (graphical user interface) to develop applications by using visual functional blocks. Simple and complex applications can be created in a short time. Bluetooth communication has been used to send commands from the smartphone to the Raspberry Pi. The GUI of the developed application is given in the Fig. 8.

In Fig. 8a; buttons, switches and slider are invisible. After the Bluetooth image has been clicked, the Bluetooth connection selection page appears. In this page, the MAC address and the name of the Raspberry Pi mounted on the cleaning robot can be seen. After selecting the Raspberry Pi and the Bluetooth connection is made, the screen given in the Fig. 8b appears. The buttons “Forward”, “Left”, “Right” and “Backward” are used to move the robot manually. Vacuum motor and the brush motor can be activated or deactivated by using ON/OFF switch on the GUI.

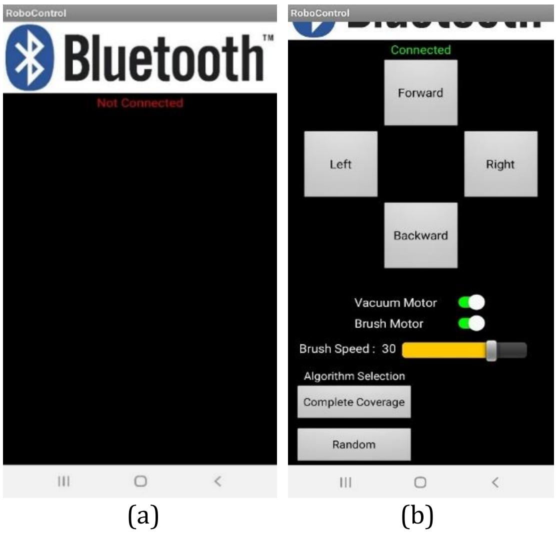


Figure 8. Android application (a) without Bluetooth connection, (b) with Bluetooth connection

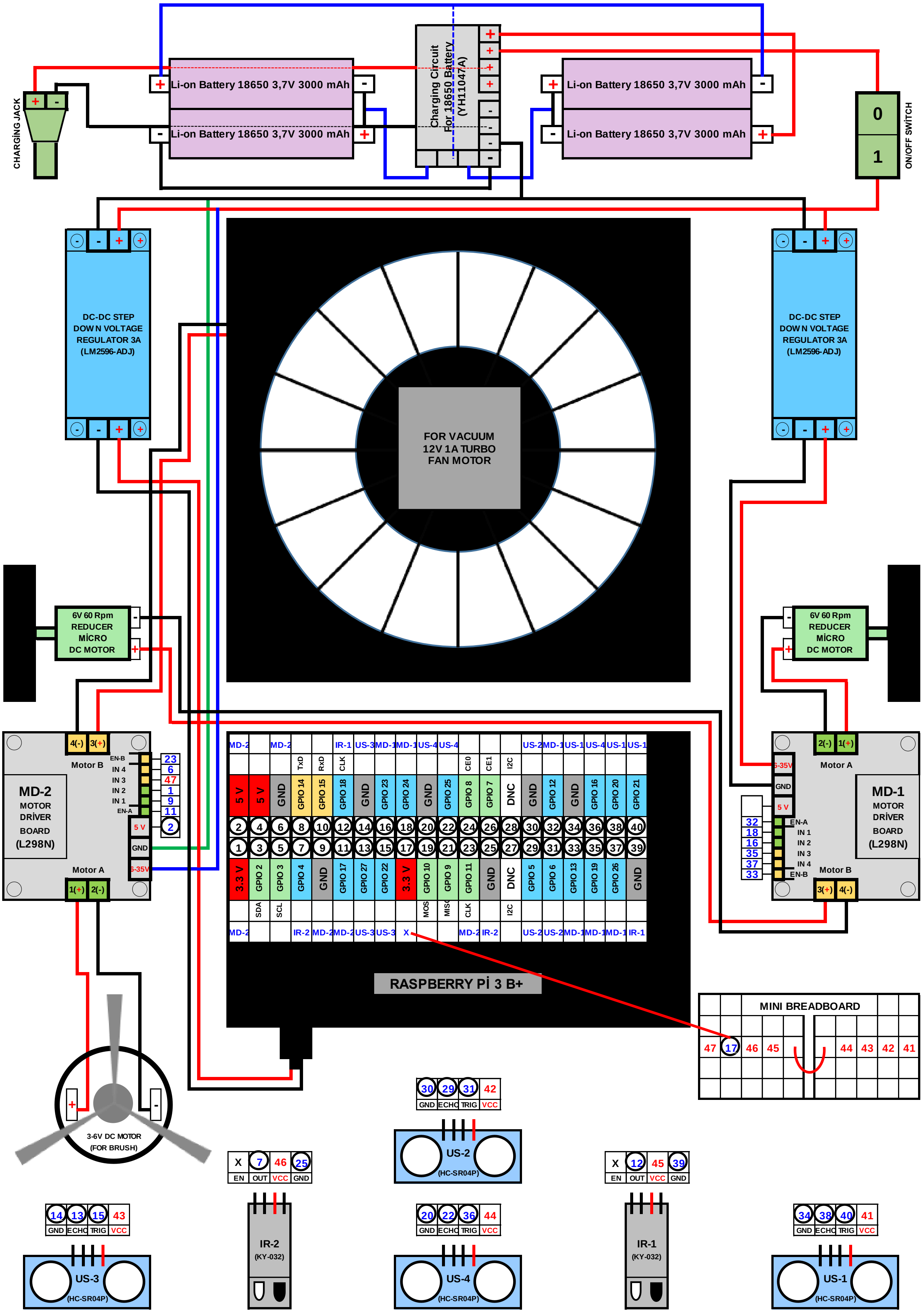
The brush speed slider only visible if the brush motor is activated. This slider can be used to adjust the speed of the rotating brush. The speed of the robot can be adjusted by adding a slider to the application since the motor speed can be controlled by PWM signals. Also the buttons under the algorithm selection are used to change current autonomous algorithm. Robot operates in autonomous mode without Bluetooth connection. When the Bluetooth connection is made, the robot stops and waits for commands from the smartphone. Closing the application will make the robot works in autonomous mode again.

2.4.2 Navigation software

Path planning for autonomous vacuum cleaner robots is very important as the efficiency of the robot depends on it (Hasan and Reza 2014). Path planning or navigation algorithms decide how and where the vacuum cleaner robot will move in an environment. The problem for this type of robots is to decide how to cover a room completely in a shortest possible time with efficient cleaning. The most efficient navigation algorithm is the one that has the mapping ability. The algorithms that create a map of the environment around the robot and try to position it on the map are called SLAM algorithms. In order to use SLAM algorithms, expensive sensors such as RPLIDAR must be used.

Since designing a cost effective robot is one of the main goals, expensive sensors such as LiDAR are not used in the design and simple navigation algorithms are preferred.

In accordance to the robot design, random walk and snake algorithms are designed. The random walk algorithm is widely used in most low-cost vacuum cleaner robots that do not have expensive sensors like RPLIDAR. The snake algorithm can be used to clean all the free space the robot can move in.


Charging Circuit
For 18650 Battery
(YH11047A)
Li-on Battery 18650 3,7V 3000 mAh
Li-on Battery 18650 3,7V 3000 mAh
Li-on Battery 18650 3,7V 3000 mAh
Li-on Battery 18650 3,7V 3000 mAh
CHARGİNG JACK
0
1
ON/OFF SWİTCH
DC-DC STEP DOWN VOLTAGE REGULATOR 3A (LM2596-ADJ)
DC-DC STEP DOWN VOLTAGE REGULATOR 3A (LM2596-ADJ)
FOR VACUUM 12V 1A TURBO FAN MOTOR
6V 60 Rpm REDUCER MİCRO DC MOTOR
6V 60 Rpm REDUCER MİCRO DC MOTOR
MD-2 MOTOR DRİVER BOARD (L298N)
4(-) 3(+) Motor B EN-B IN 4 IN 3 IN 2 IN 1 EN-A 5 V GND 6-35V Motor A 1(+) 2(-)
23 6 47 1 9 11 2
MD-1 MOTOR DRİVER BOARD (L298N)
2(-) 1(+) Motor A 6-35V GND 5 V EN-A IN 1 IN 2 IN 3 IN 4 EN-B Motor B 3(+) 4(-)
32 18 16 35 37 33
RASPBERRY Pİ 3 B+
MINI BREADBOARD
47 17 46 45 44 43 42 41
3-6V DC MOTOR (FOR BRUSH)
US-2 (HC-SR04P)
30 29 31 42
GND ECHO TRIG VCC
US-4 (HC-SR04P)
20 22 36 44
GND ECHO TRIG VCC
US-3 (HC-SR04P)
14 13 15 43
GND ECHO TRIG VCC
US-1 (HC-SR04P)
34 38 40 41
GND ECHO TRIG VCC
IR-2 (KY-032)
X 7 46 25
EN OUT VCC GND
IR-1 (KY-032)
X 12 45 39
EN OUT VCC GND

MD-2		MD-2			IR-1	US-3	MD-1	MD-1	US-4	US-4				US-2	MD-1	US-1	US-4	US-1	US-1
			TxD	RxD	CLK						CE0	CE1	I2C						
5 V	5 V	GND	GPIO 14	GPIO 15	GPIO 18	GND	GPIO 23	GPIO 24	GND	GPIO 25	GPIO 8	GPIO 7	DNC	GND	GPIO 12	GND	GPIO 16	GPIO 20	GPIO 21
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
3.3 V	GPIO 2	GPIO 3	GPIO 4	GND	GPIO 17	GPIO 27	GPIO 22	3.3 V	GPIO 10	GPIO 9	GPIO 11	GND	DNC	GPIO 5	GPIO 6	GPIO 13	GPIO 19	GPIO 26	GND
	SDA	SCL							MOS	MISO	CLK		I2C						
MD-2			IR-2	MD-2	MD-2	US-3	US-3	X			MD-2	IR-2		US-2	US-2	MD-1	MD-1	MD-1	IR-1

Figure 9. The layout of the electrical system

This algorithm has been designed considering the open-loop system and dimensions of the robot. Algorithms to be run on the robot can be selected from the mobile application. The algorithm selection process is done using multithreading to immediately change the running algorithm.

2.4.2.1 Random walk algorithm

The random walk algorithm is based on the random motion of the vacuum cleaner robot. The main aspect of this algorithm is that it does not require a precise route plan. The disadvantage of the random walk algorithm is that the robot passes the same area several times.

In random walk based algorithm, the vacuum cleaner robot moves in forward direction until an obstacle or wall is detected by the sensors. Then the robot stops, and turns arounds depending the random number generated by microprocessor. Block diagram of the random walk algorithm is given in the Fig. 10.

The random number determines the rotation time. Random rotation time (RRT) is adjusted as

$$RRT = 1\, sec + \, random\ time\ [0 - 1.5\, sec] \qquad (2)$$

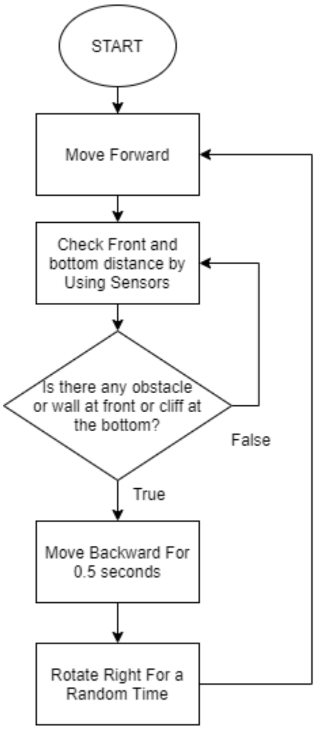

START
Move Forward
Check Front and bottom distance by Using Sensors
Is there any obstacle or wall at front or cliff at the bottom?
False
True
Move Backward For 0.5 seconds
Rotate Right For a Random Time

Figure 10. Block diagram of random walk algorithm

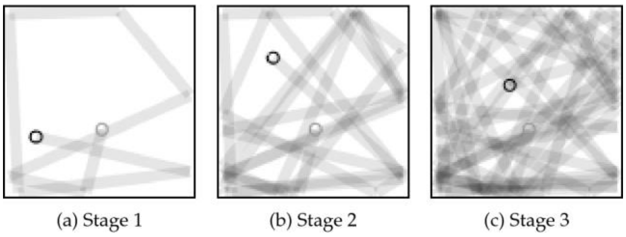

(a) Stage 1 (b) Stage 2 (c) Stage 3

Figure 11. Random walk based algorithm (Edwards and Sörme 2018)

Random walk based algorithm in a square room without any obstacle is given in the Fig. 11 at different times. The black circle in the square indicates the vacuum cleaner robot, the gray circle indicates the starting point of the robot and gray paths indicate the way that the robot has passed. Darker gray path means more times the robot passed the same area. Notice that, in this random walk based algorithm; the robot passes the same area several times which increases the time to clean the entire room completely. Also, in stage 3 in Fig. 11, there are some small areas that have not been cleaned.

2.4.2.2 Snake algorithm

In this algorithm, the vacuum cleaner robot makes S – shaped motion which is snake-like movement. This is the fastest algorithm to cover the entire room area in a certain time. However, snake algorithm has a disadvantage that; in the realization of this pattern, there may many errors occur due to the lots of stopping and rotating motion of the robot. Consider the case that the wheels are not aligned and the sensors are not accurate. The summation of small errors will create bigger errors. Due to this disadvantage, the snake pattern is preferred only in local and small areas in order to prevent positional errors accumulation (Edwards and Sörme 2018).

The snake algorithm is improved for the case when obstacles exist in a room. An algorithm has been designed and implemented based on the region filling navigation using specific templates (Oh et. al 2004). Block diagram of the improved snake algorithm is given in the Fig. 12.

At first, the robot moves forward if no obstacle detected by the sensors. According to the block diagram in Fig. 12, this algorithm has 4 different conditions based on detected obstacle or wall direction. These are;

- Condition 1: Obstacles at front and right directions
- Condition 2: Obstacles at front and left directions
- Condition 3: Obstacles at only front direction
- Condition 4: Obstacles at front, right and left directions

Height from robot body to the floor is measured by the ultrasonic cliff sensor. When height exceeds 6 cm, height flag is set. If the height flag is set, the robot assumes there is a cliff in front of it. The robot moves back 0.5 seconds per cycle until the sensor does not detect the cliff. The height flag resets after the measured height is smaller than 6 cm.

A moving direction flag is used to enable the robot cleans the room from left to right or right to left. Initially, the moving direction is set as “right”. This means, the cleaning robot starts to clean the room from where it is, to the right until it reaches a wall or an unpassable obstacle.

After reaching a wall or an unpassable obstacle, cleaning direction becomes left. The movements of the robot according to the conditions can be exlained as follows:

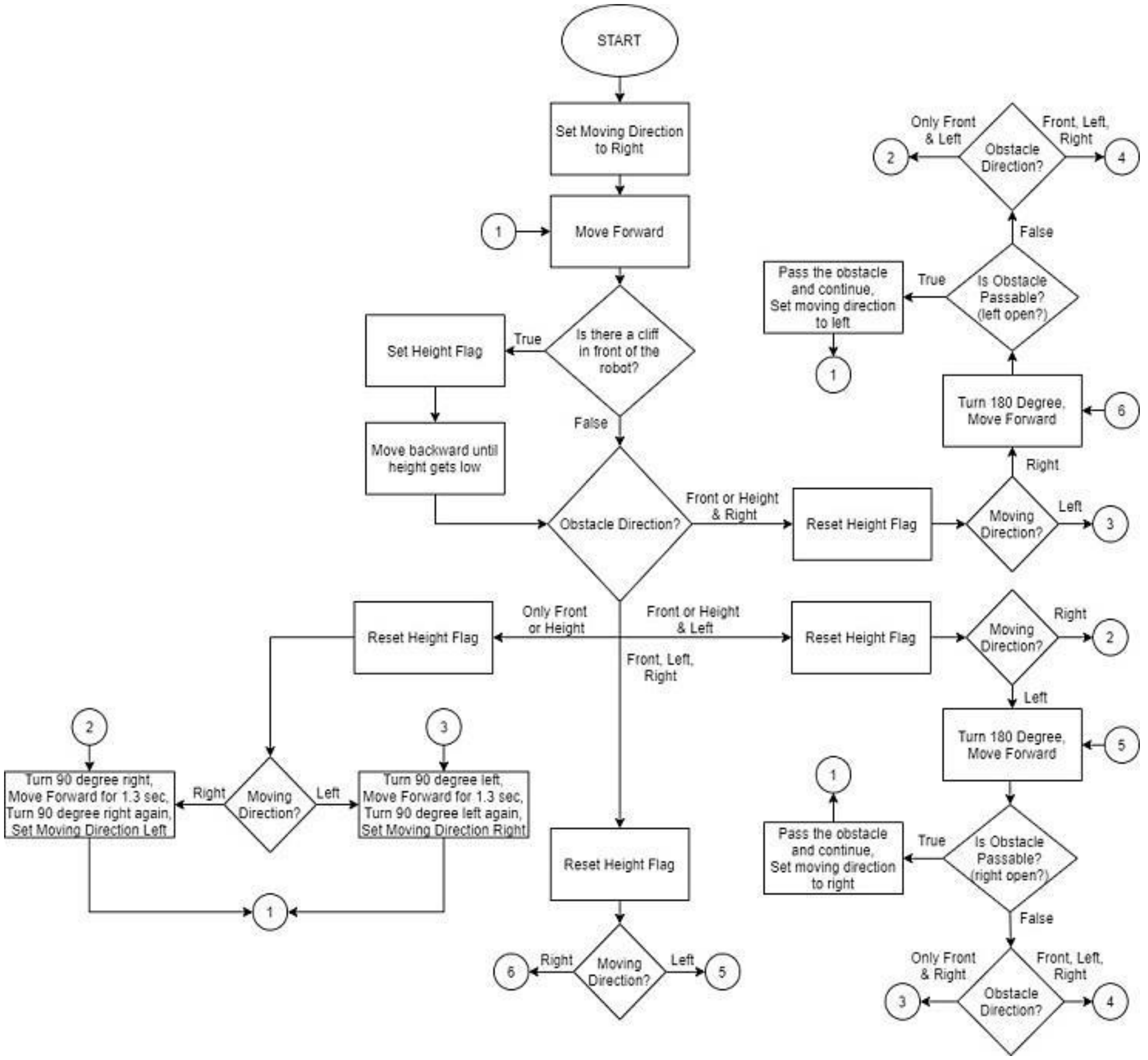

START
Set Moving Direction to Right
1
Move Forward
Is there a cliff in front of the robot?
True
Set Height Flag
Move backward until height gets low
False
Obstacle Direction?
Front or Height & Right
Reset Height Flag
Moving Direction?
Right
Left
3
Turn 180 Degree, Move Forward
6
Is Obstacle Passable? (left open?)
True
Pass the obstacle and continue, Set moving direction to left
1
False
Obstacle Direction?
Only Front & Left
2
Front, Left, Right
4
Only Front or Height
Reset Height Flag
Front or Height & Left
Reset Height Flag
Moving Direction?
Right
2
Left
Turn 180 Degree, Move Forward
5
Is Obstacle Passable? (right open?)
True
Pass the obstacle and continue, Set moving direction to right
1
False
Obstacle Direction?
Only Front & Right
3
Front, Left, Right
4
Front, Left, Right
Reset Height Flag
Moving Direction?
Right
6
Left
5
Moving Direction?
Right
Left
2
Turn 90 degree right, Move Forward for 1.3 sec, Turn 90 degree right again, Set Moving Direction Left
3
Turn 90 degree left, Move Forward for 1.3 sec, Turn 90 degree left again, Set Moving Direction Right
1

Figure 12. Improved snake algorithm block diagram

Condition-1: The robot moves according to the moving direction. If the moving direction is right, the robot cannot directly move to the right since there is an obstacle at the right side. In this case robot rotates 180 degree. After rotating 180 degrees, it moves forward until it passes the obstacle. When the obstacle is passed, the robot moves next to the obstacle and continues to clean the room by making snake movement again. If the obstacle next to the robot is an unpassable obstacle or a wall, the robot will move forward until an obstacle or wall is detected in front of it. Then it reverses the cleaning direction according to the obstacle direction and continues cleaning. If the moving direction is left, the robot can directly move to the left as there is no obstacle at the left side. Movement patterns of the robot for the condition-1 are given in Fig. 13. The dotted lines and the arrows in the figure represent the past movement of the robot until the condition satisfies.

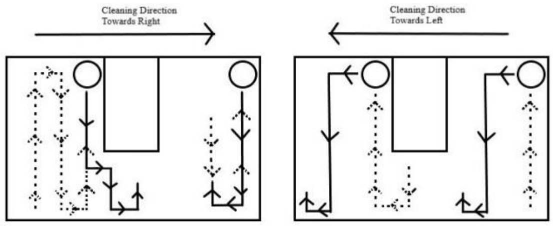

Cleaning Direction Towards Right
Cleaning Direction Towards Left

Figure 13. Condition-1 movement patterns

Condition-2: Robot acts similar to the movement given in the condition 1. The only difference in this condition is the obstacle direction (Fig. 14).

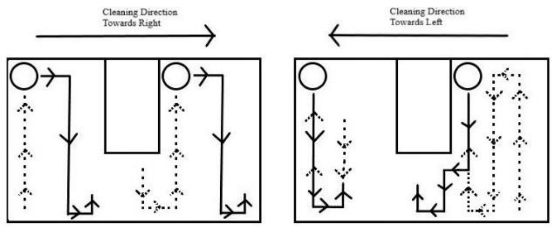

Cleaning Direction Towards Right
Cleaning Direction Towards Left

Figure 14. Condition-2 movement patterns

Condition-3: Since there is no obstacle at robot's left and right sides, the robot moves left or right according to the moving direction flag. After it turns left or right, the robot moves forward for 1.3 seconds (moving distance equals to the robot's diameter) and turns left or right again according to the moving direction flag. Then the robot moves forward and reverses the flag.

In summary, the robot makes S-shaped movement when there is no obstacles and walls at the left and right side of the robot (Fig. 15).

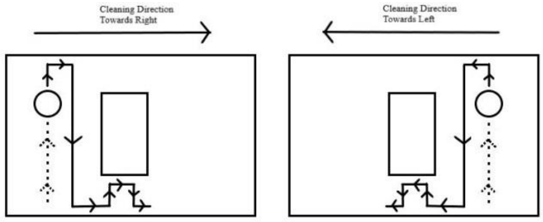

Cleaning Direction Towards Right
Cleaning Direction Towards Left

Figure 15. Condition-3 movement patterns

Condition-4: The robot rotates 180 degree and moves forward until it passes the obstacle according to the moving direction and continues to clean. If the obstacle is unpassable or there is a wall, the robot will reverse the direction of cleaning (Fig. 16).

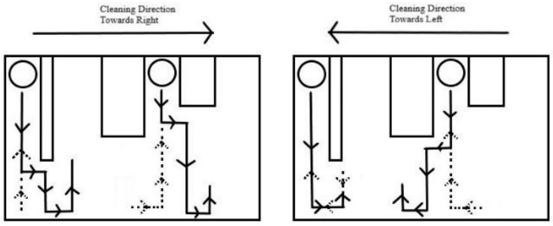

Cleaning Direction Towards Right
Cleaning Direction Towards Left

Figure 16. Condition-4 movement patterns

2.4.3 Switching Between Remote Control and Autonomous Modes

Switching between the remote control and autonomous mode needs to be as fast as possible because the user does not want to wait for the switching time. The Bluetooth connection must be checked every time to switch modes. Switching delay is not a problem for switching from remote control mode to the autonomous mode. When the Android application is closed or the Bluetooth connection is disabled, the robot switches from remote control mode to autonomous mode immediately. Since the code execution process is a serial process normally, there will be delay occurred for switching from autonomous mode to the remote control mode. In order to eliminate this delay, multithreading can be used.

Multithreading is the ability of a CPU (Central Processing Unit) to executes a process in multiple threads in parallel. It provides execution of the parts of a process independently. The speed of the computation can be significantly increased by using multithreading. In multithreading, threads share the same memory, meaning they can use the same variables defined in a process.

Two threads are used: One thread is used to check the Bluetooth connection every time while the other thread is used for the execution of autonomous algorithms. When the Bluetooth connection is requested from the smartphone and is accepted by the Raspberry Pi, the thread used for autonomous algorithm is killed. After the Bluetooth connection disabled (switching to autonomous mode), thread for autonomous algorithm starts again and other thread starts to check Bluetooth connection.

3. RESULTS

After the mechanical and electrical parts have been combined, the robot chassis has carried all the electrical components without any deformation. Implemented robot can be seen in the Fig. 17 and 18.

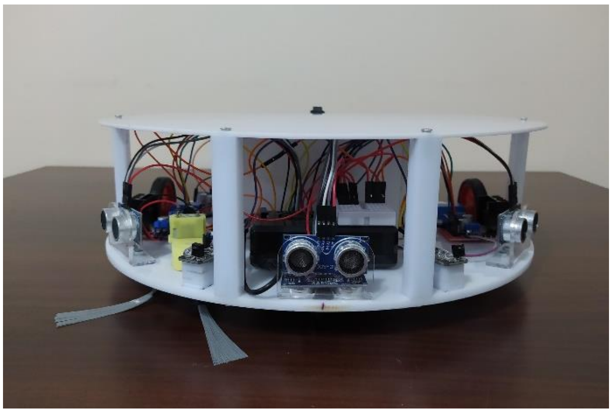


Figure 17. Front view of the robot

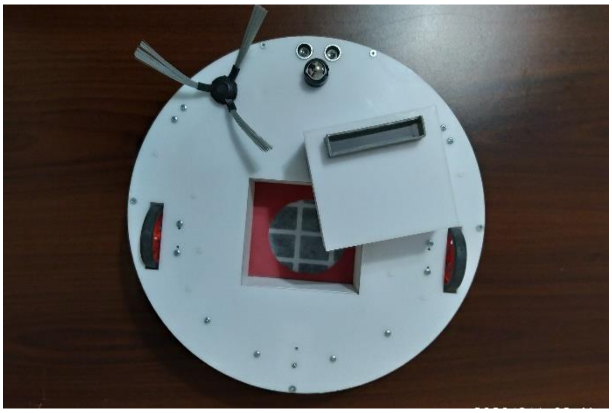


Figure 18. Bottom view of the robot

When the electricity system is energized; all electrical components operated smoothly. The short circuit test showed that the power and charging system automatically cut off the power from the batteries. In addition, a charge test has been carried out to check whether the batteries can be charged or not. As a result, the batteries have been charged from its barrel jack. The robot operated for about 1.5 hours with fully charged batteries.

The remote control mode has been tested using the designed Android application. As a result, this mode has been worked as expected. The robot has executed every given command without delay. Remote control mode has also be used for testing and calibration of the robot movement. At the end of the calibration, straight line motion has been guaranteed. The software code for the autonomous mode was tested and the results were satisfactory. The vacuum cleaner robot has successfully collected the crumbs on the ground.

The voltage across four series batteries has been measured when they are fully charged and found as 16.51 V. The current measurement has been done when the robot is working in autonomous mode and found as 0.76 A. The power consumption (PC) of the designed robot has been calculated by using the measured voltage and current:

$$PC = 16.51\,V\,x\,0.76\,A = 12.38\,W \qquad (3)$$

Table 3 shows the specifications of the implemented robot with the specifications of two low-cost models on the market, Ecovacs Deebot N79S and Roomba 675.

Table 3. Specifications of vacuum cleaner robots

	Implemented Robot	Ecovacs Deebot N79S	Roomba 675
Dimensions (Diameter x Height) in cm	33 x 12	35.3 x 8.4	33 x 9.4
Weight (kg)	3.5	4.3	3.5
Battery Life (mins)	150	95	104
Power (W)	12.38	25	33
Remote Control	No	Yes	No
Phone Control	Yes	Yes	Yes
Noise Level (dB)	52	67	65
Brush Number	One	Two	One
Navigation Algorithm	Random, Snake	Auto (Random), Edge, Spot	Random, Spot

4. DISCUSSION

Ecovacs Deebot N79S and Roomba 675 have similar specifications with the other low-cost vacuum cleaner robots in the market. If the specifications of the implemented robot and the other two robots are compared; their dimensions are almost equal but the height of the implemented robot is higher due to the placement of the dust bin and components.

The weight of the implemented robot is lighter than the Ecovacs Deebot and is equal to Roomba 675. Implemented robot consumes much less power than its counterparts. Since the robot can be controlled by smartphones, there is no need for remote control, which leads to a reduction of production costs. It's the same for the Roomba 675. Ecovacs Deebot has both remote and phone controls. The noise level of the implemented robot is lower than the others. The implemented robot and Roomba 675 have one brush while the Ecovacs Deebot has two brushes. Two brushes need more power consumption and one brush is enough for cleaning.

Three of the robots have their own random algorithms. The implemented robot has a snake algorithm that allows it to clean the entire room in a short time. The snake algorithm is superior to the random algorithm as it saves time and leaves no uncleaned space. Ecovacs Deebot and Roomba 675 have a spot algorithm that can be used to clean a specific area in a spiral motion, but the snake algorithm is also sufficient to clean certain spots. Therefore, there is no need for a spot navigation algorithm.

5. CONCLUSION

Robots are widely used in industry for the mass production of consumer and industrial goods. In order for the use of robots to become widespread in household chores as well as industry, robots must fully perform their functions, be easy to use and most importantly, must be low-cost.

In this paper, the design and implementation of a cost-effective fully functional vacuum cleaner robot is given. Inexpensive, functional and easily accessible components and materials have been selected in the mechanical and electrical design of the robot. The prototype robot can act autonomously and manually. In autonomous mode, the robot cleans the room by going around the obstacles according to the selected algorithm. With the cliff sensor, the robot has been prevented from falling through spaces such as stairs. In remote control mode, navigation can be controlled by a smartphone. A user-friendly android application has been developed where the user can control the brush and vacuum motors and select the navigation algorithm.

Random walk and snake algorithms have been used as the navigation algorithm. Since the robot does not know its exact position and cannot plan the navigation, it has to pass through the same place several times while cleaning a room. This increases the cleaning time. With the use of expensive and advanced sensors such as lidar and camera, it is possible to know the exact position of the robot in the room and plan the navigation accordingly. In order to process the information coming from the sensors, a more advanced microcontroller with more pins is also needed. That means addition of these sensors will increase the cost and the power consumption of the robot.

Author contributions

Anıl Eren: Designed the study, took the measurements, and wrote the manuscript. **Hatice Doğan:** provided the idea, supervised the study, and contributed to the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

REFERENCES

Adithya P S, Tejas R, Varun V S, & Prashanth B N (2019). Design and Development of Automatic Cleaning and Mopping Robot. IOP Conference Series: Materials Science and Engineering 577, 012126.

Asafa T B, Afonja T M, Olaniyan E A, & Alade H O (2018). Development of a vacuum cleaner robot. Alexandria engineering journal, 57(4), 2911-2920.

Blower Documentation,

https://corporate.usacanadashop.com/item/B00ZL99AFQ/Gdstime-Centrifugal-Brushless-Cooler-Blower, (Access Date: 20.06.2020)

BMS protection board YH11047A, https://tr.aliexpress.com/item/32719942192.html (Access Date:20. 06. 2020)

Chen W, Liang J, Yang Z & Li G (2019). A review of lithium-ion battery for electric vehicle applications and beyond. Energy Procedia, 158, 4363-4368

Dai G T & Chen T (2007). Design on measurement and control system of cleaning robot based on sensor array detection. IEEE International conference on control automation, 2007, 1319-1322, Guangzhou, China.

Dave M (2020). Differential Drive Mechanism, https://sites.google.com/site/mleesrobotronics/home/robotics/differential-drive-mechanism (Access Date: 01.08.2020)

Dudek G, & Jenkin M (2010). Computational principles of mobile robotics. Cambridge university press. ISBN-13: 978-0521692120.

Edwards T, & Sörme J (2018). A Comparison of Path Planning Algorithms for Robotic Vacuum Cleaners, Doctoral dissertation, Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden.

Elias M F M, Nor K M, Rahim N A, & Arof A K (2003). Lithium-ion battery charger for high energy application. National Power Engineering Conference, (PECon 2003), 283-288, Bangi, Malaysia.

Gao X, Li K, Wang Y, Men G, Zhou D & Kikuchi K (2007). A floor cleaner robot using Swedish wheels. IEEE international conference on robotics and biomimetics (ROBIO), 2007, 2069-2073, Sanya, China.

Hasan K M & Reza K J (2014). Path planning algorithm development for autonomous vacuum cleaner robots. 2014 International Conference on Informatics, Electronics & Vision (ICIEV), 1-6, Dhaka, Bangladesh.

INR18650-30Q (2014). Li-on Battery Datasheet, Samsung SDI, August 2014.

LaValle S M (2006). Planning Algorithms. Cambridge University Press. ISBN-13: 978051124133-8.

Liu C, Kang L, & Yu S-N (2011). Hardware and software integration for domestic stairs cleaning robot. SICE Annual Conference 2011, 663-670, Tokyo, Japan.

Maruri L, Martinez-Esnaola A, Landaluze J, Casas S & Fernández M (2006). Design of a prototype robot vacuum cleaner-from virtual prototyping to real development. International Conference on Informatics in Control, Automation and Robotics (ICINCO-RA), 461-468, Setúbal, Portugal.

Oh J S, Choi Y H, Park J B, & Zheng Y F (2004). Complete coverage navigation of cleaning robots using triangular-cell-based map. IEEE Transactions on Industrial Electronics, 51(3), 718-726.

Oh Y-J & Watanabe Y (2002). Development of the small robot for home floor cleaning. Proceedings of the 41st SICE Annual Conference, 2002, 3222-3223, Osaka, Japan.

Ottoviani G (2020). Differential Drive Motions, https://guiott.com/Cleaning Robot/C-Motion/C0Theory/differential-drive.jpg (Access Date: 01.08.2020)

Prabakaran V, Elara M R, Pathmakumar T & Nansai S (2017). hTetro: A Tetris inspired shape-shifting floor cleaning robot. IEEE International Conference on Robotics and Automation (ICRA), 2017, 6105-6112, Singapore.

Radu Ș (2015). Relation Between Function and Form in Vacuum Cleaners Design. Journal of Industrial Design & Engineering Graphics 10(2), 37-42.

Roh S G, Lim B, Moon H, Lee J S, Park J H, Koo J C, & Choi H R (2013). Flexible suspension mechanism for stable driving of a differential drive mobile robot, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 5518-5523, Tokyo, Japan.

Stubbs L (2019). D-Shaped vs Round Shaped Vacum Cleaners,https://bestrobotvacuumforyou.com/d-shaped-vs-round-shaped-robot-vacuum-cleaners (Access Date: 27.02.2021).

Ulrich I, Mondada F & Nicoud J D (1997). Autonomous vacuum cleaner. Robotics and Autonomous Systems 19(3-4), 233-245.