جامعة الأمير محمد بن فهد
PRINCE MOHAMMAD BIN FAHD UNIVERSITY

College of Engineering

Department of Electrical Engineering

Fall 2019

Senior Project Report

Smart Cleaner

In partial fulfillment of the requirements for the Degree of Bachelor of Science in Electrical Engineering

Team Members

	Student Name	Student ID
1	HussainAlYami	201301015
2	Ibrahim Al-Marzouk	201403781
3	Falah Almutairi	201301166
4	TurkyAlkhamis	201301566

Project Advisors:

Dr. Samir El-Nakla

Abstract

Some people are suffering with cleaning house. We all see different types of cleaning tools. As you know, most of cleaning tools need to do some effort to make the place clean as you want. So, most people know are looking for a robot which can serve them in their needs without making any effort.

This project aims to serve people who wants robot to clean their house without tension. Moreover, this robot compared to conventional cleaning tools is much easier than the old tools. In addition, there is no need to put much effort on it. It is easy to deal with.

Table of *Contents*

1. Introduction

1.1 Project Definition

To design a smart cleaner that can automatically clean the floor. It can also be controlled remotely and can clean different type of floors.

1.2 Project Objectives

1. Can be used on your setting time.
2. Increase cleaning efficiency.
3. Help old people and people with special needs.
4. Easy to use.

1.3 Project Specifications

a) Can clean different types of floor within 2 cm maximum height.
b) Can be controlled remotely with range 30 m in free space.
c) Return to home base.
d) The dimensions 21 x 21cm, and the height is7.6cm.
e) Operation time is around 15min.

1.4 Product Architecture and Components

Our project consists of a power supply, microcontroller, DC motor, vacuum motor, and infrared sensor, and wheels. We will insert a code into the microcontroller to control the robot. Also, when its finish cleaning it will return to the home base.

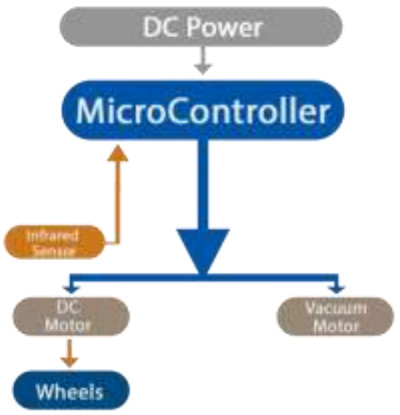


Figure 1.1(Architecture).

1.5 Applications

- Can clean all the rooms in the house.
- Can clean many types of floor within 2 cm height.
- It can clean in tiny spaces.

2. Literature Review

2.1 Project background

First of all, as an engineer, we are trying to help old people and people with special needs that have problems with cleaning the floor. In addition, the existing tools of cleaner machine is wasting power and wasting water. Also existing vacuum cleaner takes more space. We find the solution and apply it in our robot cleaner. Our robot cleaner can be used to help old people and people special needs, because if they setting on the wheel chair and they can't move around they can open the app and press any mode to clean. The robot will save power more than the traditional machine. Furthermore, our project will not take a big space.

2.2 Previous Work

We have seen three projects of vacuum cleaner in different universities. Each of these projects has one property. Such that, manual control, not remotely controlled, and they can't clean and wipe at the same time. They do not have a technology. In addition, regarding to design of the previous projects was not that nice. Fig2.2.

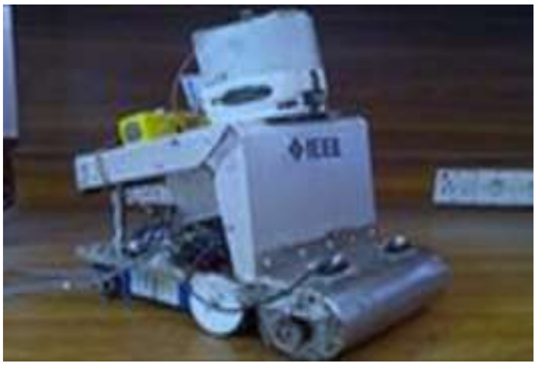


Fig2.1- Hamdard University, Pakistan.

Fig2.2- GIFT University – Pakistan.

2.3 Comparative Study

Table 2.1

Previous projects	Our project
Hard to use	Easy to use
Use by hand	Robotic
Old style	Modern technology
Does not return to home charging station	Return to home charging station.

3. System Design

3.1 Design Constraints

In our project design, we decided to make our project deferent from the other vacuum machines in order to have a unique project in a way that is easier to move from one place to another. Our concern was to make sure that our project has the ability to achieve the engineering standards including voltages and general electrical terminology *and* the ability to work in efficient way to finish what it was made for which is cleaning the floor. In the environmental part, the vacuum cleaner is made from plastic with strong silk collecting with each other to make full body made from plastic, and it does not make any harm for the environment, because our smart cleaner does not use a large voltage and will not have any smoke coming out from it. In the social part, we are trying to make our project in a way that keeps up with Saudi 2030 vision. That's why we have finished our first social part which is the communication between our project and using mobile phone. We are trying to make artificial intelligence for the project.

3.2 Design Methodology

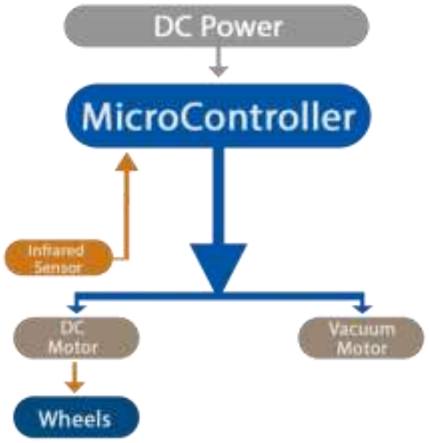


Figure 3.1 (architecture diagram).

3.3 Product Subsystems and Components

DC power: the buttery will supply all the power to the vacuum to be able to work, and we are using one lithium buttery 11.1 V 5000 mAh.

Microcontroller: from the battery, the Arduino will be able to work and give the order to all of our vacuum motors to work.

Vacuum motor and DC motor: after the Arduino works, the vacuum motor and the DC motor will work depending on the order that it got from the Arduino.

Infrared sensor: after the vacuum works, the Arduino will give the orders to the motors, and the sensors give the instructions to the Arduino if there is anything will stop the vacuum from moving the Arduino start giving orders to the motors to solve the problems and change the direction either to go right or to go left.

In our project component, we have a lot of options to make our design and we have decided to choose:

Motor micro metal gear motor: with 6 V and 1.2 A it has a cross section with 10 *12 mm and the shaped gear box output shaft is 9 mm long and 3 mm in diameter and we chose this one because it does not use a lot of power and it is small and work excellently.

Microcontroller: there are a lot of microcontrollers like raspberry and AVR but we are using Arduino Uno because we know how to program this microcontroller because we study the language in microprocessor course.

Sensors: the sharp distance sensors that are required to measure distance accurately. This IR sensor it provides much better performance than other IR alternatives. Interfacing to most microcontrollers is straightforward: the single analog output can be connected to an analog-to-digital converter for taking distance measurements, so it can implement to analog directly.

We use Sharp Analog Distance Sensor 4-30 cm. It has operation from 4.5 V to 5.5 V and 12 mA with distance range 4cm-6cm.

Figure 3.2- Sharp Analog Sensor (IR).

Pololu IR Beacons: are small infrared transceivers meant to be used in pairs to give autonomous robots simple means for detecting each other. The operating voltage range has increased to 6 – 16 V.

Our Beacons work by transmitting and detecting infrared light, much like a television remote control. In each piece there are four IR detectors. So the beacons alternate between transmitting and receiving signals. Each Beacon has a fixed NORTH direction and that can help to detect the angle very well. Our vacuum can return to home changing station. IR Beacon Transceiver has IR modulation frequency of 56 kHz and detection range of 6 inches to 15 feet with an operation voltage of 5 V. Also, we use one IR Tracing Black Sensor that has 2.5 V- 12V operation voltage and 18 mA – 20 mA current.

Figure 3.3- IR Beacons.

3.4 Implementation

1. We design our chassis in 3D printer store with help of the shop owner.
2. We decided our first stage in this course.
3. We ordered and bought our components depend on our first stage.
4. After the 3D store finished printing our Chassis, we started assembling our components and implementing the circuit.
5. We started by putting the motors. After that, we put the breadboard and we started to program our microcontroller.
6. In the second stage, we started putting the vacuum motor in our project, and we connected to both modes.
7. We put three infrared sensors that can detect the obstacles.
8. We designed a Home Base that can the vacuum return to it, when it is finish cleaning, and charge it manually.

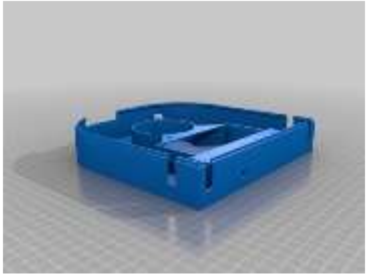


Figure 3.4- 3D design for the top cover.

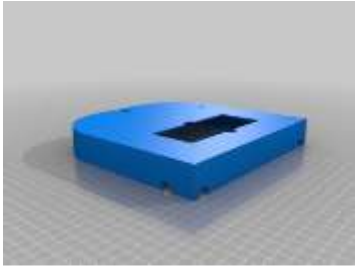


Figure 3.5- 3D design for the top cover.

Figure 3.6- Vacuum from outside.

Figure 3.7- Vacuum from inside.

4. System Testing and Analysis

4.1 Subsystem 1: Chassis design

Communication part: in our project, we are using Bluetooth module HC-06 to be able to communicate between the phone and the vacuum. This Bluetooth module have some specific rang until 9 meters or (30 ft). After that range you cannot connect with the prototype.

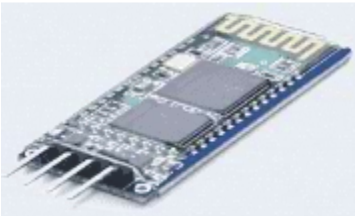


Figure 4.1(Bluetooth module).

Battery: for the first stage, we are using 1 battery, have a voltage of 11.1 V and current of 5000 mA.

Figure 4.2(Battery).

We have tested our project in the university labs after confirming that's connected and works perfectly as we want.

4.2 Subsystem 2: DC Motor

Our second subsystem is the DC motor. This subsystem is to control the movement part of the cleaner. It consists of a microcontroller, DC motor, and wheels. We have tested the servo motor, but it does not work, because it has only 180 degree to move on. Then, we decided to have the DC motor because it rotates freely.

Figure 4.3(Motor).

4.3 Subsystem 3: Auto Mode

Our third subsystem is the Auto-Mode. This subsystem controls the vacuum through the infrared sensor, and there are three sensors in the front of the vacuum, and it's detecting the obstacles. In addition, it works in one mode if it's read high on the left sensor. It turns right, and if it is read high on the right sensor and it turns left.

4.4 Subsystem 4: Return to Home Base

In our forth subsystem we worked on the Home Base. This subsystem is a station that can let the vacuum return in it, and charge manually. We used IR Beacon Transceiver, it is a sensor from two parts pair with each other one its receiver and one is the transmitter, and it is about four infrared sensors in each part, its need to face them to the north.

Figure 4.4 – Home Base.

4.5 Overall Results, Analysis and Discussion

Table 4.1

Equipment name	Voltage [V]	Current	Frequency [kHz]	Rpm	Distance
Arduino	5	200 mA	-	-	-
Motors	6	1.4 A	-	100 Rpm	
IR beacon	5	200 mA	56	-	15 to 450 cm
Line follower	5	20 mA	-	-	-
Sharp sensor	5	36 mA	-	-	4 to 6 cm
Vacuum	12	4.5 A	-	-	-

- Overall stop current = 160 mA
- Overall working current = 6.86 A
- Overall power = 82.32 W

Testing: We have tested our vacuum and we observed can working for 39 minutes.

5. Project Management

5.1 Project Plan

Table 5.1

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Write a plan for the Fall semester	1	1	All	2	1	100%
Ordered the components (sensors)	3	1	All	3	1	100%
visit the 3D printing store (to print second floor)	4	1	Hussain	4	1	100%
Meeting with the advisor	5	1	All	5	1	100%
Build the First Phase (connect the sensors)	5	1	All	5	1	100%
Write the code for the robot with the sensor	5	2	All	5	2	100%
test the code with the sensor	6	1	All	6	1	100%
design the home charge	7	2	All	7	2	100%
Prepare midterm presentation	7	1	All	7	1	100%
Design subsystem 3 (with the vacuum motor)	8	2	All	8	2	100%
write the code for the vacuum motor	10	1	All	10	1	100%
Test subsystem 3	10	1	All	10	1	100%
Test everything, and test the home charge	11	1	All	11	2	100%
Prepare final report	12	2	All	12	2	100%
Prepare final presentation	12	2	All	12	3	100%
Prepare project demo	13	3	All	13	3	100%
Submit Rpt/PPT/Brochure ….	14	2	All	14	2	100%
Write a plan for the Fall semester	1	1	All	2	1	100%
Ordered the components (sensors)	3	1	All	3	1	100%

5.2 Contribution of Team Members

Table 5.2

Task	Hussain	Falah	Turki	Ibrahim	Task Total
Search & acquire components	30%	30%	30%	10%	Search & acquire components
Design & Implement Subsystem 1	40%	20%	20%	20%	Design & Implement Subsystem 1
Design & Implement Subsystem 2	20%	35%	35%	10%	Design & Implement Subsystem 2
Design & Implement Subsystem 3	25%	25%	25%	25%	Design & Implement Subsystem 3
Design & Implement Subsystem 4	25%	25%	25%	25%	Design & Implement Subsystem 4
Write Reports & Presentations	25%	25%	25%	25%	Write Reports & Presentations

5.3 Project Execution Monitoring

List various activities:

- We met weekly with our advisor, and we gave him the feedback, and we worked on it.
- In our team we meet daily, because we have same classes, and we discuss about our project, and we share the new ideas.
- The testing part was for one week that tests the project, and make sure that we don’t have any problem with our project.

5.4 Challenges and Decision Making

Decision Making:

- When the DC motor finish from the stock we searched for a new supply, and we ordered it for the same time, and its deliver in three days, from USA.

5.5 Project Bill of Materials and Budget

Table 5.3

Item	Quantity	Unit Cost (SR)	Subtotal
Microcontroller	**1**	**179.55**	**179.55**
Wheels With Motors	**4**	**125**	**125**
3D Printer Chassis	**1**	**1200**	**1200**
Components		**725**	**725**
Total			**2604.55 SR**

6. Project Analysis

6.1 Life-long Learning

In this project we have learn new experience and skills and we have improved some skills. First of all, we learned how to design a chassis for our project to meet our specification. Second, we have improved our skill in coding and using Arduino. In addition, we have learned how to manage a project and how to divide work between us and based on skills that we have. Moreover, we have learned how to manage our time between the project work and the other courses exam and assignment. Also, we have learned how to search for parts and compare between the local market and abroad market and choose the best price and the best quality.

6.2 Impact of Engineering Solutions

The impact of Smart Cleaner in the society is that old people and people with special needs have problems with cleaning the floor. So, our Smart Cleaner will make it easily for them because it will work remotely and automatically with setting time. For economy our project will save more power than the traditional vacuum.

7. Conclusions and Future Recommendations

7.1 Conclusions

Finally, we learned a lot of things in this project, we learned how teamwork could be, and we learned how to think, and manage the time, don't waste the time, and we learned how to choose the right components, and how to compare between two components doing the same function. In addition, we learned how to calculate the power for the motors, and how many voltages we need for our projects.

7.2 Future Recommendations

- I recommend for the future to have sensor that can be sense the dust, and clean by knowing there is dust.
- I recommend for the future to have wipe function, so that can clean by water, and wipe the floor.
- I recommend for the future to have Wi-Fi connection so that we can control the smart cleaner from outside.
- I recommend for the future to have many modes that can help the vacuum to move in smart way.
- I recommend for the future to having auto stop when the vacuum reach the home base.
- I recommend for the future is to have automatically charging when the vacuum reach the home base.

8. References

[1] Gift University, http://www.gift.edu.pk/home

[2] SRM Institute of Science and Technology (formerly known as SRM University), http://www.srmuniv.ac.in/

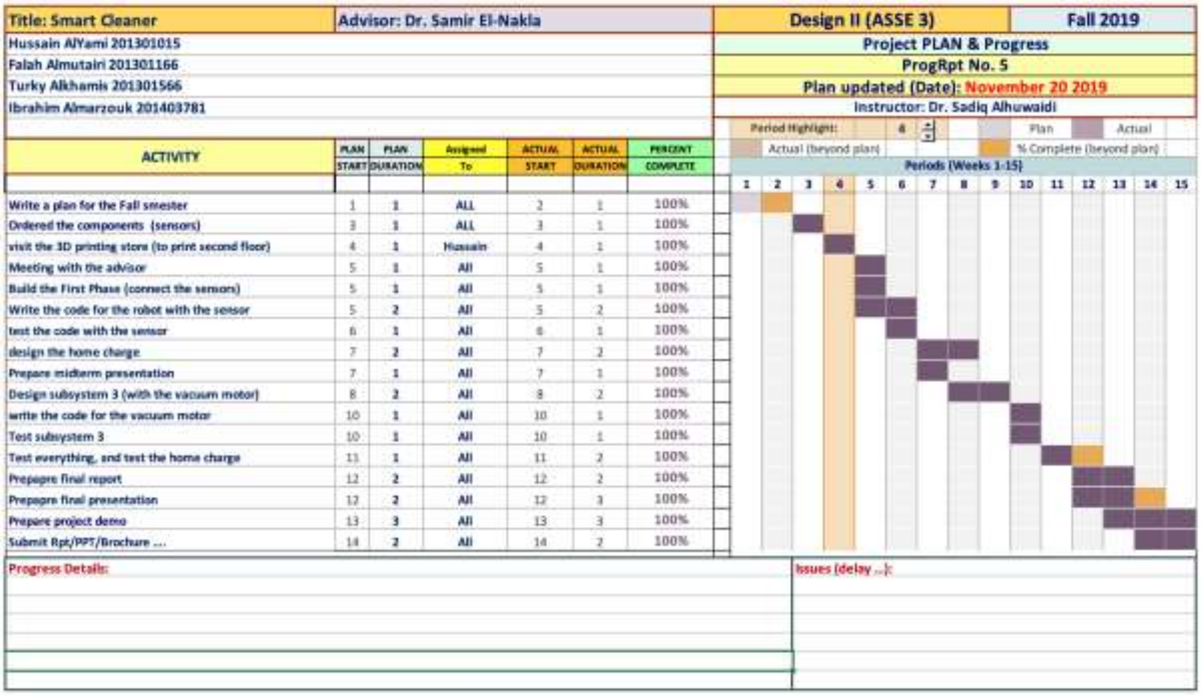
[3] https://www.youtube.com/watch?v=oIKSulS9MaE

[4] https://www.slideshare.net/AsishNayak1/wireless-floor-cleaning-robot

[5] http://www.standardsuniversity.org/wp-content/uploads/Smart-Floor-Cleaning-Robot-CLEAR.pdf

[6] Hamdard University, Madunat al-Hikmah, http://www.hamdard.edu.pk/

Appendix A: Progress Reports



Title: Smart Cleaner	Advisor: Dr. Samir El-Nakla	Design II (ASSE 3)	Fall 2019
Hussain AlYami 201301015		Project PLAN & Progress	
Falah Almutairi 201301166		ProgRpt No. 5	
Turky Alkhamis 201301566		Plan updated (Date): November 20 2019	
Ibrahim Almarzouk 201403781		Instructor: Dr. Sadiq Alhuwaidi	

Period Highlight: 4 | Plan | Actual | Actual (beyond plan) | % Complete (beyond plan)

Periods (Weeks 1-15)

ACTIVITY	PLAN START	PLAN DURATION	Assigned To	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Write a plan for the Fall smester	1	1	ALL	2	1	100%
Ordered the components (sensors)	3	1	ALL	3	1	100%
visit the 3D printing store (to print second floor)	4	1	Hussain	4	1	100%
Meeting with the advisor	5	1	All	5	1	100%
Build the First Phase (connect the sensors)	5	1	All	5	1	100%
Write the code for the robot with the sensor	5	2	All	5	2	100%
test the code with the sensor	6	1	All	6	1	100%
design the home charge	7	2	All	7	2	100%
Prepare midterm presentation	7	1	All	7	1	100%
Design subsystem 3 (with the vacuum motor)	8	2	All	8	2	100%
write the code for the vacuum motor	10	1	All	10	1	100%
Test subsystem 3	10	1	All	10	1	100%
Test everything, and test the home charge	11	1	All	11	2	100%
Prepapre final report	12	2	All	12	2	100%
Prepapre final presentation	12	2	All	12	3	100%
Prepare project demo	13	3	All	13	3	100%
Submit Rpt/PPT/Brochure ...	14	2	All	14	2	100%

Progress Details:

Issues (delay ...):

Figure 3 : Progress Reports last update 20/11/2019

Appendix B: Bill of Materials

Item	Quantity	Unit Cost (SR)	Subtotal
Microcontroller	1	179.55	179.55
Wheels With Motors	4	125	500
3D Printer Chassis	1	1200	1200
Components	-	725	725
Total			2604.55 SR

Figure 4: Bill of Materials

Appendix C: Datasheets

Micro Metal Gearmotor HP 6V with Extended Motor Shaft:

Device Specification:

- **Operating voltage:** 6 V
- **Average current consumption:** 70 mA
- **Torque:** 100 RPM

Pololu IR Beacon Transceiver:

Device Specification:

- **PCB size:** 1.35″ diameter
- **IR modulation frequency:** 56 kHz
- **Output refresh rate** 20 Hz
- **Detection range:** 6 inches to 15 feet (typical; actual max range will depend on ambient lighting conditions)
- **Supply voltage:** 6-16 V
- **Data voltage:** 5 V
- **Number of IR detectors:** 4

Sharp GP2Y0A41SK0F Analog Distance Sensor 4-30cm:

Device Specification:

- **Operating voltage:** 4.5 V to 5.5 V
- **Distance measuring range:** 4 cm to 30 cm (1.5″ to 12″)
- **Output type:** analog voltage
- **Output voltage differential over distance range:** 2.3 V (typical)
- **Update period:** 16.5 ± 4 ms
- **Size:** 44.5 mm × 18.9 mm × 13.5 mm (1.75″ × 0.75″ × 0.53″)
- **Weight:** 3.5 g (0.12 oz)

Line follower sensor IR:

Device Specifications:

- **Voltage:** 3.3 V to 5 V
- **Working current:** ≥ 20 mA
- **Size:** 42 mm x 11 mm x 12 mm

Appendix D: Program Codes

```
#define motRF 5 // RIGHTFORWORD//
#define motRB 2 // RIGHTBACKWORD//
#define motLF4  // LEFTFORWORD//
#define motLB3  // LEFTBACKWORD//
#define EN1 12 // 1 IS LEFT//
#define EN2 11 // 2 IS RIGHT//
#define SLF  A0  //LEFT FRONT SENSOR//
#define SRF  A1   //RIGHT FRONT SENSOR//
#define SMF A2     // MIDDLE FRON SENSOR//
#define Fan 13
#define LINE A3

char state;
int flag=0;
intstateStop=0;
byte spd1 = 250;
byte spd2 = 65;
int mode=0 ; //default mode is auto

// Direction Pins:
intnorthPin = 9;
intsouthPin = 7;
inteastPin = 8;
intwestPin = 6;
// Direction Pin Values:
booleannorthValue;
booleansouthValue;
booleaneastValue;
booleanwestValue;
// Direction Sensors Variables:
int north;
int south;
int east;
int west;
// Direction of Strongest Signal:
intstrongestSignal;



void setup() {

Serial.begin(9600);
   //MOTORS//
pinMode(motRF, OUTPUT);
pinMode(motRB, OUTPUT);
pinMode(motLF, OUTPUT);
pinMode(motLB, OUTPUT);
pinMode(EN1, OUTPUT);
pinMode(EN2, OUTPUT);
  //Fan//
pinMode(Fan, OUTPUT);
   //SENSOR//
pinMode(SLF, INPUT);
```

```
pinMode(SRF, INPUT);
pinMode(SMF, INPUT);
   // IR beacon //
pinMode(northPin, INPUT);
pinMode(southPin, INPUT);
pinMode(eastPin, INPUT);
pinMode(westPin, INPUT);
  //Line Flower//
pinMode(LINE, INPUT);
}

doublesdSHARP(int Sensor){
doubledist = pow(analogRead(Sensor), -0.857); // x to power of y
return (dist * 1167.9);
}

void loop() {

if(Serial.available() > 0){
state = Serial.read();
flag=0;
 }

if (state=='M'){mode=0;} // MunualMode
if (state=='A'){mode=1;} //AutoMode
if (state=='X'){mode=2;} //Shutdown
if (state=='E'){mode=3;} // ReturnMode
if(mode==0){ // MunualMode

if (state == 'Q') {
digitalWrite(Fan, HIGH);

      }
if (state == 'W') {
digitalWrite(Fan, LOW);

      }
if (state == 'F') {
digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);

       }

if (state == 'R') {
digitalWrite(motRF, HIGH);   // RIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);

       }
if (state == 'L') {
digitalWrite(motRF, LOW);   //LIFT
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
```

```
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);

    }
if (state == 'B') {

digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
      }


    // if the state is 'S' the motor will Stop
if (state == 'S' ) {
digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);

   }
   }
if(mode==1){ //AutoMode

Serial.print("SLF= ");
Serial.print(sdSHARP(SLF));
Serial.println();
Serial.print("  SRF= ");
Serial.print(sdSHARP(SRF));
Serial.println();
digitalWrite(Fan, HIGH);
digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
floatminDistanceSharp = 5;

if (sdSHARP(SLF)<= minDistanceSharp){

digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (100);


digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(200);
```

```
digitalWrite(motRF, HIGH);   // SHARPRIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);

digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
      }

if (sdSHARP(SRF)<= minDistanceSharp){



digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (100);
digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(200);


digitalWrite(motRF, LOW);   //SHARPLIFT
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);



digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);


}
if (sdSHARP(SMF)<= minDistanceSharp){


digitalWrite(motRF, LOW);   // STOP
```

```
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (150);


digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(100);

digitalWrite(motRF, LOW);   //SHARPLIFT
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(900);


digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);

}
if (sdSHARP(SLF)<= minDistanceSharp&&sdSHARP(SMF)<= minDistanceSharp){

digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (150);

digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(100);

digitalWrite(motRF, HIGH);   // SHARPRIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);

digitalWrite(motRF, HIGH);   // FORWORD
```

```
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(400);

digitalWrite(motRF, HIGH);   // SHARPRIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);

digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);


   }
if (sdSHARP(SRF)<= minDistanceSharp&&sdSHARP(SMF)<= minDistanceSharp ){
digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (150);

digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(100);

digitalWrite(motRF, LOW);   // SHARPLIFT
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);


digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(400);

digitalWrite(motRF, LOW);   // SHARPLIFT
digitalWrite(motRB, HIGH);
```

```
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);


digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
  }


if (sdSHARP(SRF)<= minDistanceSharp&&sdSHARP(SMF)<= minDistanceSharp&&sdSHARP(SLF)<=
minDistanceSharp){
digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (150);

digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(100);

digitalWrite(motRF, HIGH);   // SHARPRIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(900);


digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
  }

}
if(mode==2){   //Shutdown

digitalWrite(Fan, LOW);
digitalWrite(motRF, LOW);
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
```

```
analogWrite(EN2 , spd2);
   }

if(mode==3){ // ReturnMode

senseDirection(northPin, southPin, eastPin, westPin);
intstrongestSignal = chooseDirection(northValue, southValue, eastValue,westValue);
int Line = digitalRead(LINE);

Serial.println(strongestSignal);
Serial.print("N: ");
Serial.print(northValue);
Serial.print("   S: ");
Serial.print(southValue);
Serial.print("   E: ");
Serial.print(eastValue);
Serial.print("   W: ");

Serial.print("        ");
Serial.print(Line);
Serial.print("        ");

Serial.println(westValue);


}
}
// Read Data from Chip:
voidsenseDirection(int sensorPin1, int sensorPin2, int sensorPin3, int
sensorPin4) {
northValue = digitalRead(sensorPin1);

southValue = digitalRead(sensorPin2);

eastValue = digitalRead(sensorPin3);

westValue = digitalRead(sensorPin4);

}
// Detect Chip Strongest Direction:
intchooseDirection(boolean north, boolean south, boolean east, boolean west) {
floatminDistanceSharp = 5;
int Line = digitalRead(LINE);
if (Line == LOW){
digitalWrite(Fan, HIGH);
if (north == 0) {
digitalWrite(motRF, HIGH);   //FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
 }
if (south == 0) {
digitalWrite(motRF, LOW);   // UTURN
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
```

```
 }
if (east == 0) {
digitalWrite(motRF, HIGH);   //LIFT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
 }
if (west == 0) {
digitalWrite(motRF, LOW);   //RIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
 }
if (sdSHARP(SLF)<= minDistanceSharp){

digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (100);


digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(200);

digitalWrite(motRF, HIGH);   // SHARPRIGHT
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);

digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
     }

if (sdSHARP(SRF)<= minDistanceSharp){



digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
```

```
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (100);
digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(200);


digitalWrite(motRF, LOW);   //SHARPLIFT
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(500);



digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);


}
if (sdSHARP(SMF)<= minDistanceSharp){


digitalWrite(motRF, LOW);   // STOP
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay (150);


digitalWrite(motRF, LOW);   // BACKWORD
digitalWrite(motRB, HIGH);
digitalWrite(motLF, LOW);
digitalWrite(motLB, HIGH);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(100);

digitalWrite(motRF, LOW);   //SHARPLIFT
digitalWrite(motRB, HIGH);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);
delay(900);
```

```
digitalWrite(motRF, HIGH);   // FORWORD
digitalWrite(motRB, LOW);
digitalWrite(motLF, HIGH);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);

}
}
else if (Line == HIGH){
digitalWrite(Fan, LOW); // STOP
digitalWrite(motRF, LOW);
digitalWrite(motRB, LOW);
digitalWrite(motLF, LOW);
digitalWrite(motLB, LOW);
analogWrite(EN1 , spd1);
analogWrite(EN2 , spd2);


}

}
```

Appendix E: Operation Manual

1. Turn on the switch
2. Connect the Smart Cleaner with application by sync with the Bluetooth
3. Choose any mode, manual or auto or return