BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 318 (January 2023)
Control System I Laboratory

Final Project Report

Section: B1 Group: 01

Hygieia : An Autonomous Floor Cleaner

Course Instructors:

Shafin Bin Hamid, Lecturer
Mrinmoy Kundu, Part-Time Lecturer

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ______________________ **Full Name: Sheikh Abu Al Raihan** **Student ID: 1906066**	**Signature:** ______________________ **Full Name: Kazi Abid Hasan** **Student ID: 1906067**
Signature: ______________________ **Full Name: Arpa Roy Dastider** **Student ID: 1906071**	**Signature:** ______________________ **Full Name: Ramisa Tahsin Shreya** **Student ID: 1906072**
Signature: ______________________ **Full Name: MD Ramim Hassan Shawn** **Student ID: 1906082**	**Signature:** ______________________ **Full Name: Dipika Rani Nath** **Student ID: 1906092**

Table of Contents

1 Abstract:

The project aims to develop an innovative floor-mopping robot designed to efficiently clean rooms while ensuring comprehensive coverage. This autonomous robot is equipped with advanced navigation and mopping capabilities, enhancing its performance in household and commercial cleaning tasks.

The robot's primary functionality involves traversing a room from one end to another, including rotations of 180 degrees to maximize floor coverage. It incorporates an integrated mop system, effectively removing dirt, dust, and stains from various floor surfaces. The robot uses sensor technology and intelligent algorithms to detect and navigate around obstacles, ensuring thorough cleaning without causing damage to furniture or the environment. One distinctive feature of this robot is its ability to identify and target three corners of the room during its cleaning process. Once these corners are detected, the robot intelligently determines that it has achieved full room coverage and proceeds to conclude its final cleaning run.

The project combines elements of robotics and cleaning technology to create a sophisticated and user-friendly cleaning robot. Its ability to autonomously clean rooms, efficiently cover floor surfaces, and detect room boundaries marks a significant advancement in the field of household and commercial cleaning automation.

2 Introduction

In a world marked by advancing technology and an ever-increasing demand for efficiency, the need for innovative solutions in the realm of cleaning and maintenance is more evident than ever before. One such problem, poised at the intersection of robotics, automation and household/industrial maintenance, is the development of an intelligent floor mopping system. This project aims to introduce a comprehensive solution to address the challenges posed by floor cleaning in both household and industrial settings.

The Complexity of the Engineering Problem:

The development of a floor mopping system that navigates, cleans, and efficiently covers spaces in diverse environments presents a multifaceted engineering challenge. Several factors contribute to the complexity of this problem:

1. Navigation and Mapping: The system navigates its surrounding and follows the instruction, while moving, given by the programmer.

2. Obstacle Avoidance: To prevent collisions with obstacles and ensure user safety, the system must detect and react to objects in its path effectively.

3. Continuous Operation: The system must mop continuously while navigating, necessitating a design that balances water reservoir capacity, mop maintenance, and power efficiency.

4. Corner Detection: Accurately identifying room corners is pivotal to the system's functionality, as it triggers the final run and completion of the cleaning process.

5. Automation and Control: Implementing robust control algorithms to govern the movement of the mopping system is critical, ensuring that it follows a predetermined cleaning path efficiently.

Each of these solutions comes with its advantages and engineering challenges.

<u>Possible Alternative Solutions:</u>

Several approaches could potentially address the complexities of this engineering problem:

1. LIDAR-based Navigation: Utilizing LIDAR sensors for precise environmental mapping and obstacle avoidance, allowing for more accurate navigation.

2. Computer Vision: Leveraging computer vision techniques to identify walls, corners, and objects, providing the system with an understanding of its surroundings.

3. Machine Learning: Training the system through machine learning algorithms to adapt and optimize its cleaning pattern based on the specific environment.

4. Image Processing: Employing the cleaning robot to process the image of dust and communicate to its cleaning mechanism for giving the information about the dust.

5. Hybrid Systems: Combining traditional mop design with robotic automation for more extensive and efficient cleaning.

3 Design

3.1 Problem Formulation

3.1.1 Identification of Scope

The project addresses the challenge of optimizing floor cleaning efficiency in indoor environments. Traditional cleaning methods are often time-consuming and labor-intensive, and they may not provide comprehensive coverage, leaving certain areas untouched. Additionally, there is a growing need for automation in household and commercial cleaning tasks to enhance convenience and hygiene.

The specific problems to be tackled in this project include:

1. Inefficient Cleaning Practices: Conventional mopping and vacuuming methods often require manual intervention and lack systematic room coverage, resulting in uneven cleaning outcomes.

2. Time and Labor Intensity: Cleaning large indoor spaces can be time-consuming and physically demanding, requiring repetitive movements and significant human effort.

3. Obstacle Navigation: Effective cleaning necessitates the ability to navigate around furniture and other obstacles without causing damage.

4. Room Coverage Verification: Ensuring complete room coverage without human supervision is a significant challenge.

3.1.2 Literature Review

Before starting to develop the working, we started to read literatures about various types related to our project. Some of the reviews of our findings are described below:

1. **Types of Cleaning Robots:** We have read about various types of robots. LFR, Automatic Navigation etc.
2. **Navigation and Mapping Algorithms:** There are LIDAR based navigation and mapping technologies and image-processing based technologies which may map the surrounding more efficiently.
3. **Obstacle Detection and Avoidance:**
   We have seen some of the algorithms regarding this topic, but nothing matches our demands. So, we designed according to our own demand and that is discussed later.
4. **Cleaning Mechanisms:**
   In this section we also developed the idea and controlling its operation.

3.1.3 Formulation of Problem

We have formulated the problem considering the evolving world and reduction of human presence in cleaning because of the safety purpose. In the commercial and industrial areas there are a lot of hazardous wastages which may cause severe health issues. So, there is a great chance of using our project in the industrial areas.

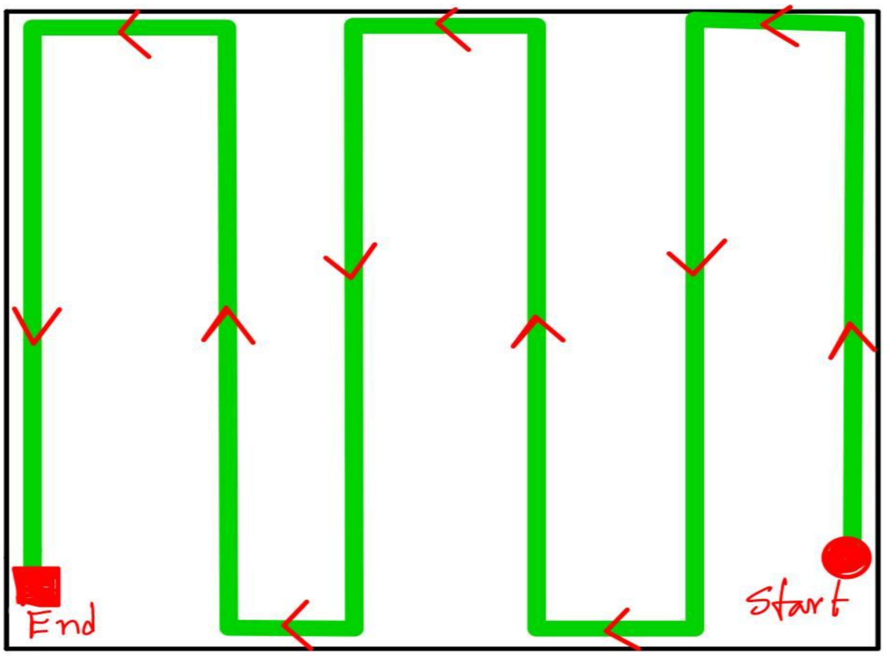
As well as, In the households, the system may reduce human effort and may give better coverage of every portion of the room.

Our basic idea pop up considering these 2 sections. Then we start to find the way to Navigate its path and obstacles avoiding while simultaneously mopping the floor.

3.2 Design Method

Navigation

The robot will start to move from one corner of the room. Then it will run linearly towards its opposite side of the room. After facing the wall, it will detect the wall and will rotate 180˚ by itself. Then it runs again towards the opposite wall. We are using an Ultrasonic sensor for detecting the end point of its single run. As soon as it finds the wall the robot stops and the Gyro sensor helps to rotate the robot and after 90˚ rotation it moves forward for 200ms so that it grabs another line of the room and then again it rotates 90˚ and move on.



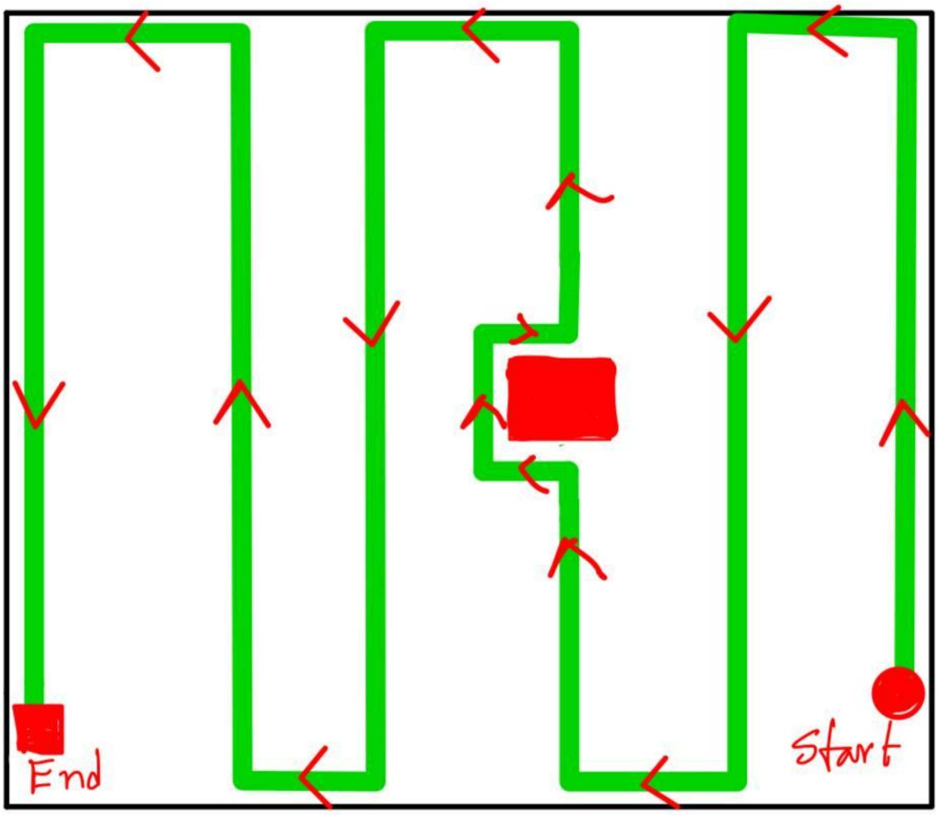
Corner Detection

The corner of the room is detected by the Ultrasonic Sensor and gyro helps the robot to rotate 180˚ which is mentioned before. We used three Ultrasonic Sensor in front, left and right to find the corners of the room. When any one of the right or left sensor senses obstacle while front sensor also senses obstacle, it is considered as a corner. When three such corners are found, the robot will identify the end point of the room and stop all of its operations.

Obstacle Avoidance

If any kind of obstacle comes in its way, the robot will detect it and avoiding the obstacles, it will move forward. For constructing the obstacle avoidance algorithm, we took the help of Ultrasonic Sensors and servo motor. There are two steps of avoiding obstacle:

i) Identifying obstacle: After sensing an object in front, the robot must decide that whether it is a wall of the room or any obstacle. To make that decision, we interfaced the Ultrasonic Sensor to a servo motor. Instantly after sensing something in front, the servo rotates 50 degree right and the Ultrasonic Sensor takes the distance in that direction. The same thing occurs at the left side. If objects are found in both sides, the robot will consider it a wall. Otherwise it is considered as an obstacle.



ii) Avoiding the obstacle: To avoid the obstacle, the robot turns 90 degree left at first. Then it goes forward as long as the right Ultrasonic Sensor senses the length of the obstacle. We used millis() function to save the time needed to overcome the obstacle. Then the robot turns 90 degree right and goes forward until the width of the obstacle is crossed. After that, it again turns 90 degree right and goes forward for the time period saved before. Finally it turns left again, goes forward and follows the roaming algorithm. Thus the robot can avoid any obstacle.

Motor Interfacing

We are using 4 pieces of 12V-300rpm DC motor to run the robot and 2 pieces of 6V-180rpm DC motor to run the mopper, again we are using a servo motor named SG-90.

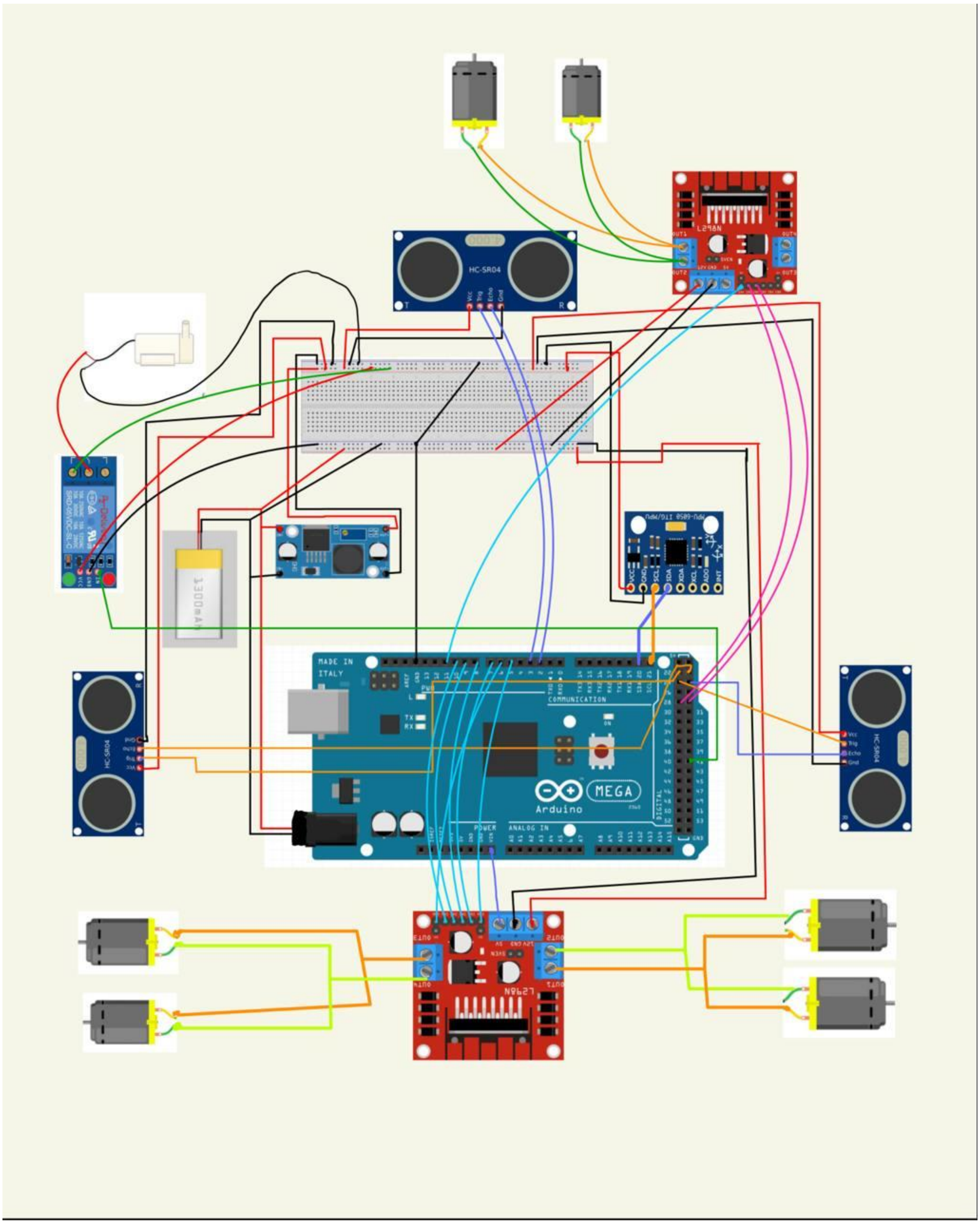
Cleaning Mechanism

The cleaning Mechanism consists of two parts:

i) Disinfectant and water spraying: This is done by using a 5V water pump. The pump triggers the outward flow of water periodically with the help of relay. This keeps the mops wet and disinfects the surface.

ii) Cleaning the floor: For this purpose, we used two dc gear motor which are connected to two circular mops. The mops rotate in opposite directions and clean the surface by scrubbing.

3.3Circuit Diagram



3.4Full Source Code of Firmware

```
// Libraries
#include <Servo.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

// sonar
#define trigPinf 3
#define echoPinf 2
#define trigPinl 22
#define echoPinl 23
#define trigPinr 24
#define echoPinr 25

const int relayPin = 41;  //Relay setup
Servo myservo;   // servo
Adafruit_MPU6050 mpu;  //gyro sensor

float distancef;
float distancel;
float distancer;
float z_angle = 0;
int count = 0;
int corner = 0;

// motors
int ENA = 5;
int ENB = 6;
int spd = 100; // range: 0 to 255
float rot = 86;
int mop_cw = 28;
int mop_ccw = 29;
int mop_speed = 11;
int motorRightA = 7; // Right motor forward
int motorRightB = 8; // Right motor backward
int motorLeftA = 9; // Left motor forward
int motorLeftB = 10; // Left motor backward

void forward() // move forward
{
  digitalWrite(motorRightA, HIGH);
  digitalWrite(motorLeftA, HIGH);
  digitalWrite(motorRightB, LOW);
  digitalWrite(motorLeftB, LOW);
}


void stp() // stop
{
  digitalWrite(motorRightA, LOW);
  digitalWrite(motorLeftA, LOW);
  digitalWrite(motorRightB, LOW);
  digitalWrite(motorLeftB, LOW);
}

void backward()
{
  digitalWrite(motorRightA, LOW);
  digitalWrite(motorLeftA, LOW);
  digitalWrite(motorRightB, HIGH);
  digitalWrite(motorLeftB, HIGH);
}


void right() // turn right
{
  analogWrite(ENA, 140);
  analogWrite(ENB, 140);
  digitalWrite(motorRightA, LOW);
  digitalWrite(motorLeftA, HIGH);
  digitalWrite(motorRightB, HIGH);
  digitalWrite(motorLeftB, LOW);
}
```

```
void left() // turn left
{
  analogWrite(ENA, 140);
  analogWrite(ENB, 140);
  digitalWrite(motorRightA, HIGH);
  digitalWrite(motorLeftA, LOW);
  digitalWrite(motorRightB, LOW);
  digitalWrite(motorLeftB, HIGH);
}

int calcdisf() // measures the distance ahead
{
  float duration, cm;
  delay(70);
  digitalWrite(trigPinf, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinf, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinf, LOW);

  duration = pulseIn(echoPinf, HIGH);

  cm = (duration / 2) * 0.0343;
  return cm;
}

int calcdisl() // measures the distance in left
{
  float duration, cm;
  delay(70);
  digitalWrite(trigPinl, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinl, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinl, LOW);

  duration = pulseIn(echoPinl, HIGH);

  cm = (duration / 2) * 0.0343;
  return cm;
}

int calcdisr() // measures the distance in right
{
  float duration, cm;
  delay(70);
  digitalWrite(trigPinr, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinr, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinr, LOW);

  duration = pulseIn(echoPinr, HIGH);

  cm = (duration / 2) * 0.0343;
  return cm;
}

float angle() // measures the turning angle in degrees
{
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
  float Z_rotation;
  static unsigned long prevTime = 0;
  unsigned long currentTime = millis();
  float dt = (float)(currentTime - prevTime) / 1000.0;
  prevTime = currentTime;

  Z_rotation = (g.gyro.z + 0.012) * 180 / 3.1416;

  z_angle = (z_angle + Z_rotation * dt);
  return z_angle;
}
```

<pre>void zigzag() // function for navigating the room
{
    //first turns left or right
    z_angle = 0;
    if (count % 2 == 0) {
      left();
    }
    else {
      right();
    }
    while (abs(angle()) <= rot)
    {
      if (count % 2 == 0) {
        left();
      }
      else {
        right();
      }
      if (abs(angle()) > rot)
        break;
    }
    z_angle = 0;
    stp();
 	
    //The goes forward for a while
    forward();
    delay(350);
    stp();
 	
    //detection of corner
    distancef = calcdisf();
    distancel = calcdisl();
    distancer = calcdisr();
 	
if((distancef<30)&&((distancel<30)||(distancer<30)))
    {
      corner=corner+1;
    }
 	
    //Then again turns left or right
    if (count % 2 == 0) {
      left();
    }
    else {
      right();
    }
    while (abs(angle()) <= rot)
    {
      if (count % 2 == 0) {
        left();
      }
      else {
        right();
      }
      if (abs(angle()) > rot)
        break;
    }
    z_angle = 0;
    stp();
    forward();
    count = count + 1;
    digitalWrite(relayPin, LOW); //throws water and
disinfectant
    delay(80);
    digitalWrite(relayPin,HIGH);
}
 	
int lookRight() // front sonar turns right and measures
distance
{
    myservo.write(72);
    delay(200);
    int distance = calcdisf();
    delay(100);
    myservo.write(122);
    return distance;
}</pre>
	<pre>int lookLeft()// front sonar turns left and measures
	distance
	{
	    myservo.write(172);
	    delay(200);
	    int distance = calcdisf();
	    delay(100);
	    myservo.write(122);
	    return distance;
	}
	void mop() //rotation of two mops
	{
	  analogWrite(mop_speed, 80);
	  digitalWrite(mop_cw, HIGH);
	  digitalWrite(mop_ccw, LOW);
	}
	
	void stop_mop() //stops rotation of mops
	{
	  digitalWrite(mop_cw, LOW);
	  digitalWrite(mop_ccw, LOW);
	}
	void setup()
	{
	  pinMode(trigPinf, OUTPUT);
	  pinMode(echoPinf, INPUT);
	  pinMode(trigPinl, OUTPUT);
	  pinMode(echoPinl, INPUT);
	  pinMode(trigPinr, OUTPUT);
	  pinMode(echoPinr, INPUT);
	  pinMode(motorRightA, OUTPUT);
	  pinMode(motorRightB, OUTPUT);
	  pinMode(motorLeftA, OUTPUT);
	  pinMode(motorLeftB, OUTPUT);
	  pinMode(relayPin, OUTPUT);
	  Serial.begin(115200);
	  myservo.attach(4);
	  myservo.write(122);
	
	  digitalWrite(relayPin, HIGH);
	  // Try to initialize!
	  if (!mpu.begin()) {
	    Serial.println("Failed to find MPU6050 chip");
	    while (1) {
	      delay(100);
	    }
	  }
	  //Serial.println("MPU6050 Found!");
	
	  // set accelerometer range to +-8G
	  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
	
	  // set gyro range to +- 500 deg/s
	  mpu.setGyroRange(MPU6050_RANGE_500_DEG);
	
	  // set filter bandwidth to 21 Hz
	  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
	  delay(100);
	}</pre>

Table: Source Code for the main program

```
void loop()
{
  float distanceR = 0;
  float distanceL =  0;
  distancef = calcdisf(); //calculates forward distance

  analogWrite(ENA, spd);
  analogWrite(ENB, spd);

  if(corner<3) //normally roams if endpoint not reached
(at endpoint, corner=3)
  {
    forward();
    mop();
  }
  else  //if endpoint reached, all operations stop
  {
    stp();
    stop_mop();
  }
  if (distancef < 30) //front sensor detects something
  {
    stp();
    delay(100);
    distancel = calcdisl();
    distancer = calcdisr();
    if((distancel<30)||(distancer<30)) //whether corner
found or not
    {
      corner=corner+1;
    }
    distanceR=lookRight();
    delay(200);
    distanceL = lookLeft();
    delay(200);

    if((distanceR<100)&&(distanceL<100)) //if the
object is wall
    {
      if(corner<3)
      {
        zigzag();
      }
      else
      {
        stp();
      }
    }
    else //the object is an obstacle
    {
      z_angle=0;
      left(); //turns left at first
      while (abs(angle()) <= rot)
      {
        left();
        if (abs(angle()) > rot)
          break;
      }

      unsigned long start_time = millis(); //tracks the
time how much time needed to cross the length of the
obstacle
      forward();
      delay(100);
      while(calcdisr()<20)
      {
        forward();
        if(calcdisr()>20)
          break;
      }
unsigned long real_time = millis();

      z_angle=0;
      right();//turns right
      while (abs(angle()) <= rot)
      {
        right();
        if (abs(angle()) > rot)
          break;
      }

      forward(); //again goes forward untill obstacle
is crossed
      delay(500);
     while(calcdisr()<20)
      {
        forward();
        if(calcdisr()>20)
          break;
      }

      z_angle=0;
      right(); //takes turn at right
      while (abs(angle()) <= rot)
      {
        right();
        if (abs(angle()) > rot)
          break;
      }

      forward(); //goes forward for the time recorded
      delay(real_time - start_time);

      z_angle=0;
      left(); //takes final turn at left
      while (abs(angle()) <= rot)
      {
        left();
        if (abs(angle()) > rot)
          break;
      }
      forward(); //resumes normal navigation
    }
  }
}
```

4 Design Analysis and Evaluation

4.1 Novelty

Autonomous System

We have designed the robot to be run autonomously. It reduces human effort to control. Here we can add machine learning to train it and also make it create its own decision.

Navigation System

The robot has its own navigation system. As it is controlled by a Gyro sensor and Ultrasonic sensor, it needs no defined path. It can be done to improve its Zig-Zag running path or modify the path by changing the angle of Gyro.

Obstacle Avoiding

The robot itself avoid obstacles on its way.

Efficient Cleaning Method

Implementing the ideas integrated in the system will result in efficient cleaning method. It will reduce human efforts and in the industrial area it will help to keep away humans from the hazardous area, which will assure the safety issues.

4.2Investigations

4.2.1 Literature Review

Most of the literature concerning floor mopping robots consists of development, applications and advancements in this field. Researchers explore methods like SLAM (Simultaneous Localization and Mapping) and sensor fusion to improve navigation accuracy, use sensor technologies like IR sensor, LIDAR sensors and ultrasonic sensors. Some sources also introduce different cleaning mechanisms, such as rotating brushes, microfiber clothes, water spraying.

4.3 Limitations of Tools

- We were asked to map the designated room for the navigation part of our robot. We will need a LIDAR sensor to meet that purpose. But LIDAR sensor costs approximately 50k BDT which is not affordable at all.
- We used 3 sonar sensors to detect the room corners and navigate. But our sonar sensor doesn't operate flawlessly. Sonars have some extent of errors.
- We were advised to operate a navigation process using image processing. The robot was expected to remember the dimensions of any room and clean accordingly. To fulfill this task, we needed Raspberry Pi as a microcontroller. But Raspberry Pi is not available for purchase in Bangladesh and costly as well.

4.4 Impact Assessment

4.4.1 Assessment of Societal and Cultural Issues

- Cultural attitudes and social norms can influence the acceptance of floor mopping robots. So, we need to evaluate the level of acceptance between different groups of people.
- Our robot must be tailored according to the feedback of users and their preferences.
- We must ensure our project is accessible to people with disabilities.

By conducting a thorough assessment of the social and cultural impact of floor mopping robots, manufacturers and developers can design and market these devices in a way that respects local values and preferences while promoting their benefits to a diverse range of users.

4.4.2 Assessment of Health and Safety Issues

- Ensuring the safety of users is necessary. So, we need to evaluate the design of the robot to minimize the risk of accidents.
- To maintain electrical safety, we need to provide clear instructions for safe charging and maintenance.

- Our robot needs to maintain relevant safety standards and regulations in the regions where it is sold or operated.

4.4.3 Assessment of Legal Issues

An assessment of legal issues for floor mopping robots is crucial to ensure compliance with relevant laws and regulations while minimizing legal risks.

- **Product Liability**: Manufacturers and developers may be liable for injuries or property damage caused by defects in the robot's design, manufacturing, or instructions. We need to provide clear and comprehensive user manuals, including safety instructions.
- **Safety Standards and Regulations**: Non-compliance with safety standards and regulations can result in fines, recalls, and legal actions. We have to conduct a thorough review of safety standards and regulations.
- **Intellectual Property Rights**: We need to conduct intellectual property due diligence to identify and respect existing patents and trademarks.
- **Data Privacy and Security**: Mishandling of user data can lead to legal consequences, including fines and damage to reputation. Project must be implemented robust data privacy measures.
- **Consumer Protection Laws**: We have to ensure that the robot adheres to consumer protection laws.
- **Government Regulations and Approvals**: Working closely with relevant government agencies to obtain the required permits, licenses, and approvals for the robot's production and distribution.

4.5 Sustainability and Environmental Impact Evaluation

- **Energy Efficiency**: We need to design robots with energy-efficient components, including low-power motors and optimized cleaning algorithms.
- **Battery Management**: We can use rechargeable batteries and renewable sources for energy sources.
- **Eco-Friendly Cleaning Solutions**: The choice of cleaning solutions can affect water quality and environmental pollution. We will encourage the use of eco-friendly and

biodegradable cleaning solutions that minimize harm to aquatic ecosystems and water sources.

- **Cleaning Efficiency**: We will adapt to efficient cleaning algorithms and mechanisms to reduce water and energy consumption.
- **Noise Pollution**: We will incorporate noise-reduction features into robot design. Comply with noise regulations and standards to minimize environmental noise pollution.

4.6 Ethical Issues

Floor mopping robots, like any technological advancement, raise various ethical issues that need careful consideration. Here are some of the ethical concerns associated with floor mopping robots:

- **Privacy Concern**: Developers should implement robust privacy protections, including clear user consent mechanisms and features that allow users to disable cameras and sensors when not needed.
- **Data Collection and Usage:** Clearly communicate data collection practices to users and obtain their informed consent. Ensure that data is used solely for the purpose of improving robot performance and not for other purposes without user consent.
- **Safety for Vulnerable Populations**: Design robots that prioritize the safety of vulnerable users. Implement features like obstacle detection, child locks, and accessibility features to ensure safe use.
- **Accessibility**: Ensure that robots are designed with accessibility features, such as tactile controls, audible alerts, and compatibility with assistive technologies, to accommodate users with diverse needs.
- **Transparency and Accountability**: We need to provide clear and understandable explanations of the robot's operation and decision-making processes.
- **Consumer Education**: Users should be educated about the ethical considerations and responsible use of floor mopping robots.

Addressing these ethical concerns through responsible design, user education, and regulatory frameworks can help ensure that floor mopping robots are developed and used in ways that respect individuals' rights, safety, and well-being while also benefiting society.

5 Reflection on Individual and Team work

5.1 Individual Contribution of Each Member

- 1906066: Sheikh Abu Al Raihan: Motor and Motor Driver interfacing, designing mopping brushes
- 1906067: Kazi Abid Hasan: Building algorithm, navigating the Bot assembling the body
- 1906071: Arpa Roy Dastider: Implementing Sonar Sensor into the navigation system, building the bot
- 1906072: Ramisa Tahsin Shreya: Operating Relay Pump with Arduino, logic buildup
- 1906082: Md Ramim Hassan Shawn: Rotating the Bot 90 &180 degree using Gyro sensor, soldering the components
- 1906092: Dipika Rani Nath: Servo Motor control & assembling the bot

5.2 Mode of Team Work

The team worked from the very beginning and became successful. The team had a lot of diversity. The individuals worked in different areas and then all six had cascaded the circuits to build the project.

5.3 Diversity Statement of Team

The team members came from different regions of the country. There were sometimes tensions between the team members, but at end of the day the members worked together for ensuring the project becomes a success

5.4 Log Book of Project Implementation

Date	Milestone achieved	Name	Individual role	Team role	Comments
Week 1	Group formed	Evan Abid Arpa	Discussed with each other to form this group	Formed a team	glad to be working in the same team

		Ramisa			
		Ramim			
		Dipika			
Week 2	Project idea finalisation	Evan	brainstorming about project idea	Came up with a project idea	glad to be working in the same team
		Abid	brainstorming about project idea		
		Arpa	brainstorming about project idea		
		Ramisa	brainstorming about project idea		
		Ramim	brainstorming about project idea		
		Dipika	brainstorming about project idea		
Week 3	Project proposal	Evan	Providing information about the project	Project proposal pitched	We were given a task to perform navigation system in our project
		Abid	Coming up with the idea mopping bot		
		Arpa	Coming up with the idea mopping bot		
		Ramisa	Preparing slide for proposal		
		Ramim	Providing information about the project		
		Dipika	Providing information about the project		
Week 4	Online meeting	Evan	Conducted the meeting	Discussing how to start our project	We got a clear idea how to work in our project
		Abid	Arranged an online meeting		
		Arpa	Discussed how to start our project		
		Ramisa	Discussed different approach		
		Ramim	Discussed different approach		
		Dipika	Contributed with important information		
Week 5	Listing and buying the components	Evan	Bought the components	Buying the components of the project	It was tough to buy these components in these humid temperature
		Abid	Bought the components		
		Arpa	Listed the price of the components		
		Ramisa	Found out the necessary components of the project		
		Ramim	Bought the components		
		Dipika	Made excel sheet of cost		
Week 6	Coding of basic components	Evan	Worked with motor driver	Working with components individually	The water pump was tough to work with
		Abid	Worked on the algorithm of the navigation		
		Arpa	Worked with sonar sensor		
		Ramisa	Worked on water pump and relay		
		Ramim	Worked on gyro sensor		
		Dipika	Worked with servo motor		
Week 7	Project progress	Evan	Updated about what we have done so far	Updating our course teachers about the progress of our project	We got important advice on the challenges we face from our respected supervisors
		Abid	Discussed the detailed implementation plan		
		Arpa	Talked about the navigation system and motor interface		
		Ramisa	Talked about how we worked on the pump and relay and mopping mechanism		
		Ramim	Discussed the challeges we have faced so far		
		Dipika	Updated about the components and our further plan for the project		

Week 8	Mounbted the pump and gyro on the bot	Evan	Figured out how to use the mopper	We started putting the component together to get a more profound working robot	The perfect 90 degree angle of rotation was challenging task
		Abid	Worked on the code on arduino		
		Arpa	Worked on the robot body		
		Ramisa	Searched how to link two arduino uno together		
		Ramim	Worked on how to rotate the bot 90 degree perfectly		
		Dipika	Searched how to link two arduino uno together		
Week 9	Linear Navigation task completed	Evan	Worked with solder iron on some components	Making sure we got a fully navigating bot	We faced some challenges as we had to debug our code
		Abid	Wrote the code on arduino for autonomous navigation		
		Arpa	Wrote the code on arduino for autonomous navigation		
		Ramisa	Helped with soldering		
		Ramim	Made the bottle for water reservoir		
		Dipika	Helped in testing the bot movement		
Week 10	Detecting three walls	Evan	Soldering the important components	To stop our autonomous mopping bot	To figure out the algorithm for stopping the bot was challenging
		Abid	Coming up with the idea on when to stop the bot		
		Arpa	Wrote the necessary code		
		Ramisa	Mounted other two sonar on to the bot		
		Ramim	Helped with adding the sonar sensor on to the bot		
		Dipika	Debugged the code		
Week 11	Completing the bot	Evan	Worked on the algorithm	To get a fully functioning bot	We faced a problem with the pump pipe as it was too wide
		Abid	Wrote the code		
		Arpa	Debugging the code of three sonar sensors		
		Ramisa	Helping with making the temporary room of the bot		
		Ramim	Testing the movements of the bot		
		Dipika	Helped with perfecting the algorithm		
Week 12	Debugging problem	Evan	Worked on the wheel to reduce manual error	Finding out the error of the bot	The project is completed
		Abid	Worked with how to solve the pumping problem		
		Arpa	Wrote code for reducing the error for navigation system		
		Ramisa	Tested the bot movements		
		Ramim	Worked with spray and mechanism for alternating pumping system		
		Dipika	Helped to calculate the error margin		
Week 13	Final project presentation	Evan	Worked in report	This is yet to be evaluated	Everyone worked hard for last 3 month to complete this project
		Abid	Gave the bot a final test run and readied the design method		
		Arpa	Gave the bot some final touch and worked in report		
		Ramisa	Designed the ppt for presentation		
		Ramim	Wrote some part of the report and worked in ppt		
		Dipika	Helped with the report		

6 Communication

6.1 Executive Summary

Introducing CleanSweep: Your Autonomous Mopping Partner

The Autonomous Mopping Bot is designed to efficiently clean floors while navigating autonomously within a given space. The project utilizes an Arduino microcontroller and motor driver for control. When an obstacle is detected, the bot employs a smart navigation strategy. The bot is equipped with a water pump connected to a water bottle and a rotating brush underneath for effective cleaning

6.2 User Manual

At first, switch on the bot. It will power up the arduino and motor driver using a lipo battery with 11.1 V DC supply. Then the robot will start moving forward while cleaning the floor

7. Bill of Materials

Component	Taka
Lipo Battery	1750
5 V Relay	50
Lipo Battery charger	380
Gyro Sensor	180
Servo Motor	110
Sonar Sensor	210
Jumper	320

Motor	120
Hexa	370
Clamp	320
Lipo Battery	1750
5 V Relay	50
Lipo Battery charger	380
Gyro Sensor	180
Servo Motor	110
Sonar Sensor	210
Jumper	320
Motor	120
Hexa	370
Clamp	320

8. Future Work:

1. Work with improved obstacle detection and avoidance.
2. Explore technologies like SLAM (Simultaneous Localization and Mapping) for better mapping.
3. Introduce the idea of machine learning for optimizing cleaning patterns.
4. Mention the potential for adapting cleaning strategies to different floor types and dirt levels.
5. Explain how the mopping bot can be integrated with smart home systems.
6. Consider voice control, remote monitoring, and scheduling features.

7. Explore the use of eco-friendly cleaning solutions and materials.
8. Investigate options for water and energy conservation.
9. Address the need for longer operating times.
10. Investigate advancements in battery technology.

9. References

- *https://circuitdigest.com/microcontroller-projects/arduino-floor-cleaning-robot*
- *https://circuitdigest.com/microcontroller-projects/build-your-own-arduino-based-smart-vacuum-cleaning-robot-for-automatic-floor-cleaning*