BATCH 45- PRATIBHA V B

GRASS CUTTING ROBOT

PROBLEM STATEMENT:

The objective of this project is to design and build an autonomous grass-cutting robot using Arduino that can navigate a designated lawn area, ensuring complete grass coverage while efficiently detecting and avoiding obstacles to prevent collisions with objects and safely maintaining the grass at a consistent height. The robot will feature a user-friendly interface for setting mowing schedules and monitoring its status while emphasizing energy efficiency, safety, and cost-effective design.

SCOPE OF THE SOLUTION:

Autonomous Navigation: The solution will enable the robot to autonomously move within a predefined lawn area, providing precise control over forward, backward, left, and right movements, ensuring comprehensive grass coverage. The navigation system will incorporate obstacle detection and avoidance mechanisms to prevent collisions with objects in the environment.

Grass Maintenance: A cutting mechanism will be developed and integrated into the robot to ensure it can consistently and evenly cut the grass to maintain a specified height without causing damage to the grass or the cutting equipment.

Safety: Comprehensive safety features will be implemented to ensure that the cutting mechanism poses no harm to humans, animals, or property. Emergency stop and shutdown procedures will be in place to halt the robot in unforeseen situations.

Energy Efficiency: The robot will be designed to optimize power consumption, enabling extended operation on a single battery charge. Where necessary, a charging and docking system will be considered to enhance operational autonomy.

User Interface: A user-friendly interface will be provided for users to set mowing schedules, define lawn boundaries, and monitor the robot's operational status.

Cost Management: The project will be executed within a reasonable budget, emphasizing cost-effective selection of components and materials to keep the solution financially viable.

REQUIRED COMPONENTS:

SOFTWARE:

1.Arduino ide

2. Fritzing

HARDWARE:

1. Arduino Uno
2. Motor Driver Shield
3. 3 Motors (BO)
4. Chassis
5. 2 Wheels
6. Ultrasonic Sensor
7. Servo Motor
8. 9V battery

CODE:

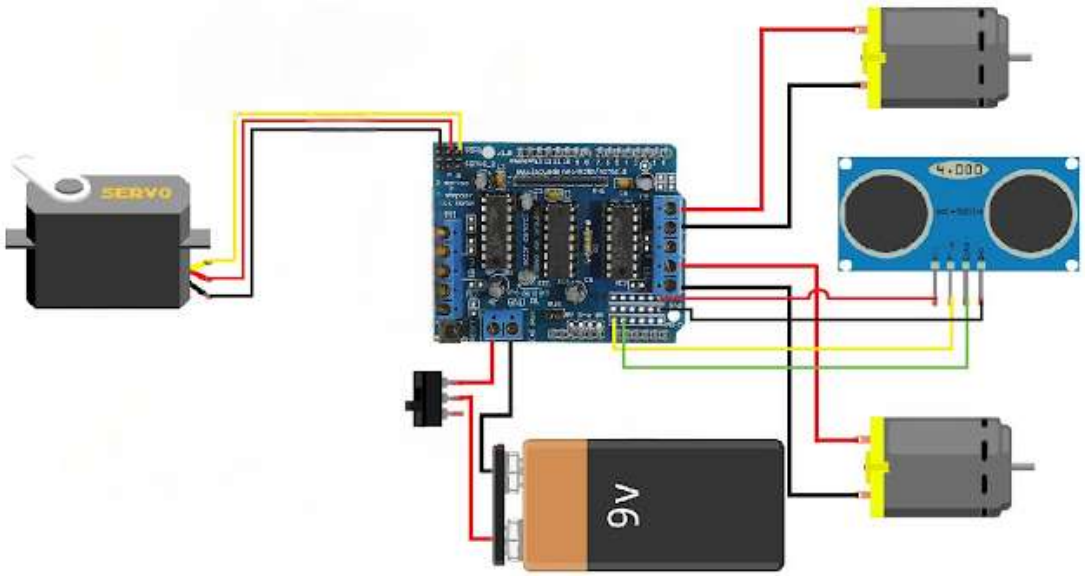
```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>

#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 200
#define MAX_SPEED 190
#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
Servo myservo;

boolean goesForward=false;
int distance = 100;
int speedSet = 0;

void setup() {

  myservo.attach(10);
  myservo.write(115);
  delay(2000);
  distance = readPing();
```

```
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}

void loop() {
 int distanceR = 0;
 int distanceL =  0;
 delay(40);

 if(distance<=15)
 {
  moveStop();
  delay(100);
  moveBackward();
  delay(300);
  moveStop();
  delay(200);
  distanceR = lookRight();
  delay(200);
  distanceL = lookLeft();
  delay(200);

  if(distanceR>=distanceL)
  {
    turnRight();
    moveStop();
  }else
  {
    turnLeft();
    moveStop();
  }
 }else
 {
  moveForward();
 }
 distance = readPing();
}

int lookRight()
{
    myservo.write(50);
    delay(500);
    int distance = readPing();
```

```
    delay(100);
    myservo.write(115);
    return distance;
}

int lookLeft()
{
    myservo.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(115);
    return distance;
    delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
  {
    cm = 250;
  }
  return cm;
}

void moveStop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);

  }

void moveForward() {

 if(!goesForward)
  {
    goesForward=true;
    motor1.run(FORWARD);
    motor2.run(FORWARD);

   for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
   {
    motor1.setSpeed(speedSet);
    motor2.setSpeed(speedSet);

    delay(5);
   }
  }
```

```
}

void moveBackward() {
   goesForward=false;
   motor1.run(BACKWARD);
   motor2.run(BACKWARD);

  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
  {
   motor1.setSpeed(speedSet);
   motor2.setSpeed(speedSet);

   delay(5);
  }
}

void turnRight() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);

  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);

}

void turnLeft() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);

  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);

}
```

SIMULATED CIRCUIT DIAGRAM (FRITZING):



VIDEO LINK:

https://drive.google.com/file/d/1cpVOFDLeMYiutpzmnkQbUZLQDhBhcOgW/view?usp=sharing

GIT REPOSITORY LINK:

https://github.com/Pratibhavb/L-T-PROJECT.git