Autonomous Fertilizer Line Follow Robot

J.P.T.Nadeeshan	MAHDSE211F-019
C.O.Jayawardana	MAHDSE211F-018
R.B.Theekshana	MAHDSE211F-026
K.R.Weerawardhana	MAHDSE211F-005
L.G.I.Vishwajith	MAHDSE211F-006

School of Computing

National Institute of Business Management

2023

DECLARATION

"I certify that this project does not incorporate without acknowledgement, any material previously submitted for a Higher National Diploma in any institution and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my project report, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations."

Group Members

J P T Nadeeshan	MAHDSE211F-019	………………………
C O Jayawardana	MAHDSE211F-018	………………………
R B Theekshana	MAHDSE211F-026	………………………
K R Weerawardhana	MAHDSE211F-005	………………………
L G I Wishwajith	MAHDSE211F-006	………………………

Submitted to

Mr.Asanka Amarasinghe ………………………………………………..

TABLE OF CONTENT

ACKNOWLEDGEMENT

We would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude we give to our RAD lecturer Mr. Asanka Amarasinghe, because he always supported and guided us while doing this project. He very well cleared all the doubts I had regarding this project. Also, we would like to especially thank my friends who helped me a lot to complete this project within the limited time. Apart from the efforts of the team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to all that have been instrumental in the successful completion of this project. Last but not least, we are eternally grateful to our lecturer for his even willingness to give us valuable advice and direction under which we executed this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depth and helped us to complete the assigned tasks. Once again, thanks to everyone who was involved with this project from beginning to end.

ABSTRACT

The aim of this project is to develop a system for fertilizing tea plantations using a line-following robot that can move efficiently across the terrain. The robot is equipped with sensors and a microcontroller to follow a specified path on the ground while simultaneously watering and fertilizing the plants.

To build the robot, we used an Arduino Uno board, an IBT_2 motor driver, two IR sensors, and a watering system consisting of a water tank, a pump, servo motor, and nozzles. The robot has Bluetooth connectivity, enabling remote control operation from a smartphone or tablet. We also utilized the PID algorithm in the control program, developed using the Arduino IDE, to ensure precise line following and efficient resource consumption.

We tested the robot's ability to follow a line course in a field while spraying fertilizer on the crops. In the future, we may include soil moisture sensors or a camera to monitor the plants.

Overall, the line-following robot is a cost-effective and efficient solution for automating fertilizer application in agriculture. The project includes recommendations for future improvements and uses, as well as a detailed explanation of the robot's design, implementation, and testing.

INTRODUCTION

The Arduino, a popular microcontroller platform, has been utilized in constructing a robot car that is specifically designed to automate the application of fertilizer to plant. use for agriculture field.

The main objective of this robot car is to reduce labor and time needed to apply fertilizer. The robot car is equipped with a line-follower for two IR sensor, allowing it to travel along a pre-determined path in the field. In addition, it has a fertilizer tank and is programmed to stop after traveling 50cm, activating a servo motor to rotate 180 degrees while spraying water through a water pump. This ensures even coverage and efficient use of resources during the application of fertilizer.

Furthermore, the robot car features Bluetooth connectivity, which enables remote control through a smartphone or tablet. This comes in handy when manual steering is required, such as steering around obstacles or returning to the starting location. The robot car is powered by two lithium-ion batteries that provide sufficient power to run the car for several hours.

To program the line-following robot car, the Arduino IDE, an open-source software development environment, was used. This has resulted in a cost-effective and efficient solution that boosts crop yield while minimizing the time and effort required in the application of fertilizer. The study includes recommendations for further improvements and alternative applications, as well as a detailed explanation of the design, implementation, and testing of the robot car.

MATERIALS

Caster & Wheels

Chassis is used for building your own Arduino robot car etc. Robot car chassis kit that is commonly used with the Arduino board.

Gear Motors

Gear motors are used in applications that require high output torque and lower output shaft rotational speed. Line follower robot has two wheels which are mounted with gear motors to maintain the speed level while following the line.

Infrared (IR) Sensors

The IR sensors mounted under the robot near the caster wheel helps inthe detecting the line. The Infrared (IR) sensors consist of Infrared (IR) LEDand Infrared (IR) photodiodes. The IR LED is called photoemitter and IR photodiode is called receiver. The IR light emitted by the LED strikes the surface and gets reflected back to the photodiode. Then the photodiode gives an output voltage which is proportional to the reflectance of the surface which will be high for a light surface and low for dark surface. Lightcolored objects reflect more IR light and dark colored objects reflect less IR light.

Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other withoutsoldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

Servo Motor

There are many different types of actuators such as servo motors etc. The servo motor is used in robotics to activate movements, giving the arm rotating. We have used servo motor to create the Arm of the water spray rotation.

Motor Driver Controller L298N

A motor controller is a device or electronic circuit that is responsible for full control of the motor such as torque control, speed control, direction control, taking feedback from the motor. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. This motor controller is basically used for controlling the motor (the wheel).

Power Supply

Main power supply for line follower robot is lithium-ion battery, a lithium-ion battery is a type of rechargeable battery that is charged and discharged by lithium ions moving between the negative (anode) and positive (cathode) electrodes. (Generally, batteries that can be charged and discharged repeatedly are called secondary batteries, whereas disposable batteries are called primary batteries.)

We used lithium-ion batteries for line follower robot since lithium-ion batteries are suitable for storing high-capacity power and they are used in a wide range application.

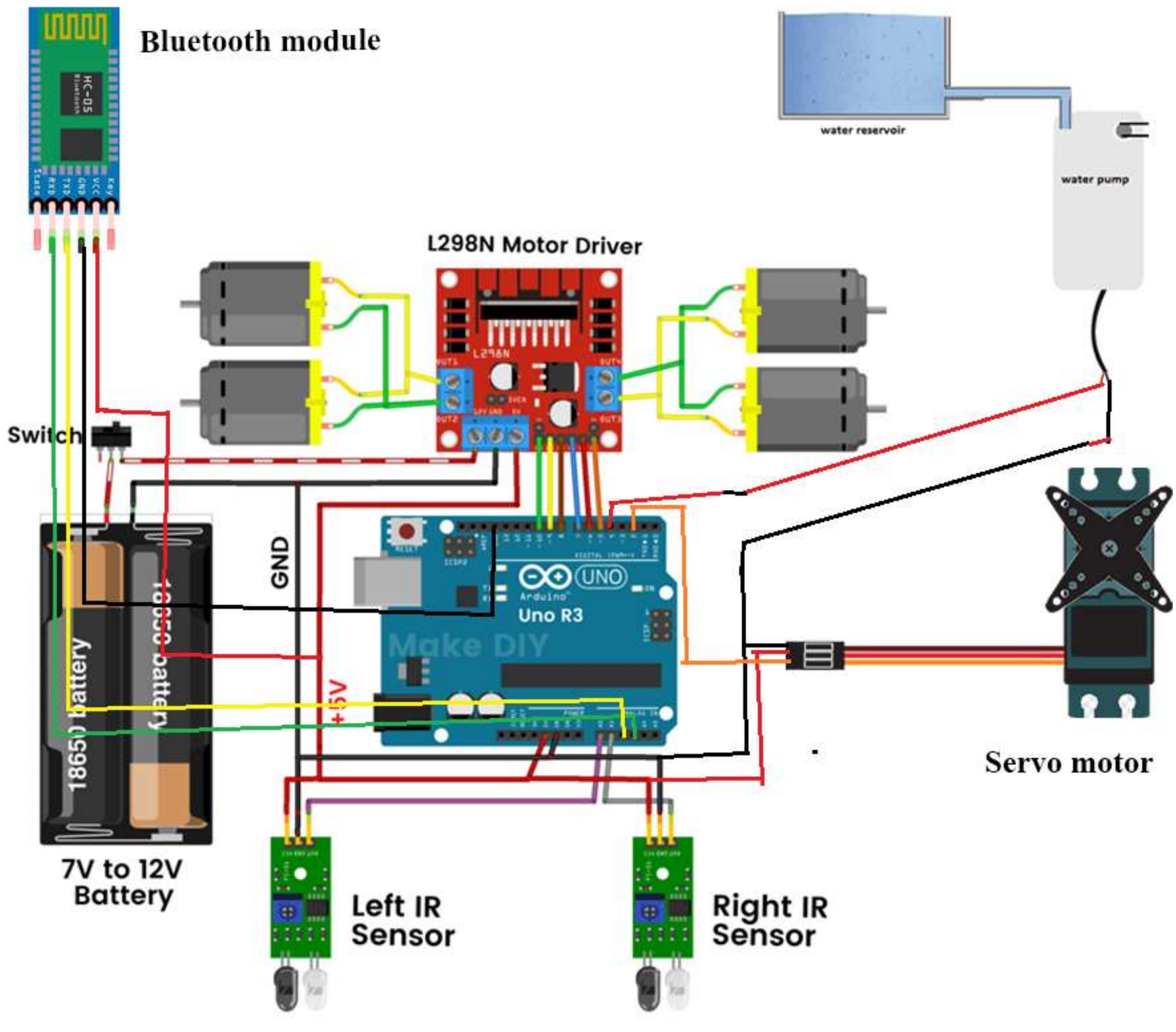
Water Pump

A water pump controlled by an Arduino microcontroller is a type of system that uses an Arduino to control the operation of a water pump. The Arduino can be used to turn the pump on and off, as well as to control the flow rate and direction of the water.

Bluetooth Module hc-06

The HC-06 is a class 2 slave Bluetooth module designed for transparent wireless serial communication. Once it is paired to a master Bluetooth device such as a PC, smartphone, and tablet, its operation becomes transparent to the user. For control robot.

METHODS


Bluetooth module
water reservoir
water pump
L298N Motor Driver
Switch
GND
+5V
Uno R3
Servo motor
7V to 12V
Battery
Left IR
Sensor
Right IR
Sensor

Design and assemble the chassis and wheels of the robot. Mount the motor driver module, ultrasonic sensor, and Arduino Uno microcontroller onto the chassis.Connect the motor driver module to the two DC motors that will control the movement of the robot. Connect the ultrasonic sensor to the Arduino board.

Program the Arduino board to control the movement of the robot using the motor driver module and to read the distance measured by the ultrasonic sensor. Attach the servo motor and peristaltic pump to the chassis. Connect the peristaltic pump to the fertilizer tank.

Program the Arduino board to control the servo motor to rotate the peristaltic pump and dispense the appropriate amount of fertilizer based on the distance measured by the ultrasonic sensor. Test the robot by placing it in a garden or agricultural field and observing its movement and fertilizer distribution.

Adjust the program and sensor readings as necessary to ensure accurate fertilizer distribution. Once the robot is operating effectively, use it to automate the fertilizer application process in the garden or field.

RESULTS

A successful project has resulted in the creation of a line-following robot car, capable of accurately navigating a pre-determined path while dispensing fertilizer to plants. The car comes equipped with an infrared sensor on an Arduino Uno microcontroller board, enabling it to drive through fields with ease.

The robot car is equipped with a spray arm that rotates 180 degrees, and when the car has traveled 50cm, the water pump activates and sprays water onto the plants. Precise line-following and efficient resource consumption are achieved through the utilization of a PID algorithm in the car's control program, developed using the Arduino IDE.

The car boasts Bluetooth connectivity, enabling remote control operation via a smartphone or tablet, enhancing convenience and flexibility. Remote car control and manual water spraying and arm rotation are possible through the Bluetooth connection.

The robot vehicle underwent rigorous testing in various terrain conditions, including flat and rugged areas, and performed effectively without incident, following the designated path accurately.

During testing, the fertilizer pump on the car evenly distributed fertilizer onto the plants, providing them with the necessary nutrients for growth. The fertilizer tank utilized by the car has a significant capacity and can cover a vast area before requiring replenishment.

Battery life testing revealed that the robot car can operate continuously for several hours before requiring a recharge. Overall, the line-following robot car is an innovative and effective way to automate fertilizer application, reducing time and effort required for the task. Bluetooth connectivity adds to the car's convenience and flexibility, allowing users to remotely control the vehicle.

DISCUSSION

The development of the line-following robot car in this project showcases the potential benefits of automation in agriculture. With its ability to accurately follow a predetermined course while applying fertilizer to crops, the robot car can significantly reduce the amount of time and labor required for this task, leading to more efficient and environmentally friendly farming practices.

The robot car's autonomous navigation feature eliminates the need for human intervention, which can be challenging in some areas due to a shortage of skilled operators. Additionally, the robot car's ability to operate continuously for several hours without breaks or rest can improve the overall efficiency of the process. The use of the open-source Arduino Mega microcontroller board offers several advantages, such as flexibility, ease of programming, and low cost, making it an ideal platform for creating robotics projects.

The Bluetooth connectivity of the robot car provides added flexibility and convenience, allowing users to operate the vehicle remotely using a smartphone or tablet. The robot car was tested successfully in various conditions, including flat and uneven terrain, and demonstrated its ability to evenly spray fertilizer onto crops, potentially increasing yields and promoting uniform plant growth.

One potential limitation of the line-following robot car is its dependence on a predetermined path. It may be unable to navigate through the field if the path is altered or blocked. However, this can be addressed by incorporating additional sensors, such as GPS or ultrasonic sensors, to enable obstacle detection and allow the robot car to move through various environments.

In summary, this project's line-following robot car demonstrates the potential for automation in agriculture, offering a more sustainable and efficient approach to fertilization. Its use of the Arduino Uno microcontroller board and Bluetooth connectivity provides flexibility and adaptability, making it an excellent platform for future robotics projects in the agriculture industry and beyond.

REFERENCES

Anon., n.d. [Online] Available at: https://www.researchgate.net/publication/327814718_PROJECT_REPORT_LINE_FOLLOWING_ROBOT

Anon., n.d. [Online] Available at: https://circuitdigest.com/electronic-circuits/ir-sensor-circuit-diagram

Anon., n.d. [Online] Available at: https://circuitdigest.com/microcontroller-projects/raspberry-pi-line-follower-robot

Appendices

```
#define ls 10//Enable L298 Pin right side
#define rf 9 //right foward
#define rb 8 //right backward
#define lf 7 //left forword
#define lb 6 //left back
#define rs 5 //Enable L298 Pin left side
#define R_S A0 //ir sensor Right
#define L_S A1 //ir sensor Left
//#define pump 12//pump moter
#include <Servo.h> //Include the Servo library
Servo myservo; //Create a Servo object
int pumpPin = 13; //The digital pin to which the pump is connected
int angle = 11; //Initial angle of the servo
unsigned long previousMillis = 0; //Variable to store the last time the servo and pump were
turned on
const long interval = 10000; //The interval between turning on the servo and pump (in
milliseconds)
char Incoming_value = 0;
int type;
void setup(){ // put your setup code here, to run once
Serial.begin(9600);
//pinMode(13, OUTPUT);
pinMode(R_S, INPUT); // declare if sensor as input
pinMode(L_S, INPUT); // declare ir sensor as input
pinMode(ls, OUTPUT); // declare as output for L298 Pin enA
pinMode(rf, OUTPUT); // declare as output for L298 Pin in1
pinMode(rb, OUTPUT); // declare as output for L298 Pin in2
pinMode(lb, OUTPUT); // declare as output for L298 Pin in3
```

```
pinMode(lf, OUTPUT); // declare as output for L298 Pin in4
pinMode(rs, OUTPUT); // declare as output for L298 Pin enB
myservo.attach(10); //Attach the servo to pin 10
13
pinMode(pumpPin, OUTPUT);
//PinMode(pump, OUTPUT);
//analogWrite(ls, 150); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
//analogWrite(rs, 150); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed
delay(1000);
}
void loop(){

if(Serial.available() > 0) {
 Incoming_value = Serial.read();
 if(Incoming_value == '8'){ type=1;}
 else if(Incoming_value == 'v'){type=0;}
 else if(Incoming_value == 'a'){type=2;}
 else if(Incoming_value == 'b'){type=3;}
 else if(Incoming_value == '7'){type=4;}
 else if(Incoming_value == '6'){type=5;}}
if(type==1){
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){forword();} //if Right Sensor and Left
Sensor are at White color then it will call forword function
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();} //if Right Sensor is Black
and Left Sensor is White then it will call turn Right function
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();} //if Right Sensor is White
and Left Sensor is Black then it will call turn Left function
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){Stop();} //if Right Sensor and Left
Sensor are at Black color then it will call Stop function
}
else if(type==3){
```

```
//auto spray
}
else if(type==4){
 //manual spray turn right
 for (angle = 180; angle >= 0; angle -= 1) { //Gradually decrease the angle of the servo
 myservo.write(angle); //Set the servo angle
 delay(15); //Wait for the servo to reach the angle
 }
}
else if(type==5){
 //manual spray turn left
 for (angle = 0; angle <= 180; angle += 1) { //Gradually increase the angle of the servo
14
 myservo.write(angle); //Set the servo angle
 delay(15);
 digitalWrite(pumpPin,HIGH);
}
}
else if(type==2){
 //stop spray
 digitalWrite(pumpPin,LOW);
}
}
void forword(){ //forword
//analogWrite(rs, 254);
//analogWrite(ls, 254);
digitalWrite(rf, HIGH); //Right Motor forword Pin
digitalWrite(rb, LOW); //Right Motor backword Pin
digitalWrite(lb, LOW); //Left Motor backword Pin
digitalWrite(lf, HIGH); //Left Motor forword Pin
}
```

```
void turnRight(){ //turnRight
digitalWrite(rf, LOW); //Right Motor forword Pin
digitalWrite(rb, HIGH); //Right Motor backword Pin
digitalWrite(lb, LOW); //Left Motor backword Pin
digitalWrite(lf, HIGH); //Left Motor forword Pin
}
void turnLeft(){ //turnLeft
digitalWrite(rf, HIGH); //Right Motor forword Pin
digitalWrite(rb, LOW); //Right Motor backword Pin
digitalWrite(lb, HIGH); //Left Motor backword Pin
digitalWrite(lf, LOW); //Left Motor forword Pin
}
void Stop(){ //stop
digitalWrite(rf, LOW); //Right Motor forword Pin
digitalWrite(rb, LOW); //Right Motor backword Pin
digitalWrite(lb, LOW); //Left Motor backword Pin
digitalWrite(lf, LOW); //Left Motor forword Pin
}
void backword(){ //stop
digitalWrite(rf, LOW); //Right Motor forword Pin
digitalWrite(rb, HIGH); //Right Motor backword Pin
digitalWrite(lb, HIGH); //Left Motor backword Pin
```

15

```
digitalWrite(lf, LOW); //Left Motor forword Pin
}
void autospray(){
 unsigned long currentMillis = millis(); //Get the current time
 if (currentMillis - previousMillis >= interval) { //Check if it's time to turn on the servo and
pump
 previousMillis = currentMillis; //Update the previousMillis variable
 //Turn on the servo
```

```
 for (angle = 0; angle <= 180; angle += 1) { //Gradually increase the angle of the servo
 myservo.write(angle); //Set the servo angle
 delay(15); //Wait for the servo to reach the angle
 }
 delay(5000); //Wait for a second
 //Turn on the pump
 digitalWrite(pumpPin, HIGH); //Set the pump pin to HIGH
 //digitalWrite(pump, HIGH);
 delay(5000); //Wait for a second
 //Turn off the servo and pump
 for (angle = 180; angle >= 0; angle -= 1) { //Gradually decrease the angle of the servo
 myservo.write(angle); //Set the servo angle
 delay(15); //Wait for the servo to reach the angle
 }
 digitalWrite(pumpPin, LOW); //Set the pump pin to LOW
 //digitalWrite(pump, LOW);
 if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){forword();} //if Right Sensor and
Left Sensor are at White color then it will call forword function
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();} //if Right Sensor is Black
and Left Sensor is White then it will call turn Right function
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();} //if Right Sensor is White
and Left Sensor is Black then it will call turn Left function
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){Stop();} //if Right Sensor and Left
Sensor are at Black color then it will call Stop function
 }
}
```