Semi-Automated Field Management Robot

1st Sahada Islam Evan
dept.CSE
United International University
Dhaka, Bangladesh
sevan203030@bscse.uiu.ac.bd

2nd MD. Shabbir Hossain Sourav
dept.CSE
United International University
Dhaka, Bangladesh
msourav203023@bscse.uiu.ac.bd

3rd Sonjoy Dey
dept.CSE
United International University
Dhaka, Bangladesh
sdey202074@bscse.uiu.ac.bd

4th Tazre Mohammad Abdullah
dept.CSE
United International University
Dhaka, Bangladesh
tabdullah181060@bscse.uiu.ac.bd

5th Tun Tun U
dept.CSE
United International University
Dhaka, Bangladesh
tu193035@bscse.uiu.ac.bd

***Abstract*—Semi-automated field management robot will help you during rainy time. It automatically detects rain and covers the cricket pitch. And it tested the soil and gave the information in mobile that it was wet or dry. If it was dry we can 'on' the water giving motor and this motor was given water in all the fields. We can also add a dark detecting sensor. If it is dark in the stadium then it will 'on' the light automatically. And we can also control the robot, monitor and access the data using our smart device wirelessly any time..**

***Index Terms*—Node MCU ESP8266, Soil moisture, Rain sensor , LDR ,Rain sensor,TT-gear motor and wheel,Mini servo,AAA Battery**

I. Introduction

This is a Semi-Automated field management robot for smart stadium.

II. Project Overview

A. Maintaining a field

The smart stadium semi-automatic field management robot is a robot that uses various electronic components to detect the current weather. When it rains, it will detect the rain and send a signal to the server to notify the user device and can control a semi-automatic field management robot to cover a cricket field without human intervention. With it, you can easily get out onto the cricket pitch even after the rain has stopped. Not only can it be used to cover pitches, but it can also be used to roll cricket pitches. Now the cricket pitch is rolled by humans, but this does require a human and you can sit in a room and have a human roll the cricket pitch easily. You can roll the cricket field outside the cricket field as well as the cricket field. This robot also has another feature that allows it to check the soil moisture and help us when we need water. It has a water pump and when the floor needs water it will notify the user server and we can put water on the floor. Also, if the soil is irrigated, it can be dried using a robot. This is useful on rainy days as the robot can tell you if the soil is irrigated or not. It can also detect how dark the stadium is. When the stadium gets dark in the evening, the stadium lights will automatically turn on, and no human intervention is required. All data is stored on the server, so you can control the robot with your smart device at any time and monitor and retrieve data wireless. Use Android apps to store data on the server. The name of these apps is BLYNK IoT. Using this app via your Android smart device, you can wireless control your robot and also control the data that your robot sees and stores. With this app, you can easily control a semi-automatic robot from top to bottom and left to right. Even if it starts to rain, the robot will send a notification to her BLYNK IOT server and the app will indicate the possibility of rain. In addition, the app also displays the percentage of soil moisture. Finally, we can easily check the total data that the robot sent to the server through this app and control the robot wireless.

Identify applicable funding agency here. If none, delete this.

III. Component list

1)Node MCU ESP8266
2)Soil moisture
3) Rain sensor
4)LDR
5)TT-gear motor and wheel
6)Mini servo
7)AAA Battery
8)Jumper Wire
9)Battery pack
10)Rain sensor
11)Bread Board

A. Solder less breadboard

A breadboard (sometimes called a plug lock) is used for building temporary circuits. It is useful to designers because it allows components to be removed and replaced easily. It is

useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit

B. Node MCU ESP8266

The Node MCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WIFI), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

C. Soil Moisture

Soil moisture sensors measure the volumetric water content in the soil. Since the direct gravimetric measurement of free-soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

D. Rain sensor

The Arduino Rain Sensor code is straightforward to understand. We just have to read the analogue data out of the sensor and we can approximate the average rainfall on top of the sensor. This can be quickly done in analogue mode. Digitally, we just need to set the potentiometer and we will get a digital output.

E. LDR

LDRs are tiny light-sensing devices also known as photoresistors. An LDR is a resistor whose resistance changes as the amount of light falling on it changes. The resistance of the LDR decreases with an increase in light intensity.

F. Motor driver L298n

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current of up to 2A. Let's take a closer look at the pinout of the L298N module and explain how it works.

G. TT-gear motor and wheel

Works for moving or rotating the robot from one place to another place.

H. Mini servo

Micro Servo Motor SG90 is a tiny and lightweight server motor with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos.

I. AAA Battery

The energy comes out of the positive and negative ends, and the machine feeds on that energy.

J. Jumper wire

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit

IV. Image

Fig. 1. Node MCU ESP8266

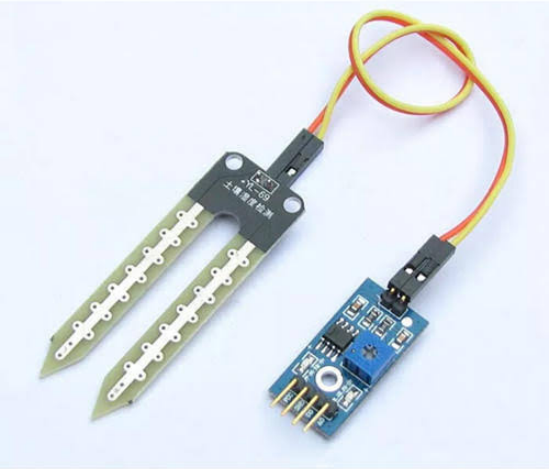

Fig. 2. Soil Moisture Sensor

V. Implementation

Node MCU ESP8266

Controls all the modules.

Soil Moisture Sensor

Check soil quality and water level of the field.

Fig. 3. Rain Sensor

Fig. 6. TT-gear motor and wheel

Fig. 4. LDR Sensor

Fig. 7. Mini Servo

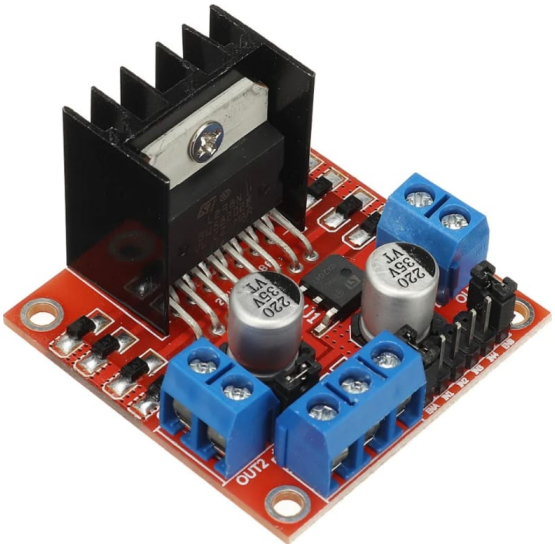

Fig. 5. Motor Driver L298n

Fig. 8. AAA Battery

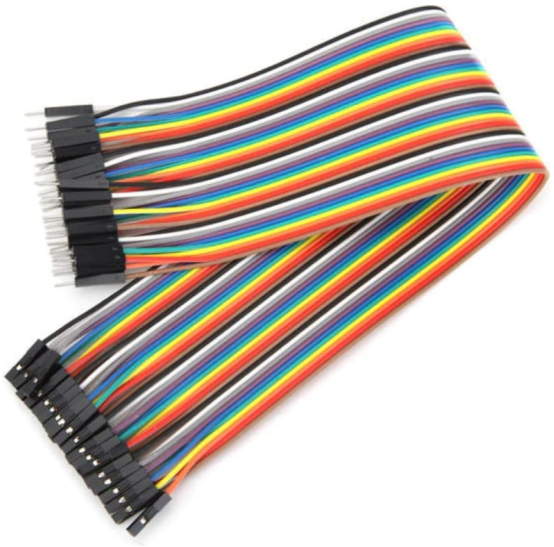

Fig. 9. Jumper Wire

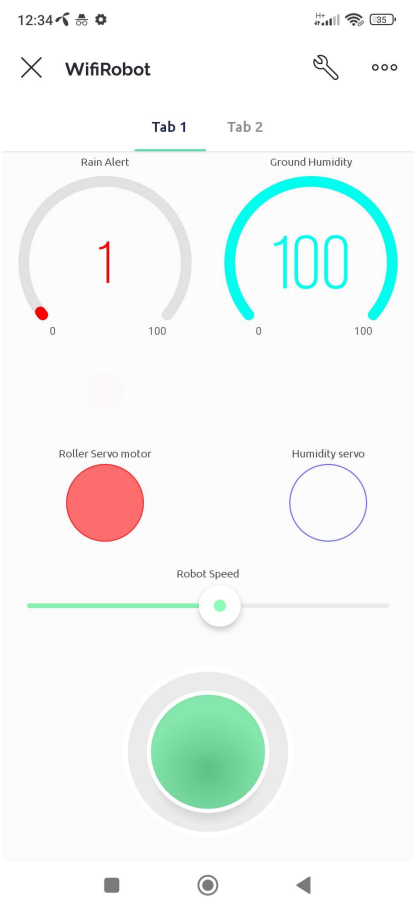

Fig. 10. Blynk IoT

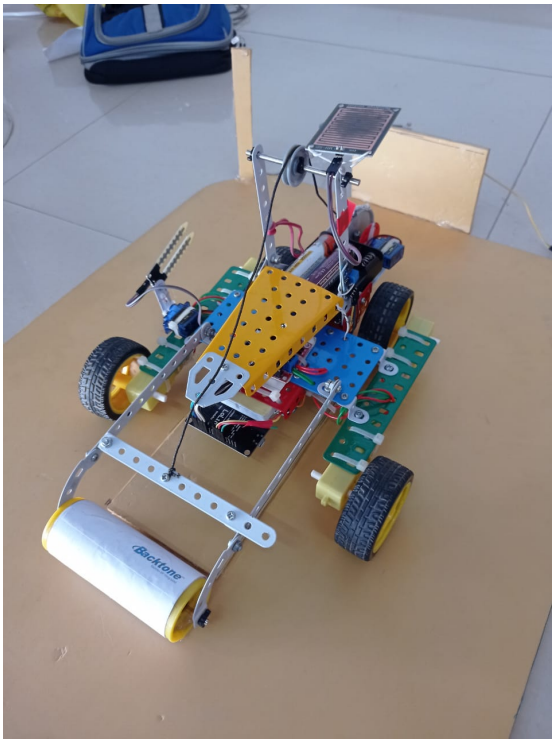

Fig. 11. Field Management Robot

Rain sensor

Detect rain and cover the field if rain starts.

LDR

To check the light level and on-off the light in the field automatically.

Motor driver L298n

controls a motor's speed, torque, direction, and resulting horsepower.

TT-gear motor and wheel

Use for moving this Semi-Automated Field Management Robot.

Mini servo

For moving soil moisture sensor and filed cover module.

AAA Battery

Supply power to all the components.

Jumper wire

For connecting all the modules with Node MCU ESP8266.

REFERENCES

[1] https://en.wikipedia.org/wiki/NodeMCU
[2] https://cb-electronics.com/products/sensor-soil-w-mod/
[3] https://store.roboticsbd.com/sensors/665-humidity-detection-sensor-module-rain-detection-for-arduino.html
[4] https://components101.com/modules/l293n-motor-driver-module
[5] https://circuitdigest.com/article/servo-motor-working-and-basics
[6] https://www.indiamart.com/ayinostech-bengaluru/light-sensor.html

APPENDIX

endlstlisting

this_is_the_car_marge_rain_soil_and_one_servo_v2

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include<Servo.h>
Servo servo;

// Define the motor pins

#define ENA D0
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4
#define ENB D5

// Variables for the Blynk widget values
int x = 50;
int y = 50;
int Speed;

int sensorpin=A0;// soil pin
int sensorvalue=0;
int outputvalue=0;

int rainSensorPin = D6;// rain

int servoOneRoler = D7;//servo
Servo servoTwo;

char auth[] = "V8TV7RxNP42-1KbXnhwCZ7aavnpF1ZPt"; //Enter your Blynk auth token
```

Fig. 12. Code

```
char auth[] = "V8TV7RxNP42-1KbXnhwCZ7aavnpF1ZPt"; //Enter your Blynk auth token
char ssid[] = "ASUS";           //WIFI Name
char pass[] = "12345678";           //WIFI Password

BlynkTimer timer;


//rain
void moisture() {
  int rainSensor = analogRead(rainSensorPin);// rain pin
  rainSensor = map(rainSensor, 100, 1023, 100, 0);
  Blynk.virtualWrite(V3, rainSensor);
  Serial.println(rainSensor);
}


void setup() {
  Serial.begin(9600);


  //Set the motor pins as output pins
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
```

Fig. 13. Code

this_is_the_car_marge_rain_soil_and_one_servo_v2

```
  //SERVO
  servo.attach(servoOneRoler);
  servoTwo.attach(15);


  // Initialize the Blynk library
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  timer.setInterval(100L, moisture);



}

//servo
BLYNK_WRITE(V5){

   servo.write(param.asInt());

}
BLYNK_WRITE(V6){

   servoTwo.write(param.asInt());

}


// Get the joystick values
BLYNK_WRITE(V0) {
```

Fig. 14. Code

```
void loop()
{
  Blynk.run();
  smartcar();// Call the main function
  timer.run();

  sensorvalue=analogRead(sensorpin);
  outputvalue=map(sensorvalue,0,1023,0,100); //100,0 pottupaaru
  //delay(1000);

  if(outputvalue>74)
  {
       //Serial.println("water your plant");
       Serial.print(outputvalue);
       Blynk.notify("water your plant");
       //delay(1000);
  }
  else if(outputvalue<45)
  {
       //Serial.println("soil is wet enough to water");
       Serial.print(outputvalue);
       Blynk.notify("soil is wet enough to water");
       //delay(1000);
  }

  Blynk.virtualWrite(V4,outputvalue);


}
```

```
Invalid library found in C:\Users\Sonjoy Dey\Documents\Arduino\libraries\libraries:
```

Fig. 15. Code