

Solar powered Automatic Pattern Design Grass Cutting Robot System based using Arduino

Dost Muhammad Khan¹, Zain Mumtaz ², Majid Saleem ², Zeeshan Ilyas ², Qinglu Ma ³ , Sahrish Ghaffar⁴ and Saleem Ullah ^{2,*}

¹Department of Computer Science & IT, The Islamia University of Bahawalpur, Pakistan

²Department of Computer Science, Khwaja Fareed University of Engineering & Information Technology, Rahim Yar Khan, 64200, Pakistan

³ School of Traffic and Transportation, Chongqing Jiaotong University, Chongqing, China

⁴Department of Computer Science, International Islamic University, Islamabad, Pakistan

*Correspondence: saleem.ullah@kfueit.edu.pk; Tel.: +92-300-7328-198

1. Supplementary Material

1.1. Mobile Application System

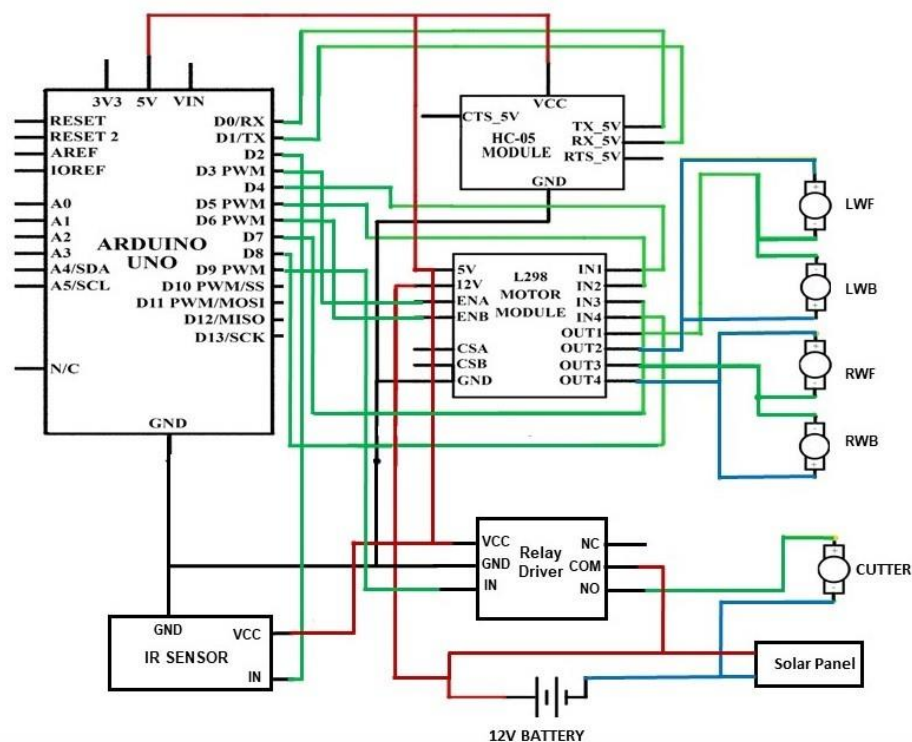


Figure 1. The circuit design of pattern design grass cutter controlling system with Android mobile application using Bluetooth.

Figure 1 shows the circuit design of the grass cutter system, which is control by an android mobile application using Bluetooth. In this scenario, the robotic grass cutting system will move in the same direction as the user presses the arrow touch button or speaks the corresponding keyword. In this task, one Arduino UNO, an HC-05 Bluetooth module, an L298N motor module, IR obstacle detection sensor, geared DC motor with encoder, simple DC motor, DC battery, solar panel, and a relay module were used.

All the explanation of Object Independent Automation System has been described in the figure 6 of main manuscript and the code is written below:

```
char Incoming_value = 0; //Variable for storing Incoming_Incoming_value
```

```

#define relay1 4 //N1
#define relay2 5 //N2
#define relay3 7 //N3
#define relay4 8 //N4
#define relay5 9 // motor control relay 5V
#define relay7 3
#define relay8 6
int isObstaclePin = 2; // This is our input pin
int isObstacle = HIGH; // HIGH MEANS NO OBSTACLE
void setup()
{
  Serial.begin(9600); //Sets the data rate in bits per second (baud) for serial data transmission
  pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(relay3,OUTPUT);
  pinMode(relay4,OUTPUT);
  pinMode(relay5,OUTPUT);
  pinMode(isObstaclePin, INPUT);
}
void loop()
{
  isObstacle = digitalRead(isObstaclePin);
  if(Serial.available() > 0)
  {
    Incoming_value = Serial.read(); //Read the incoming data and store it into variable
    Incoming_Incoming_valueue
    Serial.print(Incoming_value); //Print Incoming_value in Serial monitor
    Serial.print("\n"); //New line
    if( Incoming_value == '9' )
    {
      if (isObstacle == LOW)
      {
        Serial.println("OBSTACLE!!, OBSTACLE!!");
        digitalWrite(relay5,HIGH);
      }
      else{
        digitalWrite(relay5, LOW);
      }
    }
    else if( Incoming_value == 'S' ) {
      digitalWrite(relay1,LOW);
      digitalWrite(relay2,LOW);
    }
  }
}

```

```

    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    digitalWrite(relay5,HIGH);
}
//relay is off
else if( Incoming_value == 'A' ) {
    if (isObstacle == HIGH)
    {
        digitalWrite(relay1,HIGH);
        digitalWrite(relay2,LOW);
        analogWrite(3,255);
        digitalWrite(relay3,HIGH);
        digitalWrite(relay4,LOW);
        analogWrite(6,255);
    }
    else{
        digitalWrite(relay1,LOW);
        digitalWrite(relay2,LOW);
        digitalWrite(relay3,LOW);
        digitalWrite(relay4,LOW);
    }
    else if( Incoming_value == 'B' )
    {
        if (isObstacle == HIGH)
        {
            digitalWrite(relay1,LOW);
            digitalWrite(relay2,HIGH);
            analogWrite(3,255);
            digitalWrite(relay3,LOW);
            digitalWrite(relay4,HIGH);
            analogWrite(6,255);
        }
        else{
            digitalWrite(relay1,LOW);
            digitalWrite(relay2,LOW);
            digitalWrite(relay3,LOW);
            digitalWrite(relay4,LOW);
        }
    }
    else if( Incoming_value == 'C' )
    {
        if (isObstacle == HIGH)
        {

```

```

digitalWrite(relay1,LOW);
digitalWrite(relay2,LOW);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(6,255);
}
else{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    }}
else if( Incoming_value == 'D' )
{
    if (isObstacle == HIGH)
    {
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(3,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
    }
    else{
        digitalWrite(relay1,LOW);
        digitalWrite(relay2,LOW);
        digitalWrite(relay3,LOW);
        digitalWrite(relay4,LOW);
        }}
    else if( Incoming_value == 'O' )
    { digitalWrite(relay5,HIGH);
    }
    ///////////////    pattern's    ////////////////////////////
    else if(Incoming_value== 'T' ) //CIRCLE
    {
        if (isObstacle == HIGH)
        {
            Serial.println("OBSTACLE!!, OBSTACLE!!");
            digitalWrite(relay1,HIGH);
            digitalWrite(relay2,LOW);
            analogWrite(relay7,255);
            digitalWrite(relay3,LOW);
            digitalWrite(relay4,LOW);

```

```

    analogWrite(relay8,50);
    digitalWrite(relay5, LOW);
delay(3300);
    digitalWrite(relay1,HIGH);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,200);
    digitalWrite(relay3,HIGH);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,150);
    digitalWrite(relay5, LOW);
delay(1000);
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,200);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,150);
    digitalWrite(relay5, LOW);
delay(1000);
    digitalWrite(relay1,HIGH);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,255);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,50);
    digitalWrite(relay5, LOW);
delay(3300);
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,200);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,150);
    digitalWrite(relay5, HIGH);
delay(1000);
}
else{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    digitalWrite(relay5, HIGH);

```

```

    }}
else if(Incoming_value== 'J' ) //Spiral
{
    if (isObstacle == HIGH)
    {
        digitalWrite(relay1,HIGH);
        digitalWrite(relay2,LOW);
        analogWrite(relay7,255);
        digitalWrite(relay3,HIGH);
        digitalWrite(relay4,LOW);
        for (int i = 0; i <= 20; i++)
        {
            analogWrite(relay8, i);
            digitalWrite(relay5, LOW);
            delay(50);
        }
        delay(10000);
        digitalWrite(relay1,LOW);
        digitalWrite(relay2,LOW);
        analogWrite(relay7,200);
        digitalWrite(relay3,LOW);
        digitalWrite(relay4,LOW);
        analogWrite(relay8,150);
        digitalWrite(relay5, HIGH);
        delay(1000);
    }
else{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    digitalWrite(relay5, HIGH);
    }}
    else if(Incoming_value== 'K' ) //Square
    {
        if (isObstacle == HIGH)
        {
            Serial.println("OBSTACLE!!, OBSTACLE!!");
            digitalWrite(relay1,HIGH);
            digitalWrite(relay2,LOW);
            analogWrite(relay7,255);
            digitalWrite(relay3,HIGH);

```

```
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1500);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1900);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1500);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1900);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1500);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
```

```

digitalWrite(relay5, LOW);
delay(1900);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1500);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1900);
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,200);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,150);
    digitalWrite(relay5, HIGH);
delay(1000);
}
else{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    digitalWrite(relay5, HIGH);
    }
    //REtAGNLE
}
else if(Incoming_value== 'L' )
{
    if (isObstacle == HIGH)
    {
        Serial.println("OBSTACLE!!, OBSTACLE!!");
        digitalWrite(relay1,HIGH);
    }
}

```



```
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1700);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,LOW);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(3700);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);;
delay(1700);
digitalWrite(relay1,LOW);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,255);
digitalWrite(relay5, LOW);
delay(3700);
digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
analogWrite(relay7,255);
digitalWrite(relay3,HIGH);
digitalWrite(relay4,LOW);
analogWrite(relay8,220);
digitalWrite(relay5, LOW);
delay(1700);

digitalWrite(relay1,HIGH);
digitalWrite(relay2,LOW);
```

```

    analogWrite(relay7,255);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,220);
    digitalWrite(relay5, LOW);
    delay(3700);
    digitalWrite(relay1,HIGH);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,255);
    digitalWrite(relay3,HIGH);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,220);
    digitalWrite(relay5, LOW);
    delay(1700);
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    analogWrite(relay7,255);
    digitalWrite(relay3,HIGH);
    digitalWrite(relay4,LOW);
    analogWrite(relay8,255);
    digitalWrite(relay5, LOW);
    delay(3700);
        digitalWrite(relay1,LOW);
        digitalWrite(relay2,LOW);
        analogWrite(relay7,200);
        digitalWrite(relay3,LOW);
        digitalWrite(relay4,LOW);
        analogWrite(relay8,150);
        digitalWrite(relay5, HIGH);
    delay(1000);
}
else{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
    digitalWrite(relay5, HIGH);
    }}
else if(Incoming_value== 'M' )
{
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);

```

```
    digitalWrite(relay3,LOW);  
    digitalWrite(relay4,LOW);  
        digitalWrite(relay5,HIGH);  
}
```

```
else if (isObstacle == LOW)  
{  
    digitalWrite(relay1,LOW);  
    digitalWrite(relay2,LOW);  
    digitalWrite(relay3,LOW);  
    digitalWrite(relay4,LOW);  
    digitalWrite(relay5,HIGH);  
}  
}
```