Abstract

The main objective behind this paper is to develop a robot to perform the act of surveillance in domestic areas. Nowadays robot plays a vital role in our day-to-day life activities thus reducing human labor and human error. Robots can be manually controlled or can be automatic based on the requirement. The purpose of this robot is to roam around and provide audio and video information from the given environment and to send that obtained information to the user.

In this project, one can control the robot with the help of mobile or laptop through Internet of Things (IoT) and also can get the live streaming of video both in daytime as well as at night with the help of wireless camera from the robot. The robot can be controlled both in manual as well as in automated mode with the help of ESP32. This robot also uses various sensors that collects data and sends it to the ESP32 which controls the robot behavior. Along with the obtained live streamed video output, user can also obtain the presence of metal bombs using metal detectors. Thus, the action of surveillance can be performed. Further advancement in our project can provide surveillance even in defense areas

CONTENTS

Chapter No.	Title	Page No.
1	Introduction	1
2	System Description	2
	2.1 Block Diagram	2
3	Hardware & Software Tools	3
4	Implementation	6
5	Results	9
6	Conclusion	10
	References	11

1. INTRODUCTION

Technology has brought a dynamic and tremendous change in robotics and automation field which ranges in all kinds of areas. Surveillance is the process of close systematic observation or supervision maintained over a person, group, etc. especially one in custody or under suspicion. Thus, surveillance is mainly required in the areas such as border areas, public places, offices and in industries. It is mainly used for monitoring activities.

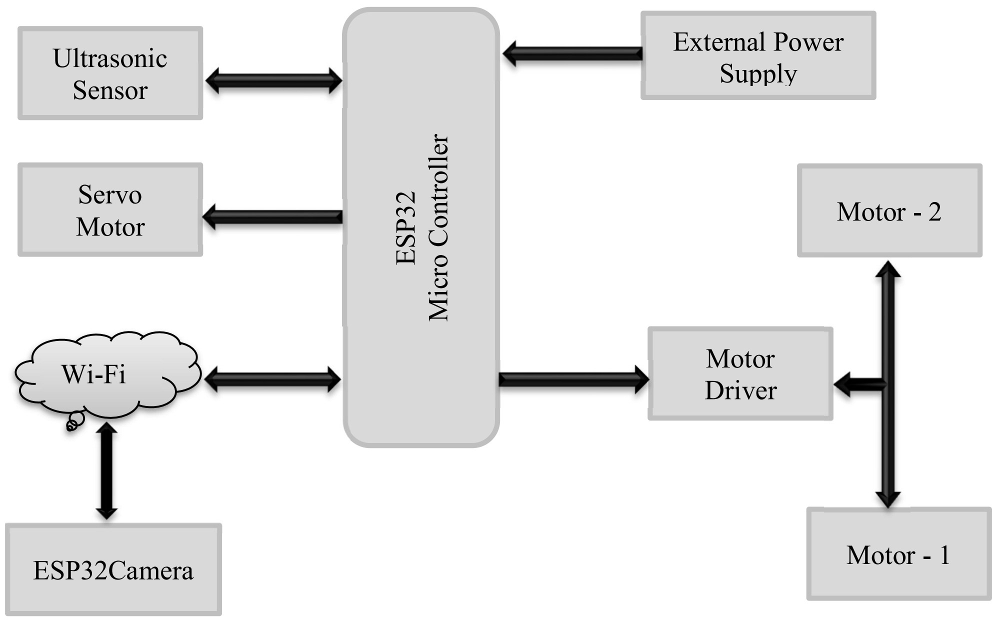
The act of surveillance can be performed both indoor as well as in outdoor areas by humans or with the help of embedded systems such as robots and other automation devices. A robot is nothing but an automatic electronic machine that is capable of performing programmed activities thus replacing human work, providing highly accurate results and easily overcoming the limitations of human beings. Thus, replacing humans in the surveillance fields is one of the great advancements in robotics.

A major disadvantage of these systems is that as soon as the objects are obscured or blinds pots, you cannot get your image information. Also, monitoring the entire area for an environment is not feasible. To increase the general approach to address this problem, the number of multi-view cameras monitoring are required. However, the cost of both hardware and system development increases.

2. SYSTEM DESCRIPTION

This project is based on a micro-controller and IOT concepts. Where we are using ESP32, Servo motors, car chassis, camera module, sensors and Motors to build this surveillance robot setup. The camera which we used to stream the video over Wi-Fi uses the internet and provide us a live feed and the amazing part is here that we can control the whole setup from the Web Server. So, for this we build an html page in which we can have the buttons that can controls like forward, backward, turn left, turn right. The Arduino uses C language for coding. The data capture through camera module, send to the desired device using internet. By using this data, the user gives further commands to the robot.

2.1 BLOCK DIAGRAM



3. Hardware & Software Tools

3.1 SOFTWARE TOOLS:

1. Arduino IDE

2. Webpage, to control the Robot and to Monitor the Video Surveillance

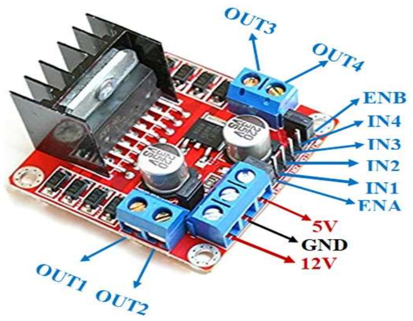
3.2 HARDWARE COMPONENTS:

1. DC Motors – 2

   Motors that operate on 12V DC power supply are used. These are rotary electrical machine that converts direct current electrical energy into mechanical energy. The motors used are of 30 rpm speed of operation.

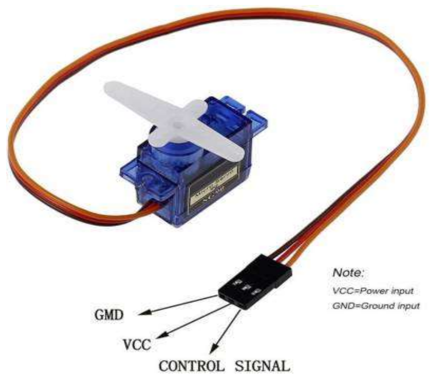
2. L298D Motor Driver

   L298D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal to run four solenoids, two DC motors or one bi-polar or unipolar stepper with up to 600 mA per channel using the L298D. These are known as the drivers in the Ada fruit Motor shield.



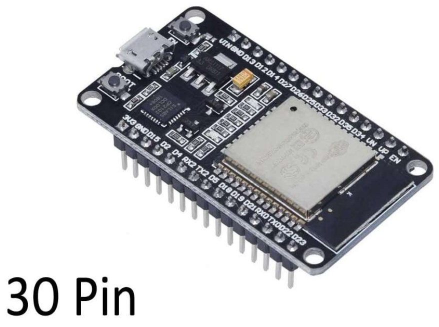
3. Servo Motor

   A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply, then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.



4. ESP32

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.

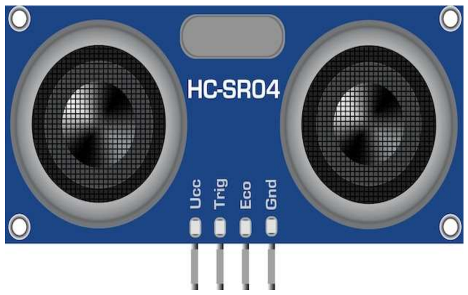


5. Jumper Wires

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

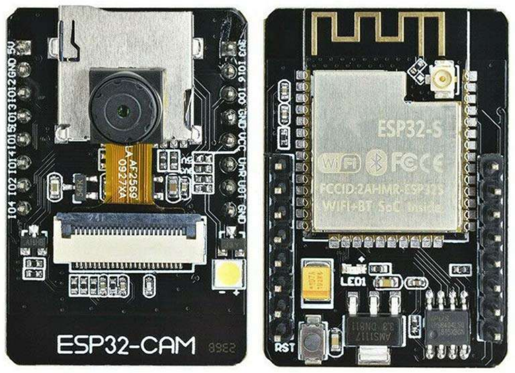
6. Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).



7. Esp32 Camera

The ESP32 CAM WiFi Module Bluetooth with OV2640 Camera Module 2MP For Face Recognization has a very competitive small-size camera module that can operate independently as a minimum system with a footprint of only 40 x 27 mm; a deep sleep current of up to 6mA and is widely used in various IoT applications.



It is suitable for home smart devices, industrial wireless control, wireless monitoring, and other IoT applications. This module adopts a DIP package and can be directly inserted into the backplane to realize rapid production of products, providing customers with high-reliability connection mode, which is convenient for application in various IoT hardware terminals.

8. Chassis Kit for Robot Model

It is a Model in which we build or place the all components to work.

9. Batteries

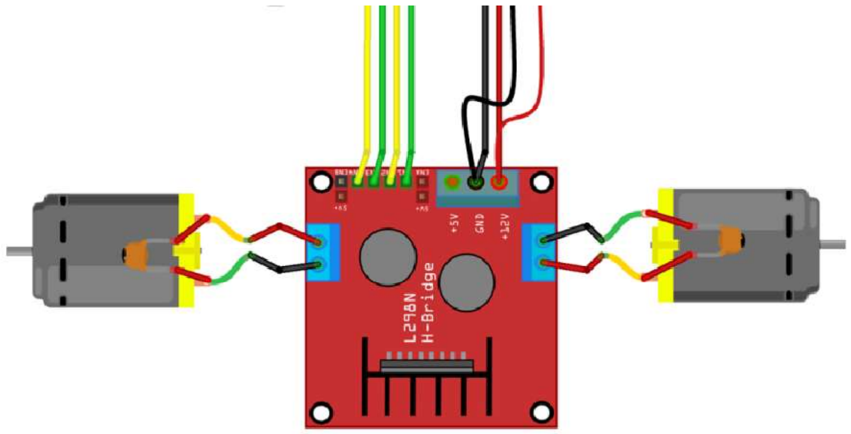
Two 6V batteries are connected in series to provide a 12V power supply for the motors. From these batteries power supply is also given to the ESP32 and other parts that require power supply for their effective performance.

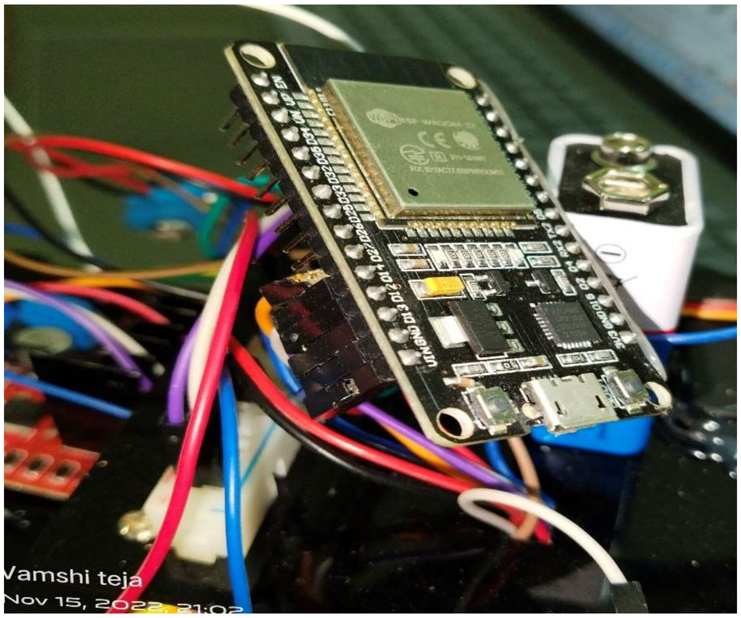
4. IMPLEMENTATION

Step 1:

Connecting the L298D Motor Driver to the DC 2 Motors

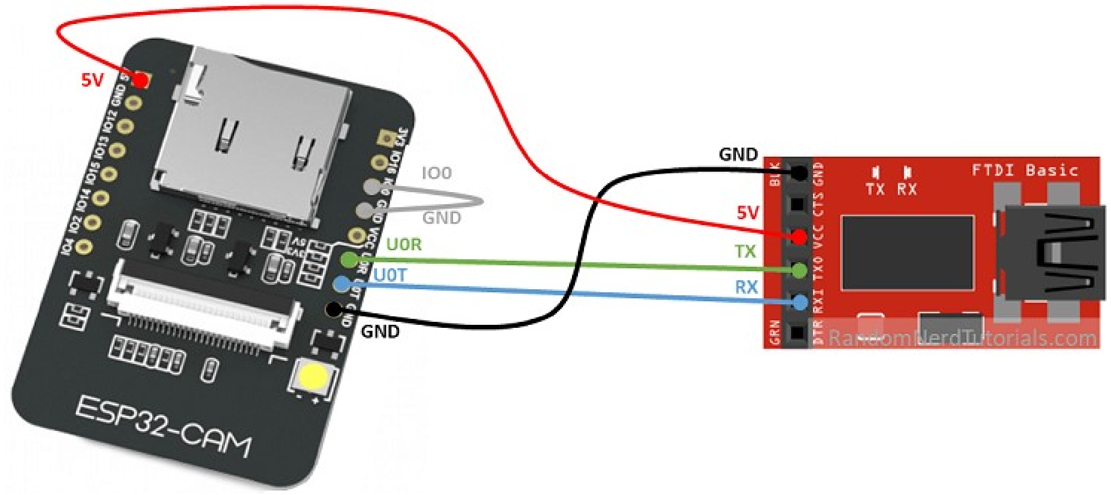
- Connect the 2 wires of the of 1st motor to the OUT1 and OUT2 of the L298D
- Connect the 2 Wires of the 2nd Motor to the OUT3 and OUT4 of the L298D
- Connect 12V and the GND to the External Battery
- Connect the IN1, IN2, IN3, IN4 to the ESP32





Step-2:

Connecting the ESP32 Camera to the model:



Why we Didn't Connected the ESP32 Camera module directly to the laptop?

Esp32 Camera AI-Thinker Module doesn't have any type of pins as like Arduino, RaspberryPi or Esp32.so to Interact or to communicate with ESP32 Module we need to have FTDI adapter to dump the code into the Esp32 Camera module.

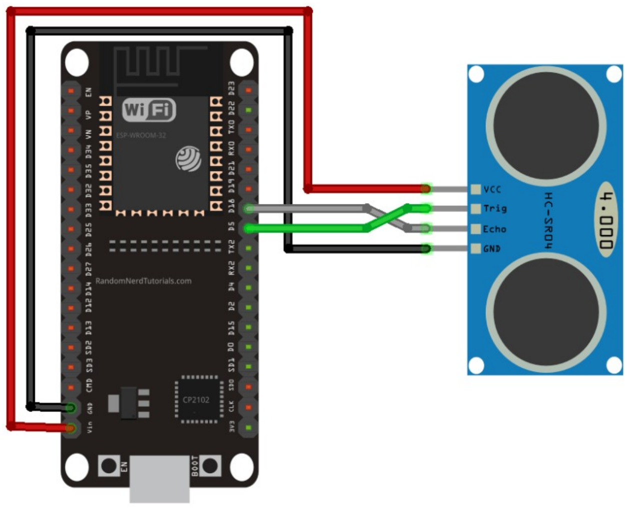
- After Dumping the Code remove the pin that is connected to the ground and IO0 wire
- Now press the RESET button on the Cam. Module
- An IP Address will be Generated copy the link and paste in the Google Chrome and you can see the live camera vision of the camera from the module.

Step-3:

Connecting the Ultrasonic sensor to the Robot:

Connecting the trigger and the echo pins to the appropriate pins to the ESp32 board.

Hence, we are using this HC-SR04 Sensor to the detect for any obstacle for the backwards of the model. As we can see from the front side of the robot car,so that its difficult whenever a obstacle found while reversing it.Obstacle may leads to damage of the model as depending on the speed of the motors.

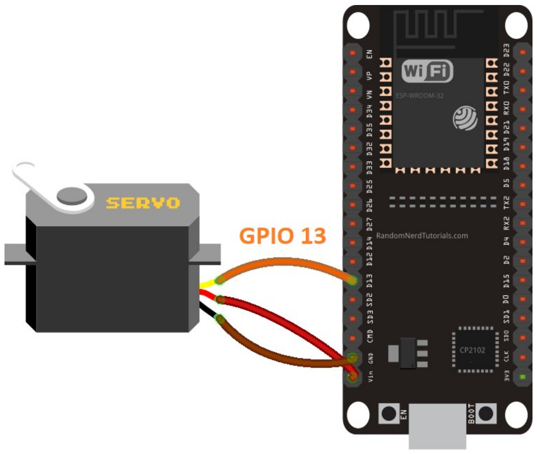


Step-4:

Connecting the Servo Motor to the ESP32:

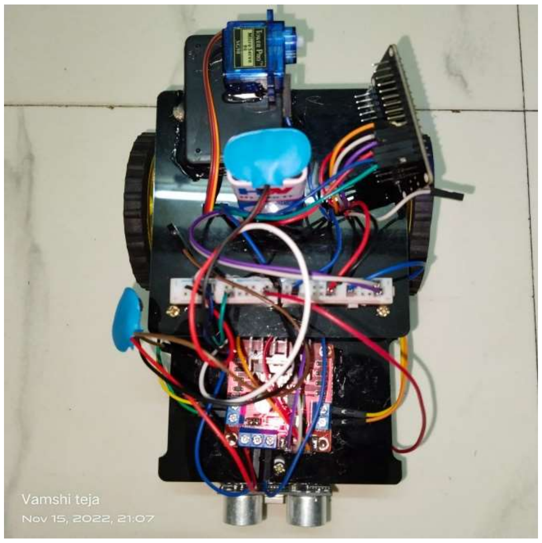
Connecting the Servo motors with Appropriate VCC, Ground and the Signal pins to the esp32.

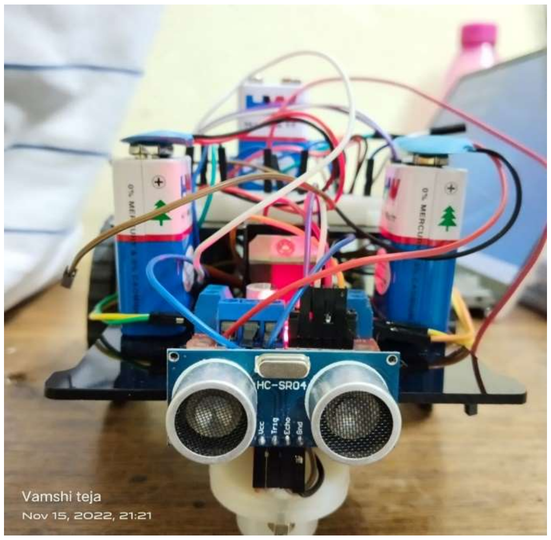
We are Using this Servo motor to adjust the camera position of top to bottom or straight according the need of the Controller which can make the flexible vision while monitoring the robot.

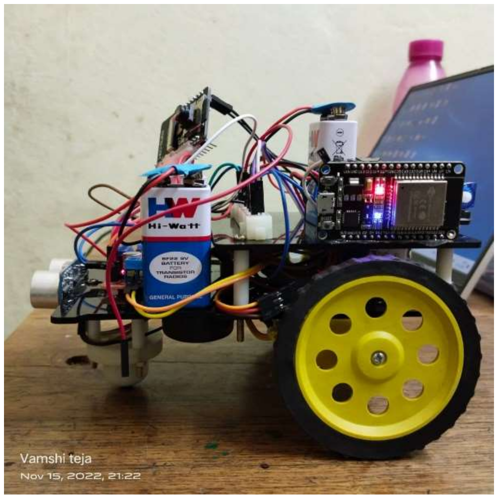


Step-5:

Combining all the Above Steps to make the model runnable. After Fixing all the Connections to the respective positions the Model looks as like the below Image

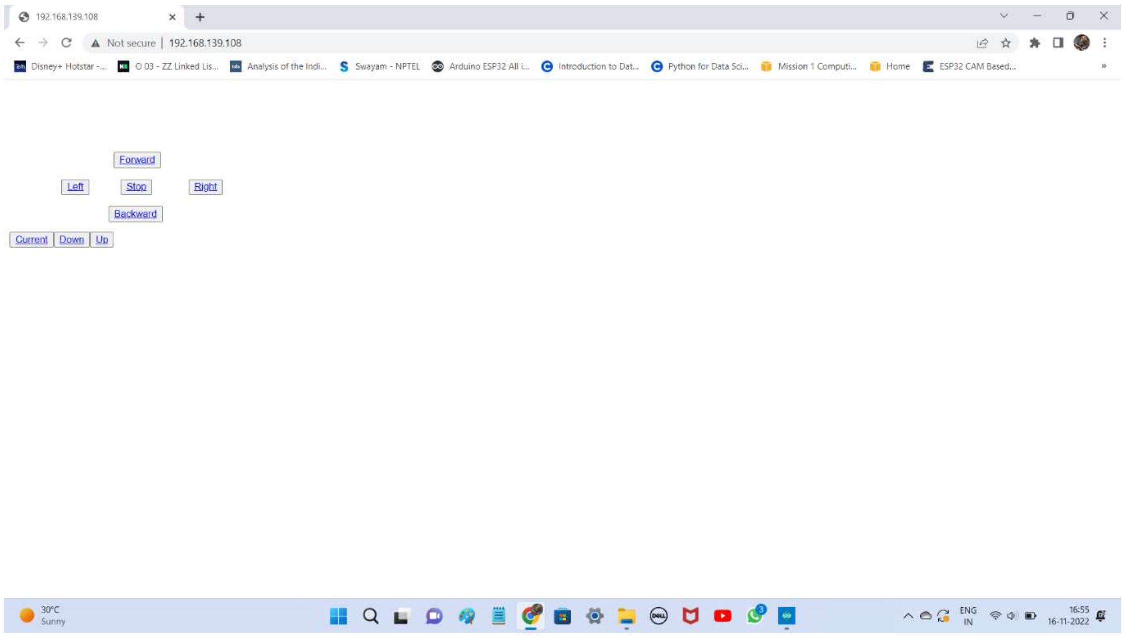






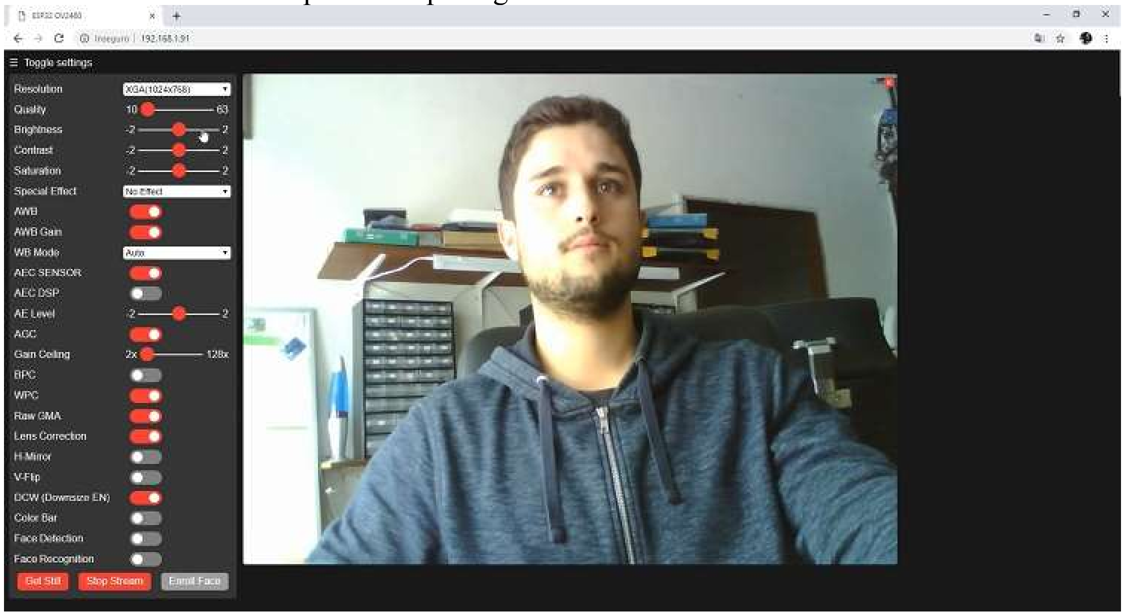
Step – 6:

Controlling the Camera from the web Server that is provide within the server.



Step-7:

ESP camera Module Output after opening the IP Address:



Resolution XGA(1024x768)
Quality 10 — 63
Brightness -2 — 2
Contrast -2 — 2
Saturation -2 — 2
Special Effect No Effect
AWB
AWB Gain
WB Mode Auto
AEC SENSOR
AEC DSP
AE Level -2 — 2
AGC
Gain Ceiling 2x — 128x
BPC
WPC
Raw GMA
Lens Correction
H-Mirror
V-Flip
DCW (Downsize EN)
Color Bar
Face Detection
Face Recognition

5. RESULT

Following are the main applications of the solar powered multifunctional robot:

- By combining camera features with the robot, we can easily monitor indoor as well as outdoor locations during daytime and at night.
- Remote areas can also be explored.
- Used to record and send video output of the required environment.

We have two results i.e., the hardware and the software result. The hardware includes the robot which runs on DC motors. The input to the motors is provided by the L293N motor driver shield. The input to the driver shield is provided by the ESP32. The navigational inputs are given by the user to the ESP32 using the Web Server via Wi-Fi. ESP32, on receiving the signal, processes it and produces the appropriate output. The communication between the Web Server and the Esp32 takes place using the Wi-Fi module which is interfaced with the Esp32 board. It provides serial communication between the Server and the model.

6. CONCLUSION

The framework for making a robot for surveillance purpose is proposed. It overcomes the problem of limited range surveillance by using the concept of IOT. We can control the robot with the help of laptop/mobile manually. Automatic monitoring can also be done. Our proposed robot is small in size thus maneuvering into area where human access is impossible. Wireless technology is one of the most integral technologies in the electronics field. This technology is used to serve our project as a supreme part of surveillance act. This provides highly efficient and a cost-effective robot that replaces human work and reduces human labor and performing monitoring works in a well effective manner

FUTURE SCOPE

The project future scope has numerous openings that could be prosecuted for various future applications for monitoring and control, etc. This robot can also be used in times of environmental catastrophes where the robot detects whether a living human being is present in that region. In domestic applications such as home security can also be implemented using this method.

- Can be Used in the Military to detect the Bombs or to know the Motion detection at the Border
- Can be used to detect for the leakages of gases by adapting with a gas sensor.

REFERENCES

1. Hou-Tsan Lee, Wei-Chuan Lin, Ching-Hsiang Huang, Yu-Jhih Huang, “Wireless indoor surveillance robot”, in Proceedings of SICE Annual Conference (SICE), 2011, p. 2164-2169.
2. Change Zheng, “Mechanical design and control system of a miniature surveillance robot”, in ICIA '09, International Conference on Information and Automation, 2009, p. 1228- 1233.
3. Kyunghoon Kim, “Intelligent surveillance and security robot systems”, in IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2010, p. 70- 73.
4. Chinmay Kulkarni, Suhas Grama, Pramod Gubbi Suresh.,” Surveillance Robot Using Arduino Microcontroller, Android APIs and the Internet”, IEEE First International Conference on Systems Informatics, 2014.
5. Ho Park, “Development of a mobile surveillance robot”, in ICCAS '07, International Conference on Control, Automation and Systems, 2007, p. 2503- 2508.
6. M.Selvam.,“Smart Phone Based Robotic Control For Surveillance Applications”, International Journal of Research in Engineering and Technology, Volume 03 ,Issue 03, Mar-2014.
7. Kota Sandeep, Kakumanu Srinath, Rammohanarao Koduri.,” Surveillance Security Robot With Patrolling Vehicle”, Volume-2, Issue3, May-Jun 2012, pp.546 – 549.
8. Shoeb Maroof Shaikh,Khan Sufiyan, Asgar Ali, Mir Ibrahim, Prof. Kalpana Bodke., “Wireless Video Surveillance Robot Controlled Using Android Mobile Device”, International Journal of Advance Foundation And Research In Science & Engineering (IJAFRSE) Volume 1,Vivruti 2015.G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references)
9. L293D Datasheet- Texas Instruments – Quadruple Half- H Driver [Online] Available: http://www.alldatasheet.com/datasheetpdf/ pdf/27189/TI/L293D.html, October 2012.
10. Manufacturer ATMEL, ATMEGA 328P Datasheet

Appendix

Code:

```
const int MR1 = 12;
const int MR2 = 14;
const int ML1 = 27;
const int ML2 = 26;
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distanceCm;
#include <Servo.h>
#include <WiFi.h>

const char* ssid     = "Redmi";
const char* password = "Karthik123";
Servo myservo;
WiFiServer server(80);

void setup()
{
    Serial.begin(115200);
    pinMode(5, OUTPUT);
    pinMode(MR1, OUTPUT);
  pinMode(MR2, OUTPUT);
  myservo.attach(13);
  pinMode(ML1, OUTPUT);
  pinMode(ML2, OUTPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);
    // set the LED pin mode

    delay(10);

    // We start by connecting to a WiFi network

    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    server.begin();

}

int value = 0;

void loop(){
 WiFiClient client = server.available();   // listen for incoming clients

  if (client) {                             // if you get a client,
    Serial.println("New Client.");           // print a message out the serial port
    String currentLine = "";                // make a String to hold incoming data from the client
    while (client.connected()) {            // loop while the client's connected
      if (client.available()) {             // if there's bytes to read from the client,
        char c = client.read();             // read a byte, then
        Serial.write(c);                    // print it out the serial monitor
        if (c == '\n') {                    // if the byte is a newline character

          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:
          if (currentLine.length() == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming, then a blank line:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();

            // the content of the HTTP response follows the header:
            client.print("<pre><br><br><br><br><br><br>"
"                  <button><a href=\"/F\">Forward</a></button><br><br>"
"         <button><a href=\"/L\">Left</a></button>      <button><a href=\"/S\">Stop</a></button>
<button><a href=\"/R\">Right</a></button><br><br>"
"                 <button><a href=\"/B\">Backward</a></button><br></pre>");
            client.print("<button><a href=\"/C\">Current</a></button>");
            client.print("<button><a href=\"/D\">Down</a></button>");
            client.print("<button><a href=\"/E\">Up</a></button>");
            // The HTTP response ends with another blank line:
            client.println();
            // break out of the while loop:
            break;
          } else
          {    // if you got a newline, then clear currentLine:
            currentLine = "";
          }
        } else if (c != '\r')
        {  // if you got anything else but a carriage return character,
          currentLine += c;      // add it to the end of the currentLine
        }

        if (currentLine.endsWith("GET /R")) {
           digitalWrite(MR1,LOW);
      digitalWrite(MR2,HIGH);
      digitalWrite(ML1,LOW);
      digitalWrite(ML2,HIGH);               // GET /H turns the LED on
        }
        if (currentLine.endsWith("GET /L")) {
          digitalWrite(MR1,HIGH);
      digitalWrite(MR2,LOW);
      digitalWrite(ML1,HIGH);
      digitalWrite(ML2,LOW);              // GET /L turns the LED off
        }
        if (currentLine.endsWith("GET /F")) {
           digitalWrite(MR1,HIGH);//MOVE FRONT
      digitalWrite(MR2,LOW); //MOVE BACK
      //LEFT MOTOR
      digitalWrite(ML1,LOW);//MOVE BACK
      digitalWrite(ML2,HIGH);//MOVE FRONT
      }
      if (currentLine.endsWith("GET /B")) {
          digitalWrite(MR1,LOW);
      digitalWrite(MR2,HIGH);
      digitalWrite(ML1,HIGH);
      digitalWrite(ML2,LOW);
      }
      if (currentLine.endsWith("GET /S"))
      {
          digitalWrite(MR1,LOW);
          digitalWrite(MR2,LOW);
          digitalWrite(ML1,LOW);
          digitalWrite(ML2,LOW);
      }
      //current=135
      //down=180
      //up=90
      //too up=0
      if (currentLine.endsWith("GET /C"))
      {
           myservo.write(135);
      }
      if (currentLine.endsWith("GET /D"))
      {
           myservo.write(180);
      }
      if (currentLine.endsWith("GET /E"))
      {
```

```
         myservo.write(90);
         digitalWrite(35,HIGH);
    }
    }
   }
   // close the connection:
   client.stop();
   Serial.println("Client Disconnected.");

   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   duration = pulseIn(echoPin, HIGH);
   distanceCm = duration * SOUND_SPEED/2;
   Serial.print("Distance (cm): ");
   Serial.println(distanceCm);
   if(distanceCm>=10)
   {
        digitalWrite(MR1,LOW);
        digitalWrite(MR2,LOW);
        digitalWrite(ML1,LOW);
        digitalWrite(ML2,LOW);
   }
   }
  }
```