SPY ROBOT

A

Project Exhibition -II

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology

In

ELECTRONIC AND COMMUNICATION ENGINEERING

Specialization in

Artificial intelligence and cybernetics

Submitted to

VIT BHOPAL UNIVERSITY (M.P.)

Submitted by

KRITIKA TRIPATHI(21BAC10032)

RAKESH LODHI(21BAC10034)

ASHISH BARPETE(21BAC10037)

Under the Supervision of

Dr.ABHISHEK JOSHI
VIT BHOPAL UNIVERSITY

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGG

BHOPAL (M.P.)-466114

Feb – 2023

VIT BHOPAL UNIVERSITY BHOPAL (M.P.) 466114

SCHOOL OF ELECTRICAL & ELECTRONICS ENGG.

CANDIDATE'S DECLARATION

I hereby declare that the Dissertation entitled "**SPY ROBOT** " is my own work conducted under the supervision of DR.ABHISHEK JOSHI, Assistant professor, SEEE at VIT University, Bhopal.

I further declare that to the best of my knowledge this report does not contain any part of work that has been submitted for the award of any degree either in this university or in other university / Deemed University without proper citation.

Kritka Tripathi (21BAC10032)

Rakesh Lodhi(21BAC10034)

Ashish Barpete(21BAC10037)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 17 Feb 2023

Abhishek Joshi

Assistant Professor,SEEE

Digital Signature of Guide

VIT UNIVERSITY BHOPAL (M.P.) – 466114

SCHOOL OF ELECTRICAL & ELECTRONICS ENGG.

CERTIFICATE

This is to certify that the work embodied in this Project Exhibition -1 report entitled **"SPY ROBOT"** has been satisfactorily completed by **Ms. KRITIKA TRIPATHI** (21BAC10032),**MR. RAKESH LODHI** (21BAC10034) and **MR.ASHISH BARPETE**(21BAC10037) in the School of Electrical & Electronics Engineering at VIT University, Bhopal. This work is a bonafide piece of work, carried out under my/our guidance in the School of Electrical & Electronics Engineering for the partial fulfilment of the degree of Bachelor of Technology.

PROJECT GUIDE

Dr. Abhishek Joshi,

Assistant Professor ,

School Of electrical and electronic engineering ,

VIT BHOPAL UNIVERSITY

Forwareded by

Dr.Soumitra K. Nayak,
Program Chair,
School Of electrical and electronic engineering ,

VIT BHOPAL UNIVERSITY

Approved by

Dr. Debashis Adhikari,

Dean,

School Of electrical and electronic engineering ,

VIT BHOPAL UNIVERSITY

Acknowledgement

In the first place we would like to record our gratitude to **Dr. Abhishek Joshi** for his supervision, advice, and guidance from the very early stage of this thesis work as well as giving us extraordinary experiences throughout the work. Above all and the most needed, he provided us unflinching encouragement and support in various ways. His words have always inspired us to work in an efficient and comprehensive way. We would like to thank him for his constant encouragement that enabled us to grow as a person. His presence has definitely improved us as a human being.

We express our gratitude towards "**Dr. Debashis Adhikari,",** Professor and Dean of SEEE, VIT Bhopal University, for providing us all the help and permitted us to work in the laboratory with no time limits.

We would also like to thank **Dr. Soumitra K. Nayak,** Assistant professor and Programme chair of SEEE, VIT Bhopal University, for giving us his valuable time and guidance.

We shall ever remain indebted to "**Dr. U Kamachi Mudali",** Vice Chancellor, VIT Bhopal University, for providing us institutional and administrative facilities during our project work at VIT.

Kritika Tripathi	21BAC10032
Rakesh Lodhi	21BAC10034
Ashish Barpete	21BAC10037

Executive Summary

As an essential constituent of many associations' security and safety precedence, surveillance has established its importance and benefits numerous times by providing immediate supervising of possessions, people, environment and property. This project deals with the design approach of an Embedded Real-Time security System for intruder observation that reinforces surveillance technology to provide essential security to our life and associated control. The proposed robotic unit is used for video surveillance of remote place as well as remotely control of the unit using Bluetooth as medium. Arduino serve the purpose of server as well as the microprocessor for the system. An embedded web server creates an easy way for monitoring & controlling any device which is at remote place. The proposed security solution hinges on our novel integration of cameras into web application. ESP32 Cam operates and controls camera for remote sensing and surveillance, streams live video. This research is focused on developing a surveillance system that helps the property owners to monitor the place to avoid intruders by using ESP32 Camera and WiFi technology for remote control. This Smart Surveillance System presents the idea of monitoring a particular place in a remote area. The proposed solution offers a cost-effective ubiquitous surveillance solution, efficient and easy to implement. The proposed system can be used in military applications just by adding few sensors like infrared sensors so as we can detect the movements. In health care applications the proposed system can be used just by changing the design of the robotic unit.

List of Figures

List of Symbols & Abbreviations

SR	Spy Robot

Table of contents

Page No.

Chapter 1

1. Introduction

Due to increased need of security most especially in the homes and places of work, there has been a demand for the security systems which are able to protect the man and property which has led to design of spy robot systems with the spy camera and obstacle avoidance technology. A robot is a machine designed to execute one or more tasks repeatedly, with speed and precision. A robot is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a computer program or electronic circuitry. Robots can be autonomous or semi-autonomous, they have replaced human in performing repetitive and dangerous tasks which humans prefer not to do, or are unable to do because of size limitations, or which take place in extreme environments such as outer space or the bottom of the sea [1]. A spy camera is a device that is capable of capturing video (and sometimes audio) of a location without the subjects' knowledge. These cameras are mainly used for surveillance activities. Surveillance is the process of monitoring a situation, an area or a person.Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always possible. There are also added risks of losing personnel in the event of getting caught by the enemy [2]. Patrol systems have recently achieved interest to address the concerns about national security. The major problem in protecting long stretches of borders is the need for large human involvement in patrolling the premises. Under the conventional patrol system, even modest-sized areas require large human resources if manual patrolling is considered alone [3]. To monitor the premises in real-time with accuracy and minimize the need for human support, multiple surveillance technologies, which complement each other are required. To address the challenges still facing by the existing surveillance techniques, we introduce the spy robot, a new spying system framework based on ESP 32 camera which can continuously monitor the barracks boundaries to prevent intrusion with minimum human involvements.

1.1 Motivation of the study

The desire to know and explore new robotic or uncover electronic projects drive us to create spy robots for reconnaissance and espionage. The motivation The desire for novelty and innovation is motivate to create spy robots for the challenge and excitement of developing new technology.related to electronic and robotics .

1.2 Objective of the work

1.2.1 security purposes to monitor a specific area and detect any potential threats or intruders.

1.2.2 surveillance purposes to gather intelligence on a target, such as a person or a location.

1.2.3 exploring hard-to-reach or hazardous areas.

1.2.4 Improved system control

Chapter 2

2. Literature review

[1] Tahzib Mashrik, Hasib Zunair, Maofic Farhan Karin [2017] proposed the Designed and Implemented Security Patrol Robot using Android Application and using it as a tool for study and research. In his paper they discussed the security robot which can be used in household or office

purposes. . Findings of the research paper is Designing of autonomous robot car of dc motor,Software application for mobile interface and Wireless communication system.

But in this paper Limitation is not able to capture picture and live streaming.

.

[2] Marapalli, Krishnakumar (202221) has discussed about An Android remote control car unit for search missions that controlled using hand gestures It will also detect the objects using ovenCV and python. Finding of this project is Python/OpenCV program and also learn application of ESP32 CAM module .

Chapter 3

3.Problem formulation and proposed methodology

- Currently, commercialized robots are mainly for safe task that is not included in any destruction crisis and may not bring harm to the robot which can be considered as not designed for being destroyed. We may need to think of the dangerous task is always bringing destruction and the robot may only use for once. However, currently the production cost of a robot that is designed purposely for dangerous task is very expensive, thus, it is not appropriate to be considering as using for one time only. Besides that, a robot for dangerous must be designed with time sensitive manner, which means that it should be more timely precise in doing a task and it should be task specific. As a conclusion, there is very high demand in producing a robot that is low production cost and perform a dangerous task in time sensitive manner. Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always possible. There are also added risks of losing personnel in the event of getting caught by the enemy. There is also a need for one to keep an eye on the people coming inside the premises of the barracks. Thus all this will lead to the design and implementation of spy robot system which will help to secure lives of personnel in barracks.

3.2 Proposed methodology

Firstly, we program ESP-32 CAM module using ARDUINO UNO & ARDUINO IDE. Next, we generate the IP address of the ESP 32 CAM through WiFi on the Arduino IDE serial monitor.

Further, we connect the ESP 32 cam module to L298N motor driver as per the circuit diagram. Next we connect the 4 BO motors to the the L298N motor driver. We also connect a 12 -20V DC power supply to power the complete setup.

After this, we power the robot through the DC power supply. Once the robot response, we now take the IP address generated on the Arduino IDE serial monitor and paste it on to the browser.

- With this we reach to a webpage where we see the live streaming provided by ESP32 CAM module and we also have LEFT, RIGHT, UP, DOWN controls to control our robot.

● 3.2.1 Components

1.Arduino :A microcontroller board contains on-board power supply, USB port to communicate with PC, and an Atmel microcontroller chip. In 2005, building upon the work of Hernando Barragán (creator of Wiring), Massimo Banzi and David Cuartielles created Arduino, an easy-to-use programmable device for interactive art design projects, at the Interaction Design Institute Ivrea in Ivrea, Italy. It is an open source hardware, anyone can get the details of its design and modify it or make his own one himself. Basically there are different types of Arduino boards are available. In this project we used Ardunio uno.

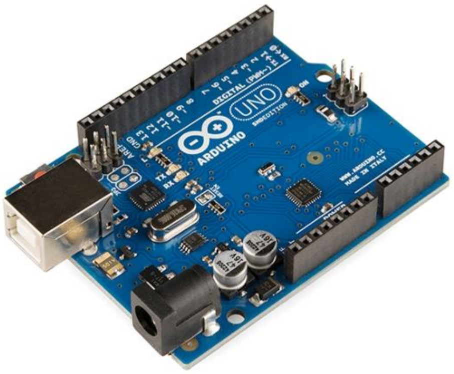


Fig 3.2.1.1Arduino

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version

2. **The Arduino IDE** :supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

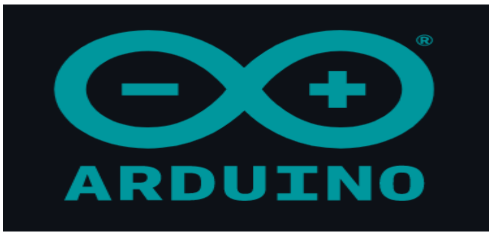


Fig 3.2.1.2 Arduino IDE

3.ESP 32 CAM MODULE : ESP32-CAM is a low-cost ESP32-based development board with onboard camera, small in size. It is an ideal solution for IoT application, prototypes constructions and DIY projects. The board integrates WiFi, traditional Bluetooth and low power BLE , with 2 highperformance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz. Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications.

●

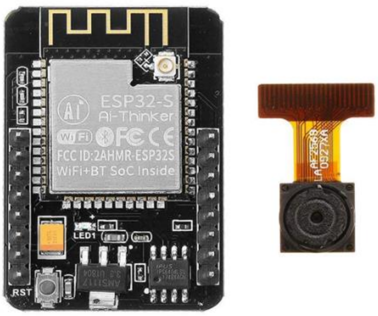


Fig 3.2.1.3 ESP32 Cam module

4.L298N MOTOR DRIVER.: The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit. 78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

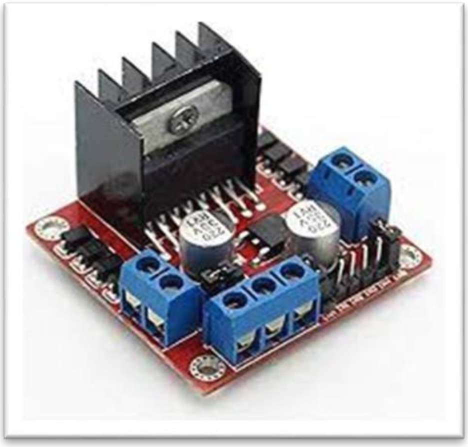


Fig 3.2.1.4 L298N Motor driver

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and

Motor .

5.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).

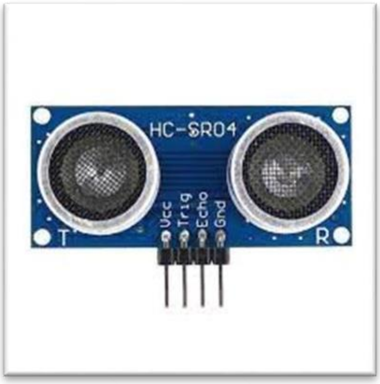


fig.3.2.1 ultrsonic sensor

6.Lithium iron battery:

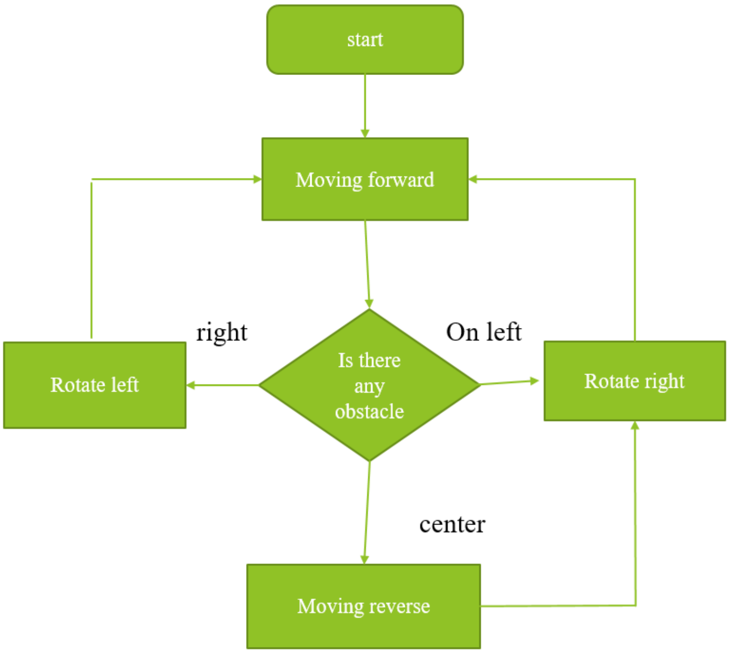
A lithium iron battery, also known as a lithium iron phosphate (LiFePO4) battery, is a type of rechargeable battery that uses lithium ions to store and release energy. It is a relatively new type of battery chemistry that has gained popularity in recent years due to its high energy density, long cycle life, and improved safety compared to other lithium-based batteries.Lithium iron batteries have several advantages over other types of batteries. They are more durable and have a longer lifespan, with a typical cycle life of up to 2000 charge and discharge cycles. They are also less prone to overheating and fire hazards, which is a significant advantage in applications such as electric vehicles and power storage systems.

Fig 3.2.2 lithium iron battery

3.3 Implementation

For better understanding of compont we make some basic robotic project obtical avoidance and line following robot .

Flow chart of obstacle avoiding robot



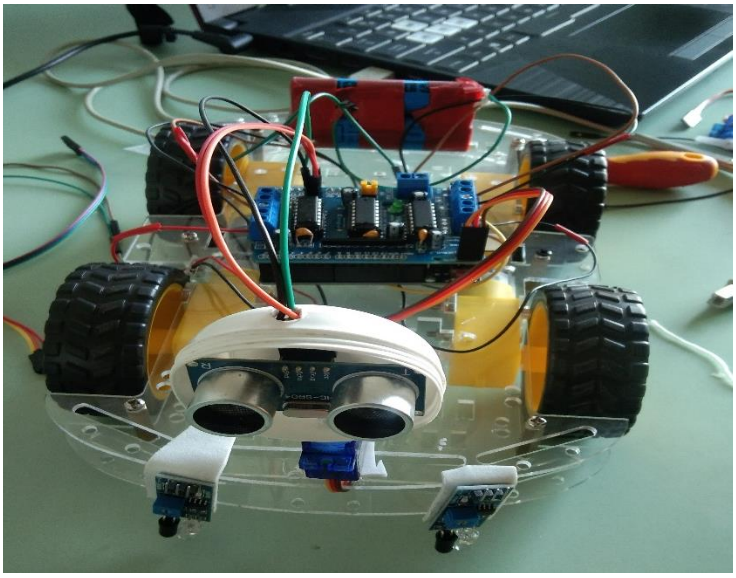
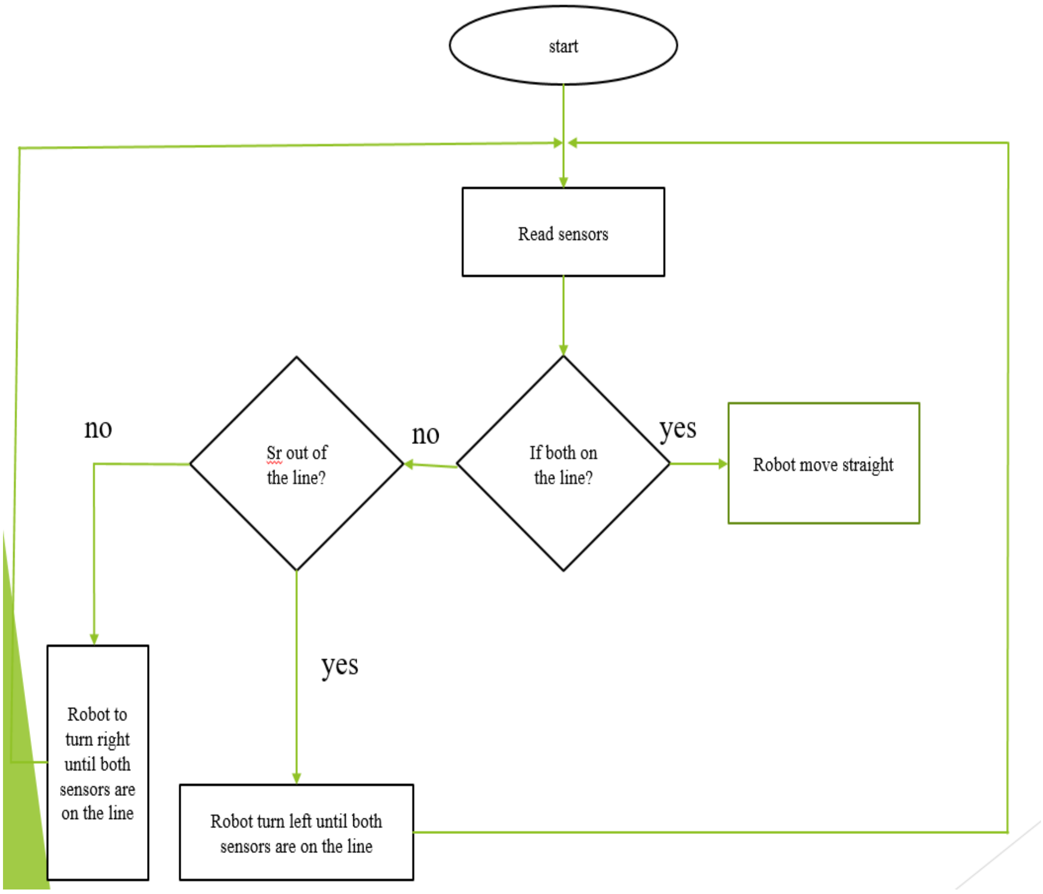


Fig 3.3.1 obstacle avoiding robot

Flowchart of line following robot


start
Read sensors
If both on the line?
yes
Robot move straight
no
Sr out of the line?
no
yes
Robot to turn right until both sensors are on the line
Robot turn left until both sensors are on the line

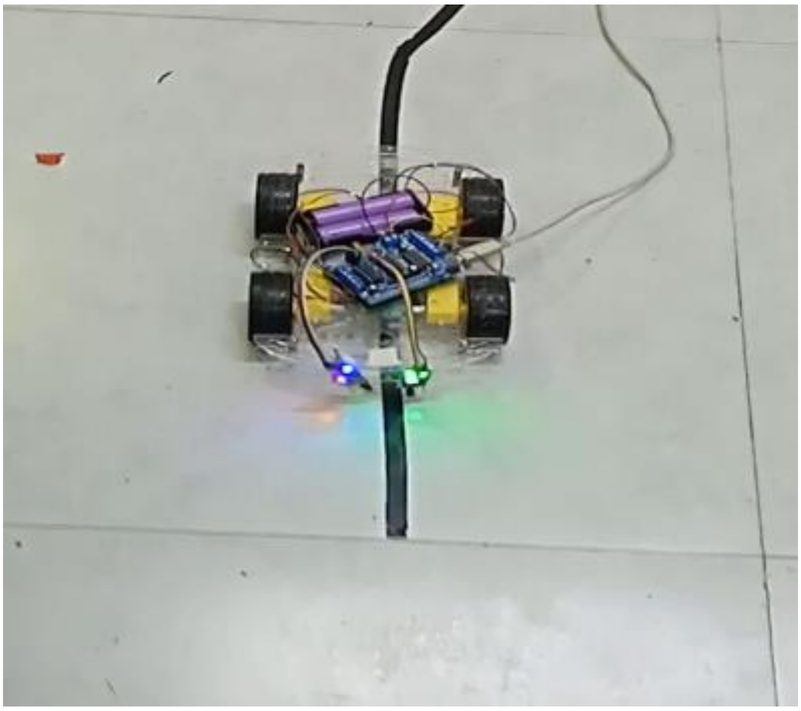


Fig 3.3.2 line following robot

- **Initialize ESP-32 module**

Overview: We have used one LDR circuit to know the day and night . LDR with a small register (220ohm) in series is connected across the 5V and GND of the Arduino Uno and from the midpoint of the LDR potential divider circuit the output of the circuit is feed to A0 of the Arduino which turn on all the street lights which are represented by Led connected to the output PWM pin (3 ,5 ,6 ,9 ,).

Four infrared receiver and sender circuits are made to detect the movements and output from the receiver is fed to the input terminal (2,4,,7,8)which corresponds to the led connected to 3 ,5 ,6 ,9 , respectively. All the object sensors are connected between 5V and GND of the Arduino UNO.

Working of the circuit:

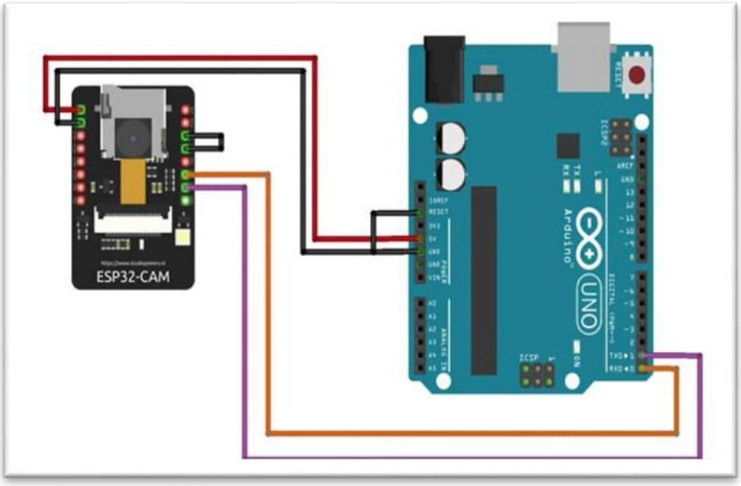


Fig.3.3.3 interfacing of esp32 to arduino uno

To upload code to the ESP32-CAM module using the Arduino IDE, we have followed these steps:

Install the ESP32 board support in the Arduino IDE:

Open the Arduino IDE and click on "File" -> "Preferences".

In the "Additional Boards Manager URLs" field, enter this URL:

https://dl.espressif.com/dl/package_esp32_index.json

Click "OK" to close the preferences window.

Go to "Tools" -> "Board" -> "Boards Manager".

Search for "esp32" and install "ESP32 by Espressif Systems".

Connect your ESP32-CAM module to your computer:

Insert your ESP32-CAM module into a breadboard or a PCB.

Connect the USB-to-UART converter to the ESP32-CAM module. The wiring should be as follows:

esp32-cam 5v -> usb-to-uart vcc

esp32-cam gnd -> usb-to-uart gnd

esp32-cam u0t -> usb-to-uart rx

esp32-cam u0r -> usb-to-uart tx

Select the board and port:

Go to "Tools" -> "Board" and select "ESP32 Wrover Module" as the board.

Go to "Tools" -> "Port" and select the port that corresponds to the USB-to-UART converter.

Upload the code:

Open your code in the Arduino IDE.

Verify that your code compiles by clicking on "Sketch" -> "Verify/Compile".

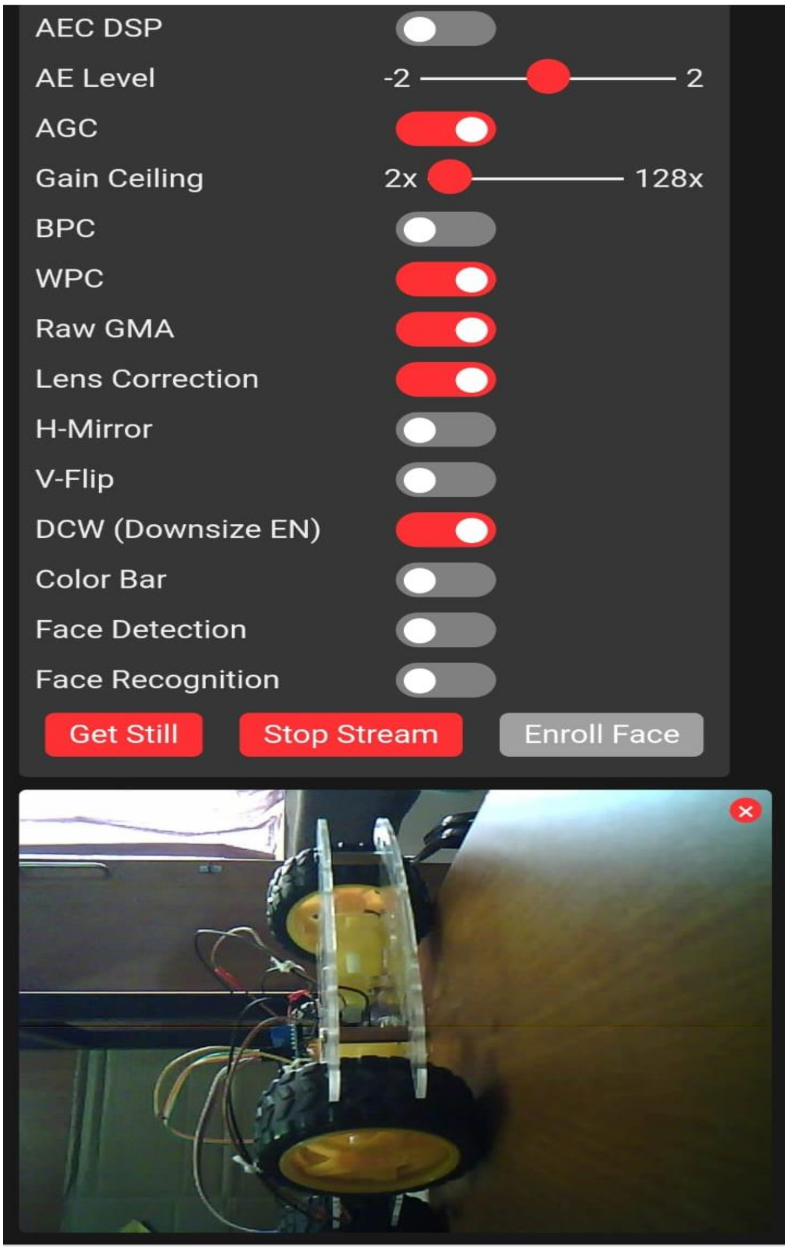
Click on "Sketch" -> "Upload" to upload the code to the ESP32-CAM module.

Monitor the serial output:

To view the serial output of the ESP32-CAM module, open the Serial Monitor by clicking on "Tools" -> "Serial Monitor".

Set the baud rate to 115200.

our code should now be uploaded to the ESP32-CAM module and we should be able to see the serial output in the Serial Monitor.



AEC DSP ○

AE Level -2 2

AGC ●

Gain Ceiling 2x 128x

BPC ○

WPC ●

Raw GMA ●

Lens Correction ●

H-Mirror ○

V-Flip ○

DCW (Downsize EN) ●

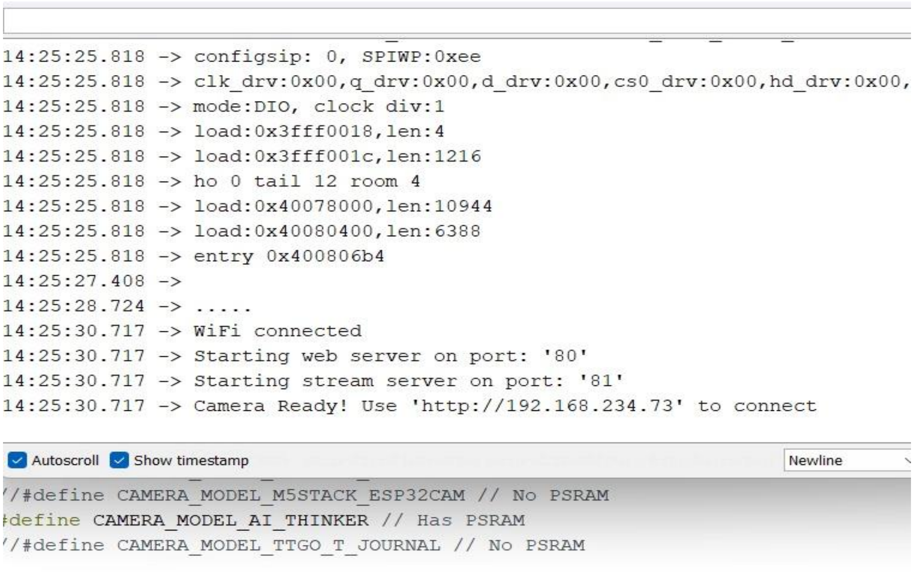
Color Bar ○

Face Detection ○

Face Recognition ○

Fig 3.3.4 open camera of esp32

After opening campera of esp32 we written the code of creating wavepage of project there are our navigation bottons,forward, left , right, backword ,stop and also to on of light.



```
14:25:25.818 -> configsip: 0, SPIWP:0xee
14:25:25.818 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,
14:25:25.818 -> mode:DIO, clock div:1
14:25:25.818 -> load:0x3fff0018,len:4
14:25:25.818 -> load:0x3fff001c,len:1216
14:25:25.818 -> ho 0 tail 12 room 4
14:25:25.818 -> load:0x40078000,len:10944
14:25:25.818 -> load:0x40080400,len:6388
14:25:25.818 -> entry 0x400806b4
14:25:27.408 ->
14:25:28.724 -> .....
14:25:30.717 -> WiFi connected
14:25:30.717 -> Starting web server on port: '80'
14:25:30.717 -> Starting stream server on port: '81'
14:25:30.717 -> Camera Ready! Use 'http://192.168.234.73' to connect
```

☑ Autoscroll ☑ Show timestamp Newline

```
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
```

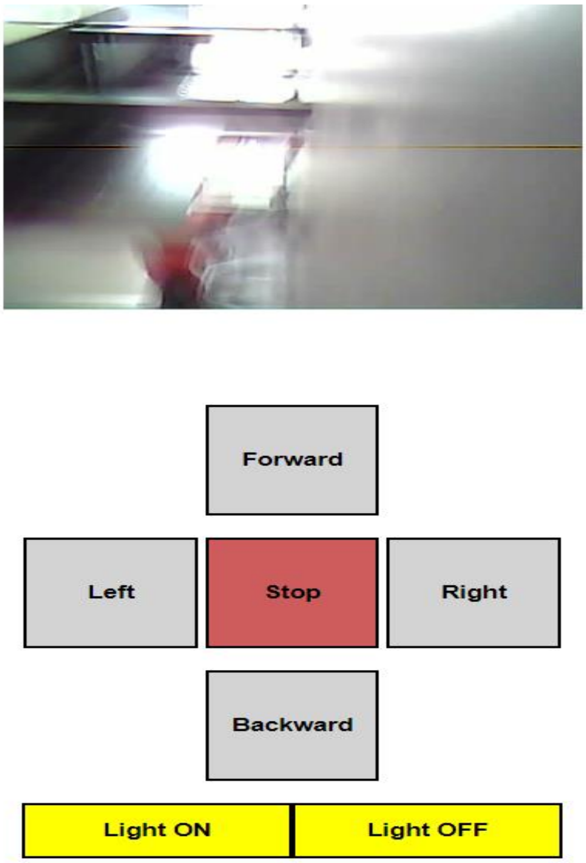


Fig 3.3.5 html page of navigation botton

Ultra sonic distance Measure :this is another feature of our robot distance measuring using ultrasonic Ultrasonic sensors measure distance by emitting high-frequency sound waves and then detecting the time it takes for the sound waves to bounce back after hitting an object. The sensor sends out a short burst of sound waves at a frequency that is too high for humans to hear (usually around 40 kHz).

The sound waves then travel through the air until they hit an object. When the sound waves hit the object, they bounce back towards the sensor, and the sensor detects the time it takes for the sound waves to return.

By knowing the speed of sound through air (which is approximately 343 meters per second at room temperature), the sensor can calculate the distance to the object by multiplying the time it took for the sound waves to travel to the object and back by the speed of sound, and then dividing by two to account for the round-trip distance.

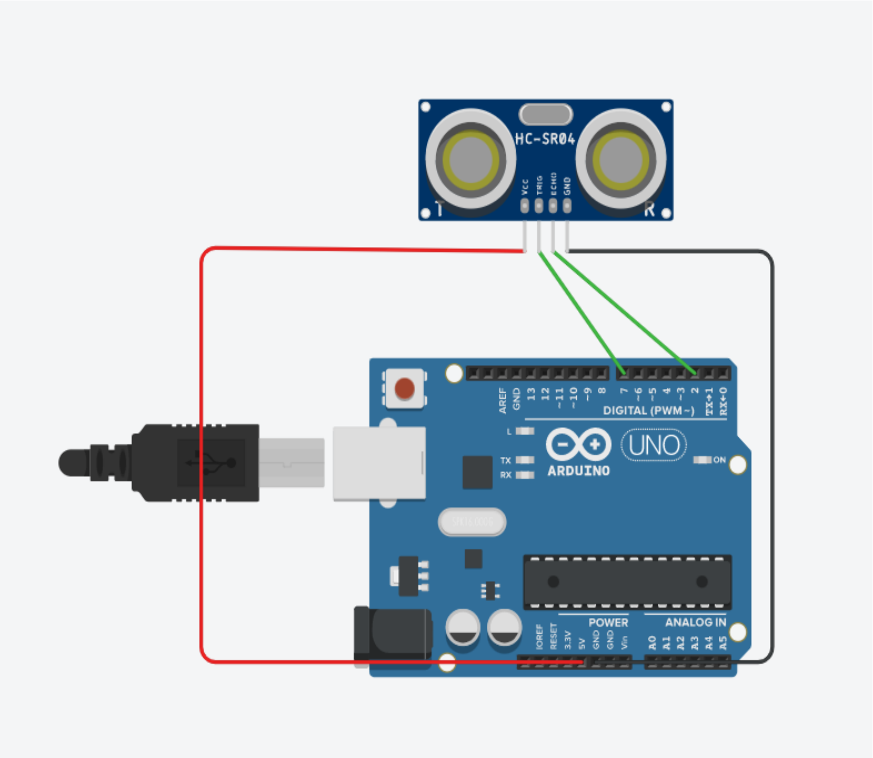


Fig 3.3.6 interfacing of ultrasonic sensor & Arduino

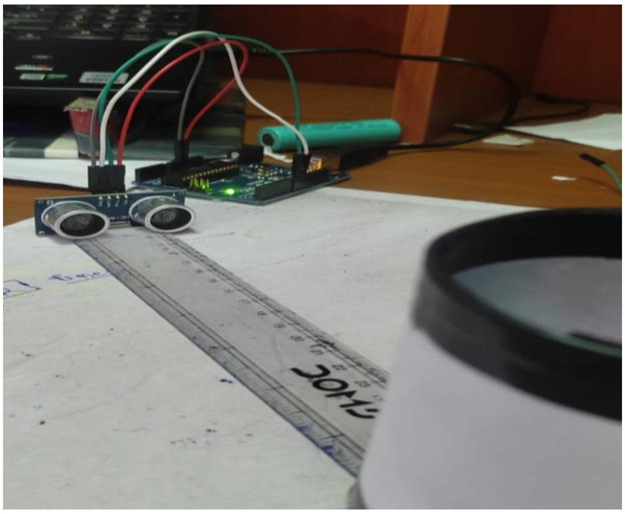
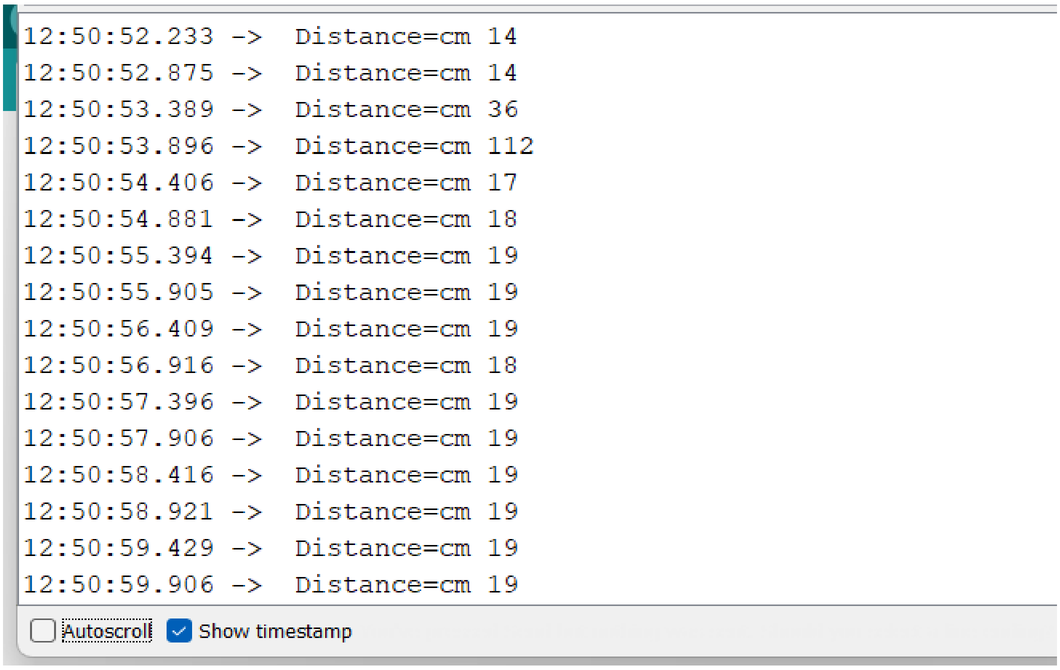


Fig 3.3.7 distance measuring



```
12:50:52.233 ->  Distance=cm 14
12:50:52.875 ->  Distance=cm 14
12:50:53.389 ->  Distance=cm 36
12:50:53.896 ->  Distance=cm 112
12:50:54.406 ->  Distance=cm 17
12:50:54.881 ->  Distance=cm 18
12:50:55.394 ->  Distance=cm 19
12:50:55.905 ->  Distance=cm 19
12:50:56.409 ->  Distance=cm 19
12:50:56.916 ->  Distance=cm 18
12:50:57.396 ->  Distance=cm 19
12:50:57.906 ->  Distance=cm 19
12:50:58.416 ->  Distance=cm 19
12:50:58.921 ->  Distance=cm 19
12:50:59.429 ->  Distance=cm 19
12:50:59.906 ->  Distance=cm 19
```

After doing all testing and learning about components we combine all just feature but there is some error to add ultrasonic sensor in the car so do it with ardiuno uno

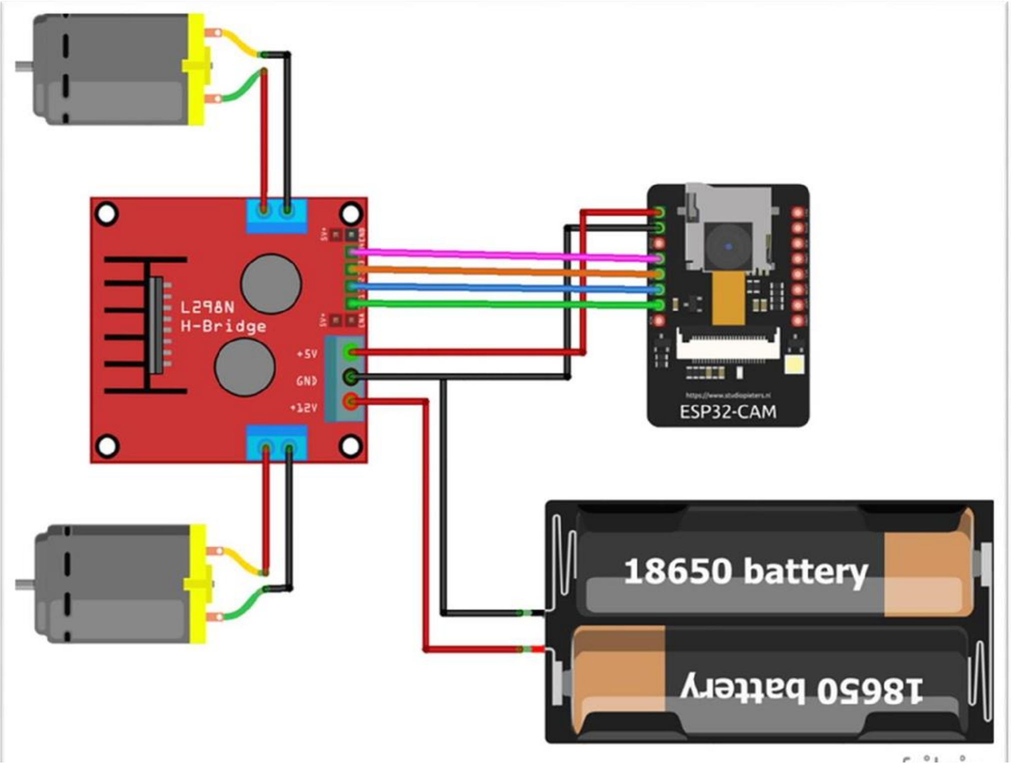


Fig 3.3.8 circuit diagram

Chapter 4

4. Results and discussion

Car is working properly and also giving the live streaming of the environment .

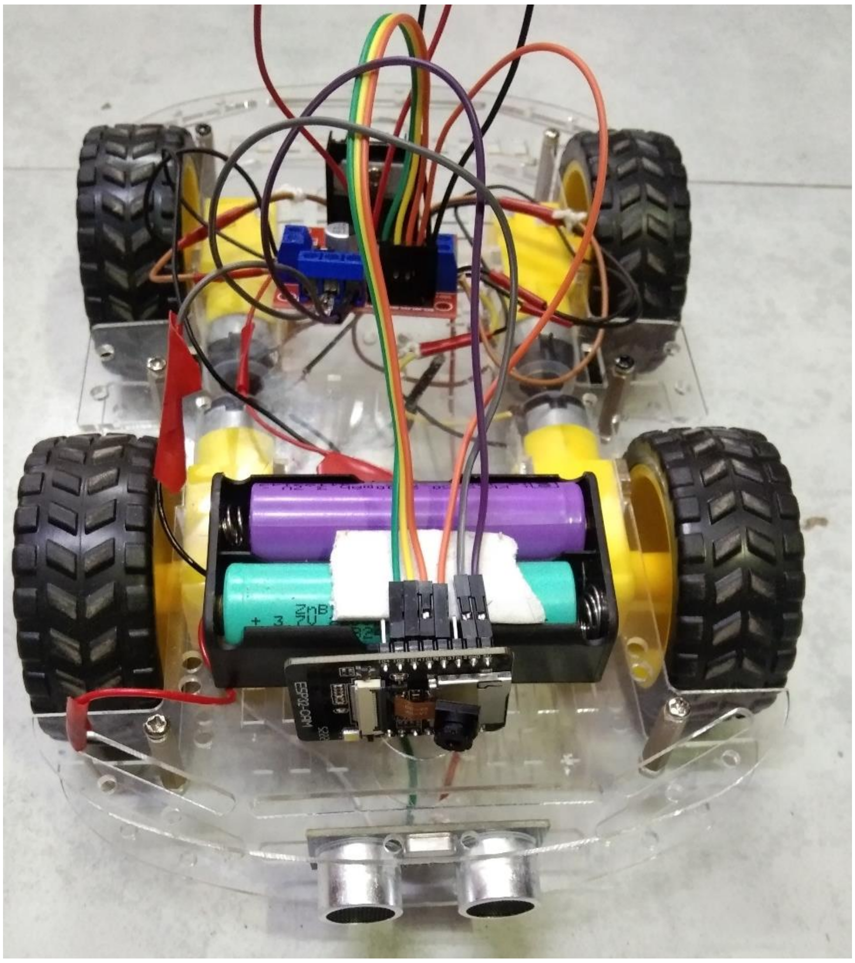


Fig 4.1 final prototype of spy robot

Chapter 5

5. Conclusion and future scope

Small Unarmed Ground Vehicle (SUGV) robot is developed for army operations and thus save life of the soldiers during a terrorist attack or any dangerous mission. The movement of the robot is controlled by the smartphone and can be moved in any direction. On top of the robot a smartphone with android application installed is mounted and the live surveillance video gained from it gives an insight to the environment of the war zone. It can be used to know exactly how many people are hidden in a room and based on that action can be taken by the armed forces. The SUGV robot is cost effective and can be easily produced and distributed. It can also be used as a home surveillance robot. In future Using this project, we will also estimate the speed of the vehicle, recognizing the number plate, recognizing the accidents took place on roads and also by using IOT we will get alerts related to any damage or fault of each particular streetlight and also we will resolving any technical issue like sorting the faulty streetlight would be easier through the application.

In the future, we are planning to add more sensors in the microcontroller This will potentially increase the processing power and speed, and will further decrease the power consumption. These additional sensors will also help the Robot to behave as an autonomous agent, when the wireless communication with the Android device is broken.

References

[1] Mashrik, Tahzib, Hasib Zunair, and Maofic Farhan Karin. "Design and Implementation of Security Patrol Robot Using Android Application." *2017 Asia Modelling Symposium (AMS)*. IEEE, 2017.

[2] Marapalli, Krishnakumar, et al. "AIGER An Intelligent Vehicle for Military Purpose." *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. Vol. 1. IEEE, 2021.

[3] Sharma, Divya, and Usha Chauhan. "War Spying Robot with Wireless Night Vision Camera." *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2020.

[4]Jing, Yuxin, et al. "AndroRC: An Android remote control car unit for search missions." *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*. IEEE, 2014.

APPENDICES

//CODE

```
//SPY ROBOT
#include "esp_camera.h"
#include <WiFi.h>
// Adafruit ESP32 Feather
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_M5STACK_PSRAM
#define CAMERA_MODEL_AI_THINKER
const char* ssid = "Redmi";   //Enter SSID WIFI Name
const char* password = "00000000";   //Enter WIFI Password
#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
```

```
#define Y5_GPIO_NUM      19
#define Y4_GPIO_NUM      18
#define Y3_GPIO_NUM       5
#define Y2_GPIO_NUM       4
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22
#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

```
#else
#error "Camera model not selected"
#endif
// GPIO Setting
extern int gpLb =  2; // Left 1
extern int gpLf = 14; // Left 2
extern int gpRb = 15; // Right 1
extern int gpRf = 13; // Right 2
extern int gpLed =  4; // Light
extern String WiFiAddr ="";
void startCameraServer();
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  pinMode(gpLb, OUTPUT); //Left Backward
  pinMode(gpLf, OUTPUT); //Left Forward
  pinMode(gpRb, OUTPUT); //Right Forward
  pinMode(gpRf, OUTPUT); //Right Backward
  pinMode(gpLed, OUTPUT); //Light
  //initialize
  digitalWrite(gpLb, LOW);
  digitalWrite(gpLf, LOW);
  digitalWrite(gpRb, LOW);
  digitalWrite(gpRf, LOW);
  digitalWrite(gpLed, LOW);

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
```

```
//init with high specs to pre-allocate larger buffers
if(psramFound()){
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}
// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}
//drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
```

```
  Serial.println("");
  Serial.println("WiFi connected");
  startCameraServer();
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  WiFiAddr = WiFi.localIP().toString();
  Serial.println("' to connect");
}
void loop() {
  // put your main code here, to run repeatedly:
}
```