WIFI Based Remote Surveillance Robot

by

JESSENTH EBENEZER (19BCE1462)
TEJAS.V (19BCE1607)

A project report submitted to Dr.Lucky Agarwal

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2006 – MICROPROCESSOR AND INTERFACING

in

B. Tech. COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY

CHENNAI - 600127

June 2021

BONAFIDE CERTIFICATE

This is to certify that this project report entitled “ **WIFI Based Remote Surveillance Robot**” is a bonafide work of by **Jessenth Ebenezer(19BCE1462)** and **Tejas.V(19BCE1607)** who carried out the Project work under my supervision and guidance for the course **CSE2006-MICROPROCESSOR AND INTERFACING**.

Dr.Lucky Agarwal
Associate Professor
School of Electronics Engineering (SENSE),
VIT University, Chennai
Chennai – 600 127.

ABSTRACT

In this project we will be making a remote control vehicle which will be used as a surveillance device using Wifi.We will be using wifi and will control the device through our computer/mobile using a web browser. The movement,capturing and processing of images and videos done by the surveillance robot will be monitored and controlled by the computer.Specific software and hardware can be modified according to the requirements needed.The robot can be used to go places where humans generally can’t go like in pipes,tunnels,underground places,etc.It can be used for many other activities such as video and image processing,military applications,etc.

Table of Contents

List of Figures

INTRODUCTION

1.1 OBJECTIVES

The objective of this project is to make a fully functional wifi based video surveillance robot.It is to perform various surveillance operations and other activities such as video and image processing,image capturing,etc.Surveillance robots are widely becoming the norm in regards to security.It is very highly in demand as it is the best surveillance devices to gather information and process the information and it will be able to upgrade itself as technology continuous to evolve.It reduces human activity in terms of operations performed during surveillances,it gathers and process information faster and delivers it to the required destination better than a human being.In the out project we will use a remote control vehicle which will be controlled through our computer to perform various operations during surveillance.It will be a wifi based project where the remote controlled car will be controlled by our computer with the help of wifi.
It will be able to do the above mentioned activities like image,video processing,etc.

1.2 SCOPE

The scope of this project provides many security companies and professionals an idea on how to make and develop surveillance robots with the same or more better operational capabilities.It gives them a picture on how to make suitable devices and models as per the environment they need or as per their demands.It also make future generations and young enthusiastic students take up an interest in the subject and helps them to take up similar projects like this.This project is also an ideal model for military drones,unmanned surveillance planes,etc.As technology is evolving the application of technology has also increased in the military field where our project can be used for applications in military operations.

IMPLEMENTATION/DESIGN

2.1 INTRODUCTION

In this project we will be using various hardware and software components.We will also be using a battery to run the remote controlled surveillance vehicle and a remote toy car or a truck.
We will integrate all of them in such a way that the software will be used to control the processes that goes on during surveillance and the battery is used to move the vehicle and the hardware is used to capture images,videos and is used to store it and send it to the computer for processing.

2.2 DESIGN APPROACH

The design approach for this project is to create a program using C language to create the controls of the vehicle and the interface between hardware and software.It uses a eps23 module camera to capture images and videos.It has 2 batteries to power the vehicle to move in specific directions given by the user.

2.3 PROPOSED SYSTEM

The ESP32-CAM module can be programmed with Arduino IDE and other IDE also. The ESP32-CAM module also has several GPIO pins to connect to external hardwares.
For this robot project, LM2596 DC-DC Buck Converter Power Supply - Step Down Module (Variable
Voltage) is used. You can always take +5V and Ground for ESP32-CAM from LM2596 Module and
+12V Power Supply directly from a Battery is used for DC geared motor through L298N motor driver. The power supply module has a trim pot which can be turned into get correct voltage for

use to ESP32 CAM Module.

In order to attach the module the chassis platform, using a hilum sheet all the modules for this

project are fixed. All connections are made as per the schematic. A 12V dry type battery also fixed over the chassis.

Install the ESP32 Add-on on Arduino IDE.Make sure that you have two tabs with code in the Arduino IDE.Connect the ESP32-CAM board to your computer using an FTDI programmer.

Important: GPIO 0 needs to be connected to GND so that you’re able to upload code.

Press the ESP32-CAM on-board RST button before upload the code.

Run Arduino IDE and click Upload to build and flash the sketch to your device.

After uploading the code, disconnect GPIO 0 from GND.

Open the Serial Monitor at a baud rate of 115200. Press the ESP32-CAM on-board RST button.

The ESP32-CAM IP address should be shown in the Serial Monitor.

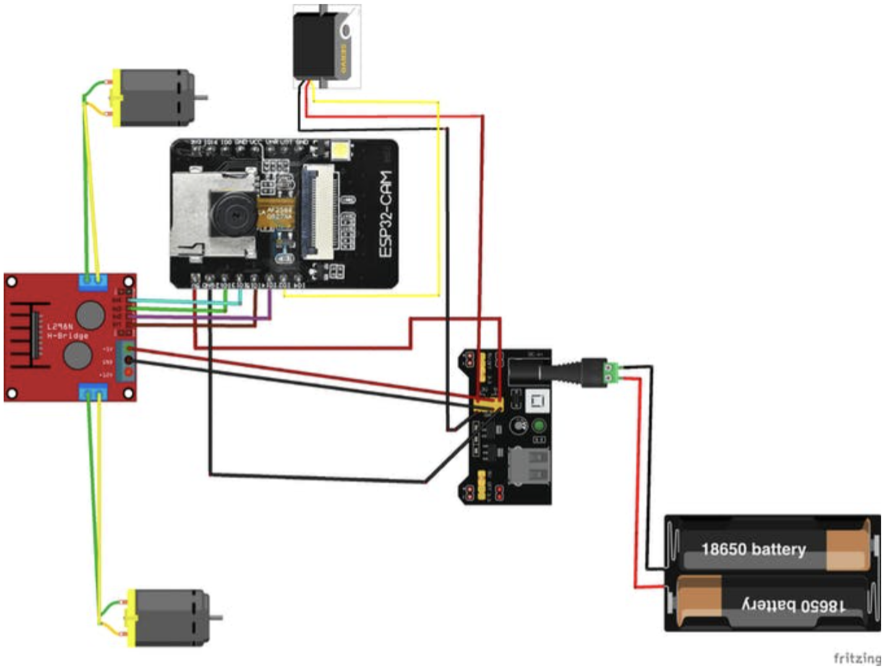
Disconnect the ESP32-CAM from the FTDI programmer.

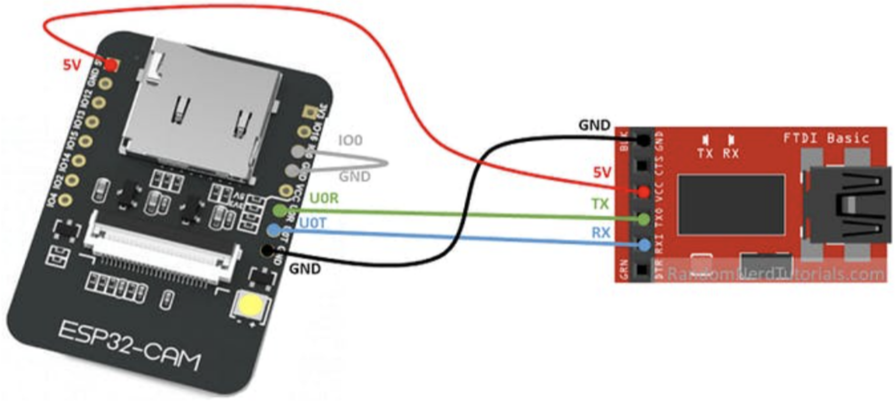
Connect the ESP32-CAM back to the Pan/Tilt platform, turn on power and press the ESP32-CAM on- board RST button.

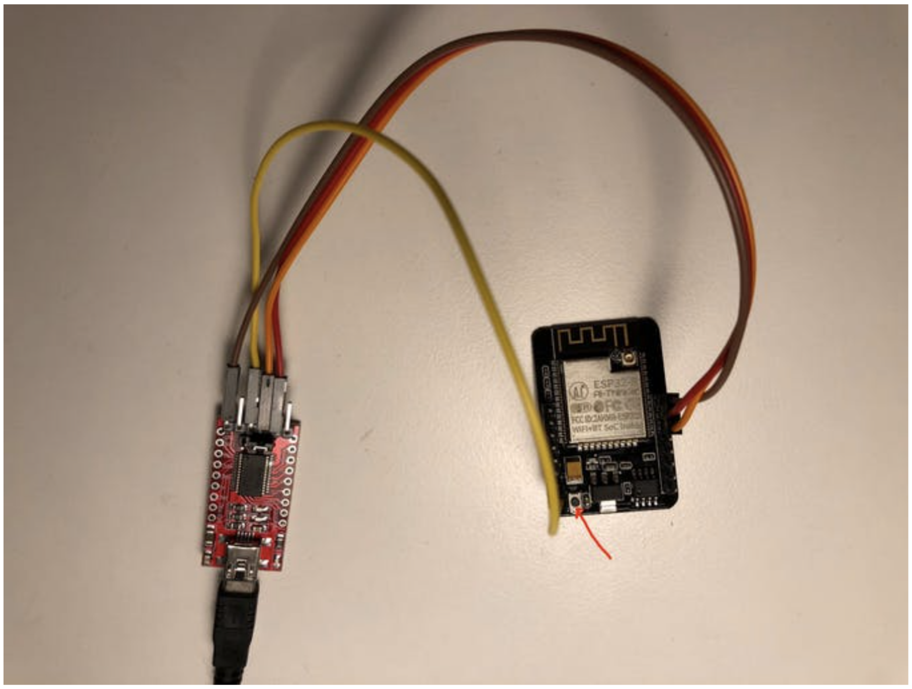
After uploading the code into the ESP32 CAM, open the smart touch app with any android mobile which is connected to the particular WiFi network. Enter the WiFi credentials in the app and you will find the IP address of the ESP 32 Camera.Open the browser either in Laptop/Android mobile which is connected with same WiFi network.Enter that IP Address, now you can see the video streaming in the Laptop/Android mobile device.Also you can navigate the robot using the navigation bars shown below the video being streamed at

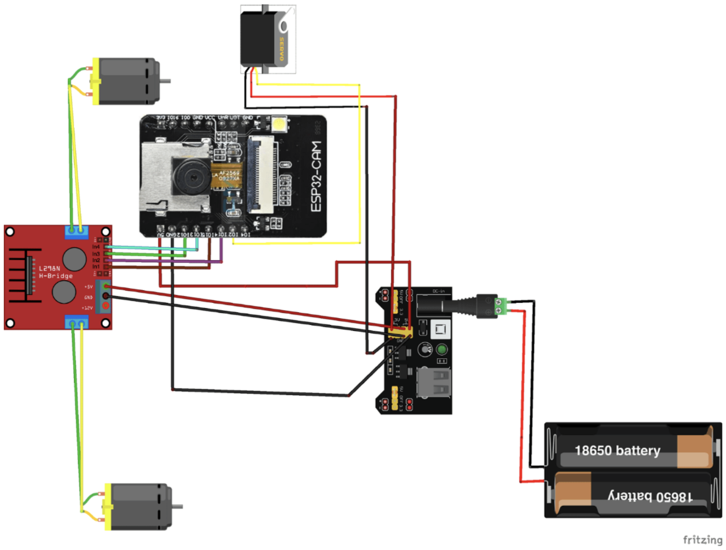
the browser of either Laptop/Android Mobile.

Circuit diagrams:


ESP32-CAM
L298N
H-Bridge
18650 battery
18650 battery
fritzing


5V
IO0
GND
U0R
U0T
GND
GND
5V
TX
RX
FTDI Basic
TX RX
BLK GND CTS VCC TXO RXI DTR GRN
RandomNerdTutorials.com
ESP32-CAM




ESP32-CAM
L298N
H-Bridge
18650 battery
18650 battery
fritzing

2.3.1 Economic Feasibility

This project is cost effective.The total cost of all the components(both hardware and software) required to build this project is around 2000 rupees.It is very cheap in making this project compared to other devices and models who perform the same or similar operations.

2.3.2 Technical feasibility:

We have gone through all the specifications and requirements needed to complete this project.

All interconnections between hardware and software have been established and we are able to build this project.As technology evolves with time we will be able to upgrade the project with better hardware and software components.We make sure that no glitches or errors occur while making this project and to make sure this remote surveillance vehicle provides the best of service.

2.3.3Operational feasibility

All operations in this project are performed through the computer which is easy to use.

The vehicle can also be controlled by the app installed on our phones.

2.4 Overview of software

C programming language:C is a general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system. By design, C provides constructs that map efficiently to typical machine instructions. It has found lasting use in applications previously coded in assembly language. Such applications include operating systems and various application software for computer architectures that range from supercomputers to PLCs and embedded systems.

2.5 Hardware Requirements

1. ESP - 32 CAM.
2. FT232RL FTDI Mini USB to TTL Serial Converter Adapter Module.
3. Dual Channel DC Motor Driver L298N.
4. Robotic Chassis Kit.
5. DC Geared Motors.
6. 12V Dry Type Battery.
7. LM2596 : DC-DC Buck Converter Power Supply - Step Down Module (Variable Voltage)
8. “ESP Touch Smart Configuration” app from google play store.
9. Any WiFI Connection.

2.6 Software Requirements

The software requirements needed for this project are C programming language,a device like a computer or an android phone where an app for example iRobbie-A can be installed to run the vehicle.

2.7 Summary

All technical,operational and economic feasibilities have been established.All requirements both hardware and software have been met.

Result and Analysis/Testing

3.1IMPLEMENTATION CODE

esp32cam_carArduino

```
/*
ESP32-CAM Remote Control
*/
const char* ssid = &quot;Your_WIFI_Network&quot;;
const char* password = &quot;Your password&quot;;
#include &quot;esp_wifi.h&quot;
#include &quot;esp_camera.h&quot;
#include &lt;WiFi.h&gt;
#include &quot;soc/soc.h&quot;
#include &quot;soc/rtc_cntl_reg.h&quot;
//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
// or another board which has PSRAM enabled
//
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_M5STACK_PSRAM
#define CAMERA_MODEL_AI_THINKER
#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_M5STACK_PSRAM)
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 25
#define SIOC_GPIO_NUM 23
#define Y9_GPIO_NUM 19
#define Y8_GPIO_NUM 36
#define Y7_GPIO_NUM 18
#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 5
#define Y4_GPIO_NUM 34
#define Y3_GPIO_NUM 35
#define Y2_GPIO_NUM 32
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM 26
#define PCLK_GPIO_NUM 21
#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 &quot;Camera model not selected&quot;
#endif
void startCameraServer();
const int MotPin0 = 12;
const int MotPin1 = 13;
const int MotPin2 = 14;
const int MotPin3 = 15;

void initMotors()
{
ledcSetup(3, 2000, 8); // 2000 hz PWM, 8-bit resolution
ledcSetup(4, 2000, 8); // 2000 hz PWM, 8-bit resolution
ledcSetup(5, 2000, 8); // 2000 hz PWM, 8-bit resolution
ledcSetup(6, 2000, 8); // 2000 hz PWM, 8-bit resolution
ledcAttachPin(MotPin0, 3);
ledcAttachPin(MotPin1, 4);
ledcAttachPin(MotPin2, 5);
ledcAttachPin(MotPin3, 6);
```

```
}
const int ServoPin = 2;
void initServo()
{
ledcSetup(8, 50, 16); // 50 hz PWM, 16-bit resolution, range from 3250 to 6500.
ledcAttachPin(ServoPin, 8);
}
void setup()
{
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // prevent brownouts by silencing them
Serial.begin(115200);
Serial.setDebugOutput(true);
Serial.println();
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_QVGA;
config.jpeg_quality = 10;
config.fb_count = 2;
} else {
config.frame_size = FRAMESIZE_QVGA;
config.jpeg_quality = 12;
config.fb_count = 1;
}
// camera init
esp_err_t err = esp_camera_init(&amp;config);
if (err != ESP_OK) {
Serial.printf(&quot;Camera init failed with error 0x%x&quot;, err);
return;
}
//drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s-&gt;set_framesize(s, FRAMESIZE_QVGA);
s-&gt;set_vflip(s, 1);
s-&gt;set_hmirror(s, 1);
// Remote Control Car
initMotors();
initServo();
ledcSetup(7, 5000, 8);
ledcAttachPin(4, 7); //pin4 is LED
```

```
Serial.println(&quot;ssid: &quot; + (String)ssid);
Serial.println(&quot;password: &quot; + (String)password);
WiFi.begin(ssid, password);
delay(500);
long int StartTime=millis();
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
if ((StartTime+10000) &lt; millis()) break;
}
/*
int8_t power;
esp_wifi_set_max_tx_power(20);
esp_wifi_get_max_tx_power(&amp;power);
Serial.printf(&quot;wifi power: %d \n&quot;,power);
*/

startCameraServer();
if (WiFi.status() == WL_CONNECTED)
{
Serial.println(&quot;&quot;);
Serial.println(&quot;WiFi connected&quot;);
Serial.print(&quot;Camera Ready! Use &#39;http://&quot;);
Serial.print(WiFi.localIP());
Serial.println(&quot;&#39; to connect&quot;);
} else {
Serial.println(&quot;&quot;);
Serial.println(&quot;WiFi disconnected&quot;);
Serial.print(&quot;Camera Ready! Use &#39;http://&quot;);
Serial.print(WiFi.softAPIP());
Serial.println(&quot;&#39; to connect&quot;);
char* apssid = &quot;ESP32-CAM&quot;;
char* appassword = &quot;12345678&quot;; //AP password require at least 8 characters.
WiFi.softAP((WiFi.softAPIP().toString()+&quot;_&quot;+(String)apssid).c_str(), appassword);
}
for (int i=0;i&lt;5;i++)
{
ledcWrite(7,10); // flash led
delay(200);
ledcWrite(7,0);
delay(200);
}
}
void loop() {
// put your main code here, to run repeatedly:
delay(1000);
//Serial.printf(&quot;RSSi: %ld dBm\n&quot;,WiFi.RSSI());
}

//SECOND PART OF THE CODE. PUT IT TO ANOTHER TAB &quot;app_httpd.cpp&quot;
//or download entire code from https://www.robbie-app.com/esp32cam_car.zip

#include &lt;esp32-hal-ledc.h&gt;
int cspeed = 200;
int noStop = 1;
int xcoord = 0;
float speed_Coeff = (1 + (xcoord/50.0));
#include &quot;esp_http_server.h&quot;
#include &quot;esp_timer.h&quot;
#include &quot;esp_camera.h&quot;
#include &quot;img_converters.h&quot;
```

```
#include &quot;Arduino.h&quot;
//#include &lt;dl_lib.h&gt;
typedef struct {
httpd_req_t *req;
size_t len;
} jpg_chunking_t;
#define PART_BOUNDARY &quot;123456789000000000000987654321&quot;
static const char* _STREAM_CONTENT_TYPE =
&quot;multipart/x-mixed-replace;boundary=&quot;
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = &quot;\r\n--&quot; PART_BOUNDARY
&quot;\r\n&quot;;
static const char* _STREAM_PART = &quot;Content-Type: image/jpeg\r\nContent-Length:
%u\r\n\r\n&quot;;
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;
static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len){
jpg_chunking_t *j = (jpg_chunking_t *)arg;
if(!index){
j-&gt;len = 0;
}
if(httpd_resp_send_chunk(j-&gt;req, (const char *)data, len) != ESP_OK){
return 0;
}
j-&gt;len += len;
return len;
}
static esp_err_t capture_handler(httpd_req_t *req){
camera_fb_t * fb = NULL;
esp_err_t res = ESP_OK;
int64_t fr_start = esp_timer_get_time();
```

```
fb = esp_camera_fb_get();
if (!fb) {
Serial.println(&quot;Camera capture failed&quot;);
httpd_resp_send_500(req);
return ESP_FAIL;
}
httpd_resp_set_type(req, &quot;image/jpeg&quot;);
httpd_resp_set_hdr(req, &quot;Content-Disposition&quot;, &quot;inline;
filename=capture.jpg&quot;);
size_t out_len, out_width, out_height;
uint8_t * out_buf;
bool s;
{
size_t fb_len = 0;
if(fb-&gt;format == PIXFORMAT_JPEG){

fb_len = fb-&gt;len;
res = httpd_resp_send(req, (const char *)fb-&gt;buf, fb-&gt;len);
} else {
jpg_chunking_t jchunk = {req, 0};
res = frame2jpg_cb(fb, 80, jpg_encode_stream, &amp;jchunk)?ESP_OK:ESP_FAIL;
httpd_resp_send_chunk(req, NULL, 0);
fb_len = jchunk.len;
}
esp_camera_fb_return(fb);
int64_t fr_end = esp_timer_get_time();
Serial.printf(&quot;JPG: %uB %ums\n&quot;, (uint32_t)(fb_len), (uint32_t)((fr_end -
fr_start)/1000));
return res;
}
// dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, fb-&gt;width, fb-&gt;height, 3);
// if (!image_matrix) {
esp_camera_fb_return(fb);
Serial.println(&quot;dl_matrix3du_alloc failed&quot;);
httpd_resp_send_500(req);
return ESP_FAIL;
}
// out_buf = image_matrix-&gt;item;
// out_len = fb-&gt;width * fb-&gt;height * 3;
// out_width = fb-&gt;width;
// out_height = fb-&gt;height;
// s = fmt2rgb888(fb-&gt;buf, fb-&gt;len, fb-&gt;format, out_buf);
// esp_camera_fb_return(fb);
// if(!s){
// dl_matrix3du_free(image_matrix);
// Serial.println(&quot;to rgb888 failed&quot;);
// httpd_resp_send_500(req);
// return ESP_FAIL;
// }
// jpg_chunking_t jchunk = {req, 0};
// s = fmt2jpg_cb(out_buf, out_len, out_width, out_height, PIXFORMAT_RGB888, 90,
jpg_encode_stream, &amp;jchunk);
// dl_matrix3du_free(image_matrix);
// if(!s){
// Serial.println(&quot;JPEG compression failed&quot;);
// return ESP_FAIL;
// }
// int64_t fr_end = esp_timer_get_time();
// return res;
//}
static esp_err_t stream_handler(httpd_req_t *req){
camera_fb_t * fb = NULL;
```

```
esp_err_t res = ESP_OK;
size_t _jpg_buf_len = 0;
uint8_t * _jpg_buf = NULL;
char * part_buf[64];
// dl_matrix3du_t *image_matrix = NULL;
static int64_t last_frame = 0;
if(!last_frame) {
last_frame = esp_timer_get_time();
}
res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
if(res != ESP_OK){
return res;
}
while(true){
fb = esp_camera_fb_get();
if (!fb) {
Serial.println(&quot;Camera capture failed&quot;);
res = ESP_FAIL;
} else {
{
if(fb-&gt;format != PIXFORMAT_JPEG){
bool jpeg_converted = frame2jpg(fb, 80, &amp;_jpg_buf, &amp;_jpg_buf_len);
esp_camera_fb_return(fb);
fb = NULL;
if(!jpeg_converted){
Serial.println(&quot;JPEG compression failed&quot;);
res = ESP_FAIL;
}
} else {
_jpg_buf_len = fb-&gt;len;
_jpg_buf = fb-&gt;buf;
}
}
}
if(res == ESP_OK){
size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
}
if(res == ESP_OK){
res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
}
if(res == ESP_OK){
res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
}
if(fb){
esp_camera_fb_return(fb);
fb = NULL;
_jpg_buf = NULL;

} else if(_jpg_buf){
free(_jpg_buf);
_jpg_buf = NULL;
}
if(res != ESP_OK){
break;
}
int64_t fr_end = esp_timer_get_time();
int64_t frame_time = fr_end - last_frame;
last_frame = fr_end;
frame_time /= 1000;
```

```
Serial.printf(&quot;MJPG: %uB %ums (%.1ffps)\n&quot;,
(uint32_t)(_jpg_buf_len),
(uint32_t)frame_time, 1000.0 / (uint32_t)frame_time
);
}
last_frame = 0;
return res;
}
enum state {fwd,rev,stp};
state actstate = stp;
static esp_err_t cmd_handler(httpd_req_t *req)
{
char* buf;
size_t buf_len;
char variable[32] = {0,};
char value[32] = {0,};
buf_len = httpd_req_get_url_query_len(req) + 1;
if (buf_len &gt; 1) {
buf = (char*)malloc(buf_len);
if(!buf){
httpd_resp_send_500(req);
return ESP_FAIL;
}
if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
if (httpd_query_key_value(buf, &quot;var&quot;, variable, sizeof(variable)) == ESP_OK
&amp;&amp;
httpd_query_key_value(buf, &quot;val&quot;, value, sizeof(value)) == ESP_OK) {
} else {
free(buf);
httpd_resp_send_404(req);
return ESP_FAIL;
}
} else {
free(buf);
httpd_resp_send_404(req);
return ESP_FAIL;
}

free(buf);
} else {
httpd_resp_send_404(req);
return ESP_FAIL;
}
int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;
if(!strcmp(variable, &quot;framesize&quot;))
{
Serial.println(&quot;framesize&quot;);
if(s-&gt;pixformat == PIXFORMAT_JPEG) res = s-&gt;set_framesize(s, (framesize_t)val);
}
else if(!strcmp(variable, &quot;quality&quot;))
{
Serial.println(&quot;quality&quot;);
res = s-&gt;set_quality(s, val);
}
//Remote Control Car
//Don&#39;t use channel 1 and channel 2
else if(!strcmp(variable, &quot;flash&quot;))
{
ledcWrite(7,val);
}
```

```
else if(!strcmp(variable, &quot;speeds&quot;))
{
if (val &gt; 255) val = 255;
else if (val &lt; 0) val = 0;
cspeed = val*2;
}
else if(!strcmp(variable, &quot;xcoord&quot;))
{
if (val &gt; 255) val = 255;
// else if (val &lt; 0) val = 0;
xcoord = val;
speed_Coeff = (1 + (xcoord/50.0));
}
else if(!strcmp(variable, &quot;nostop&quot;))
{
noStop = val;
}
else if(!strcmp(variable, &quot;servo&quot;)) // 3250, 4875, 6500
{
if (val &gt; 650) val = 650;
else if (val &lt; 326) val = 225;
ledcWrite(8,10*val);

}
else if(!strcmp(variable, &quot;car&quot;)) {
if(val == 1){
actstate = fwd;
ledcWrite(4,cspeed); // pin 12
ledcWrite(3,0); // pin 13
ledcWrite(5,cspeed); // pin 14
ledcWrite(6,0); // pin 15
delay(25);
}
if(val == 0){
actstate = stp;
ledcWrite(4,0);
ledcWrite(3,0);
ledcWrite(5,0);
ledcWrite(6,0);
}
if(val == 2){
actstate = rev;
ledcWrite(4,0);
ledcWrite(3,cspeed);
ledcWrite(5,0);
ledcWrite(6,cspeed);
delay(25);
}
if(val == 3){
ledcWrite(3,0);
ledcWrite(6,0);
ledcWrite(5,cspeed+30);
ledcWrite(4, cspeed/speed_Coeff);
delay(25);
}
if(val == 4){
ledcWrite(3,0);
ledcWrite(6,0);
ledcWrite(5, cspeed/speed_Coeff);
ledcWrite(4,cspeed+30);
delay(25);
}
```

```
if(val == 5){
ledcWrite(6,0); ledcWrite(3,0); ledcWrite(5,130);
delay(25);
}
if(val == 6){
ledcWrite(5,0); ledcWrite(4,130); ledcWrite(6,0);
delay(25);

}
if(val == 7){
ledcWrite(4,0);
ledcWrite(5,0);
ledcWrite(6, cspeed/speed_Coeff);
ledcWrite(3,cspeed+30);
delay(25);
}
if(val == 8){
ledcWrite(4,0);
ledcWrite(5,0);
ledcWrite(6,cspeed+50);
ledcWrite(3, cspeed/speed_Coeff);
delay(25);
}
if (noStop!=1)
{
ledcWrite(3, 0);
ledcWrite(4, 0);
ledcWrite(5, 0);
ledcWrite(6, 0);
}
```

```
}
else
{
Serial.println(&quot;variable&quot;);
res = -1;
}
if(res){ return httpd_resp_send_500(req); }
httpd_resp_set_hdr(req, &quot;Access-Control-Allow-Origin&quot;, &quot;*&quot;);
return httpd_resp_send(req, NULL, 0);

}

static esp_err_t status_handler(httpd_req_t *req){
static char json_response[1024];
sensor_t * s = esp_camera_sensor_get();
char * p = json_response;
*p++ = &#39;{&#39;;
p+=sprintf(p, &quot;\&quot;framesize\&quot;:%u,&quot;, s-&gt;status.framesize);
p+=sprintf(p, &quot;\&quot;quality\&quot;:%u,&quot;, s-&gt;status.quality);
*p++ = &#39;}&#39;;
*p++ = 0;
httpd_resp_set_type(req, &quot;application/json&quot;);
httpd_resp_set_hdr(req, &quot;Access-Control-Allow-Origin&quot;, &quot;*&quot;);
return httpd_resp_send(req, json_response, strlen(json_response));
}
static const char PROGMEM INDEX_HTML[] = R&quot;rawliteral(
&lt;!doctype html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta charset=&quot;utf-8&quot;&gt;
```

```
&lt;meta name=&quot;viewport&quot;
content=&quot;width=device-width,initial-scale=1&quot;&gt;
&lt;title&gt;ESP32 OV2460&lt;/title&gt;
&lt;style&gt;
body{font-family:Arial,Helvetica,sans-serif;background:#181818;color:#EFEFEF;font-
size:16px}h2{font-size:18px}section.main{display:flex}#menu,section.main{flex-
direction:column}#menu{display:none;flex-wrap:nowrap;min-
width:340px;background:#363636;padding:8px;border-radius:4px;margin-top:-10px;margin-
right:10px}#content{display:flex;flex-wrap:wrap;align-items:stretch}figure{padding:0;margin:0;
-
webkit-margin-before:0;margin-block-start:0;-webkit-margin-after:0;margin-block-end:0;-webkit
-
margin-start:0;margin-inline-start:0;-webkit-margin-end:0;margin-inline-end:0}figure
img{display:block;width:100%;height:auto;border-radius:4px;margin-top:8px}@media
(min-width:
800px) and (orientation:landscape){#content{display:flex;flex-wrap:nowrap;align-
items:stretch}figure img{display:block;max-width:100%;max-height:calc(100vh -
40px);width:auto;height:auto}figure{padding:0;margin:0;-webkit-margin-before:0;margin-block
-
start:0;-webkit-margin-after:0;margin-block-end:0;-webkit-margin-start:0;margin-inline-start:0;-
webkit-margin-end:0;margin-inline-end:0}}section#buttons{display:flex;flex-wrap:nowrap;justif
y-
content:space-between}#nav-toggle{cursor:pointer;display:block}#nav-toggle-
cb{outline:0;opacity:0;width:0;height:0}#nav-toggle-cb:checked+#menu{display:flex}.input-
group{display:flex;flex-wrap:nowrap;line-height:22px;margin:5px 0}.input-
group&gt;label{display:inline-block;padding-right:10px;min-width:47%}.input-group
input,.input-group
select{flex-grow:1}.range-max,.range-min{display:inline-block;padding:0
5px}button{display:block;margin:5px;padding:0 12px;border:0;line-
height:28px;cursor:pointer;color:#fff;background:#ff3034;border-radius:5px;font-
```

```
size:16px;outline:0}button:hover{background:#ff494d}button:active{background:#f21c21}butto
n.dis
abled{cursor:default;background:#a0a0a0}input[type=range]{-webkit-
appearance:none;width:100%;height:22px;background:#363636;cursor:pointer;margin:0}input[t
ype
=range]:focus{outline:0}input[type=range]::-webkit-slider-runnable-
track{width:100%;height:2px;cursor:pointer;background:#EFEFEF;border-radius:0;border:0
solid
#EFEFEF}input[type=range]::-webkit-slider-thumb{border:1px solid

rgba(0,0,30,0);height:22px;width:22px;border-radius:50px;background:#ff3034;cursor:pointer;-
webkit-appearance:none;margin-top:-11.5px}input[type=range]:focus::-webkit-slider-runnable-
track{background:#EFEFEF}input[type=range]::-moz-range-
track{width:100%;height:2px;cursor:pointer;background:#EFEFEF;border-radius:0;border:0
solid
#EFEFEF}input[type=range]::-moz-range-thumb{border:1px solid
rgba(0,0,30,0);height:22px;width:22px;border-
radius:50px;background:#ff3034;cursor:pointer}input[type=range]::-ms-
track{width:100%;height:2px;cursor:pointer;background:0 0;border-
color:transparent;color:transparent}input[type=range]::-ms-fill-lower{background:#EFEFEF;bor
der:0
solid
#EFEFEF;border-radius:0}input[type=range]::-ms-fill-upper{background:#EFEFEF;border:0
solid
#EFEFEF;border-radius:0}input[type=range]::-ms-thumb{border:1px solid
rgba(0,0,30,0);height:22px;width:22px;border-
radius:50px;background:#ff3034;cursor:pointer;height:2px}input[type=range]:focus::-ms-fill-
lower{background:#EFEFEF}input[type=range]:focus::-ms-fill-
upper{background:#363636}.switch{display:block;position:relative;line-height:22px;font-
size:16px;height:22px}.switch
input{outline:0;opacity:0;width:0;height:0}.slider{width:50px;height:22px;border-
```

```
radius:22px;cursor:pointer;background-color:grey}.slider,.slider:before{display:inline-
block;transition:.4s}.slider:before{position:relative;content:&quot;&quot;;border-
radius:50%;height:16px;width:16px;left:4px;top:3px;background-
color:#fff}input:checked+.slider{background-color:#ff3034}input:checked+.slider:before{-webk
it-
transform:translateX(26px);transform:translateX(26px)}select{border:1px solid #363636;font-
size:14px;height:22px;outline:0;border-radius:5px}.image-container{position:relative;min-
width:160px}.close{position:absolute;right:5px;top:5px;background:#ff3034;width:16px;height:
16px
;border-radius:100px;color:#fff;text-align:center;line-
height:18px;cursor:pointer}.hidden{display:none}
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;figure&gt;
&lt;div id=&quot;stream-container&quot; class=&quot;image-container hidden&quot;&gt;
&lt;div class=&quot;close&quot; id=&quot;close-stream&quot;&gt;×&lt;/div&gt;
&lt;img id=&quot;stream&quot; src=&quot;&quot;&gt;
&lt;/div&gt;
&lt;/figure&gt;
&lt;section class=&quot;main&quot;&gt;
&lt;section id=&quot;buttons&quot;&gt;
&lt;table&gt;
&lt;tr&gt;&lt;td align=&quot;center&quot;&gt;&lt;button id=&quot;get-still&quot;&gt;Get
Still&lt;/button&gt;&lt;/td&gt;&lt;td&gt;&lt;td&gt;&amp;nbsp&amp;nbsp&amp;nbsp&amp;nb
sp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp
&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&lt;/td&gt;&lt;td
align=&quot;center&quot;&gt;&lt;button id=&quot;toggle-stream&quot;&gt;Start
Stream&lt;/button&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td
```

```
align=&quot;center&quot;&gt;&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;
nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp
&amp;nbsp&amp;nbsp&amp;nbsp&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td
align=&quot;center&quot;&gt;&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;
nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp
&amp;nbsp&amp;nbsp&amp;nbsp&lt;/td&gt;&lt;td
align=&quot;center&quot;&gt;&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;
nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp
&amp;nbsp&amp;nbsp&amp;nbsp&lt;td&gt;&lt;/td&gt;&lt;td

align=&quot;center&quot;&gt;&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;
nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp&amp;nbsp
&amp;nbsp&amp;nbsp&amp;nbsp&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;/br&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;/br&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;/br&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Servo&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;input type=&quot;range&quot; id=&quot;servo&quot;
min=&quot;325&quot; max=&quot;650&quot; value=&quot;487&quot;
onchange=&quot;try{fetch(document.location.origin+&#39;/control?var=servo&amp;val=&#39
;+this.value);}catch(e){}&quot;&gt;&lt;t
d&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Flash&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;input type=&quot;range&quot; id=&quot;flash&quot;
min=&quot;0&quot;
max=&quot;255&quot; value=&quot;0&quot;
```

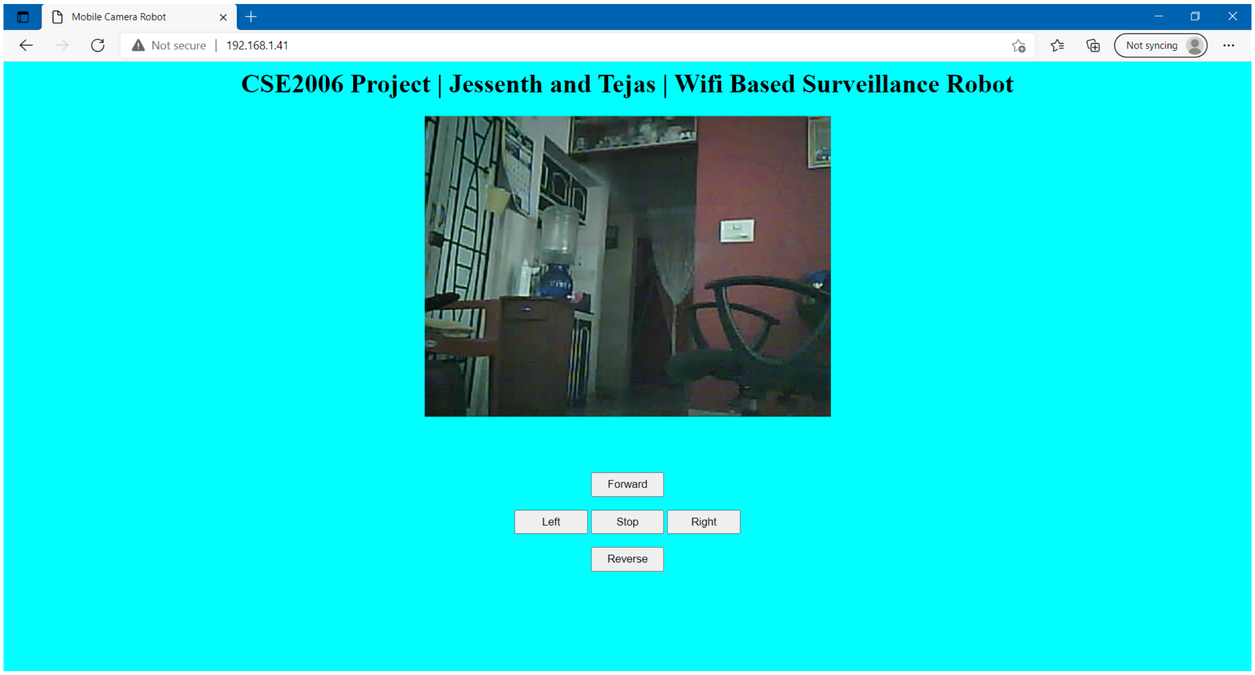
```
onchange=&quot;try{fetch(document.location.origin+&#39;/control?var=flash&amp;val=&#39;
+this.value);}catch(e){}&quot;&gt;&lt;t
d&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Resolution&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;input type=&quot;range&quot;
id=&quot;framesize&quot; min=&quot;0&quot; max=&quot;6&quot; value=&quot;5&quot;
onchange=&quot;try{fetch(document.location.origin+&#39;/control?var=framesize&amp;val=&
#39;+this.value);}catch(e){}
&quot;&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Quality&lt;/td&gt;&lt;td align=&quot;center&quot;
colspan=&quot;2&quot;&gt;&lt;input type=&quot;range&quot; id=&quot;quality&quot;
min=&quot;10&quot; max=&quot;63&quot; value=&quot;10&quot;
onchange=&quot;try{fetch(document.location.origin+&#39;/control?var=quality&amp;val=&#3
9;+this.value);}catch(e){}&quot;&gt;
&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;

&lt;/table&gt;
&lt;/section&gt;
&lt;/section&gt;
&lt;script&gt;
document.addEventListener(&#39;DOMContentLoaded&#39;,function(){function b(B){let
C;switch(B.type){case&#39;checkbox&#39;:C=B.checked?1:0;break;case&#39;range&#39;:cas
e&#39;select-
one&#39;:C=B.value;break;case&#39;button&#39;:case&#39;submit&#39;:C=&#39;1&#39;;br
eak;default:return;}const
D=`${c}/control?var=${B.id}&amp;val=${C}`;fetch(D).then(E=&gt;{console.log(`request to
${D} finished,
status: ${E.status}`)})}var c=document.location.origin;const
e=B=&gt;{B.classList.add(&#39;hidden&#39;)},f=B=&gt;{B.classList.remove(&#39;hidden&#
39;)},g=B=&gt;{B.classList.add(&#39;disabled&#39;)
,B.disabled=!0},h=B=&gt;{B.classList.remove(&#39;disabled&#39;),B.disabled=!1},i=(B,C,D)
=&gt;{D=!(null!=D)||D;let
E;&#39;checkbox&#39;===B.type?(E=B.checked,C=!!C,B.checked=C):(E=B.value,B.value=C)
,D&amp;&amp;E!==C?b(B):!D&amp;
&amp;(&#39;aec&#39;===B.id?C?e(v):f(v):&#39;agc&#39;===B.id?C?(f(t),e(s)):(e(t),f(s)):&#
39;awb_gain&#39;===B.id?C?f(x):e(x):&#39;face_rec
ognize&#39;===B.id&amp;&amp;(C?h(n):g(n)))};document.querySelectorAll(&#39;.close&#3
9;).forEach(B=&gt;{B.onclick=()=&gt;{e(B.
parentNode)}}),fetch(`${c}/status`).then(function(B){return
B.json()}).then(function(B){document.querySelectorAll(&#39;.default-
action&#39;).forEach(C=&gt;{i(C,B[C.id],!1)})});const
j=document.getElementById(&#39;stream&#39;),k=document.getElementById(&#39;stream-
container&#39;),l=document.getElementById(&#39;get-still&#39;),m=document.getElementBy
Id(&#39;toggle-
stream&#39;),n=document.getElementById(&#39;face_enroll&#39;),o=document.getElementB
yId(&#39;close-
stream&#39;),p=()=&gt;{window.stop(),m.innerHTML=&#39;Start
Stream&#39;},q=()=&gt;{j.src=`${c+&#39;:81&#39;}/stream`,f(k),m.innerHTML=&#39;Stop
Stream&#39;};l.onclick=()=&gt;{p(),j.src=`${c}/capture?_cb=${Date.now()}`,f(k)},o.onclick=(
)=&gt;{p(),e(k)},m.onc
lick=()=&gt;{const B=&#39;Stop
Stream&#39;===m.innerHTML;B?p():q()},n.onclick=()=&gt;{b(n)},document.querySelectorAll
(&#39;.default-
action&#39;).forEach(B=&gt;{B.onchange=()=&gt;b(B)});const
r=document.getElementById(&#39;agc&#39;),s=document.getElementById(&#39;agc_gain-
group&#39;),t=document.getElementById(&#39;gainceiling-

group&#39;);r.onchange=()=&gt;{b(r),r.checked?(f(t),e(s)):(e(t),f(s))};const
u=document.getElementById(&#39;aec&#39;),v=document.getElementById(&#39;aec_value-
group&#39;);u.onchange=()=&gt;{b(u),u.checked?e(v):f(v)};const
```

```
w=document.getElementById(&#39;awb_gain&#39;),x=document.getElementById(&#39;wb_
mode-
group&#39;);w.onchange=()=&gt;{b(w),w.checked?f(x):e(x)};const
y=document.getElementById(&#39;face_detect&#39;),z=document.getElementById(&#39;face
_recognize&#39;),A=docum
ent.getElementById(&#39;framesize&#39;);A.onchange=()=&gt;{b(A),5&lt;A.value&amp;&a
mp;(i(y,!1),i(z,!1))},y.onchange=()=&gt;
{return 5&lt;A.value?(alert(&#39;Please select CIF or lower resolution before enabling this
feature!&#39;),void
i(y,!1)):void(b(y),!y.checked&amp;&amp;(g(n),i(z,!1)))},z.onchange=()=&gt;{return
5&lt;A.value?(alert(&#39;Please select
CIF or lower resolution before enabling this feature!&#39;),void
i(z,!1)):void(b(z),z.checked?(h(n),i(y,!0)):g(n))}});
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;
)rawliteral&quot;;
static esp_err_t index_handler(httpd_req_t *req){
httpd_resp_set_type(req, &quot;text/html&quot;);
return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}
void startCameraServer()
{
httpd_config_t config = HTTPD_DEFAULT_CONFIG();
httpd_uri_t index_uri = {
.uri = &quot;/&quot;,
.method = HTTP_GET,
.handler = index_handler,
.user_ctx = NULL
};
httpd_uri_t status_uri = {
```

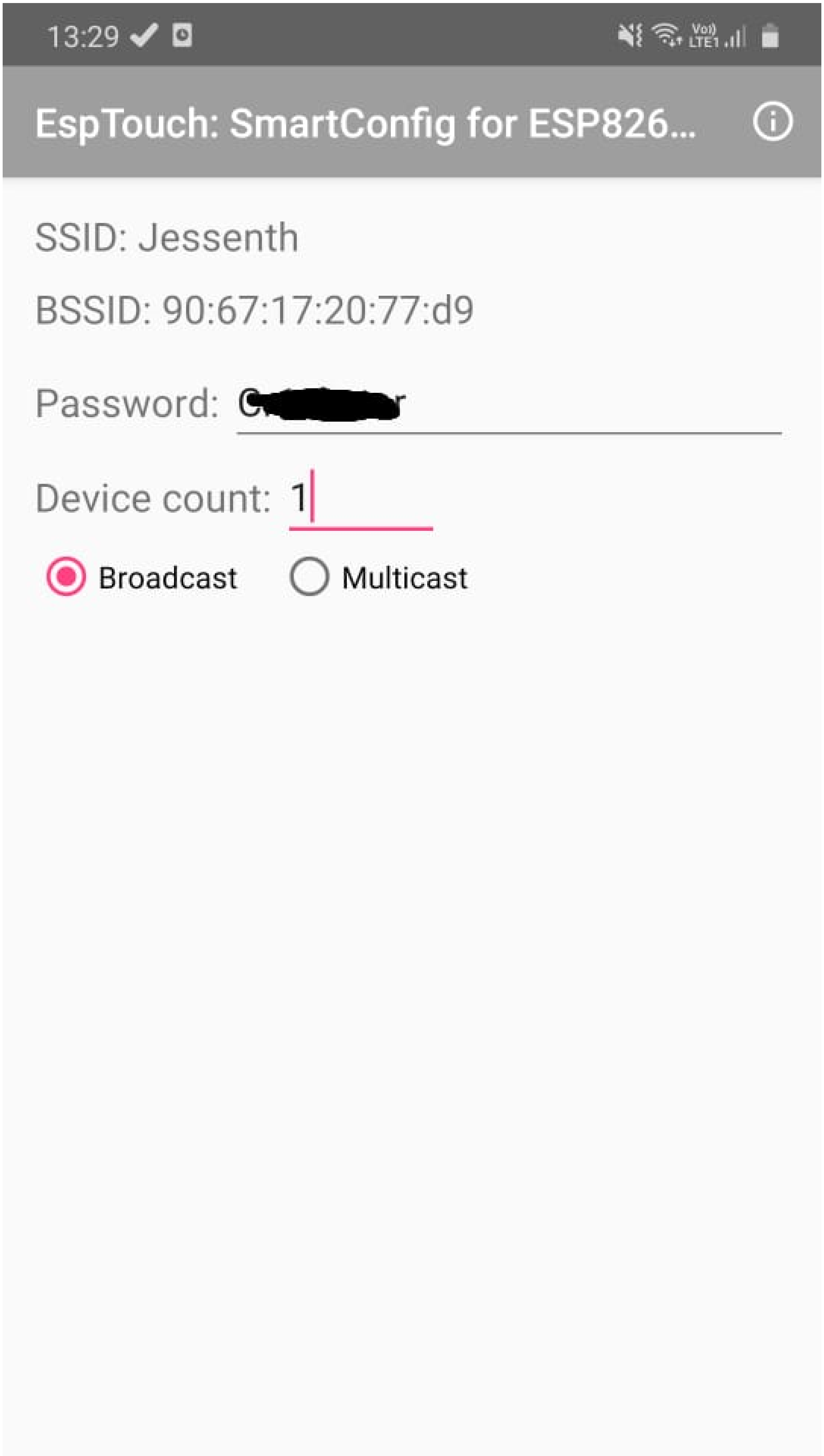
```
.uri = &quot;/status&quot;,
.method = HTTP_GET,
.handler = status_handler,
.user_ctx = NULL
};
httpd_uri_t cmd_uri = {
.uri = &quot;/control&quot;,
.method = HTTP_GET,
.handler = cmd_handler,
.user_ctx = NULL
};
httpd_uri_t capture_uri = {
.uri = &quot;/capture&quot;,
.method = HTTP_GET,
.handler = capture_handler,
.user_ctx = NULL

};
httpd_uri_t stream_uri = {
.uri = &quot;/stream&quot;,
.method = HTTP_GET,
.handler = stream_handler,
.user_ctx = NULL
};
Serial.printf(&quot;Starting web server on port: &#39;%d&#39;\n&quot;, config.server_port);
if (httpd_start(&amp;camera_httpd, &amp;config) == ESP_OK) {
httpd_register_uri_handler(camera_httpd, &amp;index_uri);
httpd_register_uri_handler(camera_httpd, &amp;cmd_uri);
httpd_register_uri_handler(camera_httpd, &amp;status_uri);
httpd_register_uri_handler(camera_httpd, &amp;capture_uri);
}
```

```
config.server_port += 1;
config.ctrl_port += 1;
Serial.printf(&quot;Starting stream server on port: &#39;%d&#39;\n&quot;, config.server_port);
if (httpd_start(&amp;stream_httpd, &amp;config) == ESP_OK) {
httpd_register_uri_handler(stream_httpd, &amp;stream_uri);
}
}
```

3.2 SCREENSHOTS



CSE2006 Project | Jessenth and Tejas | Wifi Based Surveillance Robot



EspTouch: SmartConfig for ESP826...

SSID: Jessenth

BSSID: 90:67:17:20:77:d9

Password: [redacted]

Device count: 1

● Broadcast ○ Multicast

4.0CONCLUSION AND FUTURE ENHANCEMENT

Making this project was a challenging task for us mainly debugging and implementing the C program code to perform the operations on the vehicle.We have to import a lot of other files and make sure the connection is established between the app or through the internet as well which is used to control the device.We can add more features to the project to make it more attractive and impressive.It will have a great demand in the future in various fields.It can be enhanced through better upgraded software and hardware components.The concept of this project can be used in other applications as well.

REFERENCE

P. Rai and M. Rehman, "ESP32 Based Smart Surveillance System," *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1-3, doi: 10.1109/ICOMET.2019.8673463.

H. Lee, W. Lin, C. Huang and Y. Huang, "Wireless indoor surveillance robot," *SICE Annual Conference 2011*, 2011, pp. 2164-2169.

BIODATA:-

NAME – Tejas.V
MOBILE NO :-9384855755
EMAIL:-tejas.v2019@vitstudent.ac.in

NAME – Jessenth Ebenezer
MOBILE NO :-8056187842
EMAIL:-jessenth.ebenezer2019@vitstudent.ac.in