# **HOW IT WORKS**

Firstly, power on both ESP8266 and the ESP32 CAM, during the setup process the ESP32 CAM sends it’s IP address to the server after connecting to a WiFi network of name “Robotcar” and password “Robotcar” using the “WiFi.h” Arduino library.

***Connect to WiFi***

#include "WiFi.h"

// WiFi Credentials

const char\* ssid = "Robotcar";

const char\* password = "Robotcar";

void setup() {

  Serial.begin(115200); // begin serial communication, for debugging

  // connect to WiFi

  WiFi.begin(ssid, password);

  Serial.println("");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println("");

  IP = WiFi.localIP().toString();

  Serial.println("IP address: " + IP);

  index\_html.replace("server\_ip", IP);

  server.begin(); // begin streaming server on the ESP IP Address

  configCamera(); // configure the camera settings

}

Now in the loop function (the function that runs continuously on the ESP32 CAM), the requestSettings() function is called to get the settings on the server using a get request at the same time, send the IP address as a parameter of the GET request

***Inside the loop function***

void loop() {

  // Serial.println(requestSettings());

  settings = requestSettings();

  processSettings(settings);

The requestSettings function creates an HTTPClient of http, to make a GET request to the server using the ”HTTPClient.h” Arduino library

***The requestSettings() function***

#include "HTTPClient.h"

String requestSettings() {

  String payload; // response from the server

  // Create the HTTP client

  HTTPClient http;

  // Set the URL for the request

  String url = "http://robotcar.000webhostapp.com/settings.php?IP="+IP;

  // String url = "http://192.168.109.138/robocar/settings.php?IP="+IP;

  // Send the GET request

  http.begin(url);

  int httpCode = http.GET();

  // Check the response status code

  if (httpCode > 0) {

    // Get the response payload

    payload = http.getString();

    Serial.println(payload);

  } else {

    Serial.println("Error: " + http.errorToString(httpCode));

    payload = http.errorToString(httpCode);

  }

  // Close the connection

  http.end();

  return payload;

}

This function sends the IP address as a parameter in the GET Request to settings.php page in the server and returns a payload of all the necessary settings required to run the ESP32 CAM.

In the “settings.php” file in the server establishes a connection to the database using the “./database\_connect.php” file in the server and handles the get request

***Setting.php file***

<?php

    include\_once "./database\_connect.php";

    if (isset($\_GET["IP"])) {

        $id = 1;

        $IP = $\_GET['IP']; // get the IP address sent from the ESP32 CAM

        $sql = "UPDATE `ip` SET `address` = '".$IP."', `datetime` = CURRENT\_TIMESTAMP WHERE `ip`.`id` = ".$id.";";

        // save the IP address in the database and the time it was sent

        mysqli\_query($connect, $sql);

        // run the query with the $connect variable established in database\_connect.php

        // sql query

        $sql = "SELECT \* FROM `robocar\_datastream` WHERE id = ?;";

        // select the required data from the database the ESP32 CAM needs to function

        $stmt = mysqli\_stmt\_init($connect); // create prepared statement

        if (!mysqli\_stmt\_prepare($stmt, $sql)) { // run prepared statement

            echo 'Query Failed';

        } else {

            // Bind parameters to the placeholder

            mysqli\_stmt\_bind\_param($stmt, "s", $id);

            // run parameters inside database

            mysqli\_stmt\_execute($stmt);

            $result = mysqli\_stmt\_get\_result($stmt);

            while ($row = mysqli\_fetch\_assoc($result)) {

                # code...

                $mode = $row['mode']; // variable to indicate the mode of operation

                $flash = $row['flash']; // varable to control the flashlight during streaming

                $motionDetected = $row['motion\_detected']; // varaible to store motion detected status

                $tolerance = $row['tolerance']; // variable to determine the tolerance level of the motion sensor

                // echo $mode."#".$row."#".$motion\_detected;

                echo $mode."#".$flash."&".$motionDetected.'@'.$tolerance; // send all these variables as a payload

            }

        }

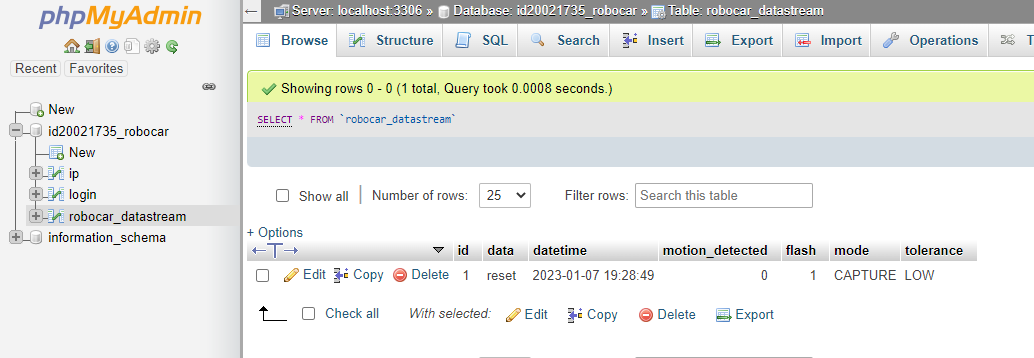
    } else {

        echo 'invalid back door';

    }

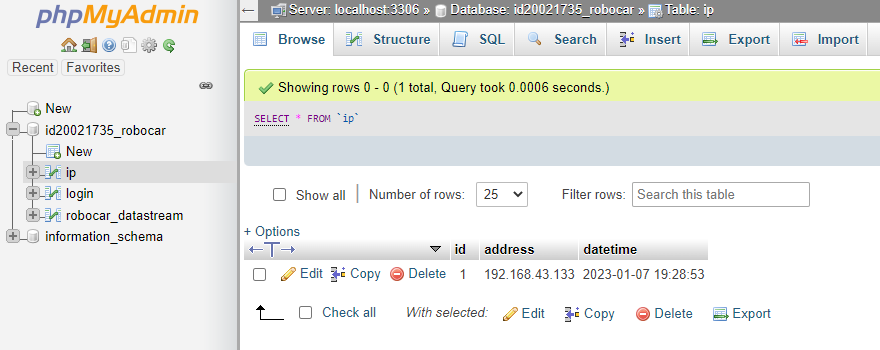
?>

The data selected from the robocar\_datatstream table in the database are “mode”, “flash”, “tolerance” and “motion\_detected”.

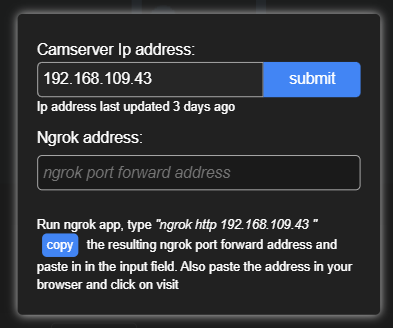


The variables “motion\_detected” and “flash” can each have two possible values of 0 or 1, for the “flash” 1 meaning ON and 0 meaning OFF, while for “motion\_detected”, 1 means motion was detected and 0 indicates the opposite. The variable “mode”, can have two possible values which are “STREAM” and “CAPTURE” while the variable tolerance can be any of three possible values, “LOW”, “AVERAGE” and “HIGH”.

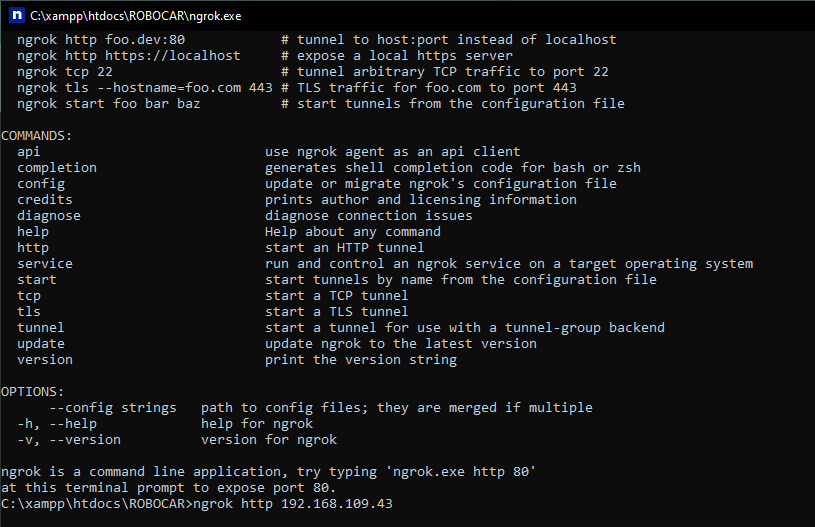
The IP address is updated in a different table in the database



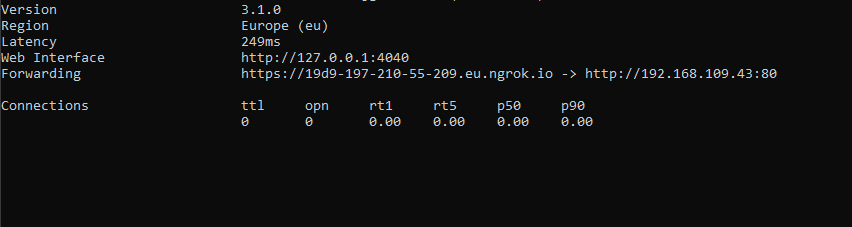
After the IP address has been updated and a user logs in, there is a prompt from the user to input the ngrok port forwarded link in an input field



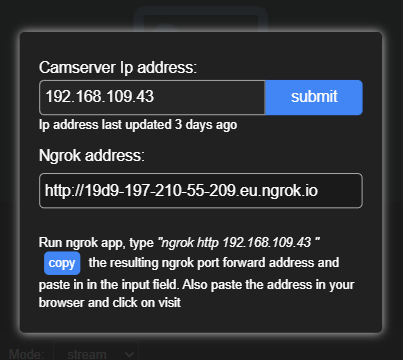
Click on copy, and open the ngrok app paste the copy text to forward the IP address of the camera



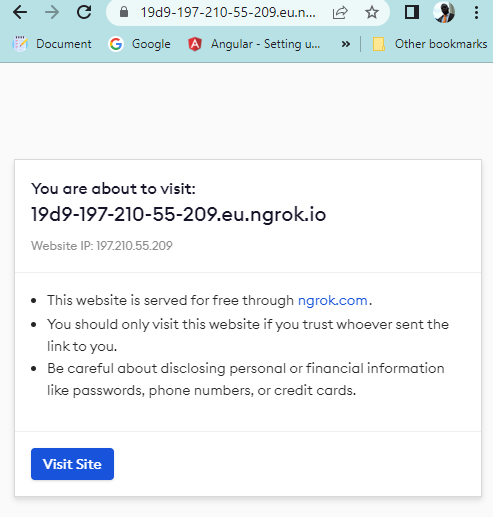
Press enter and the IP address would be forwarded to a link over the internet



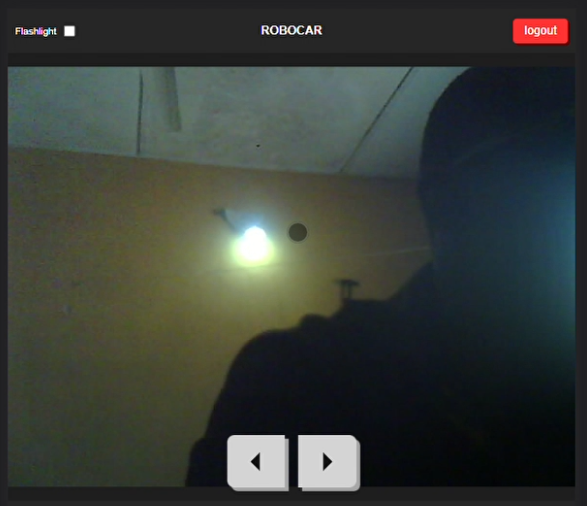
Copy the ngrok address and paste on the ngrok prompt in the website



Before clicking on submit, open the ngrok address on a different tab and click on visit



Now go back to the main website and click on visit, the stream should start right away



For the ESP8266, it connects to the same WiFi network, and makes a request to ”rx.php” file in the server

***The ESP8266 setup code***

#include "ESP8266WiFi.h"

#include "ESP8266HTTPClient.h"

#include "WiFiClient.h"

#include <Wire.h>

const int SLAVE\_ADDRESS = 8; // the slave address

const char\* ssid = "Robotcar";

const char\* password = "Robotcar";

String host = "http://robotcar.000webhostapp.com";

const int httpPort = 80;

String dataFromServer; //data to be sent to ARDUINO UNO via I2C

String mode; // mode switch connected to D0 of ESP8266

String prevMode; // variable to detect switch in operation mode

int motion; // variable to store motion detected state and send it to server

String payload; // data from server

void setup() {

  Serial.begin(115200); // begin serial communication for debugging

  WiFi.begin(ssid, password); // connect to WiFi network

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to the WiFi network");

  Wire.begin(); // begin wire communication via I2C

  pinMode(D8, INPUT); // motion sensor

}

In the loop function the ESP8266 reads the signal from the PIR motion sensor to determine whether motion was detected or not and sends this information to the server and receives a payload corresponding to the user inputs

***The loop function***

void loop() {

  motion = readMotion(); // get the motion detected state from Arduino

  makeGetRequest(); // this function updates the values of dataFromServer and mode

  delay(1000);

}

***The readMotion() function***

int readMotion(){

  int motionValue = digitalRead(D7);

  Serial.print("motion: ");

  Serial.println(motionValue);

  return motionValue;

}

***The makeGetRequest() function***

void makeGetRequest() {

  WiFiClient client;

  HTTPClient http;

  Serial.println("inside get request");

  String motionString = String(motion);

  String page = "rx.php?id=1&motion=0";

  String url = host + "/" + page;

  // String url = "http://robotcar.000webhostapp.com/rx.php?id='1'&motion='0'";

  if (http.begin(client, url)) {

    Serial.print("GET...");

    int httpCode =  http.GET();

    if (httpCode > 0) {

      Serial.printf("[HTTP] GET... code: %d", httpCode);

      Serial.println("");

        // if request is successful

      if (httpCode == HTTP\_CODE\_OK || httpCode == HTTP\_CODE\_MOVED\_PERMANENTLY) {

        payload = http.getString();

        Serial.print("payload: ");

        // find the index of th '#' character

        int delimiterIndex = payload.indexOf('#');

        dataFromServer = payload.substring(0, delimiterIndex);,

        mode = payload.substring(delimiterIndex + 1);

        Serial.println(payload);

      }

    }

    http.end();

    sendData(payload);

  } else {

    Serial.printf("[HTTP] Unable to connect");

  }

}

The payload returned from the server indicates the user inputs and the mode of operation, the user input is selected from the ”robocar\_datastream” in the “rx.php” file

***The rx.php file***

<?php

    include\_once "./database\_connect.php";

    if (isset($\_GET["id"])) {

        $id = $\_GET['id'];

        $motion = $\_GET['motion'];

        // sql query

        $sql = "SELECT `data`, `datetime`, `mode` FROM `robocar\_datastream` WHERE `id` = 1;";

        // run query

        $result = mysqli\_query($connect, $sql);

        while ($row = mysqli\_fetch\_assoc($result)) {

            # code...

            $data = $row['data'];

            $dateTime = $row['datetime'];

            $mode = $row['mode'];

            $dateTime = strtotime($dateTime) + 3600;

            $unixSeconds = date('U');

            $time = time();

            $timeElapsed = $unixSeconds - $dateTime;

            if ($timeElapsed > 5) {

                echo 'N/A#'.$mode;

            } else {

                echo $data.'#'.$mode;

            }

        }

        $sql = "UPDATE `robocar\_datastream` SET `motion\_detected` = '".$motion."' WHERE `robocar\_datastream`.`id` = 1;";

        mysqli\_query($connect, $sql);

    } else {

        # code...

        echo 'invalid back door';

    }

?>

The user input is sent to the ESP8266 if its not more than 5 seconds hold since the button was pressed, this can be determined because a timestamp is saved in the database. The user input is sent to the database using a combination of Ajax and PHP

The HTML of the buttons, all the buttons have a data variable to store the direction value

***The HTML***

<form class="car\_forward">

<button data-direction="car\_forward" id="forward" type="button" class="car-buttons">

<span class="up-arrow"></span>

</button>

</form>

<form class="car\_backward">

<button data-direction="car\_backward" id="backward" type="button" class="car-buttons">

<span class="down-arrow"></span>

</button>

</form>

***AJAX JQuery***

  let intervalId;

  // function to run on press and hold of a button

  $('.car-buttons, .camera-buttons, .reset-camera').on('mousedown touchstart', function () {

    if (getCookie("mode") === "STREAM") {

      let direction = $(this).data("direction");

      intervalId = setInterval(function() {

        // Do something on each iteration of the loop

        // let direction = this;

        console.log(direction);

        $.ajax({

          type: 'POST',

          url: 'update\_db.php',

          data: { data: direction },

          success: function(response) {

            // code to handle the response from the server

          }

        });

      }, 250);

    }

  });

  // function to run on press and hold of a button

  $('.car-buttons, .camera-buttons, .reset-camera').on('mouseup touchend', function () {

    clearInterval(intervalId);

  });

If the user press and hold any of the direction button make a POST request to update\_db.php to update the database

***Update\_db.php file***

<?php

    include\_once "./database\_connect.php";

    if (isset($\_POST['data'])) {

        # code...

        $data = $\_POST['data'];

        $sql = "UPDATE `robocar\_datastream` SET `data` = '".$data."', `datetime` = CURRENT\_TIMESTAMP WHERE `robocar\_datastream`.`id` = 1;";

        mysqli\_query($connect, $sql);

        echo 'Car Direction: ';

        echo $data;

        echo '<br>';

        echo 'successful';

    } else{

        # code...

        header("location: index.php?successful");

    }

    header("location: index.php?successful");

?>

With knowledge of how the user inputs get to the ESP8266, the ESP8266 sends the data/payload via I2C to the Arduino Uno microcontroller for processing using the sendData() function

***The sendData() function***

// called when data is requested by the master

void sendData(String data) {

  Wire.beginTransmission(SLAVE\_ADDRESS);

  Wire.write(data.c\_str()); // send the data as a character array

  Wire.endTransmission();

  Serial.println("Sent: " + data); // print the sent data

}

The Arduino receives this data with the receiveData() function from the ESP8266 and processes it

***The receiveData() function***

// called when data is received from the master

void receiveData(int numBytes) {

  String data = "";

  while (Wire.available()) { // loop through all received bytes

    data += (char)Wire.read(); // add the byte to the string

  }

  Serial.println("Received: " + data); // print the received data

  dataFromESP = data;

}

The function is declared in the setup section of the Arduino code

void setup() {

  Serial.begin(115200); // start serial for output

  pinMode(modeLed, OUTPUT); // declare pinmode for mode led indicator

  // pins for motor

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(in3, OUTPUT);

  pinMode(in4, OUTPUT);

  myservo.attach(9);  // attaches the servo on pin 10 to the servo object

  myservo.write(servoAngle);  // sets the servo position according to the scaled value

  //Begin serial communication with Arduino and ESP8266

  // serial.begin(115200);

  Wire.begin(SLAVE\_ADDRESS); // join the I2C bus as a slave

  Wire.onReceive(receiveData); // register the receive callback

  //Begin serial communication with Arduino and SIM800L

  mySerial.begin(9600);

}

The loop function of the Arduino Uno processes the data and executes the necessary user input

**The loop() function**

void loop() {

  // dataFromESP = receiveString();

  getData(); // get dataFromServer and Mode

  indicateMode();

  // use dataFromServer

  if (dataFromServer == "N/A") { moveStop(); }

  else if (dataFromServer == "car\_forward") { moveForward(); }

  else if (dataFromServer == "car\_backward") { moveBackward(); }

  else if (dataFromServer == "car\_left") { moveLeft(); }

  else if (dataFromServer == "car\_right") { moveRight(); }

  else if (dataFromServer == "cam\_left") { moveStop(); camLeft(); }

  else if (dataFromServer == "cam\_right") { moveStop(); camRight(); }

  else if (dataFromServer == "reset") { moveStop(); camCenter(); }

  else if (dataFromServer == "captured\_image") {

    moveStop(); sendTextMessage(phoneNumber, message);

  } else { delay(1000); }

  Serial.println(dataFromServer);  // print the received data to the serial monitor

  // loop delay comes from respective functions

}