* <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROBE 24</title>

<style>

body {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

min-height: 100vh;

margin: 0;

}

div {

text-align: center;

}

#gamepad {

margin-top: 40px; /\* Increased margin for spacing \*/

margin-bottom: 40px; /\* Added margin at the bottom for further spacing \*/

}

#gamepad a {

margin: 5px;

padding: 10px;

color: #000;

border: 1px solid #000;

}

a {

text-decoration: none;

}

</style>

</head>

<body>

<div>

<h1>Probe</h1>

<a href="/1"> <button>Gamepad</button> </a>

<a href="/2"> <button>Hand Following</button> </a>

<a href="/3"> <button>Collision Avoidance</button> </a>

</div>

<div id="gamepad">

<a href="/up">▲</a>

<div style="margin: 30px;">

<a href="/left">◄</a>

<a href="/stop">◯</a>

<a href="/right">►</a>

</div>

<a href="/down">▼</a>

</div>

</body>

</html>

* <img

class="img-qualcomm"

style="position: absolute; top: 0; center: 0; width: 100px"

src="https://i.postimg.cc/rw80qzNp/512px-Qualcomm-Logo-svg.png"

alt="512px-Qualcomm-Logo-svg"

/>

<img

class="img-nitt"

style="position: absolute; top: 0; left: 0; width: 50px"

src="https://i.postimg.cc/630QM43Z/NITT.png"

/>

<img

class="img-probe"

style="position: absolute; top: 0; right: 0; width: 50px"

src="https://i.postimg.cc/s2Pn03RS/probe-nit-trichy-logo.jpg"

/>

* #include <ESP8266WiFi.h>

#include <Servo.h>

#define DISABLE\_SERIAL

const char\* ssid = "narzo 50";

const char\* password = "brokenow";

WiFiServer server(80);

Servo servo;

int servo\_delay = 500;

int turn\_delay = 125;

int mode=1; // mode1 is gamepad, 2 is hand following, 3 is collision avoidance

int inputDirection = 0; // 0 is for stop, 1 is for forward, 2 is for back, 3 is for left, 4 is for right

int speed=16; //D0

int speed2 = 5; //D1

int in1= 4; //D2

int in2= 0; // D3

int in3= 14; //D8

int in4= 15; // D5

int trig=12; //D6

int echo=13; //D7

int servo\_pin = 2; // D4

int leftir=1; // tx

int rightir=3; // rx

int rightdis=0, leftdis=0, middis=0;

void webpage(WiFiClient client){

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("");

client.println("<!DOCTYPE html>");

client.println("<html lang=\"en\">");

client.println("<head>");

client.println("<meta charset=\"UTF-8\">");

client.println("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\">");

client.println("<title>Probe 24</title>");

client.println("<style>");

client.println("body {");

client.println("display: flex;");

client.println("flex-direction: column;");

client.println("align-items: center;");

client.println("justify-content: center;");

client.println("min-height: 100vh;");

client.println("margin: 0;");

client.println("}");

client.println("div {");

client.println("text-align: center;");

client.println("}");

client.println("#gamepad {");

client.println("margin-top: 20px;");

client.println("}");

client.println("#gamepad a {");

client.println("margin: 5px;");

client.println("padding: 10px;");

client.println("color: #000;");

client.println("border: 1px solid #000;");

client.println("}");

client.println("a {");

client.println("text-decoration: none;");

client.println("}");

client.println("</style>");

client.println("</head>");

client.println("<body>");

client.println("<div>");

client.println("<h1>Probe</h1>");

client.println("<a href=\"/mode1\"> <button>Gamepad</button> </a>");

client.println("<a href=\"/mode2\"> <button>Hand Following</button> </a>");

client.println("<a href=\"/mode3\"> <button>Collision Avoidance</button> </a>");

client.println("</div>");

client.println("<div id=\"gamepad\">");

client.println("<a href=\"/up\">▲</a>");

client.println("<div style=\"margin: 30px;\">");

client.println("<a href=\"/left\">◄</a>");

client.println("<a href=\"/stop\">◯</a>");

client.println("<a href=\"/right\">►</a>");

client.println("</div> ");

client.println("<a href=\"/down\">▼</a>");

client.println("</div>");

client.println("</body>");

client.println("</html>");

}

void forward(){

digitalWrite(speed,HIGH);

digitalWrite(speed2,HIGH);

digitalWrite(in1,HIGH);

digitalWrite(in3,HIGH);

digitalWrite(in2,LOW);

digitalWrite(in4,LOW);

Serial.println("Forward");

}

void backward(){

digitalWrite(speed,HIGH);

digitalWrite(speed2,HIGH);

digitalWrite(in2,HIGH);

digitalWrite(in4,HIGH);

digitalWrite(in1,LOW);

digitalWrite(in3,LOW);

Serial.println("backward");

}

void rightturn(){

digitalWrite(speed,HIGH);

digitalWrite(speed2,HIGH);

digitalWrite(in1,LOW);

digitalWrite(in3,HIGH);

digitalWrite(in2,HIGH);

digitalWrite(in4,LOW);

Serial.println("right");

}

void leftturn(){

digitalWrite(speed,HIGH);

digitalWrite(speed2,HIGH);

digitalWrite(in1,HIGH);

digitalWrite(in3,LOW);

digitalWrite(in2,LOW);

digitalWrite(in4,HIGH);

Serial.println("left");

}

void stop(){

digitalWrite(speed, LOW);

digitalWrite(speed2,LOW);

}

int distance(){

digitalWrite(trig,LOW);

delayMicroseconds(10);

digitalWrite(trig, HIGH);

delayMicroseconds(10);

digitalWrite(trig,LOW);

int tt=pulseIn(echo, HIGH);

int d =(0.0343 \* tt) / 2; // speed = 0.0343 cm/microsecond

return d;

}

void readInput(String request){

if (request.indexOf("stop") != -1){

inputDirection=0;

}

if (request.indexOf("mode1") != -1){

mode=1;

}

else if (request.indexOf("mode2") != -1){

mode =2;

}

else if (request.indexOf("mode3") != -1){

mode = 3;

}

if(mode==1){

if (request.indexOf("up") != -1){

inputDirection=1;

}

else if (request.indexOf("down") != -1){

inputDirection=2;

}

else if (request.indexOf("left") != -1){

inputDirection=3;

}

else if (request.indexOf("right") != -1){

inputDirection=4;

}

}

}

void handFollow(){

int leftSensor = digitalRead(leftir); // Read sensor values directly

int rightSensor = digitalRead(rightir);

if (leftSensor == LOW && rightSensor == LOW) {

int action = distance();

if ((action <=7) && (action >=2)) {

forward();

}

else if(action <=2) {

stop();

}

}

else if (leftSensor == LOW) { //active low logic, when it detects the presence of hand, the digital output given by IR isLOW

leftturn();

}

else if (rightSensor == LOW) {

rightturn();

}

else {

stop();

}

}

void collisonAvoidance(){

servo.write(90); //Align the motor such that ultrasonic sensor mounted on it faces frontside

delay(servo\_delay);

middis=distance();

Serial.println(middis);

if(middis <=10){ //when obstacle is close to bot

stop();

delay(servo\_delay);

servo.write(30);

delay(servo\_delay);

rightdis=distance();

Serial.println(rightdis);

delay(servo\_delay);

servo.write(150);

delay(servo\_delay);

leftdis=distance();

Serial.println(leftdis);

delay(servo\_delay);

servo.write(90);

delay(servo\_delay);

if((rightdis>10) || (leftdis>10)){

if(rightdis > leftdis){

rightturn();

delay(turn\_delay);

}

if(leftdis > rightdis){

leftturn();

delay(turn\_delay);

}

}

}

else{

forward();

}

}

void move(){

switch(inputDirection){

case 0:

stop();

break;

case 1:

forward();

break;

case 2:

backward();

break;

case 3:

leftturn();

break;

case 4:

rightturn();

break;

default:

inputDirection=0;

stop();

}

}

void setup() {

// comment for ip address

#ifdef DISABLE\_SERIAL

Serial.end(); // Disable serial communication to release the TX pin

#endif

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

server.begin();

delay(3000);

pinMode(in3, OUTPUT);

pinMode(trig, OUTPUT);

pinMode(echo, INPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in4, OUTPUT);

pinMode(speed,OUTPUT); //l293d driver had only one common enable pin

pinMode(speed2, OUTPUT);

pinMode(leftir, INPUT);

pinMode(rightir, INPUT);

servo.attach(servo\_pin,600,2400);

servo.write(0);

}

void loop() {

WiFiClient client;

client = server.available();

switch(mode){

case 1:

move();

break;

case 2:

handFollow();

break;

case 3:

collisonAvoidance();

break;

default:

inputDirection=0;

stop();

break;

}

if (client == 1) {

String request = client.readStringUntil('\n');

Serial.println(request);

client.flush();

readInput(request);

webpage(client);

}

}