Código do Software

**Trabalho feito por:**

* **Fernando Ventura**
* **Murilo Motta**
* **Pedro Egydio**
* **Rafael Nascimento**

Link do repositorio github: https://github.com/EgydioLucas/Objetos-inteligentes-

#include <ESP8266WiFi.h>

const char\* ssid = "DistribuidoraMalta";

const char\* password = "malta2013";

WiFiServer server(80);

#include <Ultrasonic.h>

#define pino\_trigger D4

#define pino\_echo D5

Ultrasonic ultrasonic(pino\_trigger, pino\_echo);

void setup() {

Serial.begin(115200);

delay(10);

// Connect to WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Start the server

server.begin();

Serial.println("Server started");

// Print the IP address

Serial.print("Use this URL : ");

Serial.print("http://");

Serial.print(WiFi.localIP());

Serial.println("/");

}

void loop() {

// Check if a client has connected

WiFiClient client = server.available();

if (!client) {

return;

}

// Wait until the client sends some data

Serial.println("new client");

while(!client.available()){

delay(1);

}

// Read the first line of the request

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

Serial.println(request);

client.flush();

// Return the response

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

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

client.println(""); // do not forget this one

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

client.println("<html>");

float distancia;

client.print("<h1><center>Bem vindo a trena eletronica! </h1></center>");

client.println("<br><br>");

client.println("<form action='' method='GET'>");

client.println("Distancia desejada: <input type='text' value='$distancia'cm>");

client.println("<br><br>");

client.println("<input type='submit' value='Enviar'>");

client.println("<br><br>");

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

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

client.println();

float diferenca;

float cmMsec;

long microsec = ultrasonic.timing();

cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);

client.print("<b>Distancia que voce esta do objeto:</b> ");

client.print(cmMsec);

client.print(" cm");

client.println("<br><br>");

diferenca = distancia - cmMsec ;

client.print("<b>A distancia desejada pelo usuario: </b>");

client.print(distancia);

client.print(" cm");

client.println("<br><br>");

client.println("<br><br>");

client.print("<b>A diferenca entre a distancia informada pelo usuario e a distancia real: </b>");

client.print(diferenca);

client.print(" cm");

client.println("<br><br>");

client.println("<br><br>");

if (diferenca+1 > 0){

client.print("<b><h2>Aproxime-se</h1></b>");

}

else if (diferenca+1 < 0){

client.print("<b><h2>Distancie-se</h2></b>");

}

else {

client.print("<b><h2>A distancia esta correta</h2></b>");

}

delay(1);

Serial.println("Client disconnected");

Serial.println("");

}