/\*\*\*\*\*\*\*

All the resources for this project:

https://www.hackster.io/Aritro

\*\*\*\*\*\*\*/

#include <SPI.h>

#include <Ethernet.h>

int Led = 6;

int MQ2 = A5;

int valor = 0;

// Your threshold value

int sensorThres = 400;

void(\* Resetea) (void) = 0;//Funcíon Reset por soft para el arduino (como si apretaramos el botón reset)

// Introduzca una direcciÛn MAC y la direcciÛn IP para el controlador

byte mac[] = {

0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress ip(172,16,48,121); // Esta direccion IP debe ser cambiada obligatoriamente

// dependiendo de la subred de su Area Local y es la que

// usara para conectarse por el Navegador.

EthernetServer server(80); // Puerto 80 por defecto para HTTP

void setup() {

Ethernet.begin(mac, ip); //inicializa la conexiÛn Ethernet y el servidor

server.begin();

pinMode(Led, OUTPUT);

pinMode(MQ2, INPUT);

Serial.begin(9600);

}

void loop() {

int analogSensor = analogRead(MQ2);

Serial.print("Pin A0: ");

Serial.println(analogSensor);

// Checks if it has reached the threshold value

if (analogSensor > sensorThres)

{

digitalWrite(Led, LOW);

tone(Led, 1000, 200);

}

else

{

digitalWrite(Led, HIGH);

noTone(Led);

}

EthernetClient cliente = server.available(); // Inicializa cliente como servidor ethernet

if (cliente) {

boolean currentLineIsBlank = true;

while (cliente.connected()) {

if (cliente.available()) {

char c = cliente.read();

if (c == '\n' && currentLineIsBlank) {

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

cliente.println("Content-Type: text/html"); // Envia el encabezado en codigo HTML estandar

cliente.println("Connection: close");

cliente.println("Refresh: 3"); // refresca la pagina automaticamente cada 3 segundos

cliente.println();

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

cliente.println("<html>");

cliente.println("<HEAD>");

cliente.println("<TITLE>Detector de gas</TITLE>");

cliente.println("</HEAD>");

cliente.println("<body bgcolor=skyblue>");

cliente.println("<hr />");

cliente.println("<center>");

cliente.println("<H1><b><i>Bienvenido al sistema detector de gas</b></i></H1>");

cliente.println("<br />");

cliente.println("<h2><p>Integrantes del grupo:");

cliente.println("<br />");

cliente.println("Saul Antonio Flores Rosales");

cliente.println("<br />");

cliente.println("Rodrigo Salomon Linares Anaya");

cliente.println("<br />");

cliente.println("Nehemias Vladimir Campos Menjivar");

cliente.println("<br />");

cliente.println("Cristopher Alejandro Lara Gonzalez</p></h2>");

cliente.println("<br />");

cliente.println("<h2>Monitorea cuando hay alguna fuga de gas</h2>");

cliente.println("<br />");

cliente.println("<h3><p>Mostrando un mensaje de advertencia cuando se detecta una fuga<br></h3>");

cliente.println("<br />");

cliente.println("<h3>Si no hay fuga se muestra un mensaje diciendo que todo se encuentra con normalidad</p></h3>");

cliente.println("</center>");

cliente.println("<br />");

cliente.println("<br />");

valor = digitalRead(MQ2); // leemos el valor de entrada

/\*int analogSensor = analogRead(MQ2);

Serial.print("Pin A0: ");

Serial.println(analogSensor);

// Checks if it has reached the threshold value

if (analogSensor > sensorThres)

{

digitalWrite(Led, LOW);

tone(Led, 1000, 200);

}

else

{

digitalWrite(Led, HIGH);

noTone(Led);

} \*/

for (int puertoAnalogo = 0; puertoAnalogo < 1; puertoAnalogo++) {

int lecturaSensor = digitalRead(MQ2); // Lee los 6 puertos analogos de A0 a A5

cliente.println("<center>");

cliente.print("<H2>Monitoreo de fuga en tiempo real:</H2>");

cliente.println("</center>");

if (analogSensor > sensorThres)

{

digitalWrite(Led, LOW);

tone(Led, 1000, 200);

cliente.print("<h3>");

cliente.print("<center>");

cliente.print("PRECAUCION Hay una fuga de gas</h3>");

cliente.print("</center>");

}

else

{

digitalWrite(Led, HIGH);

noTone(Led);

cliente.print("<center>");

cliente.print("<h3>Todo se encuentra con normalidad</h3>");

cliente.print("</center>");

}

cliente.println("<br />");

}

cliente.println("<br />");

break;

}

if (c == '\n') {

currentLineIsBlank = true;

}

else if (c != '\r') {

currentLineIsBlank = false;

}

}

}

delay(10); // Da tiempo al Servidor para que reciba los datos 15ms

cliente.stop(); // cierra la conexion

}

delay(100);

}