



UNIÓN DE ASOCIACIONES
DE INGENIEROS TÉCNICOS
INDUSTRIALES Y GRADUADOS
EN LA INGENIERÍA DE LA
RAMA INDUSTRIAL DE ESPAÑA

UNIÓN DE ASOCIACIONES DE INGENIEROS TÉCNICOS INDUSTRIALES Y GRADUADOS EN INGENIERÍA DE LA RAMA INDUSTRIAL DE ESPAÑA (UAITIE)

“CONVOCATORIA 2025”

**X PREMIO NACIONAL DE INICIACIÓN A LA INVESTIGACIÓN
TECNOLÓGICA**

AirGuard
Vigilancia de calidad del aire

AUTOR/ES:

Dario Garrido Carlón
Javier Sedano Carrillo de Albornoz
Gael Giraud Moreno
Helena Martínez Sánchez
Jaime González de Castro

BLOQUE TEMÁTICO:

Urbanismo Inteligente
Robótica

NIVEL EDUCATIVO:

4º E.S.O.

COORDINADOR:

Roa Ortiz Ana
Marzo 2025

Resumen

AirGuard es una solución innovadora y ecológica diseñada para combatir el impacto negativo de la contaminación del aire en la salud. Equipado con avanzados sensores de CO₂, humedad, gotas de lluvia y gases como CO, NH₃, NO₂, este dispositivo no solo mide la calidad del aire, sino que también alerta y educa a las personas sobre su importancia.

Gracias a la placa MKR 1010 de Arduino, AirGuard opera de forma autónoma en un coche con motores controlados por una aplicación. Además, los datos obtenidos se visualizan en tiempo real en una pantalla LCD y en la nube, pudiendo acceder desde una página web habilitada para la monitorización en tiempo real.

El Internet of Things nos Brinda ventajas como la automatización del coche que puede ser controlado desde cualquier parte a partir de un navegador como puede ser Chrome, o el monitoreo gracias a sensores conectados que permiten vigilar propiedades, permitiendo a los expertos su análisis estadístico.

La baja calidad del aire supone un peligro por causar problemas respiratorios. Una exposición prolongada a un aire de baja calidad puede provocar enfermedades como cáncer de pulmón, enfermedades cardíacas e incluso paros cardíacos especialmente en pacientes pediátricos y geriátricos. Este trabajo aborda el desafío de intentar mejorar todo lo posible la calidad del aire en los espacios interiores y exteriores.

Con AirGuard, el propósito es claro: sensibilizar y contribuir a mejorar la calidad del aire, protegiendo la salud de todos, especialmente de los más vulnerables. Una herramienta esencial para un entorno más saludable y sostenible.

1 Palabras Clave

Calidad, aire, ecológico, gases, efecto invernadero, Air Guard.



2 Índice

Resumen	2
Palabras Clave.....	2
Índice.....	3
1. Desarrollo	4
1.1 Introducción.....	4
1.2 Objetivos	5
1.3 Metodología.....	6
1.4 Resultados.....	9
1.5 Conclusión.....	19
2. Figuras	20
3. Referencias	20

La calidad del aire es un factor muy importante para la salud humana, el bienestar del medio ambiente y el desarrollo sostenible de la sociedad. La presencia de gases contaminantes en el aire, como partículas finas, dióxido de nitrógeno y ozono troposférico, puede tener efectos devastadores en la salud respiratoria y cardiovascular de las personas, además de contribuir al cambio climático. Una mala calidad del aire no solo aumenta la morbilidad y mortalidad por enfermedades respiratorias, sino que también afecta a la biodiversidad, los ecosistemas y las cosechas, lo que pone en riesgo la seguridad alimentaria. Por tanto, mantener un aire limpio es esencial para proteger la vida y garantizar un entorno saludable para las generaciones presentes y futuras.

1. Desarrollo

El AirGuard ha sido diseñado con tecnología avanzada para medir la calidad del aire en tiempo real. Equipado con sensores que detectan diversos gases contaminantes, el vehículo recoge datos de la calidad del aire mientras circula. Estos datos se envían a un servidor, dónde se procesan y almacenan, permitiendo a los usuarios monitorear la calidad del aire.

1.1 Introducción

La calidad del aire tiene un gran impacto en nuestra salud, ya que la exposición a contaminantes como las partículas finas, el dióxido de nitrógeno, el ozono y el monóxido de carbono está relacionada con enfermedades como el asma, la bronquitis y problemas del corazón. Las personas más vulnerables como los niños, ancianos y aquellos con problemas de salud previos son especialmente vulnerables a los efectos negativos de un aire contaminado. Además la exposición prolongada puede aumentar el riesgo de desarrollar enfermedades crónicas, lo que afecta a la validez de vida de las personas.

Por ejemplo, en una ciudad como Madrid, el aumento de la contaminación atmosférica se debe principalmente al tráfico rodado que representa el 46,9% de las emisiones de óxido de nitrógeno. Otras fuentes incluyen otros medios de transporte

(25,5%) relacionadas con el aeropuerto de Barajas, el sector residencial (18,1%) por los sistemas de calefacción y el sector industrial (menos del 7%).

Entre los contaminantes más nocivos para la salud destacan:

- **CO₂**: Principal gas de efecto invernadero, causa acidificación oceánica y desplaza oxígeno en espacios cerrados. Intensifica olas de calor, sequías y fenómenos meteorológicos.
- **CO**: Gas tóxico que forma carboxihemoglobina, impidiendo el transporte de oxígeno; puede ser fatal. Contribuye al cambio climático al oxidarse a CO₂. Se acumula en espacios cerrados con fuentes mal ventiladas.
- **NO₂**: Emisión de vehículos, calefacciones e industrias, afecta a personas vulnerables. Forma ozono troposférico, es precursor de lluvia ácida y contribuye al cambio climático.
- **NH₃**: Emitido por tráfico, residuos y actividades agrícolas; sin variaciones estacionales en Madrid. Irrita vías respiratorias y piel, provoca inflamación pulmonar y acidifica suelos, afectando ecosistemas y biodiversidad.

Espero que sea útil. ¿Qué opinas del resumen?

1.2 Objetivos

El objetivo principal de este trabajo es la implementación de un robot que disponga de la tecnología de IOT para monitorizar la medida de gases en distintas zonas de un área. De esta forma podremos contribuir al control de la calidad del aire.

El robot va equipado con sensores avanzados de CO₂, NH₂, CO, NO₂, temperatura y humedad. Además, se ha diseñado y programado toda una página web para que actúe como sistema de análisis en tiempo real, de tal forma que todas las medidas realizadas se puedan ver desde cualquier ordenador o dispositivo móvil. El robot tiene el potencial de detectar y cuantificar contaminantes atmosféricos en diversos entornos, permitiendo la identificación temprana de áreas con altos niveles de contaminación. Esta información facilita la toma de decisiones informadas por parte de las autoridades y la ciudadanía promoviendo acciones correctivas y preventivas para reducir la exposición a agentes nocivos.

En resumen, el AirGuard se basa en la premisa de que la integración de tecnologías de monitoreo ambiental puede desempeñar un papel crucial en la mejora de la calidad del aire, contribuyendo a la salud pública y a la sostenibilidad ecológica.



1.3 Metodología

1.3.1 Fundamentos teóricos básicos

1.3.1.1 Principio de funcionamiento del sensor de CO₂

El sensor MQ-2 funciona mediante un elemento calefactor interno que calienta un material semiconductor, generalmente dióxido de estaño (SnO₂). Este material es sensible a gases inflamables y humo y su resistencia eléctrica cambia al entrar en contacto con estos gases. Este cambio en la resistencia se traduce en una señal eléctrica que puede ser procesada. El sensor tiene dos tipos de salida:

La salida analógica proporciona un voltaje proporcional a la concentración del gas detectado.

La salida digital permite configurar un umbral de detección mediante un potenciómetro activando una señal alta o baja cuando se supera dicho límite.



Figura 1: Sensor de CO₂ para Arduino.

1.3.1.2 Principio de funcionamiento del sensor de NO, NH₃ y CO

La medición de Óxido Nítrico (NO) se realiza mediante principios electroquímicos o de espectroscopia. Los sensores electroquímicos generan una corriente eléctrica proporcional al NO a través de una reacción química en su electrodo catalítico. Los de espectroscopia detectan la absorción específica del NO usando luz infrarroja o ultravioleta.

La medición de amoniaco (NH₃) se basa en principios electroquímicos. Un sensor con electrodos y conductor de iones genera una reacción de oxidación al detectar el gas, produciendo una corriente proporcional a su concentración. Esta corriente se traduce en una señal analógica o digital según el diseño del sensor.

La medición de monóxido de carbono (CO) puede realizarse mediante sensores electroquímicos o de óxido metálico. Los primeros generan una corriente eléctrica proporcional a la concentración del CO tras una reacción química entre el gas, el

electrodo y el electrolito. Los sensores de óxido metálico emplean un semiconductor, como dióxido de estaño, cuya resistencia eléctrica varía al contacto con el CO, generando una señal eléctrica que refleja su concentración en el aire

1.3.1.3 Principio de funcionamiento de los sensores de humedad.

Los sensores de humedad detectan el vapor de agua en el aire y pueden ser capacitivos, resistivos o térmicos. Los capacitivos usan un material dieléctrico cuya capacidad cambia con la humedad, generando una señal eléctrica. Los resistivos miden la resistencia eléctrica de un material, que disminuye con más humedad. Los térmicos comparan la conductividad del aire seco con la del aire húmedo.

El sensor usado por nosotros es un DHT11 que es un sensor de tipo capacitivo. Además este sensor nos dará la medida de temperatura.



Figura 2: Sensor capacitivo de humedad DHT11

1.3.1.4 Principio de funcionamiento de un puente H

El puente H es un circuito que controla la dirección de un motor DC, permitiendo que gire hacia delante o atrás al cambiar el flujo de corriente. Consta de cuatro interruptores o transistores y el motor se encuentra en el centro. Para un sentido de giro, dos interruptores dirigen la corriente de izquierda a derecha; para el inverso, se activan los otros dos, cambiando la dirección del flujo

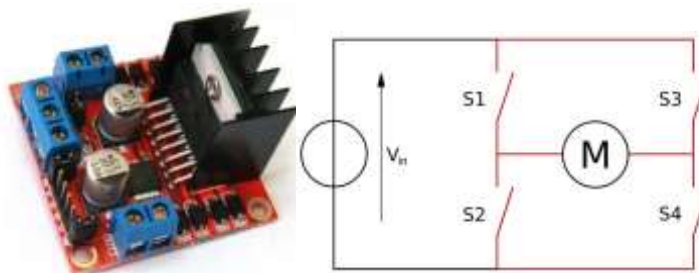


Figura 3 y 4: Esquema de conexiones de un Puente H y Puente H utilizado

1.3.1.5 Principios básicos de la impresión 3D

La manufactura aditiva crea objetos tridimensionales a partir de un diseño digital, construyéndolos capa por capa con materiales como plásticos, metales o biomateriales. Comienza con un diseño 3D en CAD (como TinkerCAD) que genera un archivo STL. Este se transforma en un laminador (como ULTIMAKER CURA) donde se ajustan parámetros de impresión, como material, temperaturas, altura de capa y velocidad, produciendo un archivo Gcode.

1.3.1.6 Principios básicos del corte por laser

El funcionamiento de la cortadora láser se basa en la capacidad de un láser para emitir un haz de luz altamente concentrado. Este haz tiene una potencia suficiente para cortar diversos materiales mediante calentamiento y vaporización. En nuestro caso hemos usado la máquina Láser de CO₂ de la marca Omtech y el tratamiento de los archivos .svg con el Software LIGHTBURN.



Figura 5: Compañera Helena Martínez realizando el corte láser de la carcasa.

1.3.1.7 Principios básicos de la programación en C con Arduino

El lenguaje de programación en Arduino está basado en C/C++. La placa MKR WIFI 1010 nos permite realizar la programación en este código con conectividad WIFI utilizando la librería creada a tal efecto WIFININA.

1.3.1.8 Principios básicos de la programación de páginas Web con HTML.

La programación con HTML (HyperText Markup Language) se basa en principios básicos que estructuran y organizan el contenido en las páginas web.

1.4 Resultados

1.4.1 Diseño de las bases del coche

Las dos placas del coche, la base con el puente H para movimiento y la superior con sensores y LCD, se diseñaron en Tinkercad. Fueron cortadas con láser Omtech en madera contrachapada, pintadas de blanco y adaptadas con chavetas para motores DC, orificios para la rueda loca y ventilación.



Figura 6: Diseño de las bases del coche.

1.4.2 Circuito de conexiones del movimiento del coche.

Para el movimiento del coche, se conectaron los motores DC al puente H y éstos se alimentaron con una pila de 9V. La Arduino es alimentada por 4 pilas de 1.5V lo que hacen 6V. Además, se ha dispuesto de un interruptor para poder desconectar la pila de 9V de los motores.

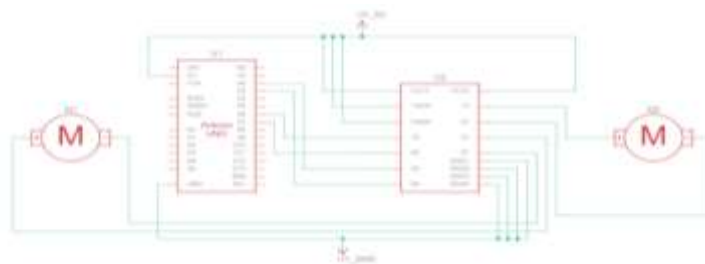


Figura 7: Esquema de conexiones de la base del móvil

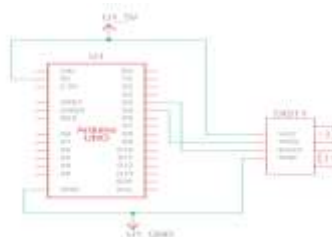


Figura 8: Esquema de conexión del sensor ultrasónico

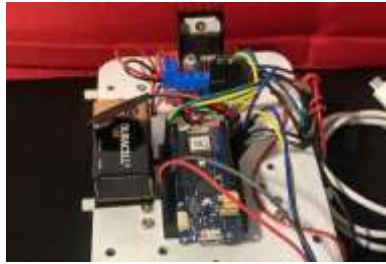


Figura 9: Imagen del puente H y motores de uno de los prototipos

1.4.3 Programa de la placa uno

El programa de la placa uno parece complicado y largo pero es debido a la cantidad de elementos, sin embargo uno por uno y por partes es bastante sencillo.

En las primeras líneas se definen el nombre del Wifi y la contraseña así como los pines del puente H y los del sensor ultrasónico.

```
#include <WiFiNINA.h>

// Configuración WiFi
char ssid[] = "Simpa";
char pass[] = "Altaseguridad";

WiFiServer server(80); // Servidor web en el puerto 80

// Definición de pines para el puente H
#define ENA 2
#define IN1 0
#define IN2 1
#define IN3 6
#define IN4 7
#define ENB 3

// Sensor HC-SR04
#define TRIG 4
#define ECHO 5

// Variables de control
bool forward = false, backward = false, left = false, right = false;
int Speed = 145;
```

En el void Setup se indican que los pines del puente H son salidas, los del ultrasónico entradas y se utilizan las funciones para hacer la conexión Wifi.

```
void setup() {
  Serial.begin(115200);

  // Configuración de pines motores
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);

  // Configuración pines sensor ultrasonido
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  // Conectar a la red WiFi
  WiFi.begin(ssid, pass);
```

```
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Conectando a WiFi...");
}

Serial.println("Conectado a WiFi");
Serial.print("Dirección IP: ");
Serial.println(WiFi.localIP());

server.begin(); // Iniciar servidor web
}
```

En el void Loop es sencillo ya que únicamente va haciendo las llamadas a las funciones que programamos después. En el caso de la placa 1 solo hay dos funciones, una llamada movimientoRover() y otra función llamada “manejarComandos” encargada del movimiento del coche a partir de los botones.

```
void loop() {
  WiFiClient client = server.available(); // Esperar conexión de un cliente

  if (client) {
    String request = client.readStringUntil('\r'); // Leer la solicitud HTTP
    client.flush();

    manejarComando(request, client);

    client.stop();
  }

  // Mantener el control del rover activo en el loop
  movimientoRover();
}
```

La función manejarComandos toma las instrucciones en función de si se ha pulsado cualquiera de los botones que después hemos configurado en la pantalla de la web. Además, dentro de esta propia función hemos llamado a la función “enviarInterfazWeb” que es la encargada de escribir el HTML en el servidor de la placa uno.

```
// ** Manejo de comandos desde la web **
void manejarComando(String request, WiFiClient client) {
  if (request.indexOf("GET /adelante/on") >= 0) forward = true;
  else if (request.indexOf("GET /adelante/off") >= 0) forward = false;
  else if (request.indexOf("GET /atras/on") >= 0) backward = true;
  else if (request.indexOf("GET /atras/off") >= 0) backward = false;
  else if (request.indexOf("GET /izquierda/on") >= 0) left = true;
  else if (request.indexOf("GET /izquierda/off") >= 0) left = false;
  else if (request.indexOf("GET /derecha/on") >= 0) right = true;
  else if (request.indexOf("GET /derecha/off") >= 0) right = false;
  else if (request.indexOf("GET /distancia") >= 0) {
    enviarDistancia(client);
    return;
  } else {
    enviarInterfazWeb(client);
  }
}
```

La función enviarInterfazWeb se encarga de escribir todo el código HTML que hemos creado y enviarlo al servidor.

```
// ** Enviar interfaz web al cliente **
void enviarInterfazWeb(WiFiClient client) {
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println();
}
```



```
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>Control del Rover</title>");

// ** Agregar estilos CSS para la imagen de fondo y los botones **
client.println("<style>");
client.println("body {");
client.println("  background-image: url('https://i.imgur.com/M6D6LP8.jpg');");
client.println("  background-size: cover;");
client.println("  background-position: center;");
//client.println("  background-attachment: fixed;"); // Fijar la imagen de fondo
client.println("  color: white;");
client.println("  text-align: center;");
client.println("  font-family: Arial, sans-serif;");
client.println("}");

//MEDIDAS MENSAJE DE ALERTA
client.println("#alerta {");
client.println("  min-height: 30px;"); // Reserva espacio para el mensaje
client.println("  font-size: 1.2em;");
client.println("}");

// ** Estilos para los botones **
client.println("button {");
client.println("  font-size: 1.2em;"); // Aumenta el tamaño en un 20%
client.println("  padding: 15px 30px;");
client.println("  margin: 10px;");
client.println("  border: none;");
client.println("  border-radius: 10px;");
client.println("  background-color: #3498db;");
client.println("  color: white;");
client.println("  box-shadow: 2px 2px 10px rgba(0,0,0,0.2);");
client.println("  cursor: pointer;");
client.println("}");
client.println("button:hover { background-color: #2980b9; }");
client.println("</style>");

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

client.println("<body>");
client.println("<h1>Control del Rover</h1>");

// Botones pulsadores con eventos JS
client.println("<button onmousedown='enviar(adelante/on)' onmouseup='enviar(adelante/off)'>Adelante</button>");
client.println("<br>");
client.println("<button onmousedown='enviar(atras/on)' onmouseup='enviar(atras/off)'>Atras</button>");
client.println("<button onmousedown='enviar(izquierda/on)' onmouseup='enviar(izquierda/off)'>Izquierda</button>");
client.println("<button onmousedown='enviar(derecha/on)' onmouseup='enviar(derecha/off)'>Derecha</button>");
client.println("<br>");
client.println("<h2>Distancia: <span id='distancia'></span> cm</h2>");
client.println("<h3 id='alerta'></h3>");
client.println("<br>");
// Enlace para acceder a la Placa 2
client.println("<h2>Otras Interfaces</h2>");
client.println("<p><a href='http://192.168.1.136'>Ver sensores adicionales (Placa 2)</a></p>");

// Incluir código JavaScript para AJAX
enviarCodigoJavaScript(client);

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

// ** Enviar código JavaScript para AJAX **
void enviarCodigoJavaScript(WiFiClient client) {
  client.println("<script>");
  client.println("function enviar(accion) { fetch('/' + accion); }");

  client.println("function actualizarDistancia() {");
  client.println("  fetch('/distancia') ;");
  client.println("  .then(response => response.text());");
  client.println("  .then(data => {");
  client.println("    document.getElementById('distancia').innerText = data;");
  client.println("    let distanciaValor = parseFloat(data);");
  client.println("    let alerta = document.getElementById('alerta');");
  client.println("    if (distanciaValor < 15) {");
```



```
client.println("    alerta.innerHTML = '<b style=\"color: red;\">PELIGRO. POSIBLE COLISION</b>';");
client.println("    } else {");
client.println("        alerta.innerHTML = ""; // Borra la alerta si la distancia es segura
client.println("    }");
client.println("    });");
client.println("}");
client.println("setInterval(actualizarDistancia, 1000);"); // Actualizar cada 1s
client.println("</script>");
}
```

La función `enviarDistancia` manda la variable `distancia` medida por el ultrasónico a la web.

```
// ** Enviar la distancia del sensor al cliente **
void enviarDistancia(WiFiClient client) {
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/plain");
    client.println();
    client.println(medirDistancia());
}
```

Las funciones `carforward`, `carbackward`, `carturnleft`, `carturnright` se encargan de escribir en los pines del puente H los valores correctos en función de cómo queremos que los motores giren para su movimiento.

```
// ** Función para mover el Rover **
void movimientoRover() {
    if (forward) carforward();
    else if (backward) carbackward();
    else if (left) carturnleft();
    else if (right) carturnright();
    else carStop();
}
```

```
// ** Funciones de movimiento del Rover **
void carforward() {
    analogWrite(ENA, Speed);
    analogWrite(ENB, Speed);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
```

```
// ** Función para medir la distancia con HC-SR04 **
float medirDistancia() {
    digitalWrite(TRIG, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG, LOW);

    long duration = pulseIn(ECHO, HIGH);
    float distance = duration * 0.034 / 2;
    Serial.print(distance);
    return distance;
}
```

1.4.4 Circuito de conexiones de la placa de sensores

En la segunda placa los sensores y actuadores conectados son: Sensor de CO₂, Sensor de CO, NO₂ y NH₃, DHT11 y LCD. Los esquemas siguientes representan como han sido conectados a la segunda placa y en qué pines.

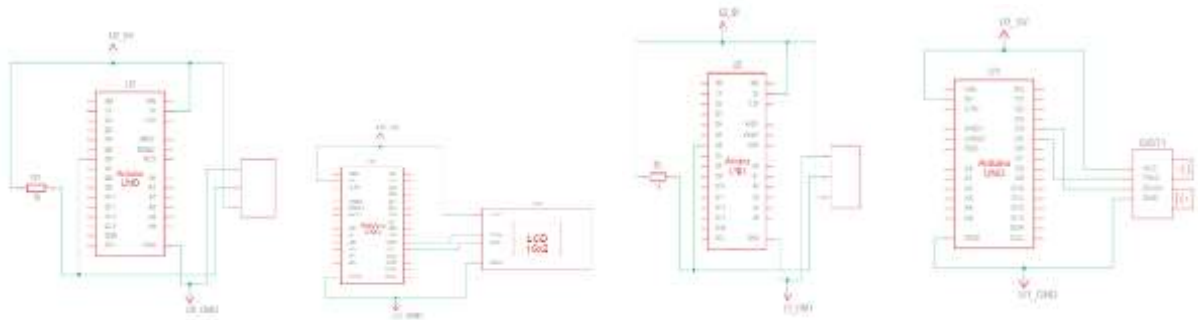


Figura 10: Esquema de conexiones del LCD, DHT11, Ultrasonico.

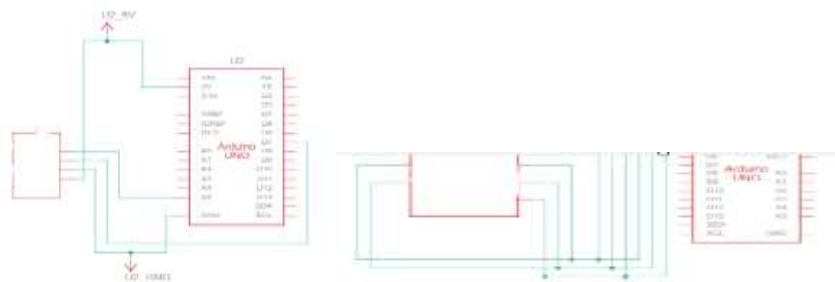


Figura 11 12 y: Esquema de conexiones del sensor de MQ de CO₂ y Esquema de conexiones del sensor de CO, NH₃, NO₂



Figura 13: Foto de la parte de arriba de la placa de sensores.

1.4.5 Programa de la placa de arriba



La placa de arriba es la que recoge la información de los sensores y los manda al servidor y a la placa LCD en tiempo real. Las primeras líneas no las copiamos porque son iguales a la placa 1.

```
// Configuración del DHT11
#define DHT_PIN 6
#define DHTTYPE DHT11

// Configuración del MQ2
#define MQ2_DIGITAL 7 // Salida digital del MQ2
#define MQ2_ANALOG A6 // Salida analógica del MQ2

// Configuración del CJMCU-6814
const int pinCO = A1;
const int pinNO2 = A2;
const int pinNH3 = A3;

//int gas = A6;
float temperatura, humedad;
int mq2Analog, mq2Digital;
int rawCO, rawNO2, rawNH3; //PARA EL CJMCU
dht11 DHT11; //se crea la instancia
LiquidCrystal_I2C lcd(0x27,16,2); // si no te sale con esta direccion puedes usar (0x3f,16,2) || (0x27,16,2) (esta es
||(0x20,16,2)

void setup() {
  Serial.begin(115200);
  // Inicializar sensores
  //DHT11.begin();
  pinMode(MQ2_DIGITAL, INPUT);
  pinMode(MQ2_ANALOG, INPUT);

  // Conectar a WiFi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando a WiFi..."); }

  Serial.println("Conectado a WiFi");
  Serial.print("IP del Servidor: ");
  Serial.println(WiFi.localIP());
  //para el DHT11
  DHT11.attach(DHT_PIN); //sensor conctado al pin 6
  //PARA LA PANTALLA LCD
  lcd.init();
  lcd.backlight(); // Enciende la retroiluminación (opcional)
  lcd.setCursor(0, 0);
  lcd.print("Hola Marte!");
  lcd.setCursor(0, 1);
  delay(5000);
  lcd.clear();

  server.begin(); // Iniciar servidor web }

void loop() {
  WiFiClient client = server.available(); // Esperar conexión de un cliente

  if (client) {
    String request = client.readStringUntil('\r'); // Leer la solicitud HTTP
    client.flush();

    // Si la solicitud es para la página web principal
    if (request.indexOf("GET / ") >= 0) {
      enviarPaginaWeb(client);
    }
    // Si la solicitud es para obtener datos JSON
    else if (request.indexOf("GET /datos") >= 0) {
      enviarDatosJSON(client);
    }

    client.stop();
  }
}

void enviarPaginaWeb(WiFiClient client) {
```




```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");

client.println("<head>");
//ESTILO DE LA INTERFAZ
client.println("<style>");
client.println("{");
client.println("  background-image: url('https://i.imgur.com/me3H8HY.png');");
client.println("  background-size: cover;");
client.println("  background-position: center;");
//client.println("  background-attachment: fixed;"); // Fijar la imagen de fondo
client.println("  color: white;");
client.println("  text-align: center;");
client.println("  font-family: Arial, sans-serif;");
client.println("}");
client.println("</style>");
// SCRIPT PARA LA ACTUALIZACION DE LOS DATOS EN LA INTERFAZ
client.println("<script>");
client.println("function actualizarDatos() {");
client.println("  var xhttp = new XMLHttpRequest();");
client.println("  xhttp.onreadystatechange = function() {");
client.println("    if (this.readyState == 4 && this.status == 200) {");
client.println("      var datos = JSON.parse(this.responseText);");
client.println("      document.getElementById('temp').innerHTML = datos.temperatura + ' C';");
client.println("      document.getElementById('hum').innerHTML = datos.humedad + ' %';");
client.println("      document.getElementById('gas').innerHTML = datos.gas + ' ppm';");
client.println("      document.getElementById('voltaje').innerHTML = datos.voltaje + ' V';");
client.println("      document.getElementById('co').innerHTML = datos.co + ' ppm';");
client.println("      document.getElementById('nh3').innerHTML = datos.nh3 + ' ppm';");
client.println("      document.getElementById('no2').innerHTML = datos.no2 + ' ppm';");
client.println("    }");
client.println("  }");
client.println("  xhttp.open('GET', '/datos', true);");
client.println("  xhttp.send();");
client.println("}");
client.println("setInterval(actualizarDatos, 2000);"); // Actualizar cada 2 segundos
client.println("</script>");
client.println("</head>");

client.println("<body>");
client.println("<h1>Monitor de Sensores</h1>");
client.println("<p><strong>Temperatura:</strong> <span id='temp'>Cargando...</span></p>");
client.println("<p><strong>Humedad:</strong> <span id='hum'>Cargando...</span></p>");
client.println("<p><strong>Nivel de Gas:</strong> <span id='gas'>Cargando...</span></p>");
client.println("<p><strong>Voltaje MQ2:</strong> <span id='voltaje'>Cargando...</span></p>");
client.println("<p><strong>Nivel de CO2:</strong> <span id='co'>Cargando...</span></p>");
client.println("<p><strong>Nivel de NH3:</strong> <span id='nh3'>Cargando...</span></p>");
client.println("<p><strong>Nivel de NO2:</strong> <span id='no2'>Cargando...</span></p>");
client.println("</body>");
client.println("</html>");
}

void enviarDatosJSON(WiFiClient client) {
  leer_estado_DHT();
  temperatura = DHT11.temperature;
  humedad = DHT11.humidity;
  mq2Analog = analogRead(MQ2_ANALOG);
  mq2Digital = digitalRead(MQ2_DIGITAL);
  rawCO = analogRead(pinCO);
  rawNO2 = analogRead(pinNO2);
  rawNH3 = analogRead(pinNH3);

  float mq2Voltage = mq2Analog * (3.3 / 1023.0); // Convertir a voltaje
  float voltageCO = rawCO * (3.3 / 1023.0);
  float voltageNO2 = rawNO2 * (3.3 / 1023.0);
  float voltageNH3 = rawNH3 * (3.3 / 1023.0);

  // Actualizar datos en la LCD
  imprime_temperatura();

  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: application/json");
  client.println();
```




```
client.print("{}");
client.print("\temperatura\n:"); client.print(temperatura); client.print(",");
client.print("\thumedad\n:"); client.print(humedad); client.print(",");
client.print("\ngas\n:"); client.print(mq2Analog); client.print(",");
client.print("\tvoltaje\n:"); client.print(mq2Voltage, 2); client.print(",");
client.print("\tco\n:"); client.print(voltageCO, 2); client.print(",");
client.print("\tnh3\n:"); client.print(voltageNH3, 2); client.print(",");
client.print("\tno2\n:"); client.print(voltageNO2, 2);
client.println("");
}

void leer_estado_DHT(){

int chk = DHT11.read(DHT_PIN); //lee el estado del sensor
Serial.print("\n Estado del sensor: ");
switch(chk){
case 0:
Serial.print("OK");
break;
case -1:
Serial.print("Checksum error");
break;
case -2:
Serial.print("Time_out error");
break;
default:
Serial.print("Error desconocido");
}
}

void imprime_temperatura(){

lcd.setCursor(0, 0);
lcd.print("Temp.: ");
lcd.print(int(temperatura));
lcd.print(" C");
lcd.setCursor(0, 1);
lcd.print("Humedad: ");
lcd.print(int(humedad));
lcd.print(" C");
// Esperar un breve tiempo antes de la siguiente actualización
delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Gas: ");
lcd.print(mq2Analog);
lcd.setCursor(0, 1);
lcd.print("CO2: ");
lcd.print(rawCO);
lcd.print(" ppm");
delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("NH3: ");
lcd.print(rawNH3);
lcd.print(" ppm");
lcd.setCursor(0, 1);
lcd.print("NO2: ");
lcd.print(rawNO2);
lcd.print(" ppm");
delay(2000);
}
```

1.4.6 Diseño de la página web de movimiento del coche

La página web se ha diseñado con una imagen de fondo que se ha colgado en un repositorio imgur.com para poder acceder a ella con el código HTML. Cuenta con cuatro botones, una zona de monitorización de la distancia y un link al servidor de la placa 2.



Figura 14 y 15: Interfaz web de la página de control de la placa 1 y Display LCD con medidas de la placa

1.4.7 Diseño de la página web de monitorización de parámetros con los sensores.

La página web de la placa 2 tiene como fondo también una imagen colgada en mismo repositorio y varias líneas dónde aparecen las medidas de los sensores.



Figura 16: Interfaz web de la página de control de la placa 2

1.4.8 Diseño de los tapacubos



Figura 17: Diseño 3D de tapacubos para impresión en PLA y tapacubos impreso y pintado con pintura acrílica

1.4.9 Diseño y corte láser de las chapas de metacrilato

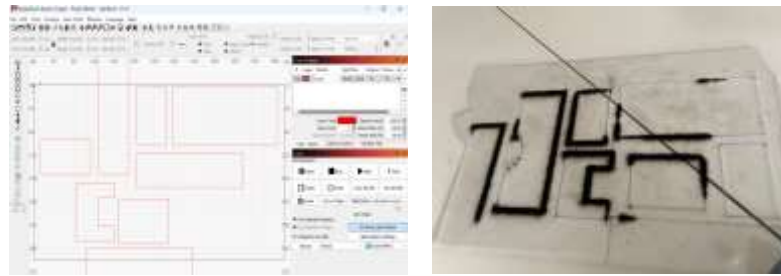


Figura 18 y 19: Corte de las piezas de Metacrilato para recubrimiento del Coche en Lightburn para cortadora láser y Corte realizado en metacrilato para cortadora láser

1.5 Conclusión

Con el coche AirGuard hemos conseguido monitorizar los gases y la temperatura y humedad en tiempo real gracias al uso de IOT pudiendo así prevenir riesgos para la salud al permitir a las entidades públicas o a los clientes actuar rápidamente. Además este dispositivo consigue facilitar el seguimiento de las regulaciones ambientales y asegura que las emisiones estén dentro de los límites permitidos. Puede ser una gran ayuda si las entidades gubernamentales o las Comunidades Autónomas contaran con él.

Se podrían implementar mejoras como puede ser la inclusión de sensores de partículas que sirven para medir y monitorear la cantidad y el tamaño de las partículas en el aire. Estos sensores detectan partículas como polvo, humo o aerosoles ayudando también a prevenir problemas respiratorios. Otro sensor que se podría implementar es el sensor de gotas de lluvia que nos permite detectar intensidad de lluvia. Los sensores de compuestos orgánicos volátiles miden sustancias químicas como formaldehído y otros compuestos orgánicos.

Además, podríamos incorporar el análisis estadístico en la propia web, tramos horarios en los que hay más contaminación, variación estacional de la contaminación...

2. Figuras

Figura 1: Sensor de CO2 para Arduino.

Figura 2: Sensor capacitivo de humedad DHT11

Figura 3: Esquema de conexiones de un Puente H

Figura 4: Puente H utilizado.

Figura 5: Compañera Helena Martínez realizando el corte láser de la carcasa.

Figura 6: Diseño de las bases del coche.

Figura 7: Esquema de conexiones de la base del móvil



Figura 8: Esquema de conexión del sensor ultrasónico

Figura 9: Imagen del puente H y motores de uno de los prototipos

Figura 10: Esquema de conexiones del LCD, DHT11, Ultrasónico

Figura 11: Esquema de conexiones del sensor de MQ de CO₂

Figura 12: Esquema de conexiones del sensor de CO, NH₃, NO₂

Figura 13: Foto de la parte de arriba de la placa de sensores.

Figura 14: Interfaz web de la página de control de la placa 1

Figura 15: Display LCD con medidas de la placa

Figura 16: Interfaz web de la página de control de la placa 2

Figura 17: Diseño 3D de tapacubos para impresión en PLA y tapacubos impreso y pintado con pintura acrílica.

Figura 18: Corte de las piezas de Metacrilato para recubrimiento del coche en Lightburn para cortadora láser.

Figura 19: Corte realizado en metacrilato para cortadora Láser

3. Referencias

De Elio Medina, J. (2013). *Estrategias de monitorización de CO₂ y otros gases en los estudios de análogos naturales* (Tesis doctoral, Universidad Politécnica de Madrid). Recuperado de [https://oa.upm.es/22541/1/Javier de Elio Medina 1.pdf](https://oa.upm.es/22541/1/Javier_de_Elio_Medina_1.pdf)

Ministerio para la Transición Ecológica y el Reto Demográfico. (n.d.). *Índice de Calidad del Aire (ICA)*. Recuperado de <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/ica.html>

Ministerio para la Transición Ecológica y el Reto Demográfico. (n.d.). *Índice de Calidad del Aire (ICA)*. Recuperado de <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/ica.html>