



**Universidad
Tecnológica
del Perú**

Presentación de plan de trabajo - Semana 11

Carrito analizador de Infraestructura de una mina y detector de CO2

Curso:

Curso Integrador I: Diseño Electrónico

Alumnos:

Rodríguez Tapia Gerardo Gabriel	U22203808
Chagua Tasaico Alexis Magno	U21310742
Santibañez Pastor Jose Alexander	1400058
Vicente Balcazar Stheveen slittle	U21216580
Girón Aguilar Aron Neil	U20209814

Docente:

Motta Zorrilla, Bryan

Lima - Peru

2025

Dividimos en dos parte el trabajo, la primera parte que solo se enfoca en la parte móvil, osea en el carro y la segunda en el detector de CO₂

Carro usando Arduino:

1. Especificaciones de diseño

- Aplicación del proyecto:

Este parte del proyecto tiene como objetivo desarrollar un vehículo controlado de forma inalámbrica mediante tecnología Bluetooth, le permite ingresar a lugares potencialmente peligrosos, como túneles o galerías en exploración, sin necesidad de exponer a los trabajadores a riesgos innecesarios.

Este carro es la base de transportar sensores (como detectores de gases, temperatura o humedad) en zonas de difícil acceso. De esta forma, se convierte en una herramienta de apoyo para labores de inspección y evaluación de condiciones ambientales subterráneas, contribuyendo a mejorar la seguridad y eficiencia en operaciones mineras.

- Explicación del Funcionamiento:

El proyecto consiste en un carro robótico controlado por un microcontrolador Arduino UNO. El funcionamiento general es el siguiente:

Control inalámbrico:

Se utiliza un módulo Bluetooth HC-05, que permite conectar el vehículo a un celular u otro dispositivo compatible. A través de una aplicación, se puede enviar comandos para mover el carro en distintas direcciones.

Movimiento del vehículo:

El Arduino recibe las señales del módulo Bluetooth y las interpreta para controlar los motores. Para esto se utiliza un driver de motores L298N, que permite manejar los 4 motores DC conectados a las ruedas. Esto le da al carro la capacidad de moverse hacia adelante, atrás, girar a la izquierda y a la derecha.

Alimentación eléctrica:

El sistema del driver es alimentado por una batería de 12V, la cual también alimenta a los motores y el arduino es alimentado por una batería de 9v.

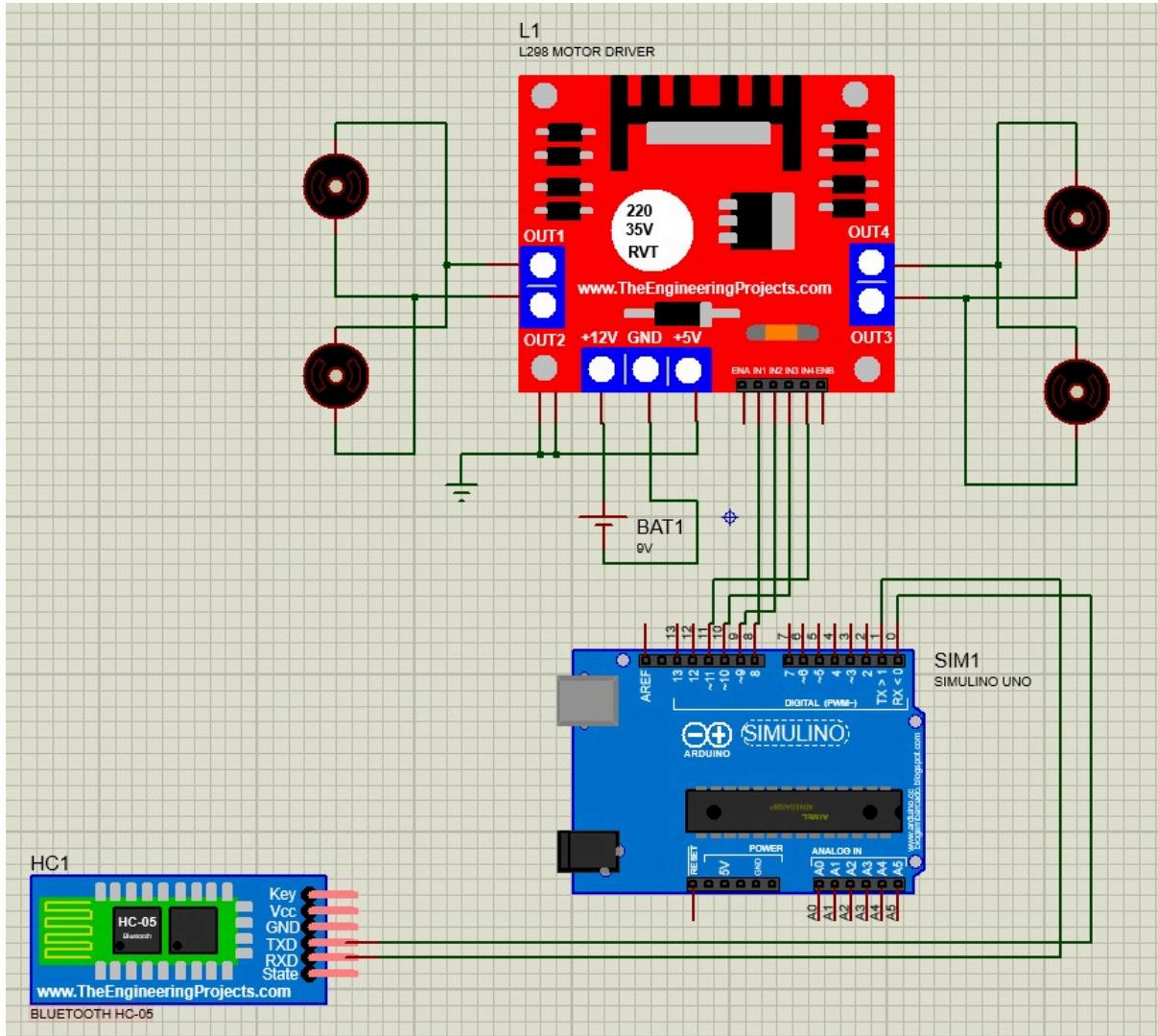
- Componentes utilizados

Componente	Funcion	Precio
Arduino UNO	la base principal para el funcionamiento del proyecto, se apoyará con el uso los sensores y actuadores para mandarlos la señal del ambiente y camino.	S/ 40

L298N	Controlador de motor, podemos manejar la dirección y velocidad de los motorreductores	S/ 11
HC-05	Manejo a control del vehículo de exploración, como prototipo nos comunicamos mediante vía bluetooth, tiene una señal en óptimas condiciones de hasta 10 metros.	S/ 25
MOTORREDUCTOR	Son 4 motorreductores para mantener un mejor equilibrio del vehículo, no necesita mucho voltaje para funcionar.	motorreductor: S/ 3.5 c/u ruedas: S/ 3 c/u
CABLE UTP	Cable económico para conectar circuitos	S/ 1

2. Creación de esquemáticos

- Diseño del circuito en el software correspondiente (Proteus, Tinkercad, Fritzing, etc.).



- Justificación técnica del diseño.

El diseño del circuito se basó en criterios de funcionalidad, simplicidad y bajo costo, buscando desarrollar un vehículo inalámbrico robusto, confiable y fácil de implementar en entornos reales como la minería.

Arduino UNO: Se eligió por su facilidad de programación, disponibilidad de pines de E/S digitales y su compatibilidad con una gran cantidad de módulos. Es ideal para prototipado rápido y control de sistemas simples como motores DC.

Módulo Bluetooth HC-05: Permite la comunicación inalámbrica entre el carro y un teléfono móvil, facilitando el control remoto sin necesidad de interfaces físicas. Es estable, fácil de emparejar y ampliamente compatible.

Driver de motores L298N: Este módulo fue elegido por su capacidad para controlar hasta dos motores DC de forma independiente. Es económico, soporta suficiente corriente para pequeños vehículos, e incluye reguladores de voltaje para alimentar el Arduino desde la misma fuente.

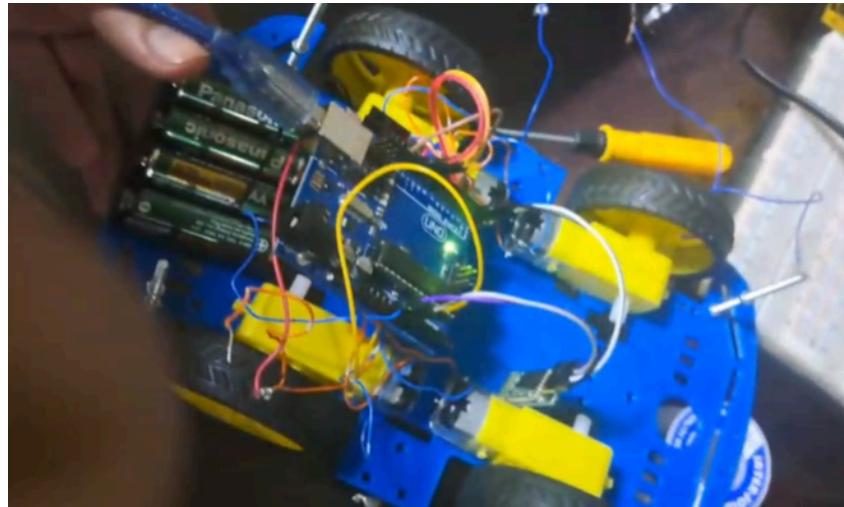
Motores DC: Proveen un método sencillo y eficiente para mover el vehículo en distintas direcciones. Su control mediante PWM permite ajustar velocidad y dirección.

3. Simulación

Debido a las limitaciones del software Proteus, no fue posible simular el módulo Bluetooth HC-05, ya que este componente no está disponible en la librería del simulador y su comunicación serial no puede emularse en tiempo real.

4. Montaje

- Fotografías del armado físico del circuito.



<https://youtu.be/5n73kNXLeBE>

- Comentarios sobre el proceso (dificultades, ajustes, observaciones).

Nuestro único problema que teníamos con el carro es que la 4 pilas AA no alimentaba a todo el circuito, después de esto quisimos alimentarlo con baterías de 9v pero se tuvo el mismo resultado, al final deducimos que el L298N, que alimenta también a los motorreductores necesita una fuente aparte del Arduino, es decir que se necesita dos fuentes, una para el módulo M298N y el Arduino.

5. Mediciones

- Resultados obtenidos en el circuito real.

Se realizó el montaje físico del circuito en una protoboard/base de pruebas, incluyendo el Arduino UNO, el módulo Bluetooth HC-05, el driver L298N y los motores DC. Luego, se alimentó el Arduino con una batería de 9V y el módulo L298N con una fuente de 12V.

El vehículo respondió correctamente a los comandos enviados desde el celular, logrando movimientos hacia adelante, atrás y giros laterales según lo programado.

- Herramientas utilizadas (multímetro, osciloscopio, etc.).

Multímetro digital: Para verificar continuidad en el cableado, medir voltajes de entrada/salida y revisar el consumo aproximado en distintas partes del circuito.

- Registro de valores relevantes.
 1. Voltaje de entrada al driver L298N: 11.9V (desde la batería de 12V).
 2. Voltaje en las salidas del L298N hacia los motores: aproximadamente 11.5V cuando están activos.
 3. Voltaje de entrada al Arduino UNO: 9.2V (desde la batería de 9V).
 4. Corriente estimada del sistema (con dos motores activos): entre 400 mA y 700 mA, dependiendo del esfuerzo que realicen los motores.

6. Comparación entre simulación y circuito físico

Debido a las limitaciones del software Proteus, no fue posible realizar una simulación completa del funcionamiento del vehículo, ya que el módulo Bluetooth HC-05 no está disponible en su librería y no puede emular su comunicación serial.

Por esta razón, la simulación solo permitió validar conexiones eléctricas básicas (como la alimentación del Arduino, el control del L298N y el comportamiento lógico de las salidas digitales).

En cambio, en el circuito físico se pudo comprobar:

- Que el Arduino interpreta correctamente los comandos recibidos por Bluetooth
- Que los motores respondan a las instrucciones enviadas desde el celular.
- Que la alimentación separada de Arduino y el L298N funciona sin conflictos.

Detector de CO2 con ESP32

1. Especificaciones de diseño

- Aplicación del proyecto:

Esta parte del proyecto permite detectar la presencia de gases nocivos como dióxido de carbono (CO2), monóxido de carbono (CO), amoníaco, y otros compuestos que pueden ser perjudiciales para la salud. Al medir estos valores con un sensor MQ135 y enviarlos mediante WiFi a un dispositivo móvil o computadora, se puede tomar decisiones preventivas sin exponer a los mineros al riesgo directo.

- Explicación del Funcionamiento:

El circuito está compuesto por un ESP32, que es un microcontrolador con conectividad WiFi integrada, y un sensor de gases MQ135. El funcionamiento general es el siguiente:

- ★ El sensor MQ135 mide la concentración de gases en el ambiente. Este sensor genera una señal analógica proporcional a la cantidad de gases presentes.
- ★ La señal analógica se conecta a uno de los pines del ESP32, que la lee con su conversor ADC (analógico a digital).
- ★ Enviarla por WiFi o Bluetooth a una aplicación móvil o página web.

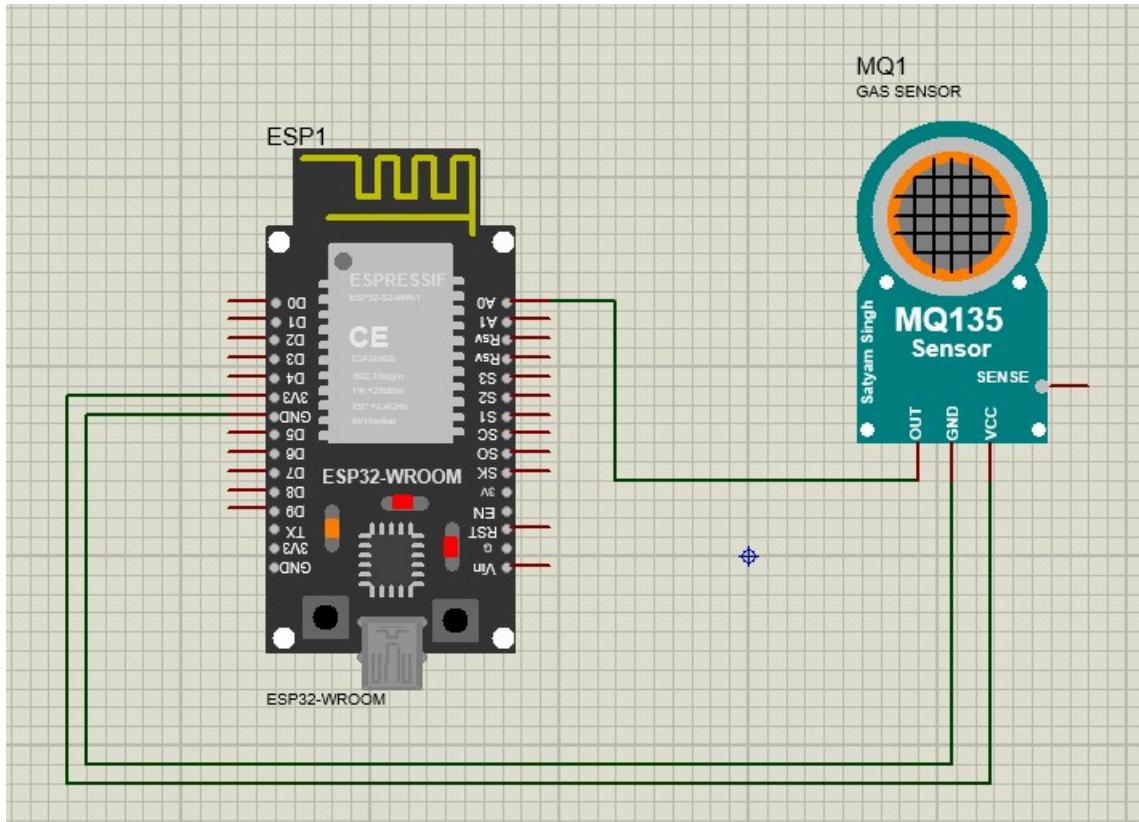
★ El sistema se alimenta por batería, lo que permite su uso en ambientes móviles o difíciles de cablear.

- Componentes utilizados

Componentes	Función	Precio
ESP32-WROOM	Microcontrolador principal del sistema. Tiene conectividad WiFi y Bluetooth, lo que permite procesar la señal del sensor y enviarla de forma inalámbrica.	S/34
MQ135	Sensor de gases que detecta CO2, CO y otros compuestos tóxicos. Genera una señal analógica proporcional a la concentración.	S/12
CABLE UTP	Cable económico para conectar circuitos	S/ 1

2. Creación de esquemáticos

- Diseño del circuito en el software correspondiente (Proteus, Tinkercad, Fritzing, etc.).



- Justificación técnica del diseño.

El diseño del circuito fue pensado para cumplir con los requerimientos de un sistema portátil y confiable de monitoreo ambiental, con foco en la detección de gases en entornos mineros cerrados o de difícil acceso. A continuación se justifica la selección de cada componente:

- ★ **ESP32:** Fue elegido por su capacidad de procesamiento, sus entradas analógicas (ADC) y su conectividad inalámbrica (WiFi y Bluetooth). Esto permite leer señales analógicas del sensor y transmitir los datos a una aplicación o servidor sin cables, ideal para situaciones remotas o peligrosas.
- ★ **Sensor MQ135:** Este sensor es capaz de detectar varios gases contaminantes como CO₂, CO, entre otros. Es económico, fácil de

usar, y entrega una señal analógica directamente proporcional a la concentración de gases presentes en el aire.

- ★ Conexión analógica entre sensor y ESP32: Se aprovechó el ADC del ESP32 para leer la salida del sensor sin necesidad de componentes adicionales, lo que simplifica el diseño y reduce el tamaño del circuito.

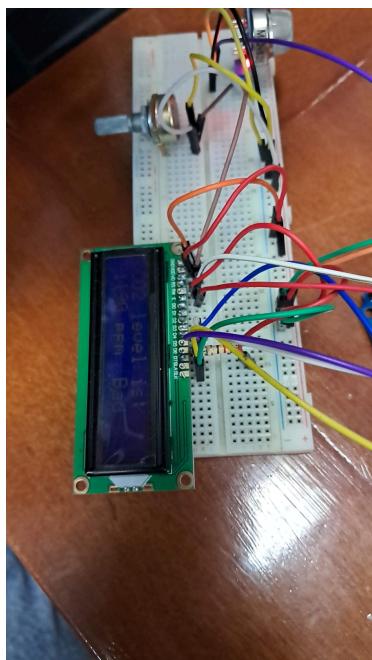
3. Simulación

En este caso, el diseño fue realizado en Proteus, con el objetivo de mostrar la conexión entre el ESP32-WROOM y el sensor MQ135. Sin embargo, debido a que Proteus no cuenta con soporte oficial para simular microcontroladores ESP32, no fue posible ejecutar una simulación funcional del sistema.

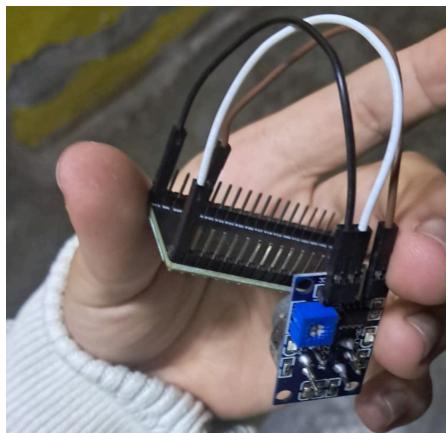
4. Montaje

- Fotografías del armado físico del circuito.

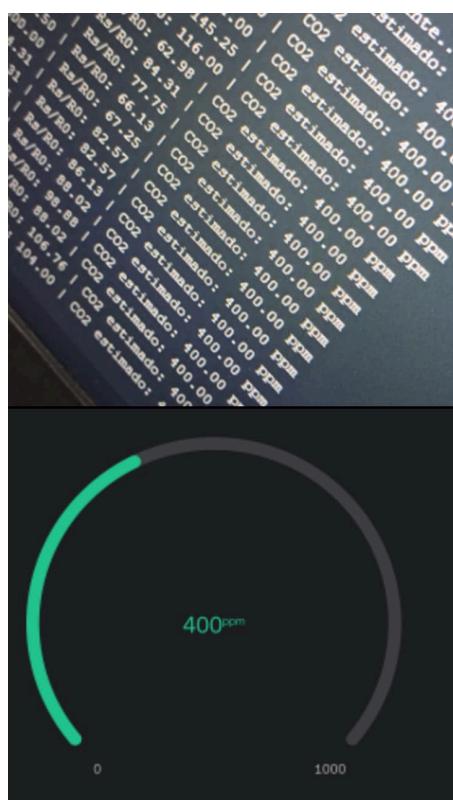
En esta imagen se ve el circuito hecho con un arduino y una lcd



En esta segunda imagen cambiamos el Arduino por ESP32 y ya no utilizamos la lcd.



En esta imagen vemos los resultados que manda el detector de CO₂ al ARDUINO IDE y también los resultados que se envía hacia una aplicación.



<https://youtu.be/K4qTDQ8ZhS8>

- Comentarios sobre el proceso (dificultades, ajustes, observaciones).

El diseño original utilizaba un Arduino UNO, una pantalla LCD y un potenciómetro para mostrar localmente los niveles de gases detectados por el sensor MQ135. Sin embargo, este enfoque obligaba a estar cerca del equipo, lo cual no es seguro en entornos mineros.

Por ello, se optó por usar un ESP32, que permite enviar los datos de forma remota mediante WiFi o Bluetooth. Este cambio mejora la seguridad al evitar que los operarios se expongan a zonas peligrosas, y permite monitorear los niveles de gases desde una ubicación segura.

5. Mediciones

- Resultados obtenidos en el circuito real.

Se realizó el montaje físico del circuito en una protoboard/base de pruebas, incluyendo el módulo ESP32-WROOM y el sensor de gas MQ135. El sensor fue conectado al ESP32 utilizando los pines de alimentación (VCC y GND) y el pin de salida analógica (OUT) al pin analógico del ESP32.

El circuito fue alimentado mediante el puerto USB del ESP32 conectado a una computadora o una fuente de 5V.

El sistema respondió correctamente a la detección de gases, mostrando variaciones en la salida analógica del sensor MQ135 conforme se modificaban las condiciones del entorno, como la presencia de humo o gases contaminantes.

- Herramientas utilizadas (multímetro, osciloscopio, etc.).

Multímetro digital: Para verificar continuidad en el cableado, medir voltajes de entrada/salida y revisar el suministro correcto de voltaje al sensor.

- Registro de valores relevantes.
 1. Voltaje de entrada al sensor MQ135: 4.96V (desde el pin 5V del ESP32)
 2. Voltaje en la salida analógica del MQ135 (pin OUT): Varía entre 0.5V y 3.8V dependiendo de la concentración de gases.
 3. Voltaje de alimentación al ESP32: 5.01V (desde el puerto USB).
 4. Consumo estimado del sistema completo: Aproximadamente 70 mA en reposo, con picos de hasta 100 mA durante la transmisión de datos vía Wi-Fi del ESP32.
- 6. Comparación entre simulación y circuito físico

Debido a las limitaciones del software Proteus, no fue posible realizar una simulación completa del funcionamiento del detector, ya que el módulo ESP32 no está disponible en su librería y no puede emular su comunicación serial.

Por esta razón, la simulación solo permitió validar conexiones eléctricas básicas