

SESIÓN 8

Construcción del chasis e inicio de la programación

Proyecto ABP: vehículo robótico controlado por Bluetooth

Pregunta orientadora

¿Cómo podemos diseñar y programar un vehículo robótico que pueda desplazarse de forma segura y ser controlado posteriormente desde un teléfono celular?

Propósito de la sesión

Iniciar la construcción física y la programación del vehículo robótico mediante el montaje del chasis, los motores, las ruedas y los principales componentes electrónicos. Además, reconocer la función del controlador de motores L298N y cargar en Arduino un programa inicial para comprobar los movimientos del vehículo.

Al finalizar esta sesión, cada equipo deberá contar con:

- El chasis ensamblado.
- Los dos motores instalados.
- Las ruedas y la rueda loca colocadas.
- Arduino UNO fijado al vehículo.
- El módulo L298N instalado.
- Las conexiones principales organizadas.
- El código básico cargado en Arduino.
- Un registro de los avances, dificultades y decisiones del equipo.



1. Recordemos nuestro desafío

En la sesión anterior conformamos equipos de ingeniería y conocimos el propósito del proyecto: construir un vehículo robótico controlado mediante Bluetooth.

A partir de esta sesión comenzaremos a convertir nuestras ideas en un prototipo real. El vehículo será construido y programado progresivamente. En la sesión 8 se trabajará principalmente en su estructura mecánica y en el código inicial de movimiento. En la sesión 9 se completarán las conexiones, se incorporará el módulo Bluetooth HC-06 y se realizarán las pruebas finales.

El proyecto se desarrolla mediante el Aprendizaje Basado en Proyectos, porque cada equipo debe investigar, distribuir responsabilidades, construir, programar, probar, identificar errores y mejorar su solución tecnológica.

2. El chasis: el esqueleto del vehículo

El chasis es la estructura sobre la cual se instalan todos los componentes del robot. Puede compararse con el esqueleto del cuerpo humano, porque sostiene y mantiene organizadas las diferentes partes del sistema.

Sobre el chasis se ubicarán:

- Los motores.
- Las ruedas.
- La rueda loca.
- La tarjeta Arduino UNO.
- El controlador de motores L298N.
- Las baterías.
- El interruptor.
- El módulo Bluetooth HC-06.
- Los cables de conexión.

Un buen montaje debe permitir que el vehículo se mantenga equilibrado, que las ruedas giren libremente y que los componentes electrónicos permanezcan firmes y protegidos.

3. Organización del equipo de ingeniería

Cada equipo estará conformado por tres estudiantes. Aunque cada integrante tendrá una responsabilidad principal, todas las decisiones deberán tomarse de manera colaborativa.

Ingeniero o ingeniera mecánica

- Revisar el diseño del chasis.
- Sujetar correctamente los motores.
- Instalar las ruedas.
- Colocar la rueda loca.
- Verificar que las partes móviles no estén bloqueadas.
- Comprobar el equilibrio del vehículo.

Ingeniero o ingeniera electrónica

- Identificar los terminales de los motores.
- Organizar los cables.
- Instalar el controlador L298N.
- Verificar la polaridad.
- Comprobar las conexiones entre Arduino y el controlador.
- Evitar cortocircuitos y cables sueltos.

Ingeniero o ingeniera de programación

- Escribir o copiar cuidadosamente el código.
- Identificar los pines utilizados.
- Verificar el programa antes de cargarlo.
- Cargar el código en Arduino.
- Registrar los errores encontrados.
- Explicar al equipo la función de cada bloque del programa.

4. Participación equitativa y enfoque de género

Los roles de ingeniería no deben asignarse de acuerdo con ideas preconcebidas sobre lo que supuestamente pueden hacer las niñas o los niños.

Una estudiante puede liderar la programación, el montaje mecánico o las conexiones electrónicas. De la misma manera, cualquier integrante puede aprender y participar en los tres campos de trabajo.

Durante la sesión deben cumplirse los siguientes acuerdos:

- Todas las personas tendrán derecho a manipular los materiales.
- Ningún integrante realizará todo el proyecto por sí solo.
- Las ideas serán escuchadas sin importar quién las proponga.
- Los errores se analizarán como oportunidades de aprendizaje.
- El liderazgo podrá ser ejercido por cualquier integrante.
- Los roles se rotarán en las sesiones posteriores.

Idea clave

La igualdad de oportunidades no significa que todas las personas hagan exactamente lo mismo al mismo tiempo, sino que tengan posibilidades reales de aprender, proponer, decidir y liderar.

5. Materiales

- 1 tarjeta Arduino UNO.
- 1 chasis para vehículo robótico.
- 2 motores reductores TT.
- 2 ruedas.
- 1 rueda loca.

- 1 módulo controlador de motores L298N.
- 1 módulo Bluetooth HC-06.
- 1 interruptor de encendido y apagado.
- 1 portapilas.
- Baterías o pilas según el montaje disponible.
- Cables de conexión.
- Tornillos y tuercas.
- Destornillador.
- Cinta aislante.
- Computador con Arduino IDE.
- Cable USB para Arduino.

6. Conociendo el controlador L298N

Arduino puede enviar instrucciones, pero sus pines no están diseñados para suministrar directamente toda la corriente que necesitan los motores. Por esta razón se utiliza el módulo L298N, cuya función es actuar como intermediario entre Arduino, la fuente de alimentación y los motores.

El L298N permite:

- Encender y apagar los motores.
- Cambiar el sentido de giro.
- Controlar dos motores al mismo tiempo.
- Regular su velocidad mediante señales PWM.
- Proteger parcialmente la tarjeta Arduino de la corriente requerida por los motores.

ENA: Controla la activación y la velocidad del motor A.

IN1 e IN2: Determinan el sentido de giro del motor A.

ENB: Controla la activación y la velocidad del motor B.

IN3 e IN4: Determinan el sentido de giro del motor B.

OUT1, OUT2, OUT3 y OUT4: Se conectan a los terminales de los motores.

GND: Corresponde a la referencia común o tierra del circuito.

7. Etapa 1: preparación del espacio de trabajo

1. Limpiesen la mesa.
2. Organicen los materiales por grupos.
3. Comprueben que tengan todos los tornillos.
4. Identifiquen los motores, ruedas y componentes electrónicos.
5. Mantengan las baterías desconectadas durante el montaje.
6. No conecten Arduino al computador hasta verificar el cableado.
7. Registren en el diario de proyecto el nombre de cada integrante y su rol.

8. Etapa 2: conexión y montaje de los motores

Paso 1. Identificar los motores

Observen los dos motores reductores TT. Cada uno posee dos terminales metálicos. A cada motor se deben conectar dos cables: uno para cada terminal. Los cables pueden soldarse o sujetarse firmemente según los materiales disponibles. No deben quedar haciendo contacto entre sí.

Paso 2. Sujetar los motores al chasis

Coloquen un motor a cada lado del chasis. Utilicen los soportes, tornillos y tuercas correspondientes. Verifiquen que:

- Los dos motores queden a la misma altura.
- Los ejes estén orientados hacia el exterior.
- Los tornillos estén firmes.
- Los cables no queden atrapados.
- Los motores no se muevan al girar las ruedas.

Paso 3. Colocar las ruedas

Instalen una rueda en el eje de cada motor. Gírenlas manualmente para comprobar que:

- No rocen con el chasis.
- No estén demasiado apretadas.
- No se desprendan con facilidad.
- Ambas puedan girar libremente.

9. Etapa 3: instalación de la rueda loca

La rueda loca es una rueda sin motor que gira libremente y permite que el vehículo mantenga el equilibrio mientras cambia de dirección.

8. Identifiquen la parte delantera o trasera del chasis donde se instalará.
9. Coloquen la base de la rueda loca.
10. Inserten los tornillos.
11. Ajusten las tuercas.
12. Comprueben que la rueda gire en diferentes direcciones.
13. Pongan el chasis sobre la mesa y verifiquen que quede nivelado.

10. Etapa 4: instalación del portapilas y el interruptor

Portapilas

El portapilas debe quedar sujeto al chasis y ubicado en una posición que no afecte el movimiento de las ruedas. Se recomienda distribuir el peso de manera equilibrada para evitar que el vehículo se incline hacia un lado.

Interruptor

El interruptor permitirá conectar o desconectar la alimentación del sistema. Debe ubicarse en un lugar accesible para que el equipo pueda apagar rápidamente el vehículo durante las pruebas.

11. Etapa 5: instalación de Arduino UNO

Arduino será el cerebro del vehículo.

14. Seleccionen una zona plana del chasis.
15. Coloquen Arduino sin que la parte inferior toque directamente elementos metálicos.
16. Utilicen tornillos, separadores o sujetadores.
17. Mantengan libre el puerto USB.
18. Comprueben que los pines sean accesibles.
19. Eviten ubicar cables debajo de la tarjeta.

12. Etapa 6: instalación del controlador L298N

Coloquen el módulo L298N en una zona firme del chasis, preferiblemente cerca de los motores.

20. Conecten los cables del motor izquierdo a una salida del L298N.
21. Conecten los cables del motor derecho a la otra salida.
22. Identifiquen claramente cuál será el motor A y cuál será el motor B.
23. Revisen que los cables estén firmes.
24. Eviten que los terminales hagan contacto accidental.

13. Conexiones entre Arduino y L298N

Controlador L298N	Arduino UNO	Función
ENA	Pin 10	Velocidad del motor A
IN1	Pin 9	Dirección del motor A
IN2	Pin 8	Dirección del motor A
ENB	Pin 5	Velocidad del motor B
IN3	Pin 7	Dirección del motor B
IN4	Pin 6	Dirección del motor B
GND	GND	Tierra común

Arduino, el L298N y las fuentes de alimentación deben compartir una referencia de tierra común mediante sus conexiones GND. Sin una tierra común, las señales de control pueden no interpretarse correctamente.

14. ¿Cómo controla el programa los movimientos?

Cada motor puede girar en dos sentidos. Arduino establece el sentido mediante combinaciones de valores HIGH y LOW.

IN1	IN2	Resultado
LOW	HIGH	Giro en un sentido
HIGH	LOW	Giro en sentido contrario
LOW	LOW	Motor detenido

La función `analogWrite()` permite establecer la velocidad utilizando valores entre 0 y 255:

- 0: motor detenido.
- 100: velocidad reducida.
- 180: velocidad media.
- 255: velocidad máxima.

Atención

La dirección real puede variar según la manera en que se hayan conectado los cables de cada motor. Si una rueda gira en sentido contrario, se pueden intercambiar sus cables o modificar los valores HIGH y LOW de su función correspondiente.

15. Código 1: prueba inicial de los motores

Este primer programa no utiliza todavía el módulo Bluetooth. Su finalidad es verificar que el vehículo pueda avanzar, retroceder, girar a la derecha, girar a la izquierda y detenerse. Cada acción se ejecuta durante cinco segundos.

```
// =====
// SESIÓN 8: PRUEBA INICIAL DE LOS MOTORES
// Vehículo robótico con Arduino UNO y L298N
// =====

// Motor A
int ENA = 10;
int IN1 = 9;
int IN2 = 8;

// Motor B
int ENB = 5;
int IN3 = 7;
int IN4 = 6;

void setup() {
  // Todos los pines del controlador funcionan como salidas
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  // El vehículo inicia detenido
```

```

    Parar();
}

void loop() {
    Adelante();
    delay(5000);

    Atras();
    delay(5000);

    Derecha();
    delay(5000);

    Izquierda();
    delay(5000);

    Parar();
    delay(5000);
}

void Adelante() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 100);

    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 100);
}

void Derecha() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);

    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
}

void Izquierda() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 255);

    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENB, 255);
}

void Atras() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);

    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENB, 255);
}

```

```
void Parar() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, 0);

  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENB, 0);
}
```

16. ¿Qué significa cada parte del código?

Variables de los pines

Indican qué pines de Arduino se conectarán al motor A y al motor B.

Función setup()

Se ejecuta una sola vez cuando Arduino se enciende o reinicia. En ella se configuran los pines como salidas.

Función loop()

Se repite continuamente mientras Arduino permanezca encendido. En el programa de prueba llama a cada movimiento en un orden específico.

digitalWrite()

Envía un estado digital de encendido o apagado a un pin.

analogWrite()

Envía una señal PWM para controlar la velocidad del motor.

delay()

Detiene temporalmente la ejecución. El número se expresa en milisegundos: 1000 equivalen a 1 segundo.

17. Procedimiento para cargar el código

25. Conecten Arduino al computador mediante el cable USB.
26. Abran Arduino IDE.
27. Creen un nuevo programa.
28. Copien el código.
29. Revisen que no falten llaves, puntos y comas o paréntesis.
30. Seleccionen la tarjeta Arduino UNO.
31. Seleccionen el puerto correspondiente.
32. Presionen Verificar.
33. Corrijan los errores que aparezcan.

34. Presionen Subir.
35. Esperen el mensaje de carga finalizada.
36. Desconecten el cable USB antes de mover el vehículo libremente.

18. Código 2: programa completo para el control Bluetooth

Este será el código final del proyecto. En la sesión 8 debe leerse, analizarse y guardarse. La conexión del módulo HC-06 y las pruebas desde el teléfono se realizarán completamente en la sesión 9.

```
#include <SoftwareSerial.h>

// Comunicación con el módulo Bluetooth HC-06
// Arduino recibe por el pin 4 y transmite por el pin 2
SoftwareSerial BT(4, 2);

// Motor A
int ENA = 10;
int IN1 = 9;
int IN2 = 8;

// Motor B
int ENB = 5;
int IN3 = 7;
int IN4 = 6;

void setup() {
  Serial.begin(9600);
  BT.begin(9600);

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  Parar();
}

void loop() {
  if (BT.available()) {
    char dato = BT.read();
    Serial.println(dato);

    switch (dato) {
      case 'a':
        Adelante();
        break;
      case 'r':
        Atras();
        break;
      case 'd':
        Derecha();
        break;
      case 'i':
        Izquierda();
        break;
      case 'p':
        Parar();
    }
  }
}
```

```

        break;
    default:
        Parar();
        break;
    }
}

void Adelante() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 255);

    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
}

void Derecha() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);

    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
}

void Izquierda() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 255);

    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENB, 255);
}

void Atras() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 255);

    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENB, 255);
}

void Parar() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, 0);

    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 0);
}

```

19. ¿Cómo interpreta Arduino las órdenes?

Letra recibida	Acción
a	Avanzar
r	Retroceder
d	Girar a la derecha
i	Girar a la izquierda
p	Detenerse

La instrucción break finaliza cada caso para evitar que se ejecuten otras acciones.

20. Prueba inicial controlada

37. Levanten el chasis para que las ruedas queden en el aire.
38. Enciendan el sistema.
39. Observen el giro de las ruedas.
40. Comprueben cada función.
41. Apaguen inmediatamente si un cable se calienta, un motor no gira, aparece olor a quemado, una rueda está bloqueada o Arduino se reinicia repetidamente.

Comprobaciones

Avanzar: Las dos ruedas deben impulsar el vehículo en la misma dirección.

Retroceder: Las dos ruedas deben invertir el movimiento.

Derecha: Las ruedas deben producir un giro hacia la derecha.

Izquierda: Las ruedas deben producir un giro hacia la izquierda.

Parar: Ambos motores deben detenerse.

21. Solución de problemas

El código no carga

- Comprueben la selección de Arduino UNO.
- Revisen el puerto USB y el cable.
- Verifiquen llaves, puntos y comas.
- Comprueben que ningún programa esté utilizando el puerto.

Un motor no gira

- Revisen las conexiones del motor.
- Comprueben los terminales del L298N.
- Revisen los pines ENA o ENB.
- Verifiquen la fuente de alimentación.
- Comprueben que la rueda no esté bloqueada.

El vehículo avanza en sentido contrario

- Intercambien los dos cables del motor correspondiente o inviertan los valores HIGH y LOW en el código.

El vehículo gira cuando debería avanzar

- Uno de los motores está girando en sentido contrario. Revisen su conexión o modifiquen la función Adelante().

Los motores funcionan con poca fuerza

- Comprueben el estado de las baterías.
- Revisen la conexión de alimentación.
- Verifiquen los valores de analogWrite().
- Reduzcan el peso excesivo y revisen el roce de las ruedas.

22. Seguridad durante la práctica

- No manipulen el circuito mientras esté energizado.
- Mantengan las baterías desconectadas durante el montaje.
- No unan directamente los terminales positivo y negativo.
- No conecten los motores directamente a los pines de Arduino.
- No dejen cables sin aislamiento.
- No fuercen las ruedas.
- Apaguen el vehículo antes de levantarlo.
- Consulten al docente antes de modificar la fuente de alimentación.

23. Diario de ingeniería

Nombre del equipo:

Ingeniería mecánica:

Ingeniería electrónica:

Ingeniería de programación:

Avances logrados:

Dificultades encontradas:

Soluciones propuestas:

Participación del equipo:

Pregunta de participación: ¿Todas las personas pudieron manipular materiales, programar o aportar ideas?

24. Evidencias del proyecto

- Un dibujo o fotografía del chasis inicial.
- Una fotografía del montaje de los motores.
- Una fotografía de Arduino y el L298N instalados.
- Una copia del código.
- El registro de errores.
- Las soluciones aplicadas.
- Una reflexión sobre el trabajo colaborativo.

25. Reto de programación

Modifiquen el código de prueba para que el vehículo ejecute esta secuencia:

42. Avance durante 3 segundos.
43. Se detenga durante 1 segundo.
44. Gire a la derecha durante 2 segundos.
45. Avance durante 3 segundos.
46. Gire a la izquierda durante 2 segundos.
47. Se detenga completamente.

```
void loop() {
  // Escriban aquí la secuencia del reto
}
```

26. Reflexión sobre el enfoque de género

Dialoguen en equipo:

48. ¿Quién tomó inicialmente las decisiones del proyecto?
49. ¿Todas las personas tuvieron la oportunidad de usar las herramientas?
50. ¿Quién trabajó con el código?
51. ¿Quién realizó las conexiones?
52. ¿Se distribuyeron las responsabilidades de manera equitativa?
53. ¿Qué cambios necesita el equipo para mejorar la participación?

Propósito de la reflexión

El objetivo no es señalar culpables, sino reconocer hábitos que pueden limitar la participación y establecer acuerdos para que todas las personas puedan desarrollar habilidades STEM.

27. Preguntas de comprensión

54. ¿Por qué se considera que el chasis es el esqueleto del vehículo?
55. ¿Qué función cumple el módulo L298N?
56. ¿Por qué los motores no deben conectarse directamente a los pines de Arduino?
57. ¿Qué función cumplen los pines ENA y ENB?
58. ¿Qué ocurre cuando se invierten los estados HIGH y LOW de un motor?
59. ¿Cuál es la diferencia entre digitalWrite() y analogWrite()?
60. ¿Para qué se utiliza la función setup()?
61. ¿Qué característica tiene la función loop()?
62. ¿Qué función cumple el comando delay()?
63. ¿Cómo puede comprobarse que un movimiento del vehículo es correcto?
64. ¿Por qué es importante registrar los errores encontrados?
65. ¿Cómo contribuyó cada rol al desarrollo del prototipo?
66. ¿Todas las personas del equipo participaron en las decisiones?
67. ¿Qué aplicación podría tener este vehículo en el contexto rural?

28. Producto de la sesión

- Chasis ensamblado.
- Dos motores instalados.
- Dos ruedas y rueda loca funcionales.
- Portapilas e interruptor colocados.
- Arduino UNO instalado.
- L298N instalado.
- Conexiones de control realizadas.
- Código de prueba cargado.
- Código Bluetooth guardado para la sesión 9.
- Diario de ingeniería actualizado.
- Distribución equitativa de responsabilidades.

29. Lista de verificación

Criterio	Sí	En proceso
Los motores están firmemente instalados	☐	☐
Las ruedas giran libremente	☐	☐
La rueda loca funciona correctamente	☐	☐
El chasis permanece equilibrado	☐	☐
Arduino está protegido y sujeto	☐	☐
El L298N está correctamente instalado	☐	☐
Los cables están organizados	☐	☐
Los pines coinciden con el código	☐	☐
El programa fue verificado	☐	☐
El código fue cargado en Arduino	☐	☐
Los integrantes cumplieron sus roles	☐	☐
Hubo participación equitativa	☐	☐
El equipo registró dificultades y soluciones	☐	☐

30. Cierre de la sesión

Construir un robot no consiste únicamente en unir piezas. Cada decisión mecánica afecta las conexiones electrónicas y cada conexión debe corresponder con una instrucción del programa.

En esta sesión el vehículo comenzó a adquirir forma y movimiento. El chasis se convirtió en la base del sistema, los motores permitieron generar desplazamiento, el L298N hizo posible controlar la potencia y Arduino ejecutó las instrucciones programadas.

En la sesión 9 se completará el montaje, se conectará el módulo Bluetooth HC-06, se configurará la aplicación móvil y se realizarán las pruebas de avance, retroceso, giro y parada. El objetivo será lograr que el prototipo responda correctamente a las órdenes enviadas desde el teléfono celular.

Documento elaborado a partir del contenido pedagógico de la Sesión 8 y de la guía técnica adjunta sobre un coche robot controlado por Bluetooth con Arduino UNO.