



Universidad Austral de Chile

FACULTAD DE CIENCIAS DE LA INGENIERÍA

**IMPLEMENTACIÓN DE SISTEMAS
EMBEBIDOS EN UN VEHÍCULO
AUTÓNOMO: DISEÑO Y
DESARROLLO**

ELEP 222 Sistemas Embebidos

Integrantes:
Pablo Cárdenas, Luis Vera.

PROFESOR
José Mardones

Índice

1. Introducción	2
2. Identificación de la problemática	3
3. Desafíos del Proyecto	3
4. Materiales	4
4.1. Smart Car - 4WD	4
4.2. Arduino Uno	5
4.3. XBee Pro S2C	6
4.4. XBee Explorer	7
4.5. Adaptador XBee para protoboard	7
4.6. Módulo L298N	8
4.7. Power Bank 3 Ultra Compact 10000mAh	9
4.8. Batería 9V	11
4.9. Joystick Arduino	11
5. Desarrollo del vehículo	12
5.1. Estableciendo conexión entre los dispositivos XBee	12
5.2. Diseño del control remoto	14
5.3. Diseño del vehículo	15
5.4. Montaje del vehículo	17
5.5. Montaje del sistema de control remoto	21
5.6. Código Arduino para el control remoto	22
5.7. Código Arduino para el vehículo	23
6. Conclusión	26
7. Anexos	27

1. Introducción

En el ámbito de los sistemas embebidos, el desarrollo de proyectos prácticos es esencial para comprender y aplicar los conceptos teóricos aprendidos en la asignatura. En este contexto, nuestro proyecto tiene como objetivo principal la creación de un automóvil utilizando la placa de Arduino Uno como una forma de explorar y aplicar los principios de los sistemas embebidos en un entorno tangible y emocionante.

Con el fin de obtener una visión global del plan de desarrollo para este proyecto, se ha elaborado un diagrama (Fig.1) que contempla los principales actuadores y controladores clave que serán empleados.

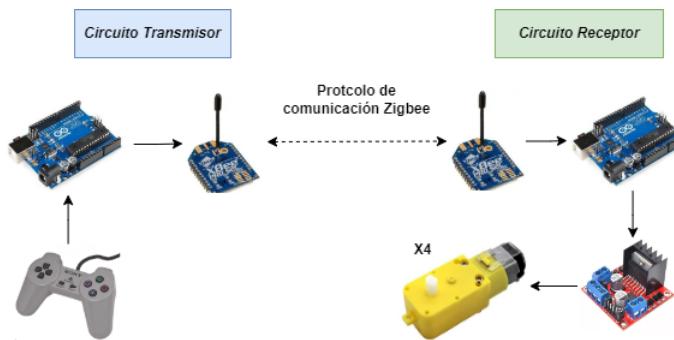


Figura 1: Diagrama general de los circuitos receptor y transmisor a desarrollar.

El automóvil que diseñaremos estará equipado con 4 motores DC controlados mediante un puente H y un Arduino uno, el cual nos brinda la capacidad de controlar la velocidad y dirección de los motores de manera eficiente. Adicionalmente, incorporaremos la placa de XBee en nuestro proyecto. Con la placa de XBee, podremos establecer una comunicación inalámbrica mediante el protocolo de comunicación Zigbee entre el automóvil (receptor) y un dispositivo externo (transmisor), como un control remoto o un sistema de monitoreo. Esta funcionalidad nos brinda la oportunidad de interactuar con el automóvil de forma remota y transmitir información en tiempo real, lo cual amplía nuestras posibilidades de control y supervisión del automóvil.

Este proyecto nos brindará la oportunidad de comprender y abordar desafíos técnicos relacionados con el diseño del hardware, la optimización del software y la gestión eficiente de los recursos disponibles en la placa de Arduino. A través de la resolución de problemas prácticos, desarrollaremos habilidades de programación, electrónica y diseño de sistemas embebidos.

2. Identificación de la problemática

La problemática que nos llevó como equipo a desarrollar este proyecto del automóvil radica en la necesidad de aplicar nuestros conocimientos teóricos de la asignatura de sistemas embebidos en un contexto práctico y emocionante. Como estudiantes de la asignatura de sistemas embebidos, reconocemos que la experiencia de trabajo en proyectos reales es fundamental para comprender a fondo los conceptos y principios aprendidos en el aula.

Nos sentimos intrigados por la posibilidad de utilizar la placa de XBee en nuestro proyecto. La integración de la placa de XBee nos permitirá establecer una comunicación inalámbrica entre el automóvil y un dispositivo externo, como un control remoto o un sistema de monitoreo. Esta característica nos brinda nuevas posibilidades y nos desafía a explorar las capacidades de comunicación inalámbrica en nuestro proyecto.

Además, este proyecto nos brinda la oportunidad de enfrentar desafíos técnicos específicos, como el diseño del hardware, la optimización del software y la gestión eficiente de los recursos en la placa de Arduino. Estos desafíos nos permitirán desarrollar habilidades prácticas y adquirir experiencia en áreas clave como programación, electrónica y diseño de sistemas embebidos.

3. Desafíos del Proyecto

Durante el desarrollo e implementación del automóvil controlado con Arduino utilizando XBee para la comunicación, surgieron varios desafíos que serán desarrollados durante la elaboración del proyecto. A continuación, se presentan los principales desafíos identificados y que serán desarrollados durante la elaboración del proyecto:

- Implementación de la comunicación XBee: Será necesario abordar el desafío de integrar y configurar los módulos XBee para establecer una comunicación inalámbrica estable y confiable. Esto requerirá configurar los parámetros de red, asignar direcciones a los nodos y garantizar una correcta sincronización entre el transmisor y el receptor.
- Control y sincronización de los movimientos del automóvil: Durante la siguiente etapa del proyecto, será crucial lograr un control preciso y sincronizado de los motores del automóvil. Esto implicará diseñar un algoritmo de control que traduzca los comandos recibidos a través de

XBee en movimientos adecuados, como avanzar, retroceder, girar a la derecha o a la izquierda.

- Optimización del consumo de energía: El uso eficiente de la energía será otro desafío a considerar en la siguiente fase del proyecto.
- Programación del Arduino: La programación del Arduino será esencial para controlar los motores de engranajes DC. Se deberá utilizar el lenguaje de programación de Arduino y el entorno de desarrollo integrado (IDE) para escribir el código necesario. La programación debe permitir enviar señales adecuadas a los controladores de los motores, controlando la velocidad y la dirección del giro de cada motor.
- Pruebas y optimización: Es importante realizar pruebas exhaustivas del sistema de control de motores para garantizar su funcionamiento correcto y confiable. Durante las pruebas, se pueden identificar posibles problemas, como vibraciones excesivas, desviaciones en la trayectoria o falta de respuesta a ciertos comandos. Estos problemas deberán ser abordados y se deben realizar ajustes y optimizaciones en el código y en la configuración de los motores para mejorar el rendimiento general.

4. Materiales

4.1. Smart Car - 4WD

El kit del automóvil cuenta con un valor de 20 mil pesos y fue adquirido a través de la plataforma de comercio electrónico Mercado Libre. Este kit incluye los siguientes componentes:

- 2 Chasis acrílicos con agujeros de montaje para facilitar la instalación de sensores, controladores y otros componentes necesarios en el proyecto.
- Tornillos necesarios para el ensamblaje y fijación de los componentes en el chasis.
- 4 Motores con caja de engranaje reductora. Estos motores están equipados con ruedas de 65 mm de diámetro y tienen las siguientes características técnicas:
 - Tensión de funcionamiento: DC 6V.
 - Corriente de funcionamiento: 120 mA.

- Tasa de reducción: 48:1.
 - RPM (con los neumáticos instalados): 240.
- Carcasa para 4 pilas AA, que proporciona la alimentación necesaria para el funcionamiento del automóvil.



Figura 2: Kit Chasis Robot De Cuatro Ruedas.

4.2. Arduino Uno

Se planea emplear dos placas Arduino Uno (Fig.3) durante el proceso de diseño del vehículo. La primera placa se instalará en el circuito transmisor y será responsable de recibir las señales de control emitidas por el usuario. Por otro lado, la segunda placa se integrará en el circuito receptor y se encargará de ejecutar los movimientos ordenados por el usuario para controlar los motores.



Figura 3: Arduinos Uno.

Especificaciones técnicas:

- Voltaje de operación: 5V.
- Microcontrolador: ATmega328P.
- Voltaje de entrada límite: 6-20V.
- Pines digitales de entrada/salida: 14 (de los cuales 6 pueden generar señales PWM).

4.3. XBee Pro S2C

Se tiene previsto la incorporación de dos módulos de comunicación inalámbrica XBee Pro S2C (Fig.4) basados en tecnología Zigbee, los cuales posibilitan la transmisión de datos sin cables entre dispositivos, optimizando la comunicación en sistemas de control y actuadores. Se propone la inclusión de un módulo XBee en el circuito emisor, mientras que otro módulo Xbee se situará en el circuito receptor.



Figura 4: Módulos de comunicación inalámbrica XBee Pro S2C.

Especificaciones técnicas:

- Protocolo de comunicación: Zigbee.
- Alcance de comunicación: Hasta 1.6 kilómetros en línea de vista (dependiendo de las condiciones del entorno).
- Voltaje de operación: 2.1V a 3.6V.
- Potencia de transmisión: 63 mW (+18 dBm).

4.4. XBee Explorer

El XBee Explorer es un dispositivo utilizado para facilitar la conexión y programación de los módulos XBee. Actúa como un adaptador que permite conectar los módulos XBee a una computadora mediante un puerto USB. El XBee Explorer proporciona una interfaz conveniente para configurar y monitorear los módulos XBee, así como para transferir datos entre la computadora y los dispositivos inalámbricos conectados.



Figura 5: Módulo XBee Explorer.

Especificaciones técnicas:

- Voltaje de operación: Compatible con voltajes de 3.3V y 5V, seleccionables mediante un puente de selección.
- Corriente máxima: Dependerá de las capacidades del módulo XBee utilizado, ya que el adaptador XBee Explorer no tiene limitaciones adicionales de corriente.

4.5. Adaptador XBee para protoboard

El adaptador XBee es una placa de expansión diseñada para facilitar la conexión y la utilización de los módulos XBee. Nos permitirá poder implementar los módulos XBee en una protoboard.



Figura 6: Adaptador XBee para protoboard.

Especificaciones técnicas:

- Voltaje de operación: El adaptador XBee para protoboard suele admitir un rango de voltaje de operación amplio, como 3.3V o 5V, dependiendo del tipo de módulo XBee utilizado. Esto permite que se integre con diferentes configuraciones de voltaje en el circuito.
- Pines de conexión: El adaptador XBee para protoboard cuenta con pines de conexión en ambos lados del adaptador. Estos pines son compatibles con las configuraciones estándar de protoboard, lo que facilita su inserción y conexión con otros componentes y circuitos.

4.6. Módulo L298N

El módulo L298N es un componente electrónico utilizado para el control de motores en proyectos de robótica y automatización, incluyendo el desarrollo del automóvil controlado con Arduino. Este módulo se encarga de proporcionar la potencia y el control necesario para operar los motores de manera eficiente y segura. El módulo L298N es un controlador de motor de doble puente H, lo que significa que puede controlar la dirección y la velocidad de dos motores de corriente continua (DC) de forma independiente.

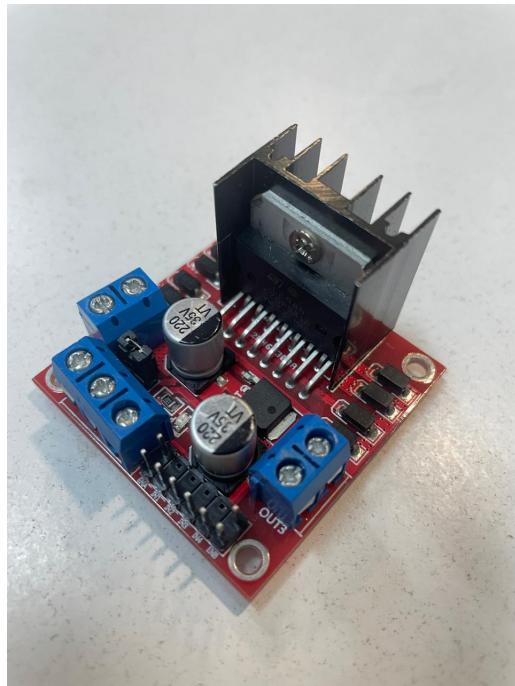


Figura 7: Módulo L298N.

Características técnicas del módulo L298N:

- Voltaje de operación: 5V a 35V.
- Corriente máxima por canal: 2A.
- Pines de conexión: 4 pines de control de dirección y 4 pines de control de velocidad (14 pines en total, incluyendo alimentación y tierra).
- Modos de funcionamiento: Control de velocidad y control de dirección.

4.7. Power Bank 3 Ultra Compact 10000mAh

Para la alimentación de los motores se optó por utilizar una batería portátil Power Bank 3 Ultra Compact de 10000mAh, la cual nos proporcionará realizar mas pruebas con el vehículo sin riesgo a quedarnos sin energía para la alimentación, ya que es posible recargar la batería.



Figura 8: Batería portátil Power Bank 3 Ultra Compact 10000mAh.

Especificaciones técnicas:

- Batería: Polímero de litio.
- Temperatura de funcionamiento: 5°C - 35°C.
- Capacidad nominal: 5500 mAh 5.1V/ 3A.
- Dimensión: 90 x 63.9 x 24.4 mm
- Puertos de entrada: Micro-USB / USB- C
- Puertos de salida: USB-C x 1, USB-A x 2.
- Entrada: Micro-USB 5V / 2A, 9V / 2A max 18W USB-C 5V / 3A, 9V / 2.5A máximo 22.5W
- Salida: USB-A 5V / 2.4A, 9V / 2.5A, 12V / 1.85A máximo 22.5W USB-C 5V / 3A, 9V / 2.5A, 12V / 1.85A máximo 22.5W Salida múltiple 5V / 3A salida múltiple al mismo tiempo
- Tiempo de carga: Aprox. 6 horas con cargador de 5 V / 2 A.

4.8. Batería 9V

Se emplea una batería de 9V alcalina Duracell para la alimentación del Arduino y el XBee que están posicionados en el vehículo.



Figura 9: Batería alcalina 9V Duracell.

4.9. Joystick Arduino

El dispositivo encargado de comandar los movimientos del vehículo será un Joystick compatible con Arduino.



Figura 10: Joystick Arduino.

Especificaciones técnicas:

- Tensión de trabajo: 5VDC.
- Dimensiones: 4.0 x 2.6 x 3.2 cm.

5. Desarrollo del vehículo

5.1. Estableciendo conexión entre los dispositivos XBee

Para poder establecer comunicación entre los dispositivos XBee, se debe conectar los XBee al computador (Fig.11), para esto se hace uso de los XBee Explorer (Fig.5), además, se utilizó el programa XCTU (Fig.12), en el cual es posible definir cual de los dos XBee corresponde al **coordinador** y cual es el **router o end device** y realizar su debida configuración.



Figura 11: Conexión dispositivo XBee al computador mediante el dispositivo XBee Explorer.



Figura 12: XCTU: Plataforma de configuración de próxima generación para Digi XBee.

Posteriormente se presentan mediante las Fig.13 y Fig.14 los parámetros utilizados para establecer la comunicación entre los XBee, para el **coordinador** y el **end device**, respectivamente.

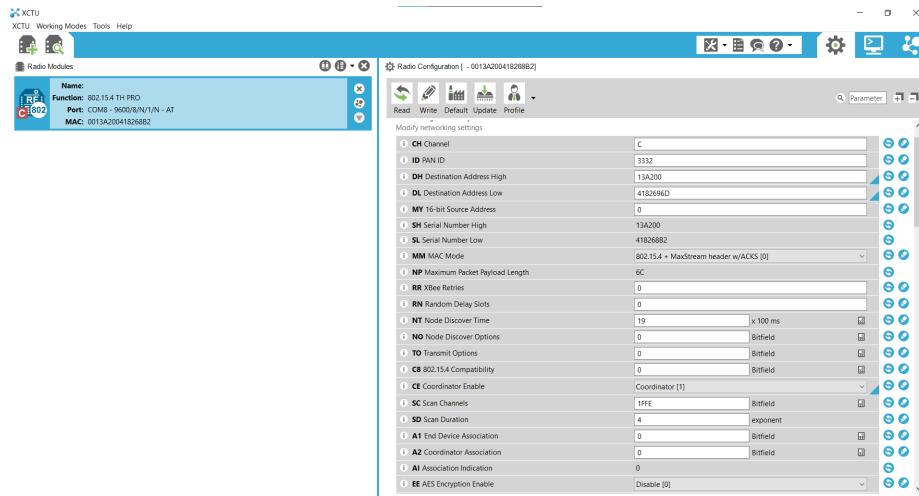


Figura 13: Parámetros para establecer la comunicación entre los XBee mediante la plataforma XCTU: Coordinador.

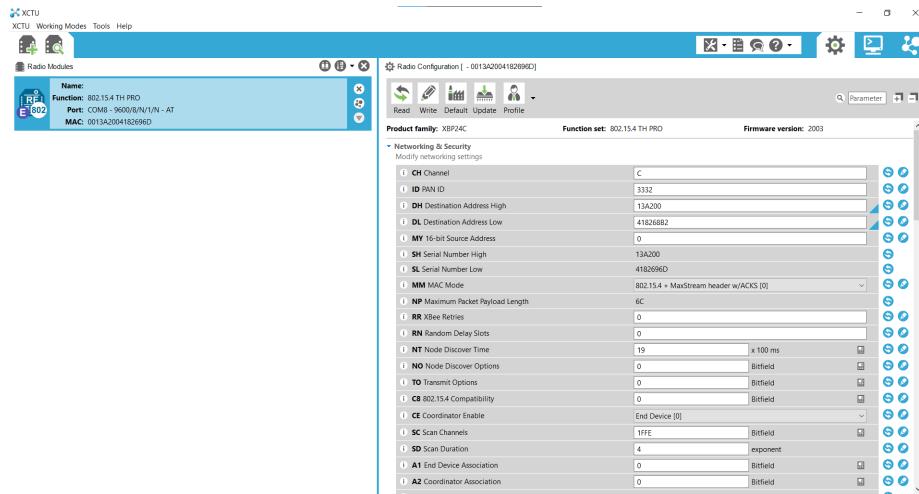


Figura 14: Parámetros para establecer la comunicación entre los XBee mediante la plataforma XCTU: End Device.

Parámetros	XBee Coordinador	XBee End device
CH Channel	C	C
ID (Operating PAN ID)	3332	3332
DH (Destination Address High)	13A200	13A200
DL (Destination Address Low)	4182696D	418268B2
BD (Baud Rate)	9600	9600

Tabla 1: Parámetros relevantes para establecer la comunicación entre los XBee.

Los parámetros que son mas relevantes para establecer la comunicación se pueden observar en la Tabla.1.

El PAN ID debe ser el mismo para cada XBee, mientras que los valores de DH y DL corresponden a los valores impresos al reverso de los XBee, el coordinador debe tener el DH y DL correspondientes al XBee con el cual se va a establecer comunicación, caso inverso para el end device.

5.2. Diseño del control remoto

El diagrama circuital para el control se puede observar en la Fig.15

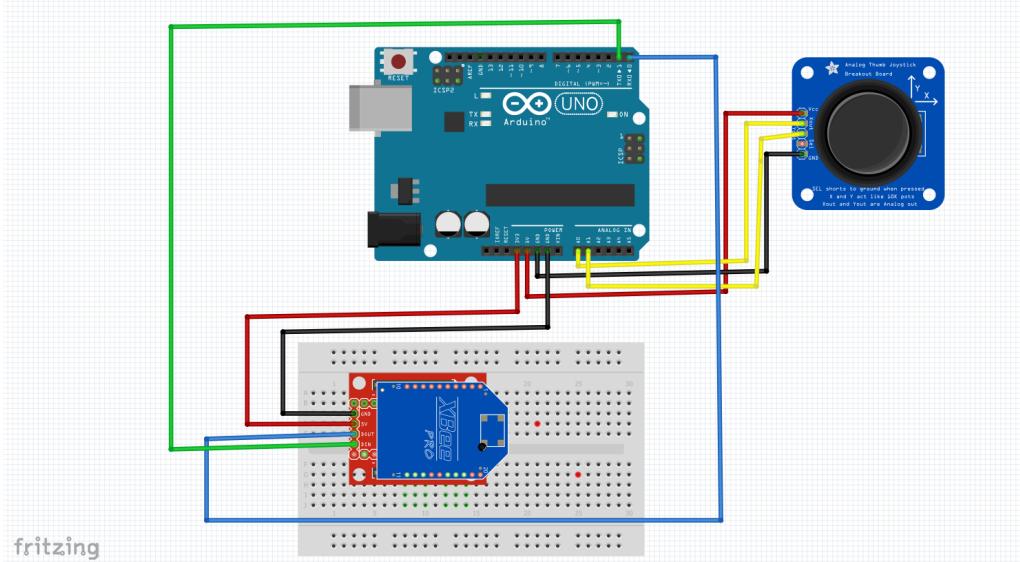


Figura 15: Circuito eléctrico para el control remoto. Los cables rojo y negro corresponden a alimentación, los cables verde y azul son para la comunicación del XBee Coordinador con el Arduino (DOUT al Pin digital 0 Rx y DIN al pin digital 1 Tx del arduino, respectivamente), mientras que los cables amarillos son para el control de los ejes X e Y, que deben ir conectados a los pines analógicos del Arduino (en este caso A0 y A1)

Para el control remoto simplemente se hará uso de un Joystick, el cual al ser desplazado hacia arriba, hará que el vehículo avance, activando en el mismo sentido los 4 motores. Si el Joystick es desplazado hacia abajo, los 4 motores se activarán en sentido opuesto al caso anterior. Si es desplazado a la izquierda o derecha, solamente se activará un par de motores de los costados, mientras que el otro par se mantendrá en reposo, de esta manera el vehículo cambiará su orientación hacia la derecha o izquierda. El XBee Coordinador será el encargado de enviarle todo los desplazamientos realizados al Joystick para controlar el vehículo al Xbee end device.

5.3. Diseño del vehículo

El diagrama circuital para el vehículo se puede observar en la Fig.16.

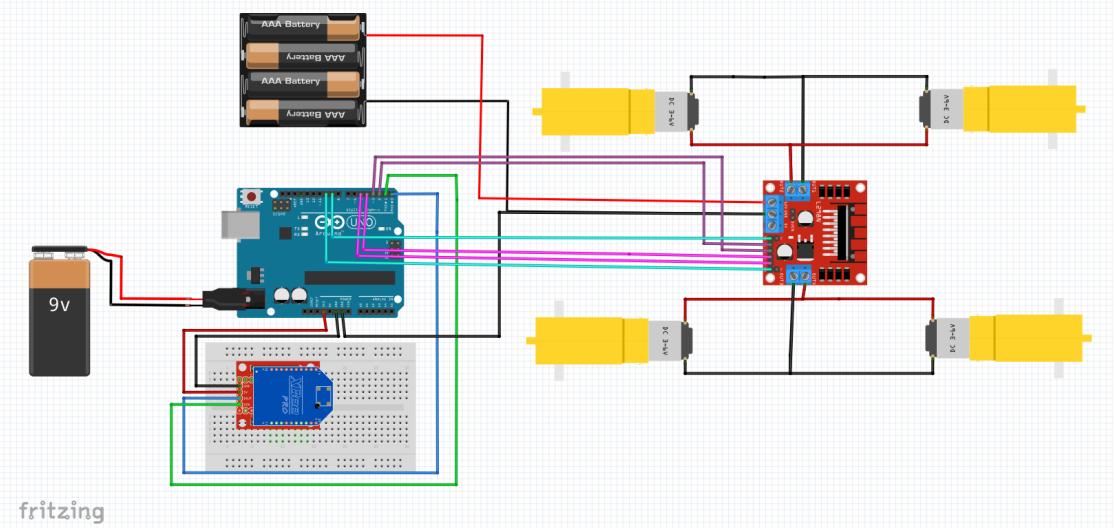


Figura 16: Circuito eléctrico para el vehículo. Los cables rojos y negros corresponden a alimentación, azul y verde para la comunicación del XBee end device con el Arduino (DOUT al pin digital 0 Rx y DIN al pin digital 1 Tx del Arduino, respectivamente), los cables celestes corresponden a los Enable (ENA) de los motores 1 y 2, los cables morados y rosados son para controlar la dirección de giro de los motores 1 y 2, respectivamente.

Se tiene una batería de 9V para la alimentación del Arduino y el XBee, mientras que para los motores se utilizó una batería portátil de — que viene representada por el porta baterías 4 slots. Se optó por separar las alimentación del Arduino de los motores a modo de buena práctica para evitar problemas de consumos de energía. Cabe destacar que el común del controlador de motores debe tener el común unificado, con el Arduino, esto quiere decir que el cable negativo de la alimentación y el GND del Arduino, deben ir juntos en el COM del L298N.

Es importante destacar que solamente se utilizó un controlador para dos motores, pero en este caso, se conectan juntos los motores de los costados, de esta manera es posible garantizar el giro de los vehículos.

El XBee end device va a recibir las señales de control enviadas por el XBee Coordinador para el movimiento del vehículo.

5.4. Montaje del vehículo

En esta sección se presentará el montaje físico del vehículo.

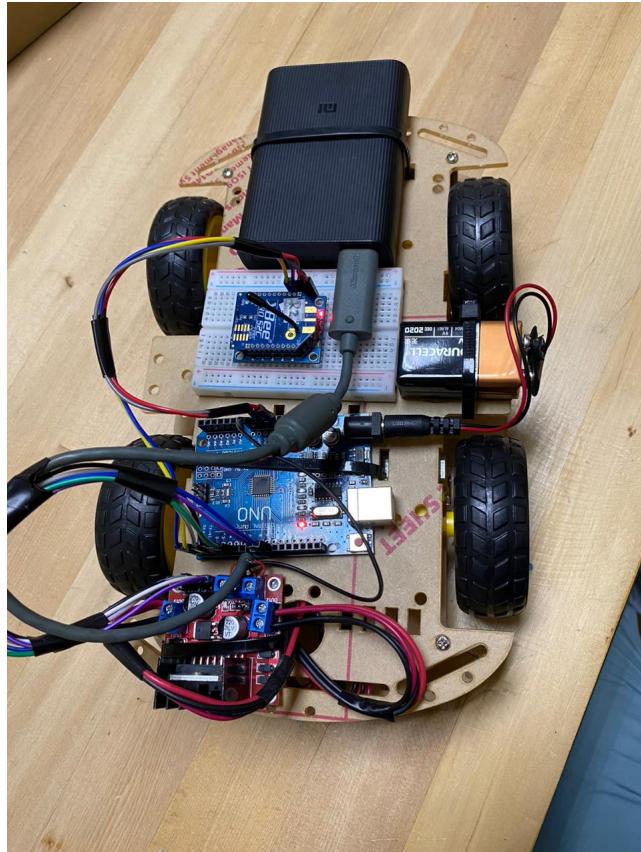


Figura 17: Fotografía del montaje realizado del vehículo. En la figura se puede observar una batería portátil para la alimentación de los motores, el controlador de motores, el módulo XBee (End device), la tarjeta de desarrollo Arduino y una batería para alimentar a estos dos últimos, además del correspondiente cableado de alimentación y comunicación.

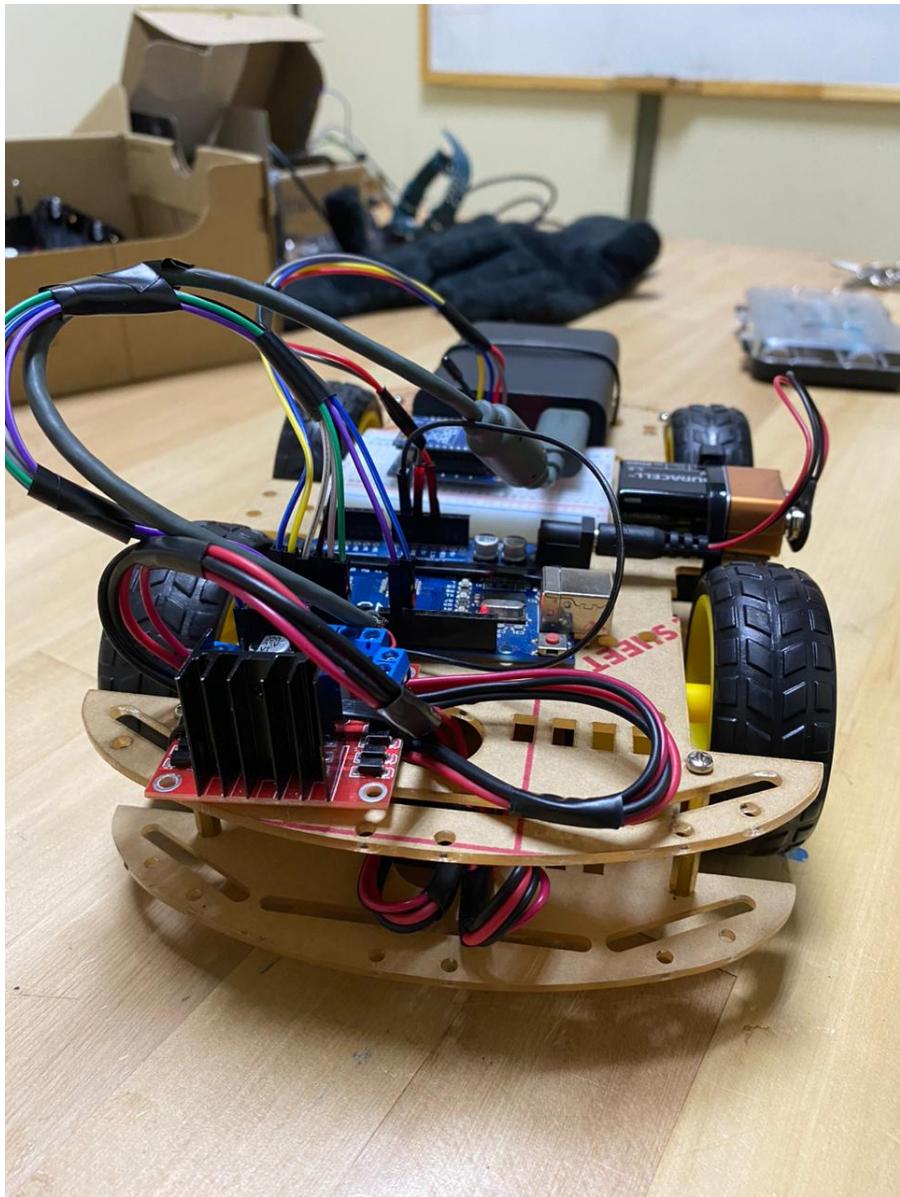


Figura 18: Fotografía del montaje realizado del vehículo.(2)

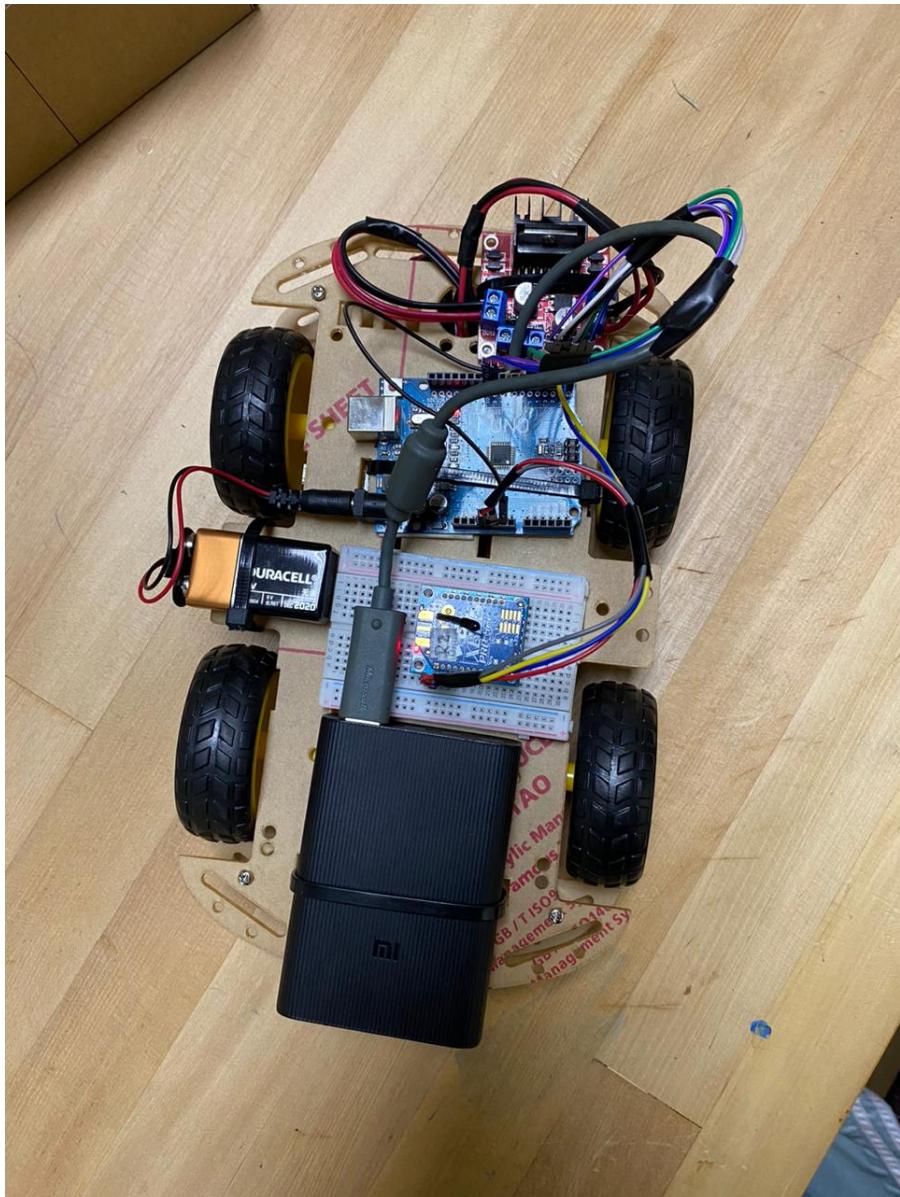


Figura 19: Fotografía del montaje realizado del vehículo.(3)

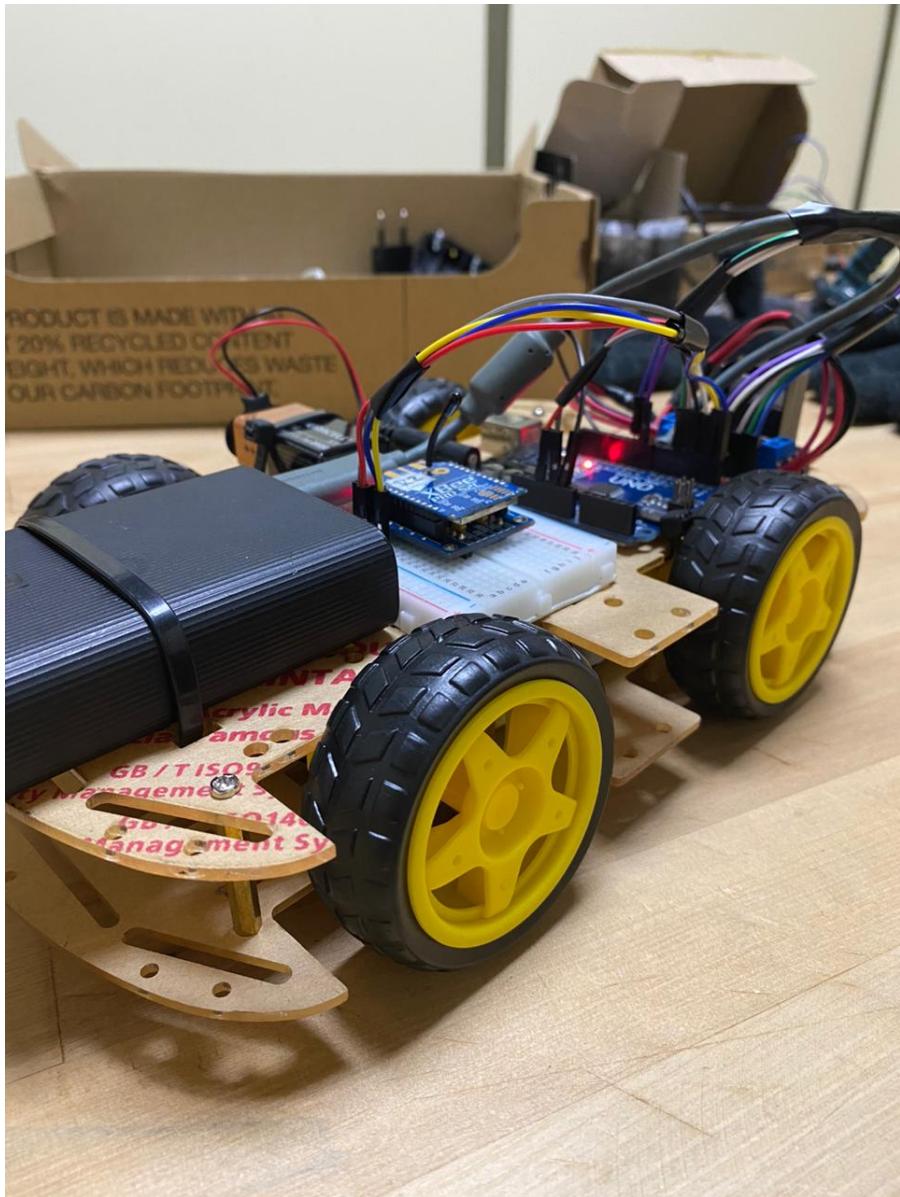


Figura 20: Fotografía del montaje realizado del vehículo.(4)

5.5. Montaje del sistema de control remoto

En esta sección se presentará el montaje físico del sistema de control remoto del vehículo.

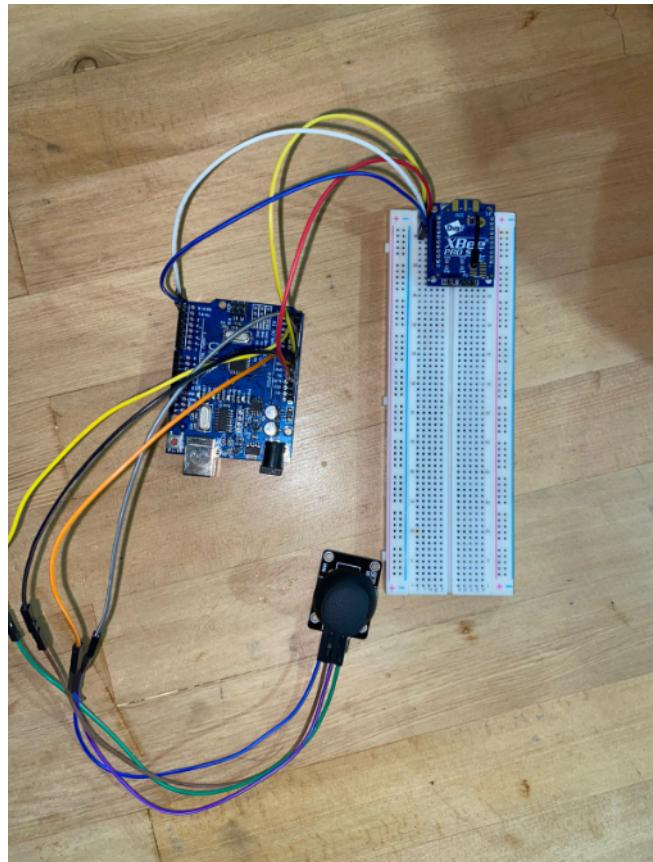


Figura 21: Fotografía del montaje realizado del control remoto. Se puede observar el modulo XBee (Coordinador), el Joystick y el Arduino. La alimentación que se utilizará en esta ocasión será proporcionada por la computadora.

5.6. Código Arduino para el control remoto

```
1 #include <SoftwareSerial.h>
2 const int joystickXPin = A0; // Pin analogico para
3 // el eje X del joystick
4 const int joystickYPin = A1; // Pin analogico para
5 // el eje Y del joystick
6
7 void setup() {
8     // Configurar la comunicacion serial con el modulo
9     // XBee
10    Serial.begin(9600);
11
12 void loop() {
13     int joystickXValue = analogRead(joystickXPin);
14     int joystickYValue = analogRead(joystickYPin);
15     // Enviar comando para avanzar durante 2 segundos
16     if (joystickYValue > 520 && joystickYValue <=
17         1023) {
18         // Enviar comando para avanzar
19         Serial.write('F');
20     } else if (joystickYValue >=0 && joystickYValue <
21         480) {
22         // Enviar comando para retroceder
23         Serial.write('B');
24     } else if (joystickXValue > 520 && joystickXValue <
25         1023) {
26         // Enviar comando para mover a la derecha
27         Serial.write('R');
28     } else if (joystickXValue >=0 && joystickXValue <
29         480) {
30         // Enviar comando para mover a la izquierda
31         Serial.write('L');
32     } else if (joystickXValue > 480 && joystickXValue <
33         520 && joystickYValue > 480 && joystickYValue <
34         520) {
35         // Enviar comando para detenerse
36         Serial.write('S');
37     }
38 }
```

```

30 // Esperar un breve periodo de tiempo
31 delay(100);
32 }

```

Este código al ser subido al Arduino del control remoto nos permitirá controlar los movimientos del vehículo según sean los movimientos del joystick. Para la configuración del joystick se utilizó como referencia la Fig.22 para determinar los intervalos de activación de los ejes.



Figura 22: Intervalos de desplazamiento del joystick.

5.7. Código Arduino para el vehículo

```

1 #include <SoftwareSerial.h>
2
3 // Definicion de los pines del controlador L298N
4 const int motorA_1 = 2; // Pin IN1 del Motor A
5 const int motorA_2 = 3; // Pin IN2 del Motor A
6 const int motorB_1 = 4; // Pin IN3 del Motor B
7 const int motorB_2 = 5; // Pin IN4 del Motor B
8 const int enableA = 9; // Pin Enable del Motor A
9 const int enableB = 10; // Pin Enable del Motor B
10
11
12 // Definicion de los comandos de control

```

```

13
14 void setup() {
15     // Configurar los pines del controlador L298N como
16     // salidas
17     pinMode(motorA_1, OUTPUT);
18     pinMode(motorA_2, OUTPUT);
19     pinMode(motorB_1, OUTPUT);
20     pinMode(motorB_2, OUTPUT);
21     pinMode(enableA, OUTPUT);
22     pinMode(enableB, OUTPUT);
23
24     // Configurar la comunicacion serial con el modulo
25     // XBee
26     Serial.begin(9600);
27 }
28
29 void loop() {
30     if (Serial.available()) {
31         char cmd = Serial.read();      //Definir una
32         // variable cmd de lectura de la comunicacion
33         // serial entre los modulos XBee
34         Serial.print("Comando recibido: ");
35
36         // Realizar el movimiento correspondiente de los
37         // motores segun el comando recibido
38         if (cmd == 'F') {
39             // Mover hacia adelante
40             digitalWrite(motorA_1, HIGH);
41             digitalWrite(motorA_2, LOW);
42             digitalWrite(motorB_1, HIGH);
43             digitalWrite(motorB_2, LOW);
44             digitalWrite(enableA, HIGH);
45             digitalWrite(enableB, HIGH);
46             delay(50);
47         } else if (cmd == 'B') {
48             // Mover hacia atras

```

```

48     digitalWrite(motorA_1, LOW);
49     digitalWrite(motorA_2, HIGH);
50     digitalWrite(motorB_1, LOW);
51     digitalWrite(motorB_2, HIGH);
52     digitalWrite(enableA, HIGH);
53     digitalWrite(enableB, HIGH);
54     delay(50);
55
56 } else if (cmd == 'R') {
57     // Mover hacia la derecha
58     digitalWrite(motorA_1, LOW);
59     digitalWrite(motorA_2, HIGH);
60     //digitalWrite(motorB_1, HIGH);
61     //digitalWrite(motorB_2, HIGH);
62     digitalWrite(enableA, HIGH);
63     //digitalWrite(enableB, HIGH);
64     delay(50);
65 } else if (cmd == 'L') {
66     // Mover hacia la izquierda
67     //digitalWrite(motorA_1, HIGH);
68     //digitalWrite(motorA_2, LOW);
69     digitalWrite(motorB_1, LOW);
70     digitalWrite(motorB_2, HIGH);
71     //digitalWrite(enableA, HIGH);
72     digitalWrite(enableB, HIGH);
73     delay(50);
74 } else if (cmd == 'S') {
75     // Detener los motores
76     digitalWrite(motorA_1, LOW);
77     digitalWrite(motorA_2, LOW);
78     digitalWrite(motorB_1, LOW);
79     digitalWrite(motorB_2, LOW);
80     digitalWrite(enableA, HIGH);
81     digitalWrite(enableB, HIGH);
82     delay(50);
83 }
84 }
```

6. Conclusión

A partir de este trabajo práctico se pudo adquirir conocimiento de los dispositivos de comunicación inalámbrica XBee. Se han verificar en primera persona las capacidades de esta tecnología que incorpora Zigbee. Se pudieron superar los desafíos de utilizar estos dispositivos establecidos al inicio de este informe, como lo son establecer la comunicación entre los módulos XBee, es importante considerar los parámetros expuestos en secciones anteriores, además, es importante que los dos módulos posean el mismo Firmware al actualizar el dispositivo, caso contrario la comunicación no será garantizada. Otra percance al desarrollo del proyecto, es el desconocimiento de que las entradas Rx y Tx del Arduino deben estar vacías a la hora de subir un Sketch, esto quiere decir que cada vez que un código sea subido al Arduino debe esté no estar conectado con el módulo XBee, de no ser así; compilará un error el monitor serial.

Una vez que fueron superados todo estos detalles, fue posible realizar la comunicación entre los dispositivos XBee, lo que llevo a que el vehículo pueda desplazarse de manera inalámbrica. Debido a esto se concluye que el trabajo fue un éxito, fue posible comandar el vehículo de manera inalámbrica de una manera precisa y estable.

7. Anexos

Evidencia de Trabajo de la Construcción del Proyecto de Automóvil con Arduino y XBee.

En esta sección, presentamos los anexos que respaldan y complementan el informe principal sobre nuestro proyecto de construcción de un automóvil controlado por Arduino y comunicación inalámbrica XBee.

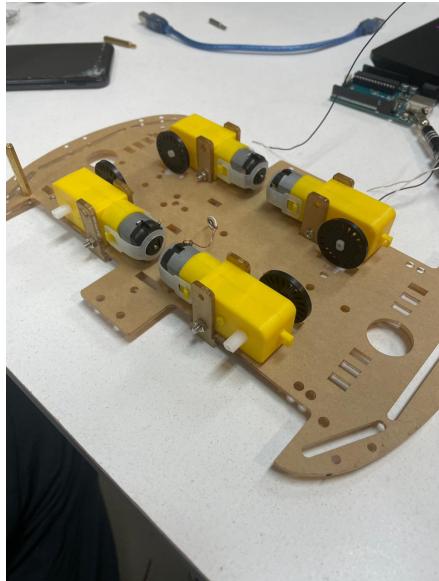


Figura 23: Instalación de los motores DC.

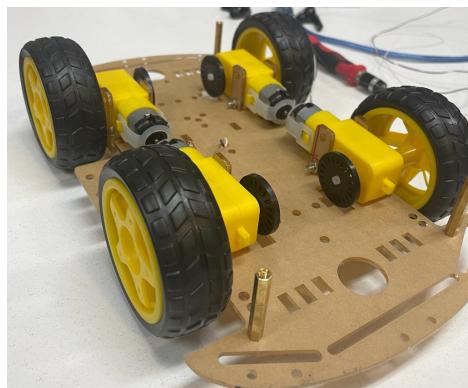


Figura 24: Colocación de las ruedas en el chasis del automóvil.