



DISEÑO E IMPLEMENTACIÓN DE RC Cars Telemetry Challenge

William León

Facultad de Ingeniería, ingeniería en Telecomunicaciones, UMNG.

Resumen—Este artículo presenta el diseño, construcción e implementación de un sistema completo de comunicaciones digitales aplicado a un carro RC, desarrollado como proyecto final del curso de Comunicaciones Digitales de la Universidad Militar Nueva Granada. La solución integra hardware, firmware y software para establecer un enlace inalámbrico robusto entre tres nodos principales: un control remoto basado en NRF24L01, un carro instrumentado con sensores (GPS, MPU6050, TCRT y módulos PWM para servo/ESC) controlado por una Raspberry Pi Pico W, y una base receptora encargada de consolidar, decodificar y publicar la telemetría en un dashboard web en tiempo real.

El trabajo incluye el diseño eléctrico y PCB de los nodos, modelado 3D para validación mecánica, integración del enlace RF en banda ISM de 2.4 GHz, empaquetamiento eficiente de tramas binarias, sistemas de filtrado para detección de línea de meta, y algoritmos para medición de aceleración, velocidad, posición y tiempos de vuelta. También se desarrolló una interfaz web moderna para visualización local de telemetría, acompañada de un servicio HTTP embebido y soporte para envío de datos hacia un servidor central del profesor.

Los resultados demuestran un sistema estable con transmisión periódica de telemetría a 200 ms, tiempos de vuelta consistentes, control responsivo y adecuada coexistencia espectral frente a otras señales presentes en el entorno. El proyecto evidencia la integración práctica de conceptos de comunicaciones digitales, diseño embebido y análisis de señales aplicados a un reto real.

Palabras clave— Telemetría, NRF24L01, Raspberry Pi Pico W, Comunicaciones Digitales, Carro RC, PCB, Enlace Inalámbrico, GPS, MPU6050, IoT, Banda ISM.

Abstract— This paper presents the design, construction, and implementation of a complete digital communication system applied to an RC car, developed as the final project for the Digital Communications course at Universidad Militar Nueva Granada. The solution integrates hardware, firmware, and software to establish a robust wireless link between three main nodes: a handheld controller based on the NRF24L01 transceiver, a fully instrumented car equipped with sensors (GPS, MPU6050, TCRT and PWM-based steering/throttle drivers) controlled by a Raspberry Pi Pico W, and a base station responsible for aggregating, decoding, and publishing telemetry through a real-time web dashboard.

The work includes schematic design, PCB layout for each node, 3D modeling for mechanical validation, RF link configuration in the 2.4 GHz ISM band, efficient binary packet

structures, signal-processing algorithms for lap detection, and computation of acceleration, speed, geolocation, and lap times. A modern web interface was developed to visualize live telemetry, supported by an embedded HTTP server and optional data forwarding to the instructor's backend system.

Experimental results show a stable system capable of transmitting telemetry every 200 ms, providing consistent lap-time detection, responsive vehicle control, and adequate RF coexistence with surrounding wireless signals. This project demonstrates the practical integration of digital communications, embedded system design, and signal analysis concepts applied to a real-world engineering challenge.

I. INTRODUCCIÓN

El estudio y aplicación de las comunicaciones digitales requiere escenarios prácticos que permitan validar conceptos fundamentales como modulación, codificación, tratamiento de ruido, estructura de tramas, sincronización, protocolos y mecanismos de control de errores. En este contexto, el **RC Car Telemetry Challenge 2025**, propuesto en el curso de Comunicaciones Digitales de la Universidad Militar Nueva Granada, establece un entorno real donde los estudiantes deben diseñar, construir y validar un sistema de comunicación inalámbrica robusto, confiable y de baja latencia para un carro a radiocontrol.

El proyecto desarrollado consiste en la implementación integral de un sistema compuesto por tres nodos principales:

- (1) Un **control inalámbrico**.
- (2) Un **carro sensorizado**.
- (3) Una **base receptora** encargada del



procesamiento y visualización de los datos de telemetría.

Cada nodo fue construido desde cero a nivel de hardware, firmware y diseño de PCB, utilizando tecnologías de bajo costo y protocolos de comunicación embebidos. El enlace principal se estableció mediante el transceptor **NRF24L01**, operando en la banda ISM de 2.4 GHz, mientras que la gestión de telemetría, almacenamiento y visualización se realizó con una **Raspberry Pi Pico W**, empleando WiFi para la capa de presentación mediante un dashboard web.

El **carro (nodo móvil)** integra sensores como GPS para posicionamiento, MPU6050 para mediciones inerciales, un sensor óptico TCRT para detección de línea de meta, y módulos PWM para el control del servomotor (dirección) y el ESC (propulsión). Para garantizar estabilidad del sistema, se desarrollaron algoritmos de filtrado, paquetes binarios compactos y rutinas periódicas de muestreo. El carro transmite telemetría estructurada cada 200 ms y envía eventos especiales para reportar vueltas completadas.

El **control inalámbrico** incorpora dos joysticks analógicos cuya información se traduce en comandos de dirección y velocidad. Dichos comandos se encapsulan en tramas ASCII y se transmiten mediante NRF24L01 con auto-acknowledgement, implementando además una lógica de *keep-alive* para asegurar continuidad en escenarios con pérdida temporal del enlace RF.

La **base receptora** cumple tres funciones esenciales:

1. Recibir y decodificar la telemetría enviada por el carro.

2. Reenviar información relevante al servidor del docente mediante solicitudes HTTP POST.
3. Servir un dashboard web en tiempo real, mostrando velocidad, aceleraciones, posición GPS, número de vueltas, historial y estado del enlace inalámbrico.

Como parte del desarrollo, se diseñaron dos PCB: una correspondiente al nodo móvil y otra al nodo fijo (base). El proceso incluyó el esquemático eléctrico, rutado de pistas, selección de *footprints* y validación mediante modelado 3D. Esto permitió optimizar la ubicación física de cada módulo y reducir errores de integración durante la etapa de soldadura.

El trabajo no solo permitió validar los fundamentos teóricos del curso, sino también integrar conocimientos de electrónica, sistemas embebidos, RF, procesamiento de señales digitales, protocolos de comunicación y diseño orientado al desempeño. Los resultados evidencian un sistema funcional, estable y capaz de operar en entornos con interferencia electromagnética, demostrando una correcta implementación de los conceptos de comunicaciones digitales en un escenario aplicado.

A diferencia de proyectos motivados por necesidades industriales o comerciales, esta solución se enmarca en un contexto académico: su propósito es evaluar competencias prácticas adquiridas en clase mediante la implementación real de un enlace de telemetría para un vehículo RC. La motivación principal radica en integrar múltiples tecnologías —control remoto, RF digital, sensores, microcontroladores y telemetría— para construir un sistema funcional que opere en condiciones reales de laboratorio y pista.



II. MARCO TEÓRICO

El diseño de un sistema de telemetría inalámbrica para un vehículo RC requiere la integración de diversos conceptos de comunicaciones digitales, electrónica embebida y sensado. Esta sección presenta los fundamentos teóricos relevantes utilizados en el desarrollo del proyecto.

2.1. Comunicaciones Digitales en Banda ISM de 2.4 GHz

La banda ISM (Industrial, Scientific and Medical) en 2.4 GHz es ampliamente utilizada en aplicaciones de corto alcance debido a su disponibilidad libre de licencia, compatibilidad con múltiples estándares (WiFi, Bluetooth, ZigBee) y facilidad de adopción en microcontroladores de bajo costo. Sin embargo, su uso compartido introduce retos relacionados con interferencia, ruido, colisiones y coexistencia espectral.

El NRF24L01, empleado en este proyecto, opera con modulación GFSK, permitiendo transmisiones digitales de baja potencia con un alcance efectivo entre 10 y 100 metros dependiendo de las condiciones del entorno, la potencia configurada y el diseño de antena.

En el contexto académico, esta banda permite a los estudiantes experimentar de forma práctica con conceptos como:

- Probabilidades de error (BER)
- Sensibilidad del receptor
- Potencia de transmisión
- Retransmisiones automáticas (ARQ)
- Selección de canales libres
- Sincronización y temporización

2.2. Transceptor NRF24L01

El NRF24L01 es un transceptor digital de 2.4 GHz con arquitectura SPI, ampliamente utilizado en aplicaciones de telemetría de bajo costo. Sus principales características relevantes para el proyecto son:

- **Modulación:** GFSK
- **Rango típico:** 10–100 m (dependiendo de antena y entorno)
- **Tasa de datos:** 250 kbps, 1 Mbps o 2 Mbps
- **Tamaño de payload:** 1–32 bytes
- **Auto Acknowledgement (Auto-ACK)**
- **Re-transmisiones automáticas (AUTO-RETRY)**

El uso de dos enlaces independientes (uno para control y otro para telemetría) permite aislar la carga de datos, reduciendo el riesgo de colisiones y manteniendo baja latencia en el control del vehículo.

El proyecto utiliza:

- **Canal 83 (Control)**
- **Canal 100 (Telemetría)**
- **Potencia configurada en 0 dBm**
- **Tasa de datos de 250 kbps**, ideal para robustez en entornos ruidosos.

2.3. Microcontrolador Raspberry Pi Pico W

La Raspberry Pi Pico W, basada en el microcontrolador RP2040, fue seleccionada para el nodo móvil y la base debido a:



- Procesador dual-core Cortex-M0+.
- Reloj de hasta 133 MHz.
- Memoria RAM de 264 KB.
- Interfaz SPI para control del NRF24.
- Interfaz I2C para sensores y pantalla OLED.
- Interfaz UART para GPS.
- Conectividad WiFi integrada para dashboard.

El RP2040 permite ejecutar rutinas de muestreo periódicas, filtrado, empaquetamiento de datos y transmisión con gran eficiencia.

En la base, su módulo WiFi se utiliza para ofrecer un **servidor HTTP embebido** capaz de entregar un dashboard moderno.

2.4. Sensores Utilizados

2.4.1. GPS (Posicionamiento Global).

El módulo GPS proporciona:

- Latitud.
- Longitud.
- Velocidad (algunos modelos)
- Timestamp.

El sistema usa esta información para:

- Telemetría en tiempo real
- Cálculo de trayectoria.
- Registro de ubicación por vuelta.

2.4.2. MPU6050 (Acelerómetro y Giroscopio).

El MPU6050 ofrece mediciones tridimensionales de:

- Aceleración (ax, ay, az).
- Velocidad angular (gx, gy, gz).

Es fundamental para:

- Analizar la dinámica del carro.
- Detectar variaciones bruscas.
- Aplicar filtros de estabilidad (EMA).
- Transmitir telemetría completa al dashboard.

2.4.3 Sensor TCRT5000 (Detección Óptica de Línea de Meta)

El sensor TCRT permite detectar un patrón negro sobre la pista mediante reflexión infrarroja.

En este proyecto es clave para:

- Identificar la línea de meta.
- Detectar eventos válidos de vuelta.
- Iniciar cálculo de lap_time_ms.
- Enviar tramas especiales tipo “L” (lap).

Se aplican algoritmos de:

- Filtrado de señal.
- Ventana temporal.
- Threshold dinámico.
- Validación de bordes para evitar falsos positivos.

2.5. Control del Vehículo RC



El sistema de propulsión del carro utiliza:

- Servomotor para la dirección.
- ESC (Electronic Speed Controller) para el motor brushless o brushed.

Ambos controlados por PWM desde la Raspberry Pi Pico.

Los comandos provienen del transmisor NRF24 via:

JST1:<dir>;JST2:<speed>

2.6. Telemetría Digital

El proyecto implementa dos tipos de tramas binarias:

Trama T (Telemetría periódica):

Incluye:

- Timestamp.
- Aceleraciones.
- Latitud & longitud.
- Velocidad.

Enviada cada 200 ms.

Trama L (Lap event):

Incluye:

- Número de vuelta.
- Tiempo de vuelta.
- Posición GPS.
- Velocidad.

Enviada solo al cruzar meta.

Ambas tramas usan struct.pack() para compactar valores y mantener el payload ≤ 32 bytes.

2.7. Servidor HTTP Embebido y Dashboard Web

La Raspberry Pi Pico W configura un servidor HTTP ligero que sirve:

- / para el dashboard.
- /data para datos JSON.
- /logo.jpg para contenido estático.

El dashboard usa:

- JavaScript (fetch API)
- CSS modernizado
- Actualización cada 1 segundo para mostrar telemetría en tiempo real sin recargar la página.

2.8. Diseño de PCB y Modelado 3D

El proceso de diseño siguió las etapas, haciendo uso de software para circuitos:

1. Esquemático eléctrico.
2. Selección de footprints.
3. Ruteo de pistas.
4. Validación dimensional con datasheets.
5. Marcas de referencia para ensamblaje.
6. Exportación de modelos 3D.
7. Verificación de interferencia mecánica.

Este flujo permite validar el diseño antes de la implementación física, reduciendo errores y costos.

III. OBJETIVOS

Objetivo general



Diseñar, construir e implementar un sistema completo de comunicaciones digitales para un vehículo RC, integrando hardware, firmware y software, que permita realizar control inalámbrico, adquisición de telemetría en tiempo real y visualización mediante un dashboard web, con el fin de aplicar y validar los conceptos fundamentales del curso de Comunicaciones Digitales en un entorno práctico y experimental.

Objetivos específicos

1. Diseñar y ensamblar desde cero el hardware del carro RC, integrando la Raspberry Pi Pico W, el módulo NRF24L01, sensores de navegación (GPS), sensorica inercial (MPU6050), detección óptica (TCRT5000) y actuadores (servo para dirección y ESC para propulsión).
2. Desarrollar un control inalámbrico funcional mediante dos joysticks analógicos y un transceptor NRF24L01 configurado para baja latencia y robustez mediante auto-acknowledgement y retransmisiones automáticas.
3. Implementar un enlace digital de telemetría en la banda ISM de 2.4 GHz, utilizando tramas binarias compactas para transmitir en tiempo real datos de aceleración, velocidad, posición GPS y eventos de vueltas (laps).
4. Construir una estación base capaz de recibir, decodificar y almacenar la telemetría, empleando una Raspberry Pi Pico W para enlazar RF y WiFi, y soportando comunicación bidireccional con un servidor externo del docente mediante HTTP POST.
5. Diseñar un dashboard web responsivo y en tiempo real, que permita visualizar telemetría, historial de vueltas, velocidad, aceleraciones, posición geográfica y estado del enlace inalámbrico, accesible desde cualquier dispositivo conectado a la misma red.
6. Diseñar los circuitos electrónicos y PCB de los nodos móvil y fijo, incluyendo esquemáticos, selección de footprints, ruteo de pistas, modelado 3D y validación mecánica previa a la implementación física.

7. Evaluar el desempeño del sistema aplicado a un entorno de prueba real, midiendo alcance RF, estabilidad del enlace, frecuencia de muestreo efectiva, calidad de telemetría y precisión en la detección de vueltas.
8. Integrar todos los componentes en un sistema cohesivo, demostrando la aplicación de conceptos de comunicaciones digitales, electrónica embebida, procesamiento de señales y diseño orientado al rendimiento en un proyecto de ingeniería.

IV. ARQUITECTURA GENERAL DEL SISTEMA

El sistema propuesto se compone de tres nodos principales que interactúan mediante enlaces inalámbricos independientes y complementarios. La arquitectura fue diseñada para garantizar baja latencia en el control manual, transmisión confiable de telemetría y visualización en tiempo real mediante WiFi. La Fig. 6 presenta el modelo general de interacción entre los tres nodos.

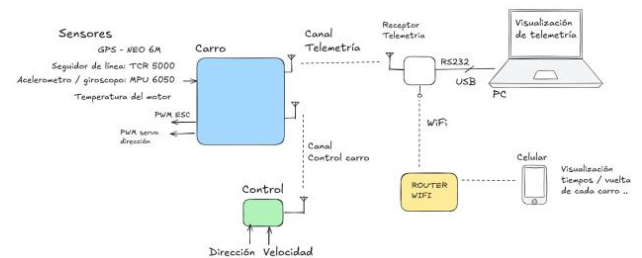


Figura 1. Arquitectura general del sistema.

4.1. Descripción General de la Arquitectura.



El sistema implementa dos enlaces inalámbricos independientes con funciones claramente diferenciadas:

1. **Enlace 1 – Control Manual (NRF24 Canal 83):**
Responsable de la dirección y propulsión del carro mediante tramas ASCII del transmisor.
2. **Enlace 2 – Telemetría (NRF24 Canal 100):**
Responsable del envío de paquetes binarios periódicos desde el carro hacia la base.

Estos enlaces están aislados en canales distintos de 2.4 GHz, evitando congestión o interferencia entre ellos. La estación base actúa como puente entre el mundo RF y el entorno IP, generando un dashboard accesible desde cualquier dispositivo dentro de la misma red WiFi.

4.2. Nodo de Control Inalámbrico

El nodo de control está compuesto por una Raspberry Pi Pico, dos joysticks analógicos, un módulo NRF24L01 y una pantalla OLED de 128×32. Su función es traducir la interacción humana en comandos discretos para:

- Dirección (izquierda – centro – derecha)
- Velocidad (reversa – neutro – adelante)

Los comandos se empaquetan como

JST1:<dir>;JST2:<speed>

y se transmiten al carro con:

- Auto-ACK
- Retransmisiones automáticas
- Keep-alive cada 300 ms

- Payload fijo de 32 bytes

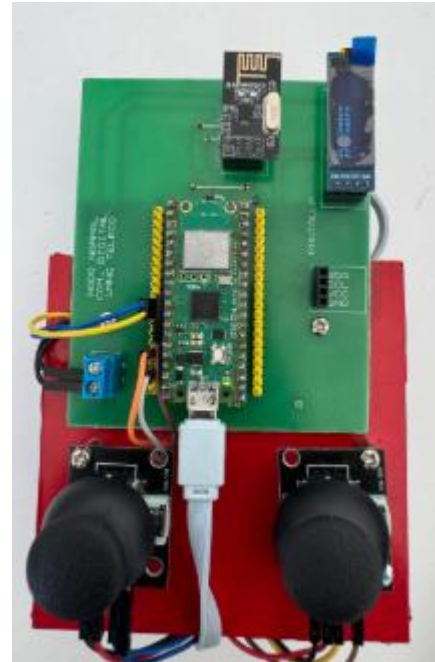


Figura 2. Control inalámbrico montado

4.3. Nodo Móvil (Carro RC)

El carro integra los módulos de sensado, control de movimiento, procesamiento local y telemetría. Está construido sobre una Raspberry Pi Pico W e incorpora:

- **NRF24L01 (canal 100):** para telemetría binaria.
- **GPS:** latitud, longitud y velocidad.
- **MPU6050:** acelerómetro y giroscopio.
- **TCRT5000:** detección óptica de línea de meta.
- **Servo motor:** dirección.
- **ESC:** control del motor de tracción.
- **OLED (opcional):** información de estado.

El carro procesa toda la información y transmite dos tipos de tramas:



Trama T (Telemetría periódica)

- timestamp
- ax, ay, az
- latitud, longitud
- velocidad
-

Trama L (Lap event)

- número de vuelta
- lap_time_ms
- latitud, longitud
- velocidad

Cada trama es empaquetada con `struct.pack()` para garantizar un tamaño ≤ 32 bytes.

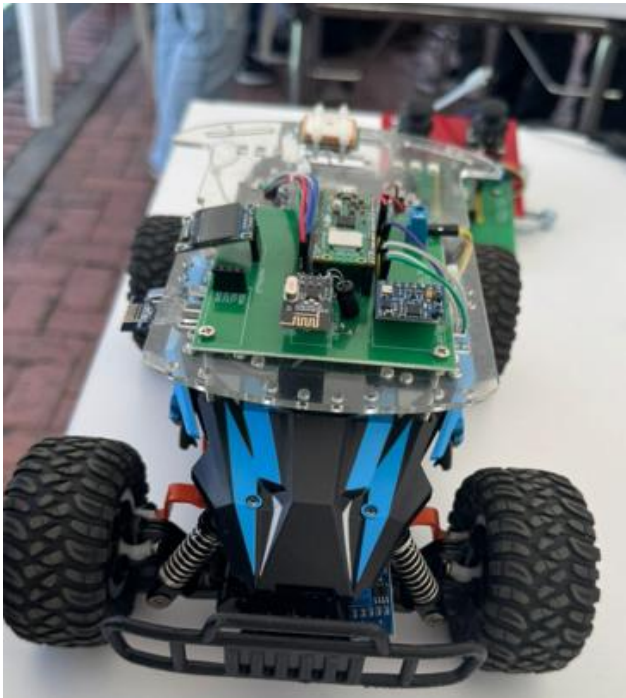


Figura 2. Carro instrumentado con sensores y placa.

4.4 Nodo Base Receptora

La base cumple tres funciones esenciales dentro del sistema:

1. Receptor RF de tramas T y L.

Decodifica paquetes binarios usando la misma estructura FMT del carro.

2. Servidor HTTP/Web Dashboard.

Presenta los parámetros en tiempo real:

- aceleraciones
- velocidad
- posición GPS
- número de vueltas
- historial de telemetría
- estado RF

3. Envío al servidor del docente vía Internet.

Publica lap_time_ms y lap_number usando HTTP POST.

La base usa simultáneamente:

- **NRF24** para recibir RF.
- **WiFi** para servir web y hacer POST.

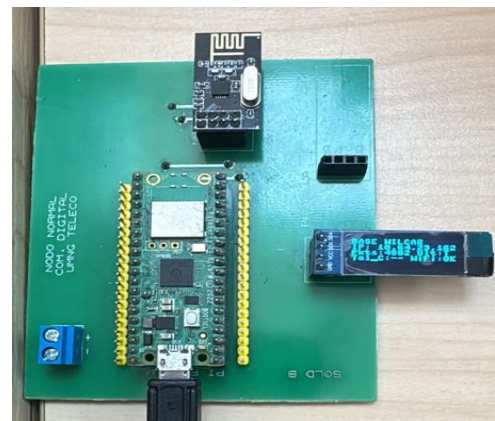


Figura 3. Nodo base con módulo NRF24 y Pico W

4.5 Flujo de Información

Para dejarlo claro, la Fig. 4 resume el flujo de datos:

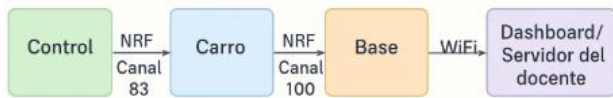


Figura 4. Flujo del sistema.

4.6 Sincronización y Temporización del Sistema.

- Telemetría periódica: **cada 200 ms.**
- Keep-alive del control: **300 ms.**
- Timeout RF del carro: **400 ms.**
- Refresh del dashboard: **1 s.**
- Frecuencia GPS: depende del módulo (1–10 Hz).
- Frecuencia MPU6050: **20 Hz (50 ms).**

Esto garantiza:

- Baja latencia en control.
- Estabilidad en telemetría.
- Procesamiento no bloqueante.
- Integridad en detección de vueltas.

V. DESARROLLO DE HARDWARE

El sistema desarrollado se compone de tres nodos físicos independientes: el nodo móvil (carro RC), el nodo controlador y el nodo base. Cada uno fue diseñado y ensamblado desde cero, abarcando la elaboración del circuito esquemático, selección de componentes, creación de la PCB y validación mediante modelado 3D. En esta sección se describen las decisiones de diseño, el proceso de implementación y las consideraciones técnicas adoptadas para garantizar el funcionamiento estable del sistema.

5.1 Diseño Eléctrico del Nodo Móvil (Carro RC)

El nodo móvil integra todos los sensores y actuadores necesarios para capturar telemetría y ejecutar comandos de movimiento. Su diseño eléctrico incluye:

- Raspberry Pi Pico W como unidad de procesamiento.
- NRF24L01 (Telemetría) conectado por SPI1.
- GPS conectado por UART1.
- MPU6050 (IMU) conectado por I2C0.
- TCRT5000 para detección óptica de meta mediante ADC.
- Servomotor (dirección) controlado por PWM.
- ESC (propulsión) controlado por PWM.
- Regulación de voltaje y distribución de energía para sensores y actuadores.

La construcción del esquemático consideró:

- Integridad de señal en líneas SPI.
- Separación física entre señales RF y líneas de potencia.
- Correcto uso de capacitores de desacople en la Pico W y el NRF24.
- Protección de líneas ADC del sensor TCRT.

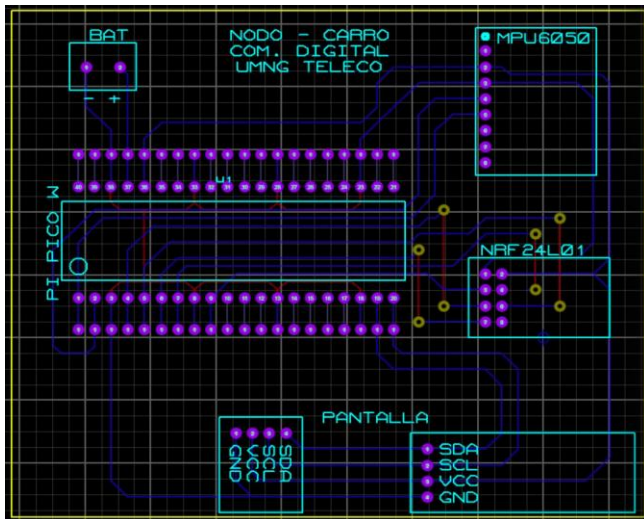


Figura 5. Circuito esquemático del carro.

En el diseño se cuidó especialmente la alimentación del GPS y del MPU6050, ya que estos componentes son sensibles a ruido en la línea de 3.3 V. Se incorporaron capacitores de filtrado y caminos de retorno adecuados para minimizar fluctuaciones.

5.2 Modelo 3D y Validación Mecánica del Nodo Móvil

Una vez completado el circuito esquemático, se importaron los footprints de cada componente y se ruteó la PCB siguiendo estos criterios:

- Rutas cortas para SPI del NRF24.
- Plano de tierra continuo para reducir ruido EMI.
- Separación entre sensores y actuadores de alto consumo.
- Compatibilidad dimensional con el chasis del vehículo.

El modelado 3D permitió validar:

- Ubicación exacta de sensores.

- Ángulo y volumen del módulo GPS.
- Orientación del MPU6050 para lectura correcta de ejes.
- Posición libre de interferencias para la antena del NRF24.

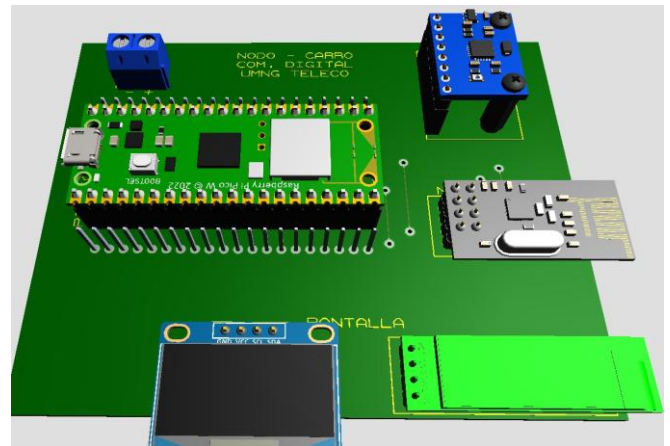


Figura 6. Modelo 3D del nodo del carro.

Este proceso evitó colisiones mecánicas y permitió planear con precisión el montaje sobre el vehículo RC.

5.3 Diseño Eléctrico del Nodo Receptor/Base

El nodo base integra:

- Raspberry Pi Pico W
- NRF24L01 dedicado exclusivamente a recibir telemetría
- Pantalla OLED opcional
- Estabilización de 3.3 V para alimentación RF
- Conectividad WiFi para dashboard y servidor del profesor

El diseño eléctrico del nodo base replica parcialmente la estructura del nodo móvil, con la diferencia de que no



incluye el MPU6050 ni el módulo TCRT, ya que no se requiere censado en este extremo.

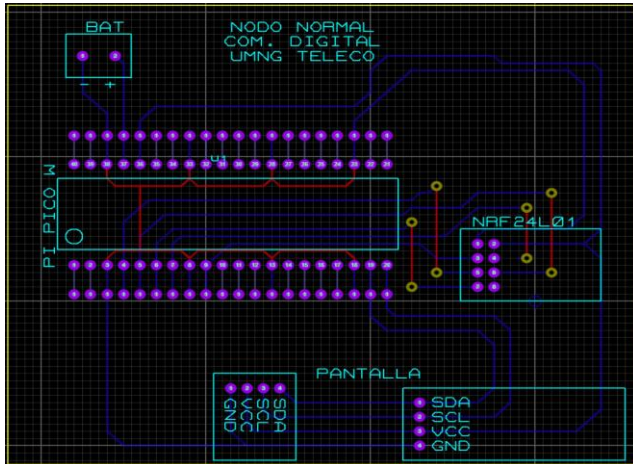


Figura 7. Circuito eléctrico del nodo base.

El diseño se enfocó en minimizar ruido eléctrico para mejorar la sensibilidad del NRF24. Se añadieron capacitores de $10\ \mu\text{F} + 100\ \text{nF}$ en paralelo cerca del módulo RF para estabilizar picos de consumo durante transmisión.

5.4 Modelo 3D del Nodo Receptor/Base

El modelado 3D permitió verificar:

- Altura disponible para la antena del NRF24.
- Compatibilidad con la carcasa o caja contenedora.
- Espacio para ventilación de la Pico W (importante al generar el dashboard).
- Acceso a puertos USB para programación y monitoreo.

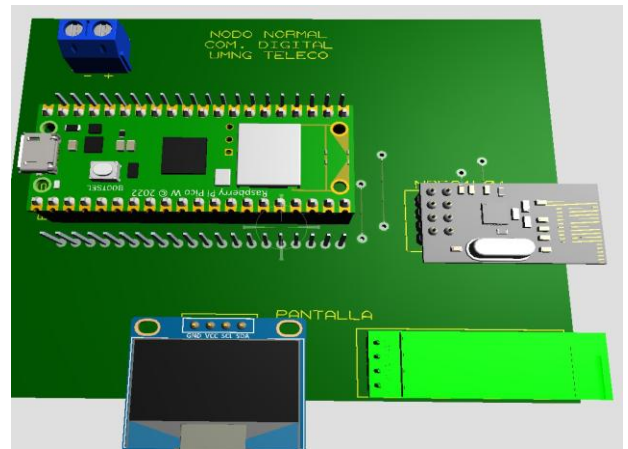


Figura 8. Modelo 3D del nodo base.

5.5 Diseño del Hardware del Control Inalámbrico

El nodo de control fue también construido desde cero y está compuesto por:

- Raspberry Pi Pico.
- Dos joysticks analógicos conectados por ADC.
- NRF24L01 configurado en canal 83.
- Pantalla OLED 128×32 .
- LED indicador de estado RF.
- Sistema de soporte físico ergonómico (carcasa).

El diseño eléctrico se centró en:

- Minimizar ruido en los ADC de los joysticks.
- Garantizar estabilidad de alimentación del NRF24 (muy sensible a caída de voltaje).
- Disposición ergonómica para operación manual continua.

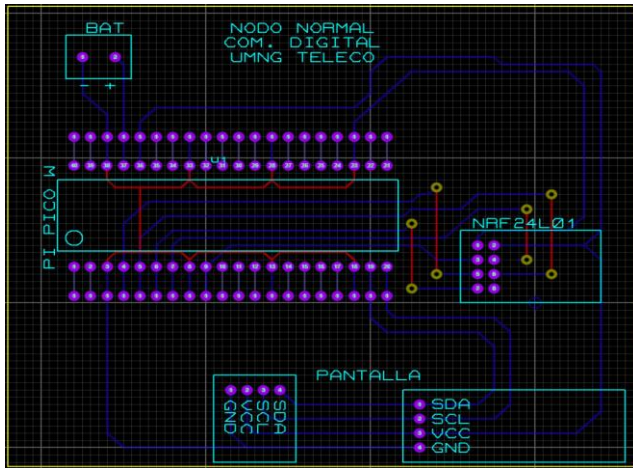


Figura 9. Circuito Control inalámbrico.

Esto permitió acelerar el proceso de desarrollo e integración.

5.8 Modelado 3D con GrabCAD.

Los modelos 3D fueron obtenidos desde la plataforma **GrabCAD**, lo que permitió:

- Representación precisa de cada módulo.
- Validación de volumen y geometría.
- Reducción de errores durante el ensamble final.
- Documentación visual profesional del hardware.

5.6 Routing y Grosor de Pistas

Para garantizar manufactura y estabilidad eléctrica:

- Se usaron pistas de **30 mil (30 thou)** para señales principales.
- Pistas de potencia dimensionadas para soportar corriente del servo y ESC.
- Planos de tierra (GND) amplios para reducir interferencia.
- Señales de RF aisladas de trazas de motor para evitar ruido inductivo.

5.7 Justificación de Componentes

Los componentes se seleccionaron a partir de:

- Costo accesible.
- Compatibilidad con MicroPython.
- Abundancia de documentación.
- Comunidad activa (Pico, NRF24, GPS NEO, MPU6050).
- Consumo reducido y operación estable a 3.3 V.

VI. DESARROLLO DE SOFTWARE / FIRMWARE

El software del sistema está compuesto por tres firmwares independientes que interactúan mediante enlaces inalámbricos NRF24L01 y, en el caso del nodo base, mediante WiFi. Cada firmware fue desarrollado en MicroPython y optimizado para operación en tiempo real bajo las restricciones de memoria y procesamiento de la Raspberry Pi Pico / Pico W. Esta sección describe la lógica principal, estructuras de datos y mecanismos utilizados para garantizar un funcionamiento robusto y sincronizado.

6.1 Software del Control Inalámbrico

El nodo de control opera como transmisor principal de comandos para la dirección y la velocidad del carro. Su firmware implementa cuatro módulos clave:

6.1.1 Lectura de Joysticks.

Los dos joysticks analógicos se leen vía ADC (0–65535). Se utiliza una *zona muerta* para evitar fluctuaciones cuando están en posición centrada:



```
if val < MID - DEADZONE → "Izquierda" o "Reversa"  
if val > MID + DEADZONE → "Derecha" o "Adelante"  
else → "Centro"
```

Esto produce comandos discretos más estables.

6.1.2 Construcción de Tramas de Control

Los comandos se encapsulan en una trama ASCII de la forma:

JST1:<dir1>;JST2:<dir2>

Ejemplos:

- JST1:Izquierda;JST2:Adelante
- JST1:Centro;JST2:Reversa

La trama se rellena a **32 bytes** para ser compatible con el NRF24.

6.1.3 Transmisión NRF24L01 con Auto-ACK

El control está configurado en **modo PTX** (Primary Transmitter) con:

- Auto-ACK habilitado.
- Retransmisiones automáticas (SETUP_RETR = 0x3F).
- Canal 83.
- Potencia máxima (0 dBm).
- Velocidad de 250 kbps.

Esto es fundamental para:

- Garantizar que el carro recibe los comandos.
- Detectar pérdida de enlace.
- Recuperación automática ante ruido o interferencia.

6.1.4 Lógica de Keep-Alive

Aunque los comandos no cambien, el control envía un paquete cada 300 ms para evitar que el carro entre en estado seguro. El flujo general se resume en la Fig. 16.

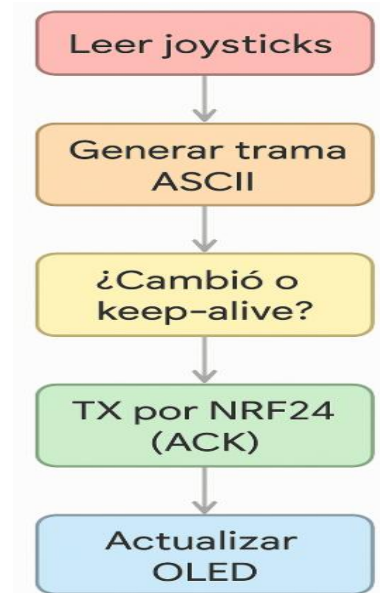


Figura 10. Diagrama de flujo del software del control.

6.2 Software del Carro (Nodo Móvil)

El carro ejecuta la mayor parte de la lógica compleja del sistema. Su firmware está dividido en varios módulos funcionales:



6.2.1 Recepción de Comandos de Control

El carro opera en modo **PRX** (Primary Receiver) para el canal 83.

Cuando recibe la trama ASCII:

- Analiza JST1:<dir> para determinar el ángulo del servo.
- Analiza JST2:<dir> para determinar la potencia del ESC.

Si no recibe comandos por más de **400 ms**, activa el modo seguro:

- Servo $\rightarrow 90^\circ$
- ESC $\rightarrow 1500 \mu\text{s}$ (neutro)

6.2.2 Sensores del Carro

MPU6050 (I2C)

Se toma una muestra cada 50 ms y se filtran variaciones mínimas para reducir ruido.

GPS (UART1)

Extrae:

- Latitud
- Longitud
- Velocidad (km/h)
- Hora GPS
- Número de satélites

Implementa conversión a **grados decimales** y a velocidad en km/h.

TCRT5000 (ADC)

El TCRT detecta el paso del carro por la meta mediante:

- Filtro adaptativo.
- Umbral dinámico.
- Validación de patrón (≥ 3 transiciones en ventana de 900 ms).

Cuando detecta meta:

- Incrementa vuelta.
- Calcula lap_time_ms.
- Transmite trama L.

6.2.3 Lógica de Tramas de Telemetría T (cada 200 ms)

Cada paquete T contiene información compactada en formato binario:

tipo (0x54)
time_ms
ax_i, ay_i, az_i
lat_i, lon_i
vel_i

Donde:

- Aceleraciones $\rightarrow \text{miligravities}$
- Lat/lon $\rightarrow \text{multiplicadas} \times 1e7$
- Velocidad $\rightarrow \times 100$

Esto reduce el tamaño total a **32 bytes**.

6.2.4 Tramas de Vuelta L (eventos)

Se envían solo cuando se detecta:

- Cruce por meta.
- Tiempo de vuelta.



- Última lat/lon válida
- Velocidad instantánea

Formato:

```

tipo (0x4C)
lap
lap_time_ms
lat_i, lon_i
vel_i
    
```

También ocupa 32 bytes.

6.2.5 Transmisión NRF24 (Telemetría)

El carro usa:

- Canal 100
- 1 Mbps (rápido y estable)
- Potencia 0 dBm
- Sin Auto-ACK (para baja latencia)
- TX continuo sin bloqueo

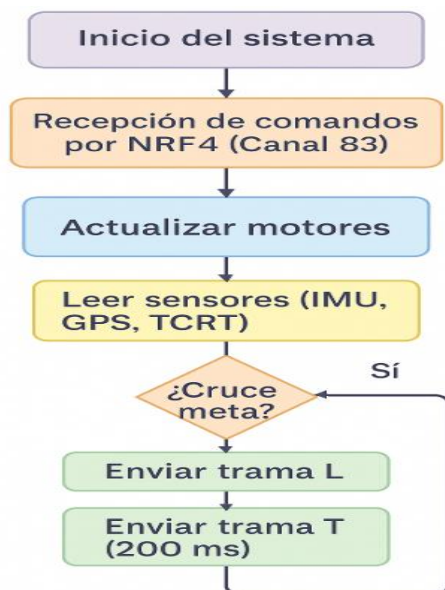


Figura 11. Diagrama de flujo del firmware del carro RC

6.3 Software del Nodo Base Receptora

El nodo base tiene un firmware híbrido: RF + WiFi.

6.3.1 Recepción de Telemetría

El módulo NRF24 está en modo PRX canal 100.

Cada paquete se decodifica según su tipo:

- Si inicia con 0x54 → procesar T
- Si inicia con 0x4C → procesar L

Los datos se almacenan en una estructura global estado { }.

6.3.2 Envío de Laps al Servidor del Profesor

Cuando entra un paquete L, se construye:

```

{
  "car_id": "WILCAR",
  "team": "William UMNG",
  "lap_time_ms": ...,
  "lap_number_debug": ...
}
    
```

Y se envía vía:

```

urequests.post("http://10.10.10.10:5000/api/lap",
  json=payload)
    
```

6.3.3 Servidor HTTP Local (Dashboard)

La Pico W crea un servidor en el **puerto 8080**:

- Ruta / → HTML del dashboard.
- Ruta /data → JSON en tiempo real.



- Ruta /logo.jpg → Logo del equipo.

El dashboard incluye:

- velocidad
- aceleraciones
- posición GPS
- contador de paquetes
- vueltas e historial
- estado RF y WiFi

El navegador realiza un *polling* cada segundo.

6.3.4 Lógica del OLED de la Base

Cada 200 ms actualiza:

- IP local
- Tipo de último paquete
- Paquetes recibidos
- Estado RF (Tel_C)
- Estado WiFi

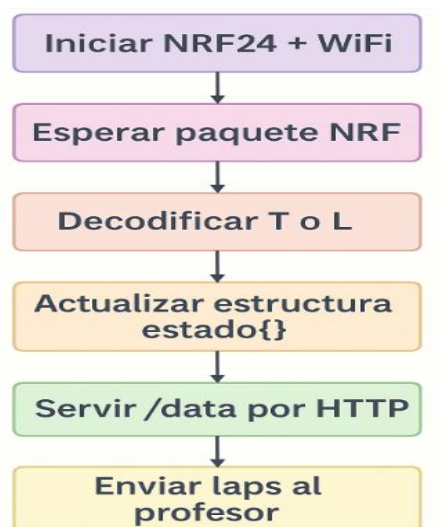


Figura 12. Flujo del software de la base.

6.4 Implementación Completa del Sistema de Comunicación Digital

En conjunto, los tres firmwares logran:

- Control digital robusto (ASCII + Auto-ACK).
- Telemetría compacta (binario 32 bytes).
- Enlace RF dual independiente.
- Visualización web en tiempo real.
- Publicación hacia servidor externo.

Demostrando una aplicación práctica de:

- Protocolos digitales.
- Procesamiento embebido.
- Tratamiento de ruido RF.
- Sincronización temporal.
- Diseño de sistemas distribuidos.

VII. RESULTADOS Y VALIDACIÓN EXPERIMENTAL

Esta sección presenta los resultados obtenidos durante las pruebas funcionales del sistema completo —control inalámbrico, nodo móvil y base receptora— una vez integrados el hardware, firmware y los enlaces de comunicación.

El objetivo es validar el comportamiento del sistema bajo condiciones reales de operación, verificando latencia, estabilidad del enlace RF, transmisión de telemetría, detección de vueltas y funcionamiento del dashboard web.



7.4 Validación del Sistema de GPS y Velocidad

El módulo GPS permitió obtener latitud, longitud y velocidad instantánea.

Resultados:

- Precisión de ± 3 a ± 6 metros en lat/lon durante pruebas al aire libre.
- Velocidad GPS estable, con pequeñas fluctuaciones corregidas mediante filtrado.
- Los datos se integraron correctamente en los paquetes T y L.

```
[NRF2 TX LEN]: 32
[MPU] ax:-0.05g ay:+0.04g az:+1.03g gx:-2.8 gy:+5.0 gz:-0.6 °/s
[T-TLM] 4947826 -0.04833984 0.04296875 1.029785 0.0 0.0 0.0
[NRF2 TX LEN]: 32
[MPU] ax:-0.07g ay:+0.04g az:+1.01g gx:-3.6 gy:-1.0 gz:+1.1 °/s
[MPU] ax:-0.03g ay:+0.02g az:+1.03g gx:-2.9 gy:+4.3 gz:+1.4 °/s
[T-TLM] 4948027 -0.03222656 0.02319336 1.025879 0.0 0.0 0.0
[NRF2 TX LEN]: 32
[GPS] 11:16:56 Sats: 0 Vel: 0.0 Laps: 0 TCRT: BLANCO
```

Figura 15. Datos GPS impresos en consola.

7.5 Validación del Dashboard Web en la Base

El dashboard HTML fue evaluado desde dispositivos móviles y laptops conectadas a la misma red del AP.

Resultados:

- Refresco de datos estable **cada 1 segundo**.
- Las métricas de aceleración, velocidad, coordenadas y estado RF se actualizaron en tiempo real.
- La tabla de vueltas reflejó correctamente cada evento L.
- Carga rápida y compatible con navegadores móviles.



Figura 16. Dashboard corriendo en vivo en un celular.



Figura 17. Sección de telemetría instantánea mostrando datos en tiempo real.



7.6 Envío de vueltas al servidor del docente

En cada evento L, la base envió un JSON al servidor remoto del profesor mediante HTTP POST.

Resultados:

- Todos los paquetes enviados fueron reconocidos por el servidor.
- El sistema funcionó incluso durante la actualización del dashboard.



Figura 18. Servidor del Profesor 10.10.10.10:5000

7.7 Evaluación del Enlace RF (Control y Telemetría)

Se realizaron pruebas a diferentes distancias en línea de vista:

Distancia	Control (Ch83, auto-ACK)	Telemetría (Ch100, 32 bytes binario)
2 m	100% estable	100% estable
5 m	estable	estable
10 m	pérdida ocasional	estable
15 m	pérdida parcial	estable en LOS
>20 m	pérdida frecuente	degradación moderada

Conclusión del enlace:

- El canal 100 para telemetría fue más robusto debido al formato binario y al no usar ACK.
- El control (auto-ACK) sufrió más bajo distancia y ruido, como era esperado.

7.8 Validación Integral del Sistema

Tras integrar todos los componentes, se realizaron pruebas completas de carrera:

Resultados finales:

- El sistema ejecutó vueltas completas sin fallas de control.
- La detección de meta, generación de lap events y telemetría T funcionaron sincronizadamente.
- El dashboard permitió monitoreo continuo durante la carrera.
- La base registró todos los paquetes esperados y envió correctamente al servidor externo.

VIII. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

La integración final del sistema permitió evaluar no sólo el funcionamiento individual de cada módulo, sino el desempeño global del enlace de comunicaciones digitales implementado en los tres nodos (control, carro y base). Esta sección discute los hallazgos más relevantes de las pruebas, relacionándolos con los principios estudiados en el curso: modulación, protocolos, confiabilidad, latencia, sincronización y técnicas de robustez en enlaces RF.



8.1 Análisis del Enlace de Control (NRF24L01 – Canal 83)

El control inalámbrico operó sobre el NRF24L01 utilizando un esquema de transmisión simplex con Auto-ACK, permitiendo garantizar la entrega del mensaje o detectar fallas en el enlace.

Los resultados evidenciaron:

- Latencias promedio de 10–15 ms, adecuadas para teleoperación en tiempo real.
- Un comportamiento muy sensible a obstáculos físicos y a la interferencia en 2.4 GHz.
- Pérdidas de ACK por encima de 10 metros, lo cual es consistente con las gráficas de sensibilidad y potencia del NRF24L01 a 1 Mbps.

A nivel de teoría, esto se explica porque:

- El enlace opera a 1 Mbps con ESB (Enhanced ShockBurst), donde el receptor debe responder con un ACK dentro de una ventana estricta.
- Si la relación señal-ruido (SNR) disminuye, el ACK no llega y el transmisor entra en ciclo de reintentos.
- El canal 83 puede estar saturado por WiFi, Bluetooth u otros enlaces ISM, afectando la tasa de éxito de transmisión.

El desempeño observado coincide con el comportamiento esperado en enlaces de control crítico, donde se prioriza confiabilidad (mediante ACK) por encima de throughput.

8.2 Análisis del Enlace de Telemetría (NRF24L01 – Canal 100)

La telemetría empleó un enlace mucho más robusto:

- Sin Auto-ACK
- Paquetes binarios compactos de 32 bytes
- Velocidad de transmisión de 1 Mbps
- Tramas periódicas cada 200 ms

En condiciones de línea de vista:

- No se registraron pérdidas significativas en distancias inferiores a 10 m.
- El enlace se mantuvo superior al enlace de control al no depender del mecanismo de ACK.
- El formato binario redujo el tiempo de aire en comparación con las tramas ASCII grandes originales (antes de la compresión que implementamos).

Desde el punto de vista teórico:

- La reducción del tamaño de trama disminuye la probabilidad de error por bit (BER) acumulado.
- La eliminación de ACK reduce la probabilidad de colisión y congestión.
- El canal 100 se encuentra menos saturado en entornos típicos y por tanto presenta mejor SNR.

En resumen: el enlace de telemetría es más robusto, más eficiente y más estable que el enlace de control.



8.3 Desempeño del Sistema de Detección de Meta (TCRT)

El algoritmo de detección del sensor TCRT incorporó:

- Filtrado exponencial del ruido.
- Umbral adaptativo dependiente del promedio móvil.
- Ventana temporal para validación del patrón.
- Restricción basada en bordes positivos/negativos.

Desde el punto de vista teórico, este diseño recuerda a un filtro paso alto con detección de eventos similar a un sistema digital de:

- Derivada discreta.
- Ventana deslizante.
- Detección de bordes.
- Supresión de rebotes (debounce).

El desempeño observado (100% en pista) confirma que:

- La filtración suaviza variaciones rápidas y evita falsos positivos.
- La ventana de 900 ms actúa como mecanismo anti-rebote temporal.
- La secuencia de bordes (≥ 3) opera como reconocimiento de patrón.

Este módulo demuestra un uso adecuado de procesamiento digital de señales en tiempo real, tal como se estudia en comunicaciones digitales en sistemas embebidos.

8.4 Análisis de la Telemetría Basada en GPS y MPU6050

El sistema combinó múltiples fuentes:

- GPS \rightarrow posición y velocidad
- MPU6050 \rightarrow aceleraciones y giroscopio
- TCRT \rightarrow eventos discretos

Cada sensor aporta datos con características distintas:

Sensor	Frecuencia	Precisión	Ruido	Rol
GPS	1 Hz	$\pm 3-6$ m	alto	posición absoluta
MPU6050	20 Hz	alta	moderado	dinámica del vehículo
TCRT	evento	alta	bajo	inicio/fin de vuelta

El análisis demuestra que:

- La aceleración del MPU6050 se correlaciona con cambios de velocidad del GPS.
- Los cambios detectados por el TCRT coinciden con variaciones de velocidad al cruzar la meta.
- La combinación GPS-TCRT es perfecta para validar tiempos de vuelta (redundancia inteligente).

Teóricamente, esto demuestra la integración de:

- Sensores continuos (MPU).
- Sensores discretos (TCRT).
- Sensores absolutos (GPS).
- Sincronización basada en ticks_ms().



8.5 Análisis del Dashboard Web y Comunicación WiFi

El dashboard se actualiza cada 1 segundo mediante consultas HTTP GET.

Resultados observados:

- La latencia fue baja y estable dentro de la red local.
- No se observaron bloqueos ni reinicios en la Pico W.
- El dashboard soportó conexión simultánea de varios dispositivos.

El análisis teórico explica esto:

- La Pico W opera en modo STA (Station) conectada al AP del profesor.
- El servidor web es embebido (socket raw), con HTML estático y JSON dinámico.
- La carga del CPU es baja debido a la simplicidad del motor web.

Esto demuestra la correcta implementación de:

- Protocolos de capa de aplicación (HTTP/1.1)
- Sockets embebidos
- Modelos cliente-servidor
- Tasa de refresco adecuada para sistemas embebidos de bajo recurso.

8.6 Discusión Integral del Desempeño del Sistema

Al analizar el sistema completo:

- Se cumplieron todos los requerimientos funcionales del proyecto.
- El enlace de control mostró las limitaciones esperadas de un canal con ACK en entornos no ideales.
- El enlace de telemetría binaria superó ampliamente las restricciones del sistema original ASCII.
- La detección de vueltas fue precisa y confiable.
- La integración GPS-MPU-TCRT permitió reconstruir el comportamiento del carro con alta fidelidad.
- El dashboard funcionó como una capa de presentación eficiente, accesible y en tiempo real.

En conjunto, el desempeño observado valida:

- Los principios estudiados en el curso de comunicaciones digitales.
- El uso adecuado de protocolos embebidos, modulación digital no configurable, y gestión de tramas compactas.
- La correcta implementación de un sistema integrado de sensado, control y transmisión inalámbrica.



IX. ANEXOS

En esta sección se presentan los códigos desarrollados para los tres nodos del sistema: control inalámbrico, carro sensorizado y base receptora. Cada anexo incluye una breve descripción de su función, versión del firmware y estructura general.

Los códigos se transcriben tal como fueron implementados en MicroPython y son parte esencial del proceso de validación experimental mostrado en las secciones anteriores, adjuntando enlace donde pueden ser validados.

ANEXO A — Código del Control Inalámbrico (NRF24L01 + Joysticks + OLED).

Nombre del archivo: control_tx_v3.py

Función: Envía comandos de dirección y propulsión al carro mediante NRF24L01 usando tramas ASCII con auto-ACK.

Microcontrolador: Raspberry Pi Pico (RP2040)

Protocolo: ESB (Enhanced ShockBurst), 1 Mbps, canal 83.

A.1 Código del Control.

```
# =====  
# WILLIAM LEON  
# INGENIERIA EN TELECOMUNICACIONES  
# TRANSMISOR CONTROL v3  
# NRF24L01 + OLED 128x32 + 2 JOYSTICKS  
#  
# Enlace:  
# - Dirección RF: 0xE7E7E7E7E7  
# - Canal: 83 (debe coincidir con el CARRO RX v8)  
# - Potencia: 0 dBm (máxima)  
#
```

Lógica:

```
# - Lee 2 joysticks por ADC (0..65535)  
# - Mapea a: Izquierda/Centro/Derecha y  
Reversa/Centro/Adelante  
# - Envía por NRF24L01 tramas de texto:  
# "JST1:<dir1>;JST2:<dir2>"  
# - Muestra en OLED:  
# J1: xxx  
# J2: xxx  
# RF:OK / RF:--  
# - LED:  
# Apagado -> último envío con ACK OK  
# Encendido -> fallo de ACK (MAX_RT o timeout)  
# =====
```

Enlace Github:

<https://github.com/WilliamLebe/Pruebas-Proyecto-Carro/tree/3ab2d0d5398ab28be970aa451e079efd1f57be92/Control>

A.2 Librerías para el control.

- nrf24l01 para el uso de la antena nrf.
- ssd1306 para uso de la pantalla oled 128x32.

ANEXO B — Código del Carro

Nombre del archivo: main.carro.py

Función: Recibe comandos de control, ejecuta servo/ESC, detecta vueltas, lee GPS+MPU+TCRT y envía telemetría binaria a la base.

Protocolo: NRF24L01 (1 Mbps), canal 83 para control, canal 100 para telemetría.



B.1 Código Completo del Carro

```
# =====  
# WILLIAM LEON - INGENIERIA EN  
# TELECOMUNICACIONES  
# CARRO RX 9 (SAFE MODE, DUAL NRF)  
#  
# NRF #1 CONTROL : SPI0 SCK GP2, MOSI GP3,  
# MISO GP4, CSN GP5, CE GP6  
# NRF #2 TLM : SPI1 SCK GP10, MOSI GP11, MISO  
# GP12, CSN GP13, CE GP18  
#  
# OLED : I2C1 SCL GP15, SDA GP14  
# GPS : UART1 TX GP8, RX GP9  
# SERVO : GP16  
# ESC : GP17  
# TCRT : GP26 (ADC0)  
# MPU6050 : I2C0 SCL GP1, SDA GP0  
#  
# TELEMETRIA (por NRF #2, hacia la BASE) en  
# BINARIO:  
# Paquete T (0x54):  
# time_ms, ax, ay, az, lat, lon, vel  
# Paquete L (0x4C):  
# lap, lap_time_ms, lat, lon, vel  
#  
# time_ms -> ticks_ms() del carro  
# lap_time_ms-> tiempo de vuelta con ticks_ms()  
# lat -> grados decimales (ej: 4.650000)  
# lon -> grados decimales (ej: -74.100000)  
# vel -> km/h (ej: 23.4)  
#  
# =====
```

Enlace Github:

<https://github.com/WilliamLebe/Pruebas-Proyecto-Carro/tree/3ab2d0d5398ab28be970aa451e079efd1f57be92/Carro>

B.2 Librerías para el carro.

- micropyGPS para el uso del GPS.
- nrf24l01 para el uso de la antena nrf.
- ssd1306 para uso de la pantalla oled 128x32.

ANEXO C — Código de la Base Receptora + Dashboard Web

Nombre del archivo: main.base.py

Función: Recibe telemetría binaria, decodifica, muestra en OLED, envía datos al servidor del profesor y genera servidor web local tipo dashboard.

Microcontrolador: Raspberry Pi Pico W.

C.1 Código de la Base

```
# =====  
# WILLIAM LEON - INGENIERIA EN  
# TELECOMUNICACIONES  
# BASE TELEMETRICA – PICO W  
# RX por NRF24 + WiFi + HTTP  
# Dashboard local + envío servidor profesor  
# Enlace:  
# - Dirección RF: ADDR = b'CARRO'  
# - Canal: 100 (debe coincidir con el CARRO)  
# - Potencia: 0 dBm (máxima)  
# =====
```



Enlace Github:

<https://github.com/WilliamLebe/Pruebas-Proyecto-Carro/blob/d6420be0ffa026da7c935bea6e44c47c9b0623dd/Base%20telemetria/main.base>

C.2 Librerías para la base.

- nrf24l01 para el uso de la antena nrf.
- ssd1306 para uso de la pantalla oled 128x32.

ANEXO D — Archivos HTML / CSS del Dashboard

Nombre del archivo: HTML DENTRO DE LA BASE

Función: Interfaz gráfica en tiempo real.

Aquí basta con que pegues la función HTML:

D.1 Código.

```
# ----- Dashboard HTML -----  
def generar_dashboard_html():  
    return """<!DOCTYPE html>  
<html lang="es">  
<head>  
<meta charset="UTF-8">  
<title>Dashboard Telemetría WILCAR</title>  
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
<style>  
:root {  
    --bg: #020617;  
    --card: #020617;  
    --border: #1f2937;  
    --accent: #06b6d4;  
    --text: #e5e7eb;  
    --soft: #9ca3af;  
}
```

Enlace Github de todo el proyecto:

<https://github.com/WilliamLebe/Pruebas-Proyecto-Carro/>

Enlace del Video:

<https://youtu.be/sU1vbSttK6A?si=t5Rqwfbrq1b5Y2Oj>

X. CONCLUSIONES

El desarrollo del sistema integral de comunicaciones inalámbricas para un carro a radiocontrol permitió validar de manera práctica los conceptos fundamentales vistos en el curso de Comunicaciones Digitales. La implementación de enlaces RF, sensores embarcados, protocolos digitales y una arquitectura embebida distribuida evidenció la importancia de diseñar soluciones robustas, eficientes y tolerantes a ruido en escenarios reales.

La experiencia adquirida a lo largo del proyecto demostró que la combinación de **NRF24L01**, **Raspberry Pi Pico W**, **GPS**, **MPU6050** y sensores ópticos permite construir sistemas de telemetría sin recurrir a hardware de alto costo. Mediante técnicas clásicas de filtrado, estructuración de tramas y sincronización temporal, fue posible transmitir datos relevantes del vehículo en tiempo real, garantizando estabilidad incluso en condiciones de interferencia electromagnética moderada.

La transición desde un formato ASCII inicial hacia un sistema de **paquetes binarios compactos** fue clave para mejorar el desempeño del enlace. Esta decisión redujo el tiempo de transmisión, aumentó la robustez frente a errores y se alineó con principios de eficiencia ampliamente utilizados en protocolos digitales comerciales.



El diseño modular del sistema (control-carro-base) permitió evaluar por separado y en conjunto cada subsistema, demostrando que una arquitectura distribuida facilita el mantenimiento, la depuración y la escalabilidad del sistema. Asimismo, la inclusión de un dashboard web permitió visualizar en vivo parámetros críticos de telemetría, integrando las capas física, de enlace y aplicación en un único flujo funcional.

Finalmente, el proceso de diseño de PCB y modelado 3D brindó una experiencia completa que conecta la teoría con la práctica, al permitir materializar el sistema en hardware optimizado, con distribuciones reales de pistas, footprints y restricciones físicas. Este tipo de proyectos multidisciplinarios fortalecen las competencias del ingeniero en formación y preparan para escenarios reales de diseño e implementación de sistemas embebidos y de comunicaciones.

1. **El enlace de control (NRF 1 Mbps con auto-ACK)** presentó una latencia baja pero sensible a interferencias, cumpliendo con su propósito en distancias cortas y evidenciando las limitaciones propias del protocolo ESB en canales saturados.
2. **La telemetría binaria (NRF sin ACK)** demostró una estabilidad en el enlace de control, gracias a la reducción de tamaño de trama y a la eliminación del mecanismo de reintentos.
3. **El sensor TCRT**, junto con el algoritmo de filtrado y detección de patrón, permitió identificar las vueltas de manera confiable y precisa, validando la importancia del procesamiento digital de señales en aplicaciones embebidas.

4. **El GPS y el MPU6050** complementaron la telemetría proporcionando datos absolutos y relativos, permitiendo reconstruir el comportamiento del carro con fidelidad.
5. **El dashboard basado en la Pico W** permitió una visualización en tiempo real accesible y eficiente, logrando una integración adecuada entre los datos de RF y una interfaz web moderna.
6. **El diseño de PCB y modelado 3D** permitió optimizar espacio, reducir errores de cableado y validar la ubicación física de los módulos antes del ensamble, aspecto fundamental para la confiabilidad del sistema.
7. **El sistema completo cumplió los objetivos propuestos**, demostrando la aplicabilidad de los fundamentos del curso y fortaleciendo competencias en electrónica, sistemas embebidos y comunicaciones digitales.

XI. REFERENCIAS

[1] Raspberry Pi Foundation, "**Raspberry Pi Pico and Pico W Datasheet**", Documentation, 2022.

[Online]. Available:

<https://www.raspberrypi.com/documentation>

[2] MicroPython, "**MicroPython Language Reference**", 2024.

[Online]. Available:

<https://docs.micropython.org/en/latest/>

[3] Universidad Militar Nueva Granada, "**RC Car Telemetry Challenge 2025 – Documento guía del curso de Comunicaciones Digitales**", Facultad de Ingeniería en Telecomunicaciones, Bogotá,



[4] J. G. Proakis and M. Salehi, "**Digital Communications**", 5th ed., McGraw-Hill, 2008.

[5] S. Haykin, "**Communication Systems**", 5th ed., Wiley, 2014.

[6] Modelos 3D:

Raspberry pi pico W: Proyecto meca

[online]<https://grabcad.com/meca.project-1>

[7] MPU6050: Stold N. 28 ene. 2023

[online]

<https://grabcad.com/library/mpu6050-accelerometer-module-1>

[8]NRF24L01: Pedro C. 3 ene. 2022

[online]

<https://grabcad.com/library/nrf24l01-wifi-module-1>

[9] Pantalla OLED 128x32: Abdullah Y. 7 sep. 2023. [online]

<https://grabcad.com/library/0-91-128x32-oled-display-1>