

Grado en Ingeniería en Tecnologías Industriales

Trabajo Fin de Grado

Telemetría para un vehículo de competición Formula Student. Equipo emisor

Autor:
Raúl Morán Guerra

Director:
José Daniel Muñoz Frías

Madrid
Agosto 2023

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título Telemetría para un vehículo de competición Formula Student. Equipo emisor

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2022/2023. es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Raúl Morán Guerra Fecha: 30/08/2023



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por MUÑOZ FRIAS JOSE DANIEL - ***7384**
el día 30/08/2023 con un certificado emitido
por AC FNMT Usuarios

Fdo.: José Daniel Muñoz Frias Fecha: ……30…/ ……08…/ ……2023…



Grado en Ingeniería en Tecnologías Industriales

Trabajo Fin de Grado

Telemetría para un vehículo de competición Formula Student. Equipo emisor

Autor:
Raúl Morán Guerra

Director:
José Daniel Muñoz Frías

Madrid
Agosto 2023

TELEMETRÍA PARA UN VEHÍCULO DE COMPETICIÓN FORMULA STUDENT. EQUIPO EMISOR

Autor: Morán Guerra, Raúl.
Director: Muñoz Frías, José Daniel
Entidad Colaboradora: ISC FS Racing Team, ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este Trabajo Fin de Grado presenta el diseño e implementación de un sistema de telemetría para un vehículo de competición Formula Student, que monitoriza variables críticas mediante un microcontrolador transmisión inalámbrica RF de alta tasa de datos, permitiendo mejorar el rendimiento y la seguridad.

El sistema recoge y empaqueta datos de múltiples sensores, los transmite vía RF 2.4GHz sincronizadamente, e incrementa sustancialmente la información disponible sobre el vehículo.

Palabras clave: telemetria; Formula Student; nRF24, Teensy 4.1; sensores

1. Introducción

Formula Student es una competición que reúne a estudiantes de ingeniería de todo el mundo para diseñar, construir y competir con vehículos de tipo Formula. Los equipos deben demostrar no solo el rendimiento del vehículo, sino también habilidades de ingeniería, diseño y gestión [1].

Un aspecto clave para mejorar el rendimiento es entender en profundidad el comportamiento dinámico del vehículo y eso se consigue mediante un sistema de telemetría. En temporadas anteriores se implementó un sistema basado en LoRa, pero las limitaciones en tasa de datos motivaron el desarrollo de una solución más avanzada.

2. Objetivos del proyecto

Este proyecto tiene como objetivo diseñar e implementar un nuevo sistema de telemetría para el vehículo IFS-05 con las siguientes metas:

- Encontrar un protocolo de comunicación inalámbrica con mayor tasa de datos que LoRa, pero que no sacrifique el alcance.
- Diseñar e implementar una solución propia basada en microcontrolador para la recogida y envío de datos.

- Desarrollar un sistema de envío en el que los datos se envíen de forma sincronizada.

3. Arquitectura de comunicaciones del sistema

El sistema consta de una unidad de telemetría (TCU) con microcontrolador Teensy 4.1 que recoge datos de múltiples sensores procedentes de la unidad de control del vehículo (VCU) y del acumulador a través de un bus CAN [2]. Se seleccionó el protocolo nRF24 (2.4GHz, 2Mbps) para la transmisión inalámbrica de los datos [3].

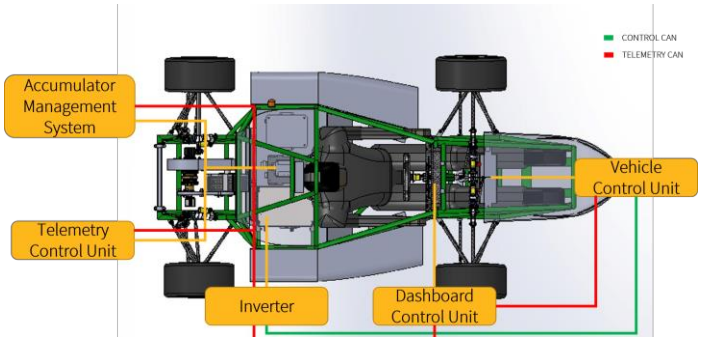


Figura 1 – Arquitectura de comunicaciones del IFS-05. Elaboración propia.

La TCU lee suspensiones, frenos, corriente, IMU y GPS entre otros sensores. Los datos se empaquetan de forma sincronizada y se envían al receptor en la pista. Para poder obtener todas estas medidas, se ha diseñado una PCB que integra el microcontrolador Teensy 4.1, añadiendo las entradas y salidas necesarias para la comunicación con el resto de unidades de control a través del bus CAN y la medida de variables mediante los sensores mencionados anteriormente, así como el módulo de comunicación inalámbrica nRF24L01.

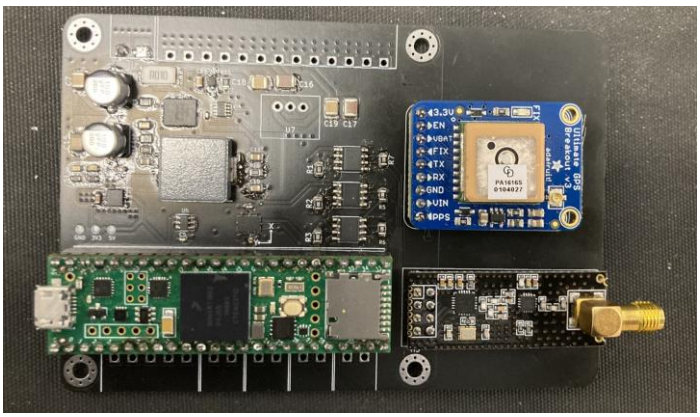


Figura 2 – PCB unidad de telemetría previo a su montaje en el vehículo.

4. Resultados

El sistema fue probado con éxito en competición, monitorizando variables críticas en tiempo real. Se observó un incremento sustancial en la tasa de datos respecto al sistema previo, permitiendo obtener más información.

Los datos sincronizados facilitarán el análisis del comportamiento dinámico y la correlación con simulaciones. Esto permitirá la evolución sistemática de prototipos futuros.

5. Conclusiones

El sistema de telemetría desarrollado cumple los objetivos planteados al inicio del proyecto. La elección de nRF24 incrementó notablemente la tasa de transmisión de datos. El hardware y software personalizados basados en Teensy 4.1 resultaron escalables. Finalmente, el empaquetamiento sincronizado de los datos facilitará su análisis detallado.

En conclusión, el sistema mejora sustancialmente la información disponible sobre el vehículo durante las pruebas dinámicas, incrementando el rendimiento y la seguridad.

6. Referencias

[1] Formula Student Germany. Rules 2023 v1.1. 2023.

[2] Robert Bosch GmbH. CAN Specification 2.0. September 1991. url: http: //esd.cs.ucr.edu/webres/can20.pdf (visitado 30-08-2023).

[3] nRF24L01 Single Chip 2.4GHz Transceiver Product Specification. 2007. url: https://cdn.sparkfun.com/datasheets/Wireless/Nordic/nRF24L01_Product_Specification_v2_0.pdf (visitado 24-06-2023).

TELEMETRY FOR FORMULA STUDENT RACING VEHICLE. TRANSMITTER SIDE.

Author: Morán Guerra, Raúl
Supervisor: Muñoz Frías, José Daniel.
Collaborating Entity ISC FS Racing Team, ICAI – Universidad Pontificia Comillas

ABSTRACT

This Bachelor's Thesis presents the design and implementation of a telemetry system for a Formula Student racing vehicle, which monitors critical variables using a microcontroller and high data rate wireless RF transmission, allowing improved performance and safety.

The system collects and packages data from multiple sensors, transmits it wirelessly via 2.4GHz RF synchronously, and substantially increases available information about the vehicle.

Keywords: telemetry; Formula Student; nRF24, Teensy 4.1; sensors

1. Introduction

Formula Student is a competition that brings together engineering students from around the world to design, build and compete with Formula-style vehicles. Teams must demonstrate not only vehicle performance, but also engineering, design and management skills [1].

A key aspect to improve performance is thoroughly understanding the vehicle's dynamic behavior, achieved through a telemetry system. In previous seasons a LoRa-based system was implemented, but data rate limitations motivated developing a more advanced solution.

2. Project objectives

This project aims to design and implement a new telemetry system for the IFS-05 vehicle with the following goals:

- Find a wireless communication protocol with higher data rate than LoRa, without sacrificing range.
- Desing and implement a custom microcontroller-based solution for data collection and transmission.
- Develop a system where data is transmitted in a synchronized way.

3. System architecture

The system consists of a telemetry control unit (TCU) with a Teensy 4.1 microcontroller that gathers data from multiple sensors from the vehicle control unit (VCU) and the battery through a CAN bus [2]. The nRF24 protocol (2.4GHz, 2Mbps) was selected for wireless data transmission [3].

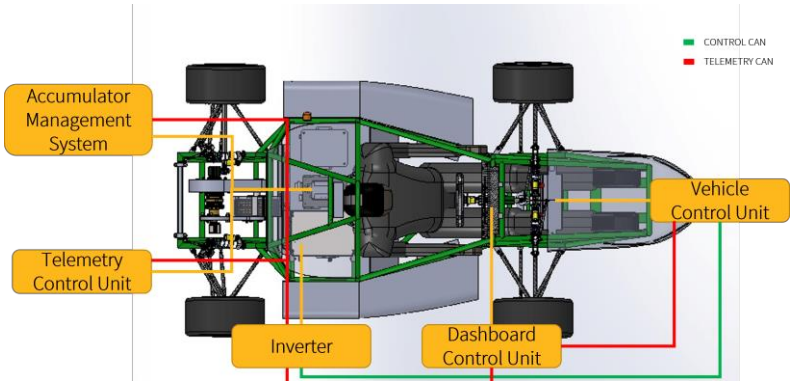


Figure 1 - IFS-05 Communication Architecture

The TCU reads suspensions, brakes, current, IMU and GPS among other sensors. Data is packaged synchronously and sent to the trackside receiver. To obtain all these measurements, a PCB was designed integrating the Teensy 4.1 microcontroller, adding the necessary inputs and outputs for communication with other control units via CAN bus and measurement of variables through the sensors, as well as the nRF24L01 wireless communication module.

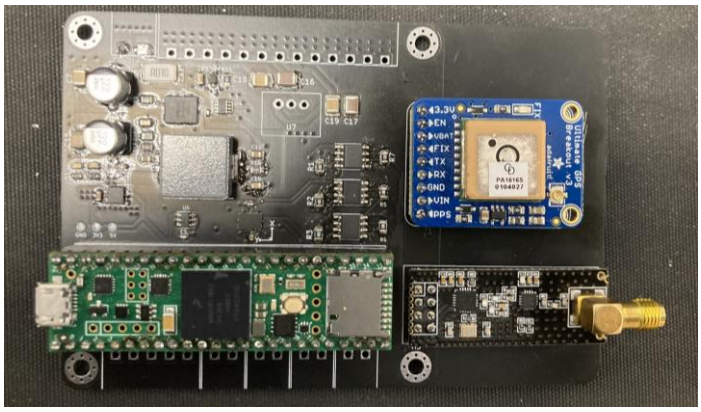


Figure 2 - Telemetry unit PCB prior to installation in vehicle

4. Results

The system was successfully tested in competition, monitoring critical variables in real time. A substantial increase in data rate was observed compared to the previous system, allowing more information to be obtained.

The synchronized data will facilitate analysis of dynamic behavior and correlation with simulations. This will enable systematic evolution of future prototypes.

5. Conclusions

The developed telemetry system meets the objectives set at the beginning of the project. Choosing nRF24 significantly increased the data transmission rate. The custom Teensy 4.1 based hardware and software proved scalable. Finally, the synchronized data packaging will facilitate detailed analysis.

In conclusion, the system substantially improves information available about the vehicle during dynamic testing, increasing performance and safety.

6. References

[1] Formula Student Germany. Rules 2023 v1.1. 2023.

[2] Robert Bosch GmbH. CAN Specification 2.0. September 1991. url: http://esd.cs.ucr.edu/webres/can20.pdf (accessed 30-08-2023).

[3] nRF24L01 Single Chip 2.4GHz Transceiver Product Specification. 2007. url: https://cdn.sparkfun.com/datasheets/Wireless/Nordic/nRF24L01_Product_Specification_v2_0.pdf (accessed 24-06-2023).

Agradecimientos

A mi padre Raúl, por ser mi estrella y mi ejemplo.
A mi madre Lola, mi hermano Álvaro, mi tía Silvia y mis abuelos Nemesio y Mari, por ser siempre mi apoyo.

Al ISC y en especial al Consejo del IFS-05, por compartir tantas horas trabajando por un sueño que por fin se ha hecho realidad.

Índice general

1. Introducción **1**
- 1.1. Formula Student . . . 1
- 1.2. Telemetría en el ISC (estado del arte) . . . 2
- 1.3. Objetivo del proyecto . . . 4
- 1.4. Alineación con los ODS . . . 4

2. Descripción del sistema **7**
- 2.1. Sistema emisor . . . 7
- 2.2. Comunicación CAN . . . 9
- 2.3. Comunicación inalámbrica . . . 10

3. Diseño del hardware del sistema de telemetría **17**
- 3.1. Protocolo de comunicación inalámbrica . . . 17
  - 3.1.1. Bluetooth . . . 18
  - 3.1.2. ZigBee . . . 18
  - 3.1.3. Redes móviles (4G/5G) . . . 18
  - 3.1.4. nRF24 . . . 19
- 3.2. Comunicaciones . . . 22
  - 3.2.1. SPI . . . 22
  - 3.2.2. CAN . . . 22
- 3.3. Microcontrolador . . . 23
- 3.4. Sensores . . . 25
  - 3.4.1. Sensores de suspensión y posición del acelerador . . . 25
  - 3.4.2. Sensor de presión de frenos . . . 26
  - 3.4.3. Sensor de corriente . . . 26
  - 3.4.4. Unidad de Medida Inercial (IMU) . . . 26
  - 3.4.5. GPS . . . 27
  - 3.4.6. Sensores de temperatura del líquido refrigerante . . . 28
- 3.5. Circuito de alimentación . . . 28
- 3.6. Diseño PCB . . . 32
- 3.7. Soldadura de los componentes en la PCB . . . 35

3.8. Montaje en el IFS-05 36
3.8.1. Arquitectura de comunicaciones 36
3.8.2. Elección de la antena y posición en el vehículo 36

4. Diseño del software del sistema de telemetría **39**
4.1. Estructura del código del sistema emisor 39
4.2. Recepción de los datos 49

5. Análisis de los resultados **51**

6. Conclusiones **57**

Anexo A. Diseño de PCBs **59**

Anexo B. Análisis económico **65**
B.1. Fabricación PCBs 65
B.2. Gastos de personal 65
B.3. Gastos de equipamiento y material 66
B.4. Gastos totales 67

Bibliografía **69**

Índice de figuras

1.1. Arquitectura de comunicaciones del IFS-04. Elaboración propia . . 3

2.1. Esquema del sistema de telemetría completo. Elaboración propia . 8
2.2. Evolución del protocolo CAN [5] 9
2.3. Estructura de un mensaje CAN 2.0A [5] 10
2.4. Primera zona de Fresnel [7] 11
2.5. Pérdidas por espacio libre para diferentes frecuencias a una distancia de 700m. Elaboración propia 13
2.6. $r_{\max}$ de la primera zona de Fresnel para diferentes frecuencias. Elaboración propia 14
2.7. Regiones ITU [14] 16

3.1. Comparativa tasa de datos versus alcance [16] 18
3.2. Estructura de un paquete nRF24L01 [18] 20
3.3. Diagrama de bloques del transceptor nRF24L01 [18] 20
3.4. Módulo nRF24L01PA/LNA 21
3.5. Pinout módulo nRF24L01PA/LNA [19] 21
3.6. Modulación FSK [21] 22
3.7. Circuito de los transceptores CAN 23
3.8. Placa de desarrollo Teensy 4.1 24
3.9. Diagrama de bloques del microcontrolador i.MX RT1060 [24] 24
3.10. Potenciómetro lineal Variohm ELPM75 [25] 25
3.11. Sensor de presión Variohm EPT3100 [26] 26
3.12. Esquema circuito IMU 27
3.13. Diagrama de bloques del módulo GPS [29] 27
3.14. Sensor de temperatura ds18B20 [30] 28
3.15. Eficiencia del convertidor CC-CC para diferentes valores de corriente [32] 29
3.16. Esquema convertidor CC-CC conmutado ($Vin = 24$V y $Vout = 5$V) 31
3.17. Esquema regulador lineal ($Vin = 5$V y $Vout = 3.3$V) 32
3.18. Diseño PCB unidad de telemetría 32
3.19. Vista superior e inferior en tres dimensiones 33

3.20. Circuito de alimentación 33
3.21. Transceptores CAN 34
3.22. PCB unidad de telemetría previo a su montaje en el vehículo 34
3.23. PCB unidad de control previo su montaje en el vehículo 35
3.24. Arquitectura de comunicaciones del IFS-05 36
3.25. Antena omnidireccional 8dBi 37
3.26. Fotografía de la antena montada en el IFS-05 37
3.27. Fotografia de la posición del piloto respecto a la antena 38

4.1. Estructura del código de la unidad de telemetría 40

5.1. Porcentaje del pedal del acelerador - FSS Endurance 52
5.2. Corriente a la salida del acumulador - FSS Endurance 52
5.3. Tensión del bus de continua - FSS Endurance 53
5.4. Temperatura motor e inversor - FSS Endurance 53
5.5. Revoluciones del motor - Telemetria IFS-04 54
5.6. Revoluciones del motor - Telemetría IFS-05 55

A.1. Esquema circuito completo de la unidad de telemetría 60
A.2. Esquema general de la PCB de la unidad de telemetría 61
A.3. Esquema circuito completo de la unidad de control 62
A.4. Esquema general de la PCB de la unidad de control 63

Índice de tablas

1.1. Puntuaciones *Formula Student* según la norma *FSG* 2023 1.1 [1] . 2

2.1. Bandas de frecuencia dentro del espectro radioeléctrico 15
2.2. Resumen de las bandas de frecuencia ISM definidas por la ITU-R [13] 15

3.1. Comparativa protocolos de comunicación inalámbrica. Elaboración propia 19
3.2. Cálculo del valor de la inductancia **L1** 29

B.1. Comparación entre JLCPCB, PCBWay y Eurocircuits 66
B.2. Gastos de equipamiento 66
B.3. Gastos totales 67

Índice de código

4.1. Inicialización variables CAN 41
4.2. Inicialización variables telemetría 41
4.3. Inicialización timers 42
4.4. Definición variables sensores 42
4.5. Inicialización del programa 43
4.6. Bucle de ejecución del programa 44
4.7. Función CANparse 45
4.8. Función nrf24send 46
4.9. Función nrf24write 48

Capítulo 1

Introducción

1.1. Formula Student

Formula Student es un evento internacional que reúne a estudiantes de ingeniería, que representando a su universidad diseñan, fabrican y compiten con un vehículo de tipo Formula de dimensiones reducidas. El objetivo de la competición no es solo diseñar el mejor vehículo en términos de rendimiento y prestaciones, sino que también se hace énfasis en otros aspectos del mundo ingenieril, como es el control de presupuestos y la gestión de procesos. De esta forma, las pruebas que componen la competición se pueden dividir en dos categorías: pruebas estáticas y pruebas dinámicas.

Las pruebas estáticas se dividen en tres y aportan 325 puntos de los 1000 totales de la competición. En el *Engineering Design* se presentan y defienden las decisiones ténicas que han resultado en el diseño del vehículo. En el *Cost and Manufacturing* se desglosan todos los costes de fabricación, simulando una producción en serie. Por último, en el *Bussiness Plan*, se presenta una idea de negocio en torno al vehículo y se estudia su viabilidad.

Por otra parte, las pruebas dinámicas ponen a prueba el rendimiento en pista del vehículo. Para poder participar en ellas es necesario que el vehículo cumpla todos los puntos de la normativa vigente en el año de competición, los cuales son revisados en diversas inspecciones técnicas. Tras ello, se suceden pruebas de agarre y estabilidad en circuito (*Skidpad*), pruebas de aceleración en una recta de 75m (*Acceleration*), manejo del vehículo (*Autocross*) y resistencia (*Endurance*) Sobre esta última prueba se mide la eficiencia del vehículo, puediendo sumar hasta 100 puntos más al total. Todas estas pruebas cuya puntuación se recoge en la Tabla 1.1 se pueden resumir en un objetivo principal para todos los equipos: reducir el

tiempo por vuelta del vehículo.

Nombre	Tipo	Puntuación
Engineering Design	Prueba estática	150 puntos
Cost and Manufacturing	Prueba estática	100 puntos
Business Plan Presentation	Prueba estática	75 puntos
Skidpad	Prueba dinámica	75 puntos
Acceleration	Prueba dinámica	75 puntos
Autocross	Prueba dinámica	100 puntos
Endurance + Efficiency	Prueba dinámica	325 + 100 puntos

Tabla 1.1: Puntuaciones *Formula Student* según la norma *FSG* 2023 1.1 [1]

Para lograr una reducción en el tiempo por vuelta, es esencial comprender a fondo el comportamiento dinámico del vehículo mientras se desplaza por el circuito. Este entendimiento se basa en la habilidad de medir y monitorizar con precisión las variables cruciales que influyen en el rendimiento del vehículo, como tensiones, corrientes, temperaturas, etc, que ofrecen la respuesta del vehículo a diferentes condiciones y estímulos en tiempo real.

Este proceso de recogida de datos no solo es relevante durante las competiciones, sino también durante las etapas de prueba. Al poder visualizar y analizar estas variables en tiempo real, se pueden validar los resultados obtenidos de las simulaciones previas, permitiendo una correlación precisa de los modelos teóricos con lo que realmente sucede en la pista. Con esta información en mano, el equipo tiene la capacidad de hacer modificaciones en el vehículo para mejorar su rendimiento.

Además de la optimización del rendimiento, la telemetría juega un papel vital en la seguridad del piloto. Al monitorizar constantemente las variables y parámetros clave, el equipo puede identificar y prever posibles problemas o fallos que puedan surgir. Esta anticipación es esencial, ya que permite tomar medidas preventivas y, en última instancia, garantizar que el piloto esté protegido.

1.2. Telemetría en el ISC (estado del arte)

La telemetría se define como el uso de las telecomunicaciones para mostrar o almacenar medidas de distintas fuentes a una distancia desde el punto o puntos de medida [2].

El prototipo de la temporada 2021-2022 del ISC, el IFS-04, fue el primero en incorporar un sistema de telemetría fiel a la definición anterior. En la Figura 1.1 se puede ver la arquitectura de los sistemas de comunicación de este prototipo. Este sistema recogía fundamentalmente las variables eléctricas del vehículo (tensión del bus DC, corriente del acumulador, corriente de fase, etc), así como temperaturas del motor y del inversor, recorrido del pedal del acelerador y presión de frenos. Estas variables se enviaban a través de un bus CAN, del cual recogían información tanto la pantalla del piloto para avisar en caso de algún valor anómalo, como la unidad de telemetría, la cual estaba equipada con un módulo de comunicación inalámbrica basado en LoRa [3]. Este módulo hacía las veces de emisor en una comunicación unidireccional con el otro módulo, el cual se encontraba en la pista para recibir los paquetes de datos y almacenarlos en un archivo **.txt**. Estos datos eran después procesados mediante un script desarrollado en Matlab. Con este sistema se consiguió diagnosticar diversos problemas que sufrió el coche durante la temporada, pero debido a la limitación de la tecnologia LoRa se decidió buscar una alternativa.

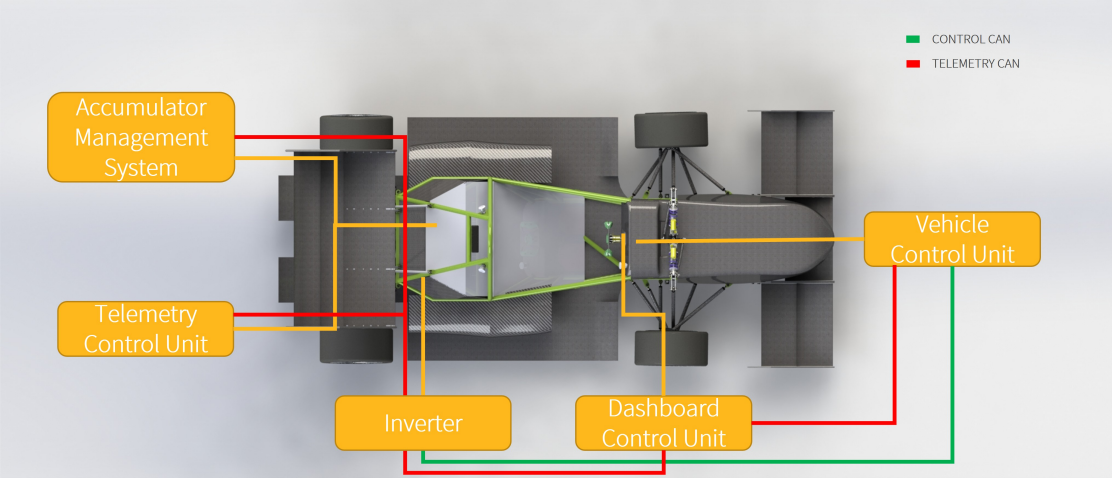


Figura 1.1: Arquitectura de comunicaciones del IFS-04. Elaboración propia

Al comienzo de la temporada 2021-2022 se escogió LoRa principalmente por su conectividad a larga distancia, facilitada por su baja frecuencia de operación a 868 MHz. Al ser la primera vez que se implementaba un sistema de telemetría en el prototipo, existía la preocupación de que debido a las distancias en los circuitos, se perdiera la comunicación con el vehículo. Este alto rango sin embargo implicaba una frecuencia de envío entre cada paquete muy baja, lo que provocó que se perdiera información esencial para entender el estado del vehículo.

En relación a la recogida, empaquetamiento y envío de datos, el sistema de telemetría del IFS-04 se encargaba de reenviar los datos que leía del bus CAN de telemetría, sin establecer ninguna frecuencia de envío para cada medida, de forma que los datos no estaban realmente sincronizados.

1.3. Objetivo del proyecto

Con todo lo expuesto anteriormente, se puede concluir que el principal objetivo de este Trabajo de Fin de Grado es diseñar e implementar un sistema de telemetría para el prototipo Formula Student de la temporada 2022-2023, el IFS-05. Este sistema tendrá que ser capaz de resolver los problemas que surgieron en el IFS-04, permitiendo al equipo mejorar su puntuación tanto en las pruebas estáticas, como en las dinámicas.

Al correlacionar los resultados en tiempo real con las simulaciones, el equipo puede asegurarse de que cualquier mejora implementada en el prototipo no solo sea teóricamente sólida, sino que también sea eficaz en la práctica. Este proceso iterativo permitirá una evolución sistemática de los prototipos futuros, basada en datos cuantificables y experiencias reales en la pista. Para conseguir este objetivo, las principales metas a alcanzar son las siguientes:

1. Encontrar y probar una tecnología de comunicación inalámbrica que ofrezca una mayor tasa de datos que LoRa, aunque esto resulte en una disminución del rango de la señal. Se evaluarán diferentes protocolos de comunicación y se elegira el que mejor se ajuste a estas necesidades.

2. Diseñar e implementar una solución propia basada en microcontrolador para la recogida y envío de datos. Se buscará la escalabilidad para su integración con prototipos futuros.

3. Desarrollar un sistema de envío en el que los datos estén empaquetados de tal forma que exista una sincronización temporal entre ellos, que permita su posterior análisis para prototipos futuros.

1.4. Alineación con los ODS

Aunque este proyecto esté centrado en el desarrollo de un sistema de telemetría para un vehículo Formula Student, todas las mejoras y conclusiones derivadas de él pueden aplicarse a cualquier campo que requiera un sistema de telemetría, tal y como se expone en [2]. De esta forma la contribución de un sistema de telemetría a los Objetivos de Desarrollo Sostenible es muy amplia, cubriendo varios objetivos:

1. ODS 2: Hambre Cero: con un sistema de telemetría se puede monitorizar la cadena de suministro de alimentos y se puede garantizar que todas las personas tienen acceso a ellos.

2. ODS 3: Salud y Bienestar: igual que con los alimentos, con un sistema de telemetría se puede garantizar que tanto medicamentos como atención médica lleguen a todo el que los necesita.

3. ODS 7: Energía Asequible y No Contaminante: con un sistema de telemetría se puede hacer un seguimiento del uso de los distintos tipos de energía, haciendo el suministro más sostenible.

4. ODS 9: Industria, Innovación e Infraestructura: se puede mejorar el crecimiento de la industria mediante un sistema de telemetría que permita reducir el desperdicio y aumentar la productividad.

5. ODS 11: Ciudades y Comunidades Sostenibles: monitorizando el uso de los recursos en las ciudades se puede conseguir que se reduzca el desperdicio y aumente la eficiencia energética. Esto supondrá un aumento en la calidad de vida de las personas.

En conclusión, aunque este proyecto se centre en telemetría para vehículos, sus hallazgos e innovaciones pueden extrapolarse para avanzar en múltiples ODS y mejorar la sostenibilidad global.

Capítulo 2

Descripción del sistema

2.1. Sistema emisor

El sistema emisor desempeña un papel crítico en el funcionamiento del sistema de telemetría. Este, representado en la Figura 2.1, tiene como núcleo principal la Telemetry Control Unit (TCU). La TCU no sólo recopila datos directamente de sensores que se encuentran en su cercanía, sino que también tiene la capacidad de recibir información de otras unidades de control. Estas unidades, como la Vehicle Control Unit (VCU) o el Accumulator Management System (AMS), recopilan y procesan datos de diversos sensores distribuidos en distintas áreas del vehículo. Luego, estos datos son enviados a la TCU mediante un bus CAN para ser posteriormente transmitidos.

Además de la recolección de datos, la TCU tiene la responsabilidad de procesar y empaquetar adecuadamente esta información para su transmisión inalámbrica. Esta etapa es crucial, ya que garantiza que los datos se transmiten de forma clara y coherente al receptor. Es importante destacar que, independientemente de la fuente original de los datos, ya sea directamente desde un sensor o de una unidad de control como la VCU o la AMS, el proceso de empaquetado y transmisión llevado a cabo por la TCU es el mismo, como se verá en el capítulo de diseño del software.

El esquema de la Figura 2.1 destaca la relevancia de la VCU en el proceso de telemetría del vehículo. Esta unidad toma como entrada la posición del pedal del acelerador, interpretando las acciones del piloto. Una vez procesada esta información, envía una consigna de par al inversor a través del bus CAN designado como CAN INV. Este bus no sólo sirve para las instrucciones, sino también para que el inversor transmita valiosa información como la tensión del bus de continua, la corriente de fase, y datos acerca de la temperatura y velocidad del motor.

A medida que estos datos se generan, se retransmiten al bus de telemetría, conocido como CAN TEL. A este flujo de información, se suman datos provenientes de los sensores de suspensión delanteros. Paralelamente, la AMS aporta al sistema enviando datos cruciales acerca de la tensión mínima de las celdas, así como la temperatura máxima y la corriente de salida del acumulador. Todo este conjunto de datos es finalmente recibido por la TCU para su respectivo procesamiento.

Adicionalmente, la TCU tiene otras funciones inherentes. Esta unidad se encarga de obtener lecturas de los sensores de suspensión traseros, además de registrar datos de un acelerómetro y un GPS. Una vez que la TCU recibe todos estos datos, los organiza y transmite a frecuencias determinadas, basándose en la relevancia y urgencia de cada pieza de información, asegurando que los datos críticos sean enviados con la prioridad necesaria.

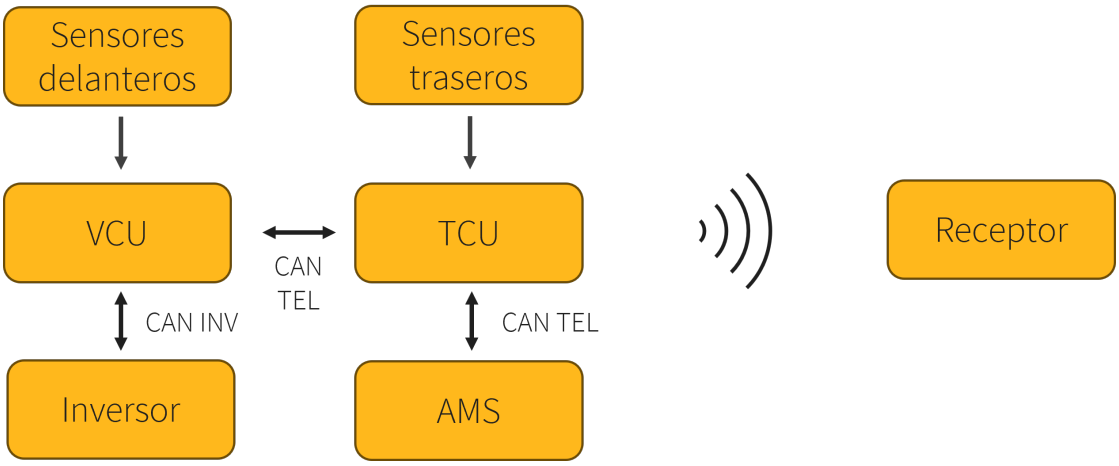


Figura 2.1: Esquema del sistema de telemetría completo. Elaboración propia

En el lado del receptor será necesario un software visualizador aparte del hardware de recepción para poder mostrar la información recopilada. Este software mostrará en tiempo real las métricas y valores recopilados, permitiendo un análisis detallado y una toma de decisiones más informada. Además, puede ofrecer características adicionales, como el almacenamiento de datos para futuras referencias, alertas en tiempo real para parámetros que exceden los límites seguros, y herramientas analíticas para estudiar tendencias o patrones en los datos.

2.2. Comunicación CAN

Como se muestra en la Figura 2.1 las unidades de control se comunican entre ellas a través del protocolo CAN (*Controller Area Network)*. Este protocolo de comunicación fue desarrollado por la empresa Bosch en la década de 1980 para reducir el cableado dentro de los vehículos, ya que se compone de dos cables (*CAN Low* y *CAN High*)[4] sobre los que se mide la diferencia de tensiones. Cuando la tensión diferencial entre las dos señales es 0V, se está transmitiendo un 1 digital y cuando es distinta de 0V se transmite un 0 digital. En la Figura 2.2 se puede ver la evolución de este protocolo. Actualmente la versión de CAN empleada en el IFS-05 es la 2.0 con una tasa de datos máxima de 1 Mbit/s.

En relación a los mensajes que se envían a través de este bus, como se muestra en la Figura 2.3, la estructura que siguen es la siguiente:

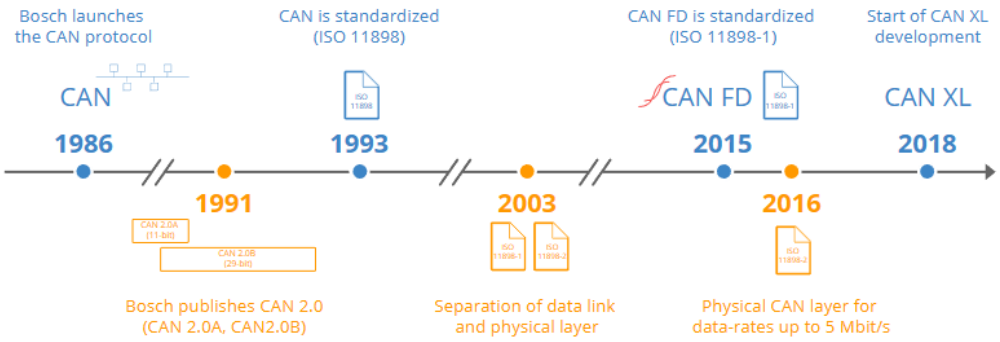


Figura 2.2: Evolución del protocolo CAN [5]

1. Start of Frame (SOF): Indica el comienzo de un nuevo mensaje.
2. Identificador (ID): Representa la dirección o el tipo del mensaje. Existen dos variantes:
   a) CAN 2.0A (Estándar): 11 bits.
   b) CAN 2.0B (Extendido): 29 bits (11 bits estándar + 18 bits adicionales).
3. Bit RTR (Remote Transmission Request): Este bit define si el mensaje es un pedido de datos (bit dominante) o una respuesta a una solicitud (bit recesivo).

4. Bit IDE (Identifier Extension): Indica si el mensaje utiliza el formato estándar (bit recesivo) o el formato extendido (bit dominante).

5. Campo de control (Control Field): Contiene información sobre la longitud del campo de datos.

6. Campo de datos (Data Field): Puede ser de 0 a 8 bytes, dependiendo de lo que se haya especificado en el DLC.

7. Campo CRC (Cyclic Redundancy Check): Es una secuencia de 15 bits utilizada para la detección de errores. Después del CRC, hay un bit recesivo que actúa como delimitador.

8. Bit ACK (Acknowledgment): Los nodos que reciben el mensaje correctamente transmitirán un bit dominante durante la ranura de ACK. Si el transmisor recibe un bit recesivo en lugar de un bit dominante, entiende que el mensaje no fue reconocido por ningún nodo.

9. Delimitador ACK: Un bit recesivo que sigue al bit ACK.

10. Fin de trama (End of Frame, EOF): Representado por 7 bits recesivos, indica el fin del mensaje.

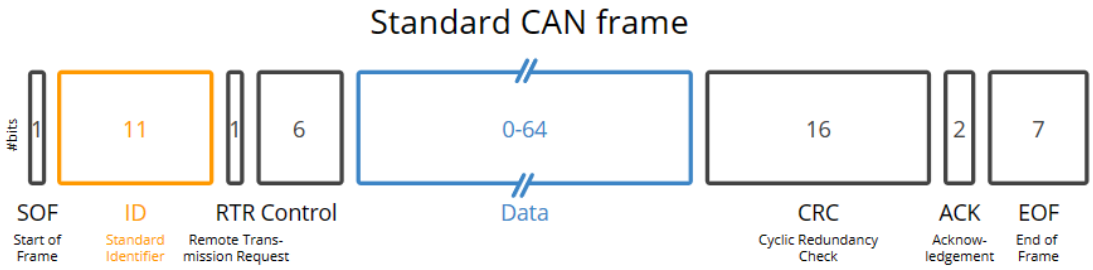


Figura 2.3: Estructura de un mensaje CAN 2.0A [5]

2.3. Comunicación inalámbrica

Como se mencionó en la introducción, es esencial identificar una tecnología de comunicación inalámbrica que ofrezca una tasa de transferencia adecuada para las variables que se van a transmitir. Independientemente de la tecnología escogida es crucial considerar las restricciones inherentes a estas comunicaciones. Dado que son inalámbricas, dependen de la dispersión de ondas electromagnéticas en una

frecuencia específica. Esto impone varias limitaciones en el enlace.

Al tratarse de una onda electromagnética, la potencia de la señal se reduce en función del cuadrado de la distancia a la misma. Dicha potencia varía según el material atravesado. Materiales con poca conductividad generan más pérdidas en el campo que el aire, mientras que los de alta conductividad reflejan el campo. Así, cualquier obstáculo físico atenúa la potencia del campo en el receptor.

Cuando las ondas electromagnéticas se propagan de una antena a otra, la mayor parte de la energía se transporta en la primera zona de Fresnel [6]. Esta es el elipsoide circular con las antenas en sus puntos focales. Desde una perspectiva geométrica, la calidad de un canal inalámbrico no solo está definida por una línea directa de visión, sino que depende de una primera zona de Fresnel libre de obstrucciones. Además de la importancia de la primera zona de Fresnel, otros factores como la atenuación atmosférica, las interferencias y las reflexiones influyen en la calidad de la transmisión.

Otro factor importante a tener en cuenta es la banda de frecuencia. A mayores frecuencias, aunque la tasa de datos es mayor, la señal se vuelve más vulnerable a los obstáculos. Asimismo, la potencia de la señal emitida debe superar el nivel de ruido ambiental para que, al llegar al receptor, pueda ser diferenciada del ruido. Si la potencia es insuficiente, el receptor no podrá distinguir la señal del ruido circundante.

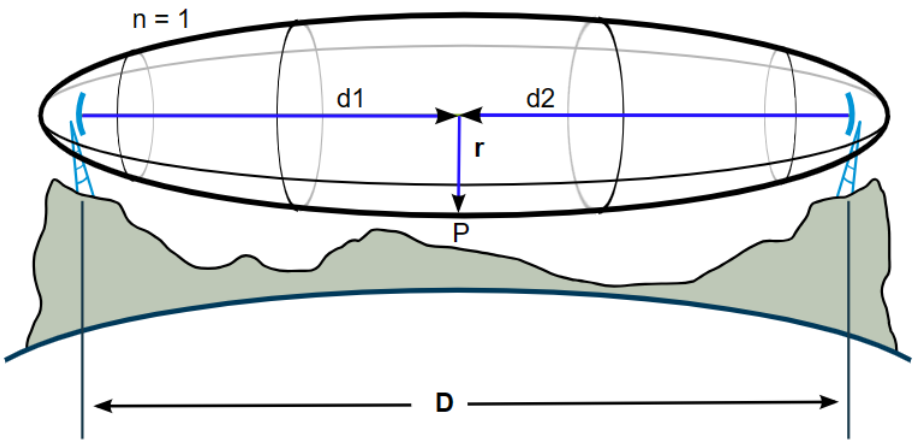


Figura 2.4: Primera zona de Fresnel [7]

Todas estas consideraciones son recogidas por la ecuación 2.1, que modela todas las pérdidas y ganancias desde el transmisor hasta el receptor.

$$P_{\mathrm{RX}} = P_{\mathrm{TX}} - L_{\mathrm{TX}} + G_{\mathrm{TX}} - L_{\mathrm{FSPL}} - L_{\mathrm{DIV}} + G_{\mathrm{RX}} - L_{\mathrm{RX}} \tag{2.1}$$

donde:

- P_{RX} = potencia recibida
- P_{TX} = potencia transmitida
- L_{TX} = pérdidas en el lado del transmisor (antena, cables, etc)
- G_{TX} = ganancia de la antena del transmisor
- L_{FSPL} = pérdidas en el espacio libre
- L_{DIV} = otras pérdidas
- G_{RX} = ganancia de la antena del receptor
- L_{RX} = perdidas en el lado del receptor (antena, cables, conector, etc)

De todos estos términos el que más afecta a la potencia recibida para este proyecto son las pérdidas en el espacio libre. Estas pérdidas se pueden modelar según la siguiente ecuación [8, 9]:

$$L_{\mathrm{FSPL}} = 20 \times \log(\frac{4 \times \pi \times d}{\lambda}) \tag{2.2}$$

Donde:

- λ = longitud de onda de la señal
- d = distancia entre transmisor y receptor

Usando esta ecuación para diferentes frecuencias pertenecientes a las bandas ISM (ver Tabla 2.2) y planteando una distancia media de 700m entre el transmisor y el receptor se obtiene la gráfica de la Figura 2.5.

Por otra parte, la ecuación 2.3 permite calcular el radio de la sección de una zona n de Fresnel:

$$r_{\mathrm{n}} = \sqrt{\frac{\lambda \times n \times d_1 \times d_2}{d_1 + d_2}} \tag{2.3}$$

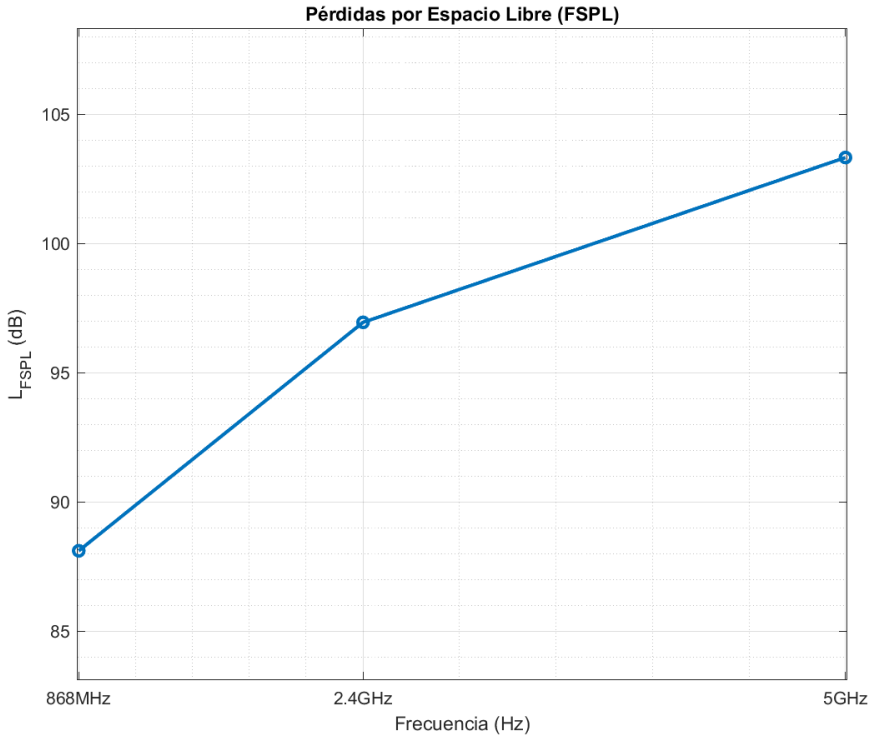


Figura 2.5: Pérdidas por espacio libre para diferentes frecuencias a una distancia de 700m. Elaboración propia

Donde d_1 y d_2 son la distancia de un punto de la elipse a las antenas. El radio máximo de la sección de la elipse que define la primera zona de Fresnel ($n = 1$) para una distancia $d = 700$m entre transmisor y receptor se encuentra en $d_1 = d_2 = \frac{d}{2} = 350$m. De esta forma la ecuación 2.3 queda:

$$r_{\max} = \frac{\sqrt{\lambda \times d}}{2} \tag{2.4}$$

El valor máximo del radio de la primera zona de Fresnel es de suma importancia para el enlace entre transmisor y receptor [10]. Permite definir el área del círculo centrado entre la línea que une transmisor y receptor, la cual debe de estar libre de obstrucciones. Por ello a mayor radio, mayor área crítica que habrá que mantener libre. Usando esta ecuación para las mismas frecuencias que 2.2, se obtiene la gráfica de la Figura 2.6.

Comparando los resultados de las Figuras 2.5 y 2.6, se puede ver como al aumentar la frecuencia de la señal, aumentan también las pérdidas en el espacio libre, pero el radio máximo de la primera zona de Fresnel disminuye.

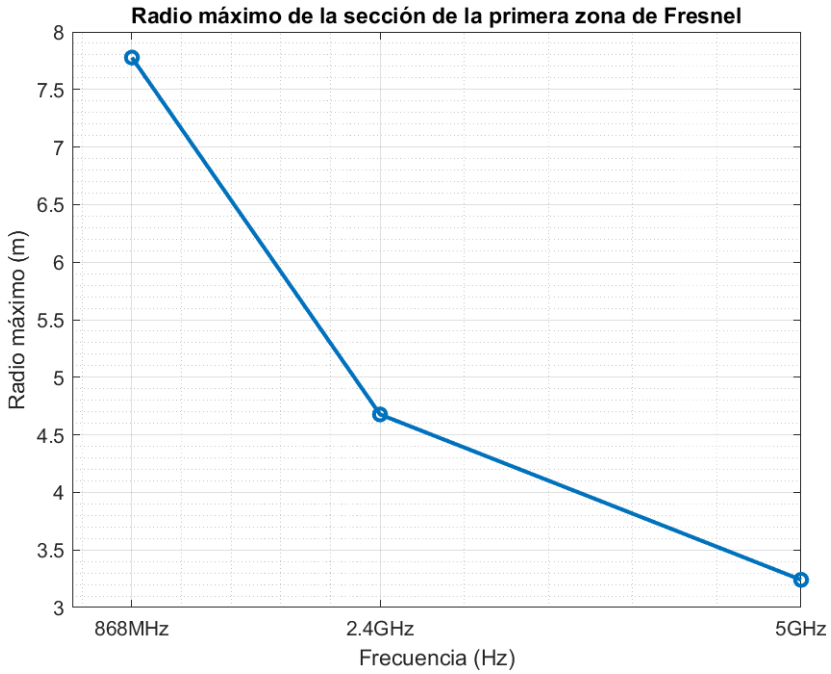


Figura 2.6: $r_{\max}$ de la primera zona de Fresnel para diferentes frecuencias. Elaboración propia

Por último, es esencial estar al tanto de la normativa sobre emisiones electromagnéticas. En España, el Estado gestiona el espectro radioeléctrico. El Cuadro Nacional de Atribución de Frecuencias (CNAF) [11] especifica los propósitos de cada banda, cubriendo desde 8,3 kHz hasta 3000 GHz. Para maximizar el aprovechamiento del espectro destinado a comunicaciones, se clasifica en diferentes bandas, asignando un servicio específico a cada una. En la Tabla 2.1 se detalla la división de frecuencias, además del uso típico de cada una.

En el contexto internacional, es importante mencionar las bandas ISM (*Industrial, Scientific and Medical*). Estas son bandas de radiofrecuencia reservadas para uso no comercial en áreas industrial, científica y médica y ofrecen la ventaja de poder ser utilizadas sin necesidad de licencia, siempre que se respeten ciertos límites de potencia. La International Telecomunications Union (ITU) [12] divide el mundo en tres zonas diferentes como se puede ver en la Figura 2.7 para administrar el espectro electromagnético. Cada una de las zonas tiene su propia asignación de frecuencias. Una dificultad de emplear frecuencias dentro de las bandas ISM es que numerosos dispositivos las usan, elevando el riesgo de interferencias. En la Tabla 2.2 se recogen las bandas ISM definidas por la ITU-R.

Bandas de frecuencia	Frecuencias	Uso típico
Very Low Frecuencies	3 kHz - 30 kHz	Navegación, servicios marítimos
Low Frecuencies	30 kHz - 300 kHz	Radio AM, navegación
Medium Frecuencies	300 kHz - 3 MHz	Radio AM
High Frecuencies	3 MHz - 30 MHz	Radiodifusión internacional
Very High Frecuencies	30 MHz - 300 MHz	TV, radio FM
Ultra High Frecuencies	300 MHz - 3 GHz	TV, telefonía móvil, Wi-Fi, radar
Super High Frecuencies	3 GHz - 30 GHz	Satélites, radar, Wi-Fi, 5G
Extreme High Frecuencies	30 GHz - 300 GHz	Comunicaciones por satélite, radar

Tabla 2.1: Bandas de frecuencia dentro del espectro radioeléctrico

Rango de frecuencias	**Aplicaciones comunes**	**Disponibilidad**
0.135MHz - 0.138MHz	Radiofrecuencia (RFID)	Mundial
6.765MHz - 6.795MHz	Radiofrecuencia (RFID)	Mundial
13.553MHz - 13.567MHz	Radiofrecuencia (RFID)	Regional
26.957MHz - 27.283MHz	Radio CB	Mundial
40.66MHz - 40.70MHz	Sensores y alarmas	Mundial
433.05MHz - 443.79MHz	Control remoto, domótica	Mundial
868MHz - 928MHz	RFID, comunicaciones industriales	Regional
2.4GHz - 2.5GHz	Wi-Fi, Bluetooth, Microondas	Mundial
5.725GHz - 5.875GHz	WLAN, Radares de corta distancia	Regional
24GHz - 24.25GHz	Comunicaciones de alta velocidad	Mundial
61GHz - 61.5GHz	Comunicaciones inalámbricas	Regional
122GHz - 123GHz	Investigación científica	Regional
244GHz - 246GHz	Investigación científica	Regional

Tabla 2.2: Resumen de las bandas de frecuencia ISM definidas por la ITU-R [13]

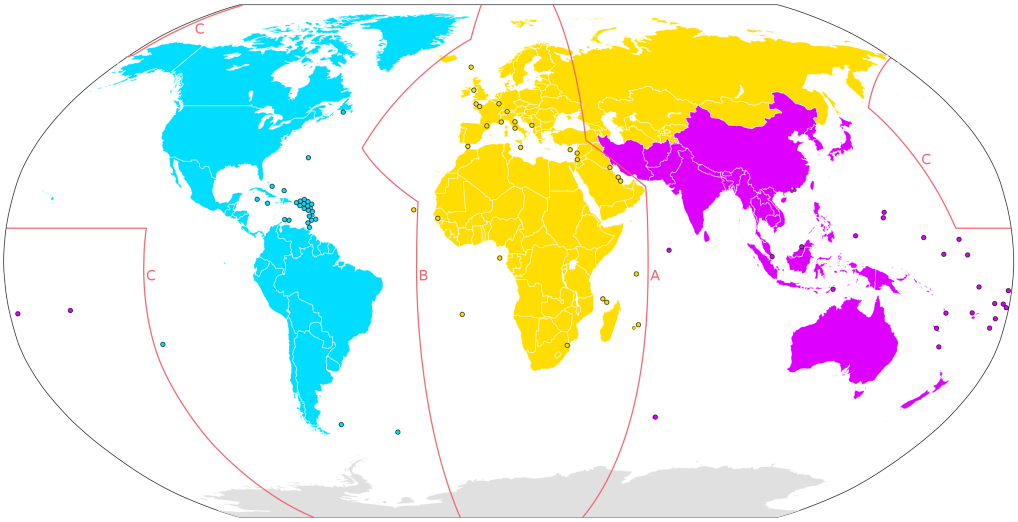


Figura 2.7: Regiones ITU [14]

Capítulo 3

Diseño del hardware del sistema de telemetría

En este capítulo, se detallará el hardware seleccionado para el sistema de telemetría del IFS-05. Tal y como se esbozó en el capítulo previo con la Figura 2.1, este sistema de telemetría constará de una unidad de telemetría (TCU). Esta unidad funcionará como emisor, transmitiendo los datos proporcionados tanto por sensores conectados directamente a la TCU como por aquellos datos recibidos vía el bus CAN de telemetría (CAN TEL) procedentes de otras unidades de control del vehículo.

Para incorporar y unificar todos los componentes que se describirán a continuación, se realizará el diseño de una PCB específica. El diseño de esta PCB será fundamental para garantizar no sólo la integración física de los componentes, sino también para asegurar la eficiencia en las comunicaciones. El diseño se realizará con la suficiente flexibilidad para permitir futuras actualizaciones o modificaciones en el sistema.

3.1. Protocolo de comunicación inalámbrica

Uno de los objetivos principales de este proyecto es encontrar un protocolo de comunicación inalámbrica que tenga un rango de alcance suficiente para poder mantener la comunicación entre el vehículo y la estación de recepción durante el desarrollo de las diferentes pruebas de la competición y que además posea una tasa de datos lo suficientemente amplia para poder enviar todas las variables medidas de los diferentes sensores. En la Figura 3.1 se muestra una comparativa entre alcance y tasa de datos de estos protocolos. A continuación, se presentarán diversas alternativas y se analizarán sus propiedades.

3.1.1. Bluetooth

Esta tecnología opera en la banda ISM (Tabla 2.2) de 2.4GHz y se emplea principalmente para la transmision de datos entre dispositivos, especialmente de voz. Permite conexiones punto a punto y ofrece tasas de datos de 1-3Mb/s, lo que cubriría las necesidades actuales de transmisión de datos. Sin embargo, su rango es limitado, pues apenas alcanza los 20m, por lo que queda descartado para esta aplicación.

3.1.2. ZigBee

ZigBee es un protocolo de comunicación inalámbrica enfocado en aplicaciones digitales que requieren bajo consumo de energía. Se fundamenta en el estándar IEEE 802.15.4 [15] destinado a redes inalámbricas de área personal. Aunque opera en las bandas de frecuencia ISM, similar a Bluetooth, en Europa utiliza principalmente la frecuencia de 868MHz. Aunque su velocidad de transmisión máxima es de 255 kbps, está optimizado para enviar paquetes pequeños y por lo tanto no es adecuado para este sistema.

3.1.3. Redes móviles (4G/5G)

Otra alternativa a considerar es el uso de redes inalámbricas 4G o 5G, ya que estas están diseñadas para entornos móviles y ofrecen altas tasas de datos. El principal problema que presentan este tipo de redes es su dependencia de licencias comerciales, obligando a los usuarios finales a depender de los proveedores de este tipo de servicios. Esto puede suponer un problema, ya que no se puede garantizar la cobertura de la red en las diferentes ubicaciones de los eventos ni la tasa de datos en caso de congestión de la red.

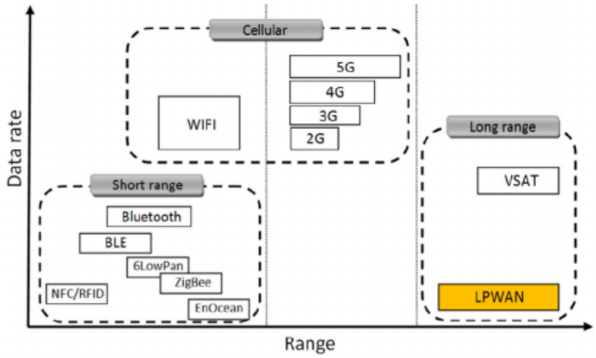


Figura 3.1: Comparativa tasa de datos versus alcance [16]

3.1.4. nRF24

nRF24 es un protocolo diseñado por Nordic Semiconductor[17] que opera en la banda ISM de 2.4GHz. Presenta una tasa de datos de hasta 2 Mbps y un alcance de 1000m. Aunque no proporciona las altas tasas de datos que ofrecen tecnologías como Wi-Fi o Bluetooth, es especialmente adecuado para aplicaciones que requieren transmisiones simples y confiables entre dispositivos. Su capacidad de configurar múltiples canales de comunicación lo hace útil en entornos donde diferentes dispositivos necesitan comunicarse simultáneamente sin interferencias. Aparte de estas carácterísticas de rango y tasa de datos que se ajustan a las necesidades del proyecto, este protocolo dispone de librerías para la plataforma Arduino, lo que facilitaría su integración.

Una vez presentados los distintos candidatos a sustituir a LoRa como protocolo de comunicación inalámbrica para el sistema de telemetría del IFS-05, se han recogido sus características principales en la Tabla 3.1. Analizando estas características, el protocolo que mejor cumple el requisito inicial de equilibrar tasa de datos con rango es nRF24. Dentro de este protocolo de comunicación se usará el transceptor nRF24L01. Este transceptor cuyo diagrama de bloques se puede ver en la Figura 3.3, utiliza el protocolo *SPI* para comunicarse con el microcontrolador de la unidad de control. Tal y como describe su datasheet [18] emplea un sistema de corrección de errores con un campo de control junto con un campo de redundancia cíclica (CRC). Esto permite detectar si ha habido algún error en el transmisión o recepción de los datos.

Protocolo	LoRa	Bluetooth	ZigBee	4G	nRF24
Frecuencia	868 MHz	2.4 GHz	868 MHz	700 MHz-2.6 GHz	2.4 GHz
Alcance (aprox.)	10 km	Hasta 100 m	Hasta 100 m	Hasta 30 km	Hasta 1000 m
Tasa de datos	50 kbps	1-3 Mbps	Hasta 256 kbps	Hasta 100 Mbps	Hasta 2 Mbps
Consumo	Bajo	Moderado	Bajo	Alto	Bajo
Licencia	Libre	Libre	Libre	Comercial	Libre

Tabla 3.1: Comparativa protocolos de comunicación inalámbrica. Elaboración propia

La estructura de los paquetes enviados a través del módulo nRF24L01, como se muestra en la Figura 3.2 es la siguiente:

1. Preamble: es una secuencia de bits utilizada para detectar niveles 0 y 1 en el receptor. Puede tomar los valores 01010101 si el primer bit de la dirección es 0 ó 10101010 si el primer bit es 1. Esto asegura que haya suficientes transiciones para estabilizar el receptor.

2. Address: Es la dirección del receptor para que detecte el paquete correcto.

3. Packet Control Field: es el campo de control del paquete que permite detectar si el paquete recibido es nuevo o ya se ha retransmitido. Si un paquete tiene el mismo identificador que el paquete anterior el transceptor compara el CRC de ambos paquetes. Si ambos son iguales, el último paquete recibido se considera una copia del anterior y se descarta.

4. Payload: es el contenido del paquete definido por el usuario. Tiene un tamaño máximo de 32 bytes.

5. CRC: es el mecanismo de detección de errores y se calcula a partir del campo *Address*, *Packet Control Field* y *Payload*.



Figura 3.2: Estructura de un paquete nRF24L01 [18]

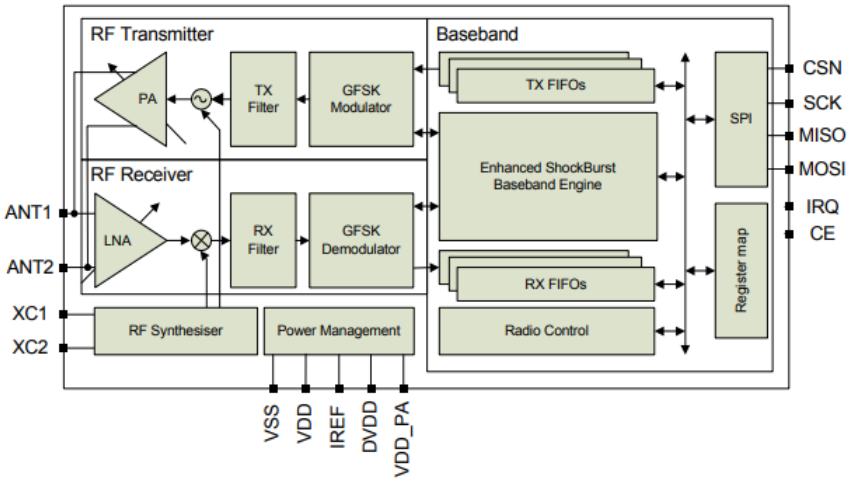


Figura 3.3: Diagrama de bloques del transceptor nRF24L01 [18]

Para implementar este protocolo en la PCB de la unidad de telemetría se empleará el módulo nRF24L01PA/LNA que se puede ver en la Figura 3.4. Este se caracteriza por incorporar un conector SMA para montar una antena externa y

por incluir un circuito integrado para amplificar la potencia de la señal del emisor y así conseguir un alcance aproximado de 1km.

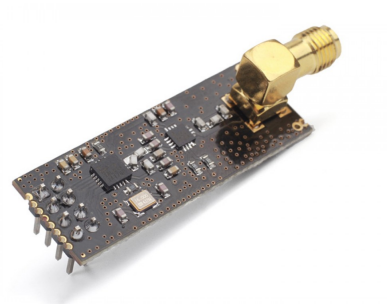


Figura 3.4: Módulo nRF24L01PA/LNA

Como ya se ha mencionado, este módulo se comunicará con el microprocesador a través del protocolo *SPI*. En la Figura 3.5 se muestran las conexiones del módulo. Estas conexiones son las necesarias para la comunicación *SPI* (SCK, MOSI, MISO y CS), alimentación (3.3V), GND y CE (para transmitir o recibir los datos).

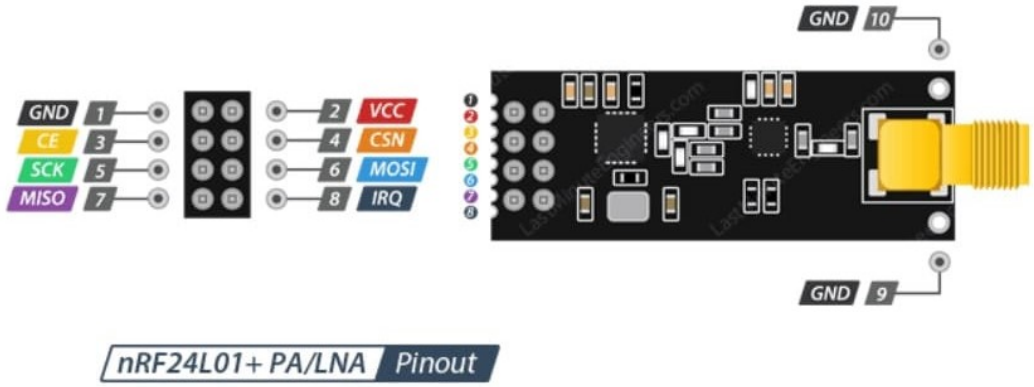


Figura 3.5: Pinout módulo nRF24L01PA/LNA [19]

Para modular las señales el módulo nRF24L01 hace uso de la técnica GFSK (*Gaussian Frequency Shift Keying*) [20]. Este tipo de modulación consiste en transmitir los datos mediante cambios discretos en la frecuencia de la señal portadora (ver Figura 3.6). Añadiendo un filtro gaussiano a la señal, los cambios de frecuacia son suaves, reduciendo el ancho de banda de la señal transmitida. Esto hace que el transceptor sea menos susceptible a interferencias. Dado que GFSK es una técnica de modulación digital, presenta una relativa inmunidad al ruido. Esto facilita que el nRF24L01 establezca comunicaciones en entornos con alta interferencia.

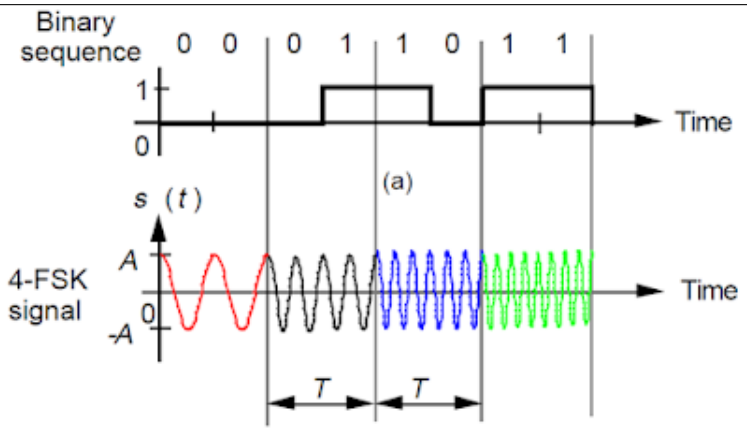


Figura 3.6: Modulación FSK [21]

3.2. Comunicaciones

3.2.1. SPI

El protocolo *SPI* (*Serial Peripheral Interface)* es necesario para las comunicaciones con los sensores integrados en la PCB de la unidad de control de telemetría, debido a su idoneidad para las distancias cortas. Este protocolo se compone de cuatro señales para su correcto funcionamiento. Emplea dos canales para la transmisión de datos (MOSI y MISO), uno para el reloj (SCK) y otro para añadir múltiples dispositivos al mismo bus (CS). El *Teensy 4.1* ya incorpora este protocolo de forma nativa, por lo que no habrá que añadir ningún integrado externo.

3.2.2. CAN

El microcontrolador *Teensy 4.1* ya integra el protocolo *CAN*, eliminando la necesidad de un chip adicional que se comunique mediante un protocolo intermedio como el *SPI*. Sin embargo, para facilitar la comunicación con otras unidades de control del vehículo, se requiere un transceptor. Para este propósito, se optó por tres unidades del modelo SN65HVD230 de Texas Instruments [22] permitiendo expandir el número de buses CAN en futuras iteraciones. La Figura 3.7 muestra el esquema del circuito dentro de la unidad de telemetría (TCU).

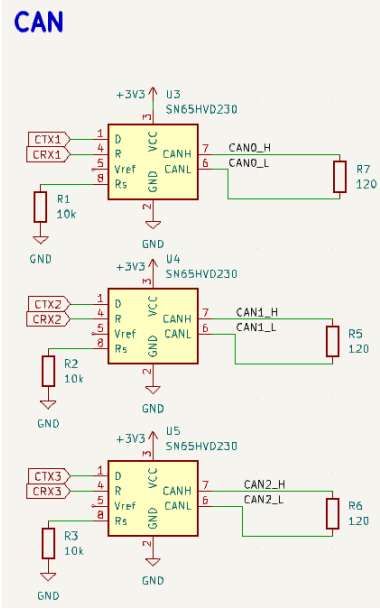


Figura 3.7: Circuito de los transceptores CAN

3.3. Microcontrolador

El microcontrolador se encargará del correcto funcionamiento de la unidad de telemetría mediante la coordinación de las tareas de lectura, procesado y envío. Leerá la información de los sensores conectados directamente a la TCU y los datos recibidos a través del bus CAN de telemetría. Procesará esos datos y los empaquetará para enviarlos a través del módulo nRF24L01. Para este proyecto será necesario un microcontrolador con las suficientes entradas digitales y analógicas para la lectura de los diferentes sensores montados en el vehículo, así como un protocolo de comunicación *SPI* para la comunicación con el módulo nRF24L01 y *CAN* para la comunicación con el resto de unidades de control.

El modelo elegido para esta tarea es el *Teensy 4.1* [23]. Es una tarjeta de desarrollo producida por PJRC equipada con un microcontrolador ARM Cortex-M7 a 600MHz (i.MX RT1060) [24], convirtiéndolo en una de las tarjetas de desarrollo de bajo costo más potentes del mercado. Además, posee numerosas opciones de conectividad, contando con 55 entradas/salidas digitales, 18 analógicas con una resolución de 12 bits a 3.3V, 3 buses *SPI* y 3 buses *CAN*. En la Figura 3.8 se puede ver la placa de desarrollo *Teensy 4.1* y en la Figura 3.9 el diagrama de bloques del microcontrolador. A la hora de programar se empleará el entorno de Arduino debido a la fácil integración con las librerías de los años anteriores.



Figura 3.8: Placa de desarrollo Teensy 4.1

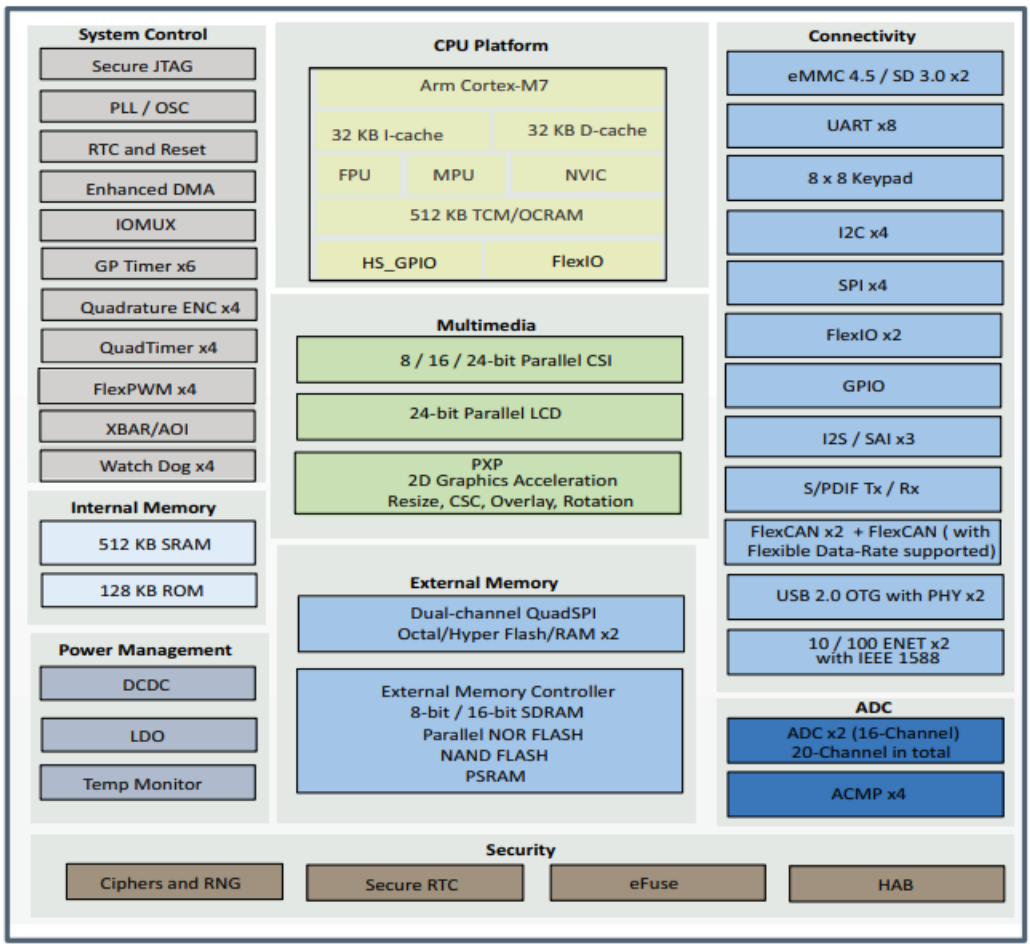


Figura 3.9: Diagrama de bloques del microcontrolador i.MX RT1060 [24]

3.4. Sensores

El microcontrolador discutido en la sección previa tiene la responsabilidad de registrar diversas variables mediante sensores. Parte de estos sensores se integran directamente en la PCB de la unidad de telemetría, mientras que otros se instalan directamente en el vehículo.

3.4.1. Sensores de suspensión y posición del acelerador

Son potenciómetros lineales modelo ELPM75 de la empresa Variohm [25]. Al medir el recorrido lineal de su parte móvil respecto de la parte fija (ver Figura 3.10), se emplean tanto en el pedal del acelerador (2 sensores exigidos por normativa) como en los amortiguadores (4 sensores). La medida del pedal del acelerador la recoge la unidad de control del vehículo (VCU) y la convierte en porcentaje de par para solicitar al inversor a través del bus *CAN* de control del inversor, CAN INV. Esta medida también es reenviada a través del bus *CAN* de telemetría, CAN TEL, para poder visualizar y compararar el par solicitado por el piloto con otras variables, principalmente las revoluciones del motor o en un futuro la velocidad de cada rueda motriz para comprobar si existe deslizamiento entre el neumático y el asfalto.

Por otra parte, los sensores montados en los amortiguadores permiten medir el recorrido de la compresión o elongación de los muelles, esencial para conocer el comportamiento del vehículo en las curvas, especialmente su estabilidad. Dado que los circuitos de Formula Student son muy revirados, conseguir un prototipo estable en las curvas aumenta la manejabilidad y por tanto se puede reducir el tiempo por vuelta.



Figura 3.10: Potenciómetro lineal Variohm ELPM75 [25]

3.4.2. Sensor de presión de frenos

Este sensor se acopla a la línea de frenos (ver Figura 3.11) y se encarga de medir la presión una vez el piloto pisa el pedal del freno. El modelo utilizado es el EPT3100 también de la empresa Variohm [26]. Al igual que con los sensores de posición del pedal del acelerador, este sensor está conectado a la VCU quien reenvía los datos a traves de CAN TEL.



Figura 3.11: Sensor de presión Variohm EPT3100 [26]

3.4.3. Sensor de corriente

Para medir la corriente del bus de continua que sale del acumulador se emplea un sensor de efecto Hall modelo DHAB S/118 [27]. Mide un máximo de 350A y devuelve una señal analógica que recibe el microcontrolador de la unidad de monitorización del acumulador (AMS). Esta medida es también reenviada a través del bus *CAN* de telemetría para su registro y visualización.

3.4.4. Unidad de Medida Inercial (IMU)

Este sensor estará implementado directamente en la PCB de la unidad de telemetría y se comunicará con el microcontrolador *Teensy 4.1* mediante el protocolo *SPI*. Consta de un acelerómetro que se encargará de medir las aceleraciones lineales en los tres ejes, así como de un giroscopio que medirá las velocidades angulares. Esto servirá para conocer el movimiento del vehiculo y determinar la magnitud de las fuerzas a las que está sometido. Se ha elegido el sensor BMI088 [28] de la empresa Bosch debido a su robustez y precisión frente a las vibraciones, convirtiéndolo en un candidato perfecto para este tipo de aplicaciones. La Figura 3.12 muestra el esquema de la IMU en la PCB de la unidad de telemetría (TCU).

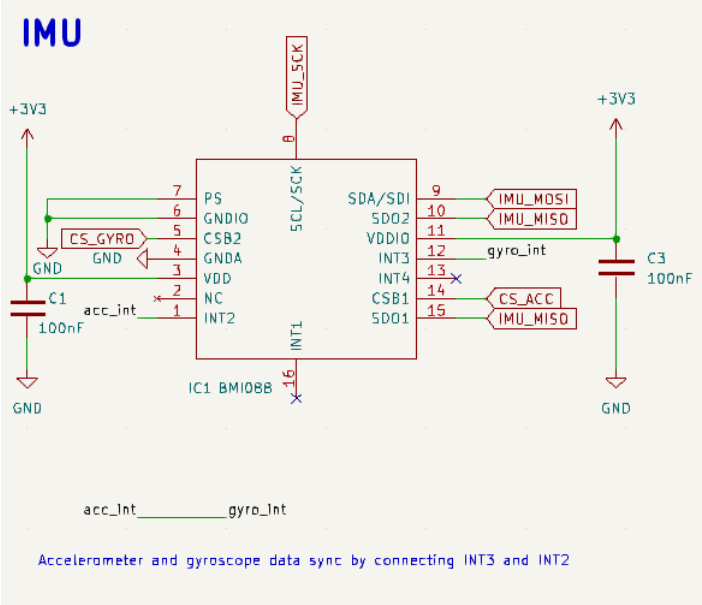


Figura 3.12: Esquema circuito IMU

3.4.5. GPS

Se incorporará un GPS a través del cual se obtendrá la posición del vehículo (latitud y longitud), además de velocidad con una precisión de 0,1 m/s. Presenta una tasa de actualización configurable entre 1Hz y 10Hz. Se ha escogido el modelo PA1616S montado en la placa de desarrollo de Adafruit [29] que se comunicará a través de UART con el *Teensy 4.1*. En la Figura 3.13 se puede ver un diagrama de bloques del módulo GPS.

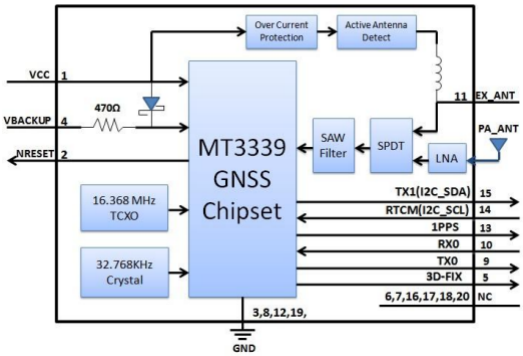


Figura 3.13: Diagrama de bloques del módulo GPS [29]

3.4.6. Sensores de temperatura del líquido refrigerante

Por último, se han montado cuatro sensores de temperatura, uno a la entrada y otro a la salida del los dos radiadores del circuito de refrigeración del motor y del inversor. Se han elegidos los sensores digitales DS18B20[30] (ver Figura, principalmente por su alto rango de temperaturas y el uso del protocolo 1-Wire[31]. Este protocolo permite conectar varios sensores en serie a un solo pin del microcontrolador, ahorrando cableado.

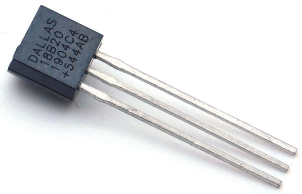


Figura 3.14: Sensor de temperatura ds18B20 [30]

3.5. Circuito de alimentación

El circuito de alimentación del sistema de telemetría es el responsable de alimentar todos los sensores y componentes electrónicos descritos en los apartados anteriores, garantizado su funcionamiento de forma estable protegiéndolo de posibles picos de tensión provenientes de la batería de bajo voltaje del vehículo. Esta batería que opera a 24V está conectada a un circuito de protección para evitar que un cortocircuito en cualquiera de las unidades de control pueda dañar la batería.

Dado que los sensores y los componentes electrónicos funcionan a 5V o 3.3V, será necesario diseñar un circuito de alimentación que convierta los 24V (tensión máxima 28.8V) a estos valores de tensión. Para ello se introducirán convertidores CC-CC para alimentar estos sistemas. Para conseguir 5V a partir de 24 se empleará un convertidor CC-CC conmutado. El modelo escogido es el MIC26903 de la empresa Microchip [32]. Este convertidor ha sido seleccionado debido a su alta eficiencia como se puede ver en la Figura 3.15 y a su corriente máxima de salida de 9A a 5V. En la Figura 3.16 se puede ver el esquemático del circuito completo. Siguiendo las indicaciones del datasheet se han añadido condensadores tanto a la entrada como a la salida de del convertidor para controlorar los picos de tensión.

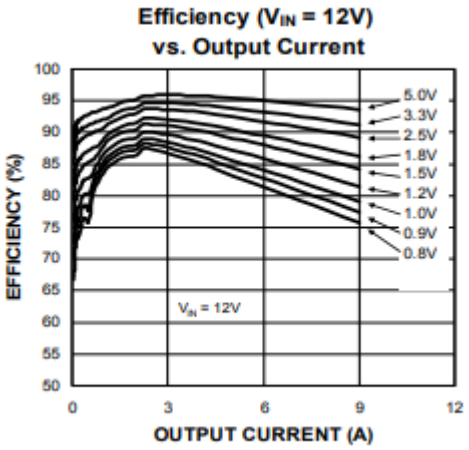


Figura 3.15: Eficiencia del convertidor CC-CC para diferentes valores de corriente [32]

Para controlar el rizado a la salida se ha añadido una inductancia **L1** cuyo valor se obtiene según la siguiente ecuación:

$$L = \frac{V_{\text{out}} \times (V_{\text{in(max)}} - V_{\text{out}})}{V_{\text{in(max)}} \times f_{\text{sw}} \times 0{,}2 \times I_{\text{out(max)}}} \tag{3.1}$$

V_{out}	5V
$V_{\text{in(max)}}$	28.8V
f_{sw}	600kHz
$I_{\text{out(max)}}$	9A

Tabla 3.2: Cálculo del valor de la inductancia **L1**

Con los parámetros de la Tabla 3.2 se obtiene un valor de 38,2μH.

Por ultimo es necesario ajustar los valores de *R15* y *R16* para que el convertidor pueda medir la tensión a la salida y de ese modo cerrar el lazo de realimentación. Siguiendo el datasheet, la ecuación es la siguiente:

$$V_{out} = 0{,}8 \times \left(1 + \frac{R_{15}}{R_{16}}\right) \tag{3.2}$$

Para conseguir 5V a la salida los valores obtenidos de estas resistencias son:

$$R15 = 6k8\Omega$$

$$R16 = 1k3\Omega$$

Para conseguir la tensión de 3.3V se introducirá un regulador lineal de 5V-3.3V. Para esta tarea se ha escogido el modelo LD39200 de STM[33]. En la Figura 3.17 se muestra el esquema del circuito completo. Los condensadores **C9** y **C10** se han añadido para eliminar cualquier posible pico de tensión proveniente del regulador CC-CC conmutado de 24V-5V.

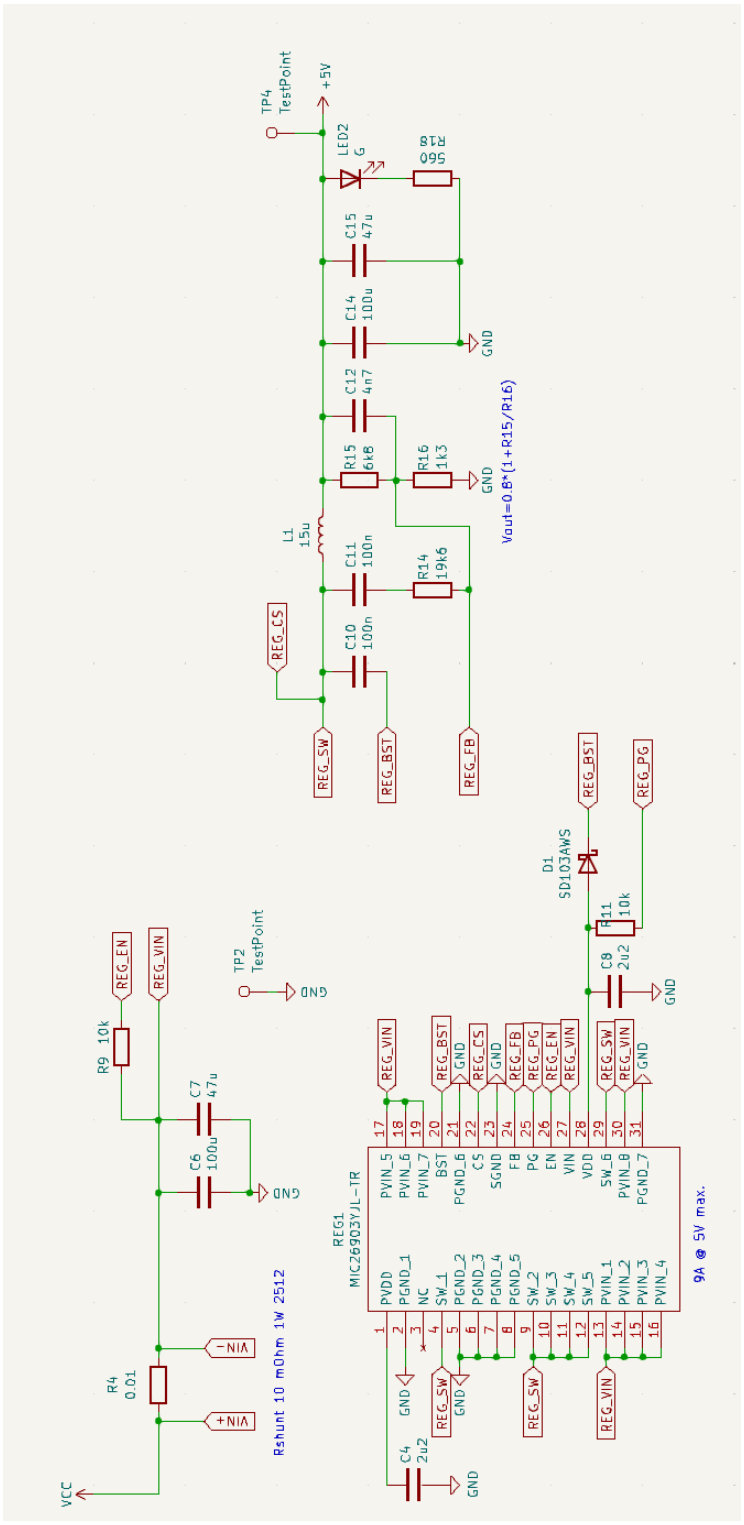


Figura 3.16: Esquema convertidor CC-CC conmutado ($Vin = 24\text{V}$ y $Vout = 5\text{V}$)

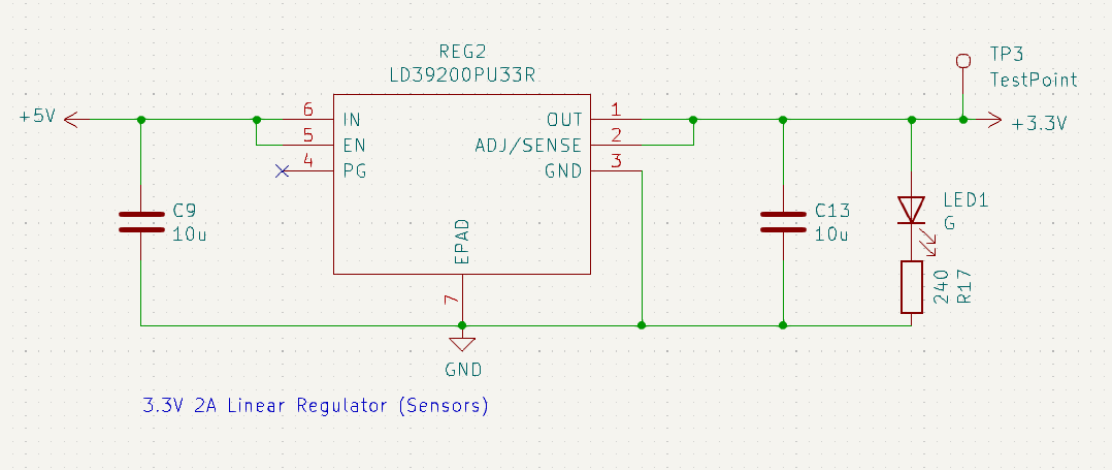


Figura 3.17: Esquema regulador lineal ($Vin = 5V$ y $Vout = 3.3V$)

3.6. Diseño PCB

Tras presentar y justificar los componentes elegidos para la PCB de la unidad de telemetría (TCU), en este apartado se mostrará el diseño final. Debido a que aunque desempeñen funciones diferentes en el vehículo, comparten casi el mismo hardware, este diseño será empeado tambien para la unidad de control del vehículo (VCU), con alguna modificación.

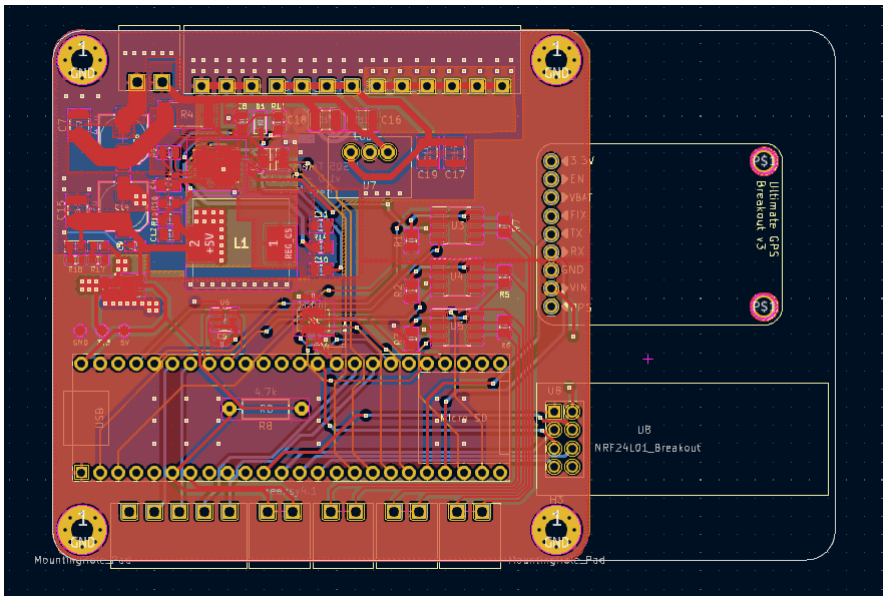


Figura 3.18: Diseño PCB unidad de telemetría

La Figura 3.18 muestra la PCB de la unidad de telemetría. La PCB se ha realizado con cuatro capas con unas dimensiones de 65mm por 101mm.

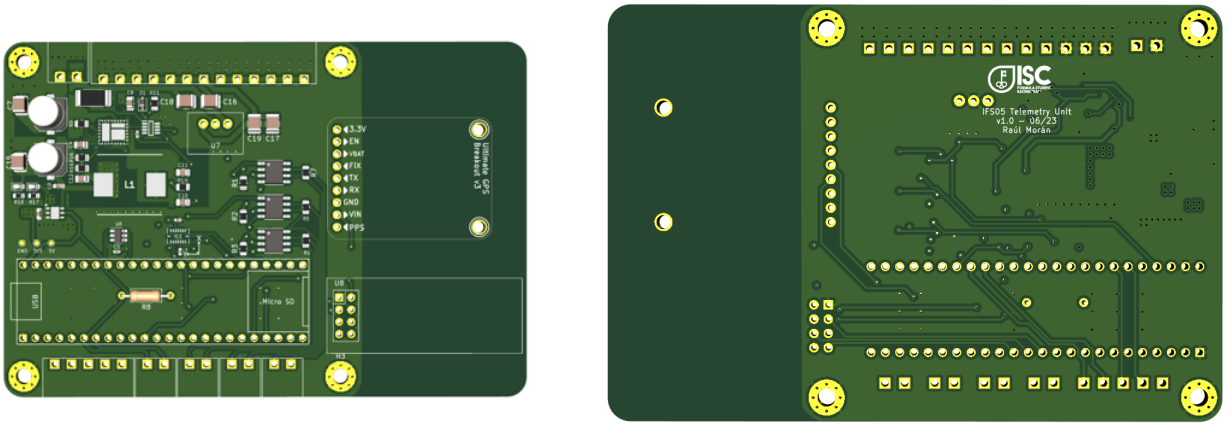


Figura 3.19: Vista superior e inferior en tres dimensiones

La placa se ha diseñado con los conectores en los extremos para facilitar el montaje y conexionado dentro del coche. Los principales elementos a destacar de esta PCB son los siguientes:

- **Circuito de alimentación**. Situado en la zona superior izquierda de la placa, se trata del circuito diseñado en la sección 3.5.

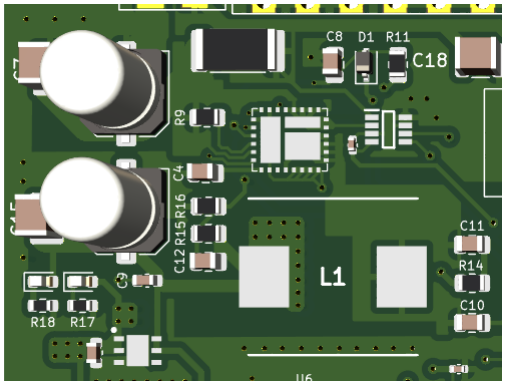


Figura 3.20: Circuito de alimentación

- **Transceptores CAN**. Situados en la parte central de la PCB tienen su salida *CAN High* y *CAN Low* para cada bus (ver sección 3.2.2) en los tres pares de conectores ubicados en la parte inferior derecha de la PCB.

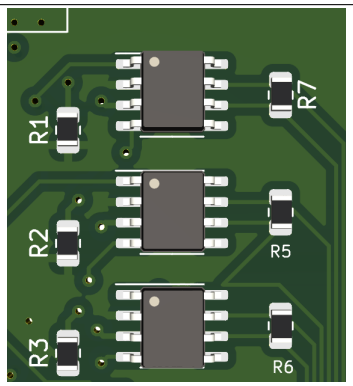


Figura 3.21: Transceptores CAN

- **Teensy 4.1**, **GPS** y **nRF24L01**. Se han añadido los zócalos para montar la tarjeta de desarrollo *Teensy 4.1* (3.3), el módulo GPS (3.4.5) y el módulo de comunicación inalámbrica nRF24L01 (3.1.4).

El resto de conexiones restantes en la PCB son entradas para los diferentes sensores que están conectados a la unidad de telemetría, como se describe en la sección 3.4.

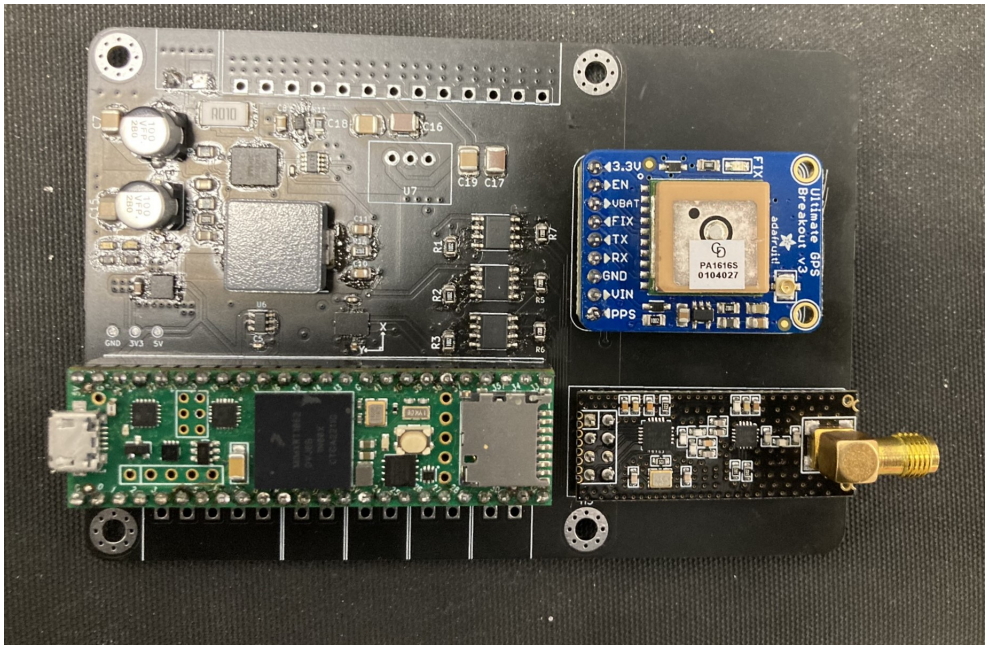


Figura 3.22: PCB unidad de telemetría previo a su montaje en el vehículo

3.7. Soldadura de los componentes en la PCB

Una vez recibida la PCB y los componentes se procedió a soldarla, pudiendo ver el resultado final en la Figura 3.22. Debido a que la gran mayoria de los componentes excepto los zócalos para el *Teensy 4.1*, el módulo de GPS y el módulo de comunicación inalámbrica nRF24L01PALNA son SMD, se ha empleado la soldadura por refusión (*reflow* en inglés) para soldarlos. Esta técnica consiste en la aplicación de una pasta de soldadura sobre la huella de los componentes de la placa y calentarla mendiante un horno o plancha para fundirla y así conseguir uniones soldadas entre los componentes y la PCB. Para facilitar la soldadura de componentes con huellas no estándar, como ha sido el caso del convertidor CC-CC conmutado (MIC26903) situado encima de la bobina L1 en la Figura 3.20, se ha hecho uso de un stencil o plantilla de la PCB sobre la que aplicar la pasta de estaño y así asegurar que no se producen cortocircuitos.

Como se ha mecionado al principio de la sección, debido a la gran similitud de hardware entre la TCU y la VCU se ha aprovechado el diseño de la primera para que con algunas modificaciones se adapte a los necesidades de esta PCB. La Figura 3.23 muestra la PCB de la unidad de control del vehículo (VCU) después de su fabricación. La principal diferencia con la placa de la unidad de telemetría está en la eliminación de las conexiones de los módulos de GPS y nRF24L01.

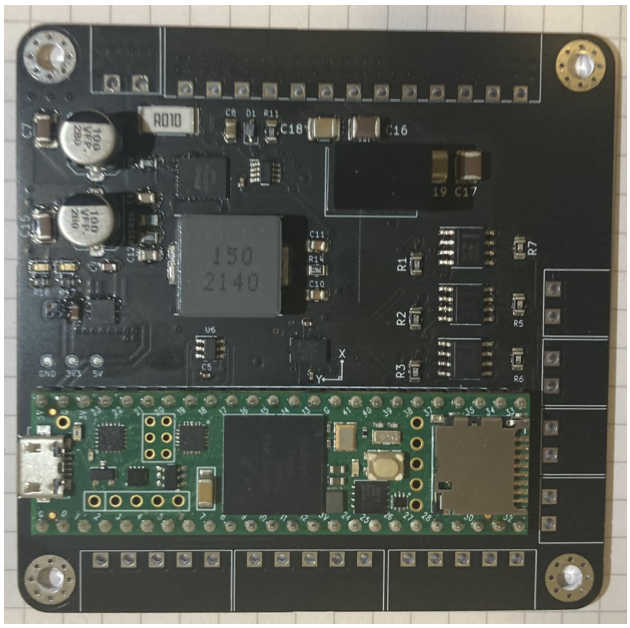


Figura 3.23: PCB unidad de control previo su montaje en el vehículo

3.8. Montaje en el IFS-05

3.8.1. Arquitectura de comunicaciones

Tras el diseño y la fabricación de las PCBs de la unidad de telemetría y la unidad de control se procede a su montaje en el IFS-05. Este paso es crucial para consolidar todo el sistema y permitir su funcionamiento en un entorno real. Debido a la disposición de los sensores mencionada en 2.1 la arquitectura de comunicaciones será la misma que en el IFS-04 con la VCU en la parte delantera del prototipo y la TCU en la parte trasera.

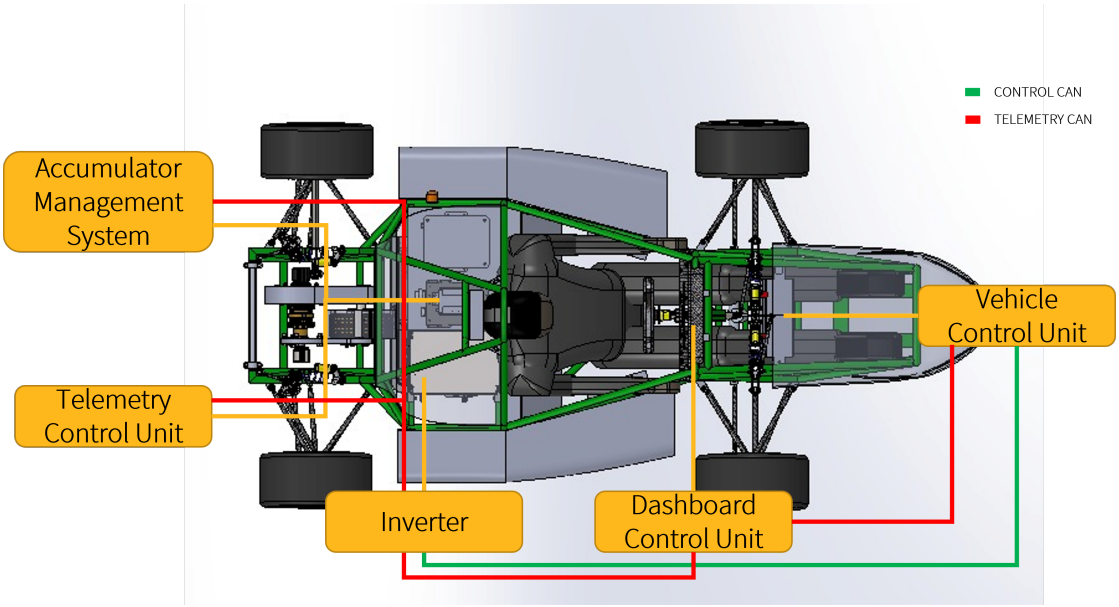


Figura 3.24: Arquitectura de comunicaciones del IFS-05

3.8.2. Elección de la antena y posición en el vehículo

Siguiendo la ecuación 2.1 usar una antena de alta ganancia aumentaría la potencia de la señal recibida y por lo tanto mejoraría la calidad del enlace entre el transmisor y el receptor. Esta opción sin embargo se ha descartado debido a que el ancho del haz (*beam width* en inglés) es muy reducido [9]. Como la posición relativa del vehículo y por tanto del emisor no siempre va a ser la misma con respecto al receptor, la antena debe ser omnidireccional polarizada en el eje horizontal. La Figura 3.25 muestra el modelo de antena elegido.



Figura 3.25: Antena omnidireccional 8dBi

A la hora de elegir la posición de la antena en el prototipo, se ha tenido en cuenta todo lo expuesto en la seción 2.3. Dado que el chasis y demás elementos mecánicos del vehículo están fabricados en acero, pueden reflejar la señal del emisor y aumentar las pérdidas descritas en la ecuación 2.1. Por ello se ha dedidido colocar la antena en la parte delantera del vehículo, como se ve en la Figura 3.26, debido a que es la zona más libre de obstrucciones.



Figura 3.26: Fotografía de la antena montada en el IFS-05

Dado que esta temporada se acortó la distancia entre ejes en el nuevo chasis, con el IFS-05 el piloto adoptó una postura más ergida, situándose a 55^{o} con la horizontal, cuando en el IFS-04 se encontraba a 30^{o}. Esta mejora de la ergonomía del piloto ha permitido situar la antena en esa posición y como se puede ver en la Figura 3.27 no supone una molestia a la hora de pilotar el vehículo.



Figura 3.27: Fotografia de la posición del piloto respecto a la antena

Capítulo 4

Diseño del software del sistema de telemetría

En este capítulo se describirá el software que se ejecutará en la unidad de telemetría. Este código se encargará de coordinar el envío a través del módulo de comunicación inalámbrica nRF24L01 (descrito en la sección 3.1.4), de los diferentes datos obtenidos tanto de los sensores directamente conectados a la TCU como los recibidos del resto de unidades de control a través del bus CAN de telemetría (CAN TEL).

Debido a la compatibilidad de la placa de desarrollo *Teensy 4.1* con el entorno de Arduino y por tanto con sus librerías, se ha optado por elegir esta plataforma. Para programar se empleará la plataforma PlatformIO instalada como extensión en Visual Studio Code.

4.1. Estructura del código del sistema emisor

A la hora de programar el código de este proyecto se ha seguido la estructura definida en la Figura 4.1:

- **AMS.h y AMS.cpp**. Se encarga de procesar los mensajes recibidos a través del bus *CAN* de telemetría con las variables del acumulador (Tensión de celda mínima, temperatura de celda máxima y corriente del bus de continua).

- **MotorInv.h y MotorInv.cpp**. Procesa los datos procedentes del inversor (temperatura del motor y temperatura del inversor) reenviados por la VCU a través del bus *CAN* de telemetría.

- **Pedals.h y Pedals.cpp**. Recibe las variables de porcentaje del pedal del acelerador y porcentaje del pedal de freno, enviadas por la VCU a través del bus *CAN* de telemetría.
- **SuspensionF.h y SuspensionF.cpp**. Recibe las lecturas de los potenciometros lineales montados en los amortiguadores delanteros (ver 3.4.1) enviadas por la VCU a través del bus *CAN* de telemetría.
- **IMUR.h e IMUR.cpp**. Obtiene los valores de aceleración lineal en los ejes X, Y, Z y velocidades angulares de la unidad inercial o IMU integrada en la PCB de la unidad de telemetría (ver 3.4.4).
- **Refri.h y Refri.cpp**. Obtienen los valores de temperatura de los cuatro sensores digitales montados a la entrada y salida del radiador de cada circuito de refrigeración (ver 3.4.6) conectados directamente a la TCU.
- **SuspensionR.h y SuspensionR.cpp**. Recibe las lecturas de los potenciómetros lineales montados en los amortiguadores traseros (ver 3.4.1), conectados directamente a la unidad de telemetría.



Figura 4.1: Estructura del código de la unidad de telemetría

- **main.cpp**. Es el archivo principal del programa. En él se inicializan todas las variables, se configuran todos los sensores y se inicializa el módulo de comunicación inalámbrica nRF24L01 para enviar los datos al receptor. A continuación se comentarán las principales secciones de este archivo:

  A la hora de establecer la comunicación *CAN* entre las distintas unidades de control, se emplea la librería *FlexCANT4* desarrollada por PJRC para sus placas de desarrollo *Teensy* [34].

```
1
2 // ---------- VARIABLES DEL CAN ----------
3
4 FlexCAN_T4<CAN2, RX_SIZE_256, TX_SIZE_16> CAN_TEL;        //
    CAN Telemetria
5
6 CAN_message_t msg_rx;
7 CAN_message_t msg_tx;
8
```

Código 4.1: Inicialización variables CAN

  Esta librería permite inicializar directamente cada uno los buses *CAN* integrados en el microcontrolador. Para este proyecto, como la TCU solo está conectada al bus CAN TEL, solo es necesario incializar uno. Tras inicializar el bus de telemetría, se crea una estrucura para el mensaje de CAN. Esta estructura permite agrupar los distintos términos que componen el mensaje CAN (ver Figura 2.3), como son el ID del mensaje, el tamaño o el buffer del que leer o escribir.

```
1 // ----------- VARIABLES DE TELEMETRIA ---------
2
3 RF24 radio(PIN_RF24_CE, PIN_RF24_CSN);
4 byte rf24_tx[6] = "1SNSR";
5
6 typedef struct payloadData
7 {
8   float buffer[8] = {0};
9
10 } payload;
11
12 payload telemetry_payload;
13
```

Código 4.2: Inicialización variables telemetría

Para controlar el módulo de comunicación inalámbrica nRF24L01 se hará uso de la librería RF24 [35]. Se creará un objeto para el módulo instalado en la PCB, indicando los pines de CS y CE y se definirá una dirección para comunicarse con el receptor. Por último se creará una variable para almacenar los mensajes que se van a enviar.

```
// ---------- TIMERS ----------

Metro timer_send_payload_MI = Metro(100);
    // Enviar telemetria motor_inversor
Metro timer_send_payload_AMS = Metro(100);
    // Enviar telemetria acumulador
Metro timer_send_payload_IMU = Metro(200);
    // Enviar telemetria IMU
Metro timer_send_payload_pedals = Metro(200);
    // Enviar telemetria pedal acelerador y freno
Metro timer_send_payload_GPS = Metro(200);
    // Enviar telemetria GPS
Metro timer_send_payload_suspension = Metro(100);
    // Enviar telemetria sensores de suspension
Metro timer_send_payload_coolant_temp = Metro(2000);
    // Enviar telemetria temperatura refri
Metro timer_gps = Metro(100);
    // Leer datos GPS
Metro timer_suspension = Metro(50);
    // Leer sensores de suspension
Metro timer_coolant_temperature = Metro(1000);
    // Leer sensores temperatura refri

```

Código 4.3: Inicialización timers

En esta sección se inicializan los contadores para enviar las diferentes variables según su importancia. Las variables que más frecuencia de envío requieren serán las eléctricas provenientes tanto del motor y el inversor como las del acumulador, seguidas de las lecturas de los sensores de la suspensión. Para inicializar los timers se hará uso de la librería Metro [36].

```
// ------ VARIABLES SENSORES ------

Bmi088Accel accel(SPI1, ACC_CS);
Bmi088Gyro gyro(SPI1, GYRO_CS);

TMP100Sensor temperatures(0x4E);

MOTOR_temperature motor_temperature;
```

```
IGBT_temperature igbt_temperature;
INVERTER_temperature inverter_temperature;
N_actual n_actual;
DC_BUS_VOLTAGE dc_bus_voltage;
I_ACTUAL i_actual;
REFRI refri;

CURRENT_SENSOR current_sensor;
CELL_MIN_V cell_min_v;
CELL_MAX_TEMP cell_max_temp;

THROTTLE_PEDAL throttle_pedal;
BRAKE_PEDAL brake_pedal;

```

Código 4.4: Definición variables sensores

Por último antes de inicializar el programa se definirán los objetos para cada sensor usando las clases presentadas anteriormente. Para la IMU se hará uso de la librería BMI088[37].

```
void setup()
{
  // Initialize serial connection.
  Serial.begin(115200);
  delay(100);

  // Inicializarcion de los buses CAN
  Serial.println("Initializing CAN...");
  CAN_TEL.begin();
  CAN_ACCU.begin();
  CAN_TEL.setBaudRate(CAN_TEL_KBPS);
  CAN_ACCU.setBaudRate(CAN_ACCU_KBPS);

  // Configure the NRF24 tranceiver.
  Serial.println("Configure NRF24 ...");
  nrf24_setup();
  // Show debug information for NRF24 tranceiver.
  radio.printDetails();

  // ---------- SECUENCIA DE ARRANQUE ----------
  // Config de sensores

  // TMP100 temperaturas
  temperatures.setup();
  // DS18B20 temperaturas refri
  refri_sensors.begin();
  // BMI088 IMU
  IMU_setup();
```

```
29    // GPS
30    GPS.begin(9600);
31    GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA);
       // specify data to be received (minimum + fix)
32    GPS.sendCommand(PMTK_SET_NMEA_UPDATE_10HZ);
       // set update rate (10Hz)
33    GPS.sendCommand(PGCMD_ANTENNA);
       // report data about antenna
34    delay(DELAY_CONFIG);
35  }
36
```

Código 4.5: Inicialización del programa

En este apartado se inicializarán el bus *CAN* de telemetría, el módulo nRF24L01 y todos los sensores definidos previamente. Para el GPS se usará la librería de AdafruitGPS [38].

```
1  void loop()
2  {
3    // Read and log CAN messages
4    CANparse();
5    // Read sensors
6    IMUread();
7    // Read GPS
8    GPSread();
9    // Coolant temperatures read
10   refriread();
11   // Suspension sensors read
12   suspread();
13   // Send telemetry
14   nrf24send();
15 }
16
```

Código 4.6: Bucle de ejecución del programa

Por último este bucle se ejecutará indefinidamente. En él se realizarán las tareas de leer los datos de los sensores conectados directamente a la PCB de la unidad de telemetría, se procesarán los mensajes recibidos por el bus *CAN* de telemetría y se enviaran a través del módulo de comunicación inalámbrica nRF24L01 según la prioridad establecida con los timers.

- Función **CANparse**. Será la encargada de clasificar los mensajes que la TCU recibe a través del bus *CAN* de telemtría procedentes de la VCU y el acumulador. Esto lo hará en función de los IDs de cada mensaje recogidos en la Tabla

```
void CANparse()
{

  while (CAN_TEL.read(msg_rx))
  {

    // Identify CAN messages and process them

    raw_data = (msg_rx.buf[0] << 8) + msg_rx.buf[1];

    if (msg_rx.id == ID_t_motor)
    {
      data = (raw_data - 9452.4) / 52.33;
      motor_temperature.load(data);
    }

    else if (msg_rx.id == ID_t_igbt)
    {
      data = (raw_data - 17621.0) / 84.043;
      igbt_temperature.load(data);
    }
    else if (msg_rx.id == ID_t_air)
    {
      data = (raw_data - 10367.0) / 48.169;
      inverter_temperature.load(data);
    }
    else if (msg_rx.id == ID_n_actual)
    {
      data = (raw_data * 6000) / 32767.0;
      n_actual.load(data);
    }
    else if (msg_rx.id == ID_dc_bus_voltage)
    {
      data = raw_data;
      dc_bus_voltage.load(data);
    }
    else if (msg_rx.id == ID_i_actual)
    {
      data = (raw_data * 400) / 1070;
      i_actual.load(data);
    }
    else
      switch (msg_rx.id)
      {
      case ID_torque_total:
        data = raw_data;
        throttle_pedal.load(data);
        break;
      case ID_s_freno:
```

```
50          data = raw_data;
51          brake_pedal.load(data);
52          break;
53
54        case ID_CORRIENTE_ACUM:
55          data = raw_data;
56          current_sensor.load(data);
57          break;
58        case ID_V_CELDA_MIN:
59          data = raw_data;
60          cell_min_v.load(data);
61          break;
62        case ID_T_CELDA_MAX:
63          data = raw_data;
64          cell_max_temp.load(data);
65          break;
66        }
67    }
68 }
69
```

Código 4.7: Función CANparse

Será necesario procesar los datos recibidos en el mensaje antes de guardarlos en el objeto creado al inicio del programa. Este es el caso de los datos provenientes del motor y el inversor cuya conversión se hara siguiendo el datasheet del inversor [39].

- Función **nrf24send**. Agrupará los datos recibidos según su origen y los enviará precedidos de un identificador, siguiendo la frecuencia establecida con los timers declarados previamente.

```
1  void nrf24_send()
2  {
3
4    if (timer_send_payload_MI.check())
5    {
6      telemetry_payload.buffer[0] = ID_MOTOR_INVERSOR;
7      telemetry_payload.buffer[1] = motor_temperature.write
       ();
8      telemetry_payload.buffer[2] = igbt_temperature.write
       ();
9      telemetry_payload.buffer[3] = inverter_temperature.
       write();
10     telemetry_payload.buffer[4] = n_actual.write();
11     telemetry_payload.buffer[5] = dc_bus_voltage.write();
12     telemetry_payload.buffer[6] = i_actual.write();
13     // Send the data ...
14     nrf24_write(telemetry_payload);
15   }
```

```
16
17  if (timer_send_payload_AMS.check())
18  {
19    telemetry_payload.buffer[0] = ID_ACUMULADOR;
20    telemetry_payload.buffer[1] = current_sensor.write();
21    telemetry_payload.buffer[2] = cell_min_v.write();
22    telemetry_payload.buffer[3] = cell_max_temp.write();
23    // Send the data ...
24    nrf24_write(telemetry_payload);
25  }
26
27  if (timer_send_payload_IMU.check())
28  {
29    telemetry_payload.buffer[0] = ID_IMU_REAR;
30    telemetry_payload.buffer[1] = accel.getAccelX_mss();
31    telemetry_payload.buffer[2] = accel.getAccelY_mss();
32    telemetry_payload.buffer[3] = accel.getAccelZ_mss();
33    telemetry_payload.buffer[4] = gyro.getGyroX_rads();
34    telemetry_payload.buffer[5] = gyro.getGyroY_rads();
35    telemetry_payload.buffer[6] = gyro.getGyroZ_rads();
36    // Send the data ...
37    nrf24_write(telemetry_payload);
38  }
39
40  if (timer_send_payload_pedals.check())
41  {
42    telemetry_payload.buffer[0] = ID_PEDALS;
43    telemetry_payload.buffer[1] = throttle_pedal.write();
44    telemetry_payload.buffer[2] = brake_pedal.write();
45    // Send the data ...
46    nrf24_write(telemetry_payload);
47  }
48
49  if (timer_send_payload_GPS.check())
50  {
51    telemetry_payload.buffer[0] = ID_GPS;
52    telemetry_payload.buffer[1] = GPS.speed;
53    telemetry_payload.buffer[2] = GPS.latitude_fixed;
54    telemetry_payload.buffer[3] = GPS.longitude_fixed;
55    telemetry_payload.buffer[4] = GPS.altitude;
56    // Send the data ...
57    nrf24_write(telemetry_payload);
58  }
59
60  if (timer_send_payload_coolant_temperature.check())
61  {
62    telemetry_payload.buffer[0] = ID_SENSORES_REFRI;
63    telemetry_payload.buffer[1] = refri.write_refri(1);
64    telemetry_payload.buffer[2] = refri.write_refri(2);
```

```
65      telemetry_payload.buffer[3] = refri.write_refri(3);
66      telemetry_payload.buffer[4] = refri.write_refri(4);
67      // Send the data ...
68      nrf24_write(telemetry_payload);
69    }
70  }
71
```

Código 4.8: Función nrf24send

- Función **nrf24write**. La función nrf24send llama a esta función cada vez que llena el buffer de envío.

```
1  void nrf24_write(payload txpayload)
2  {
3
4    // Stop listening on the radio (we can't both listen
       and send).
5    radio.stopListening();
6
7    Serial.print("Bytes packed: ");
8    Serial.println(sizeof(txpayload));
9
10   if (radio.write(&txpayload, sizeof(txpayload)))
11   {
12     Serial.print("Payload sent successfully. Retries=");
13     Serial.println(radio.getARC());
14     for (int i = 0; i < 8; i++)
15     {
16       Serial.println(telemetry_payload.buffer[i]);
17     }
18   }
19   else
20   {
21     Serial.print("Failed to send payload. Retries=");
22     Serial.println(radio.getARC());
23   }
24
25   // Start listening again.
26   radio.startListening();
27 }
28
```

Código 4.9: Función nrf24write

4.2. Recepción de los datos

El equipo receptor también contará con un módulo de comunicación inalámbrica nRF24L01PALNA. Se encargará de clasificar los mensajes recibidos en función del ID que se encuentra en la primera posición del vector que compone el buffer. Una vez clasificados los mostrará en tiempo real y los irá escribiendo en un fichero .csv que posteriormente será tratado con un script desarrollado en Matlab que permitirá obtener las gráficas de cada variable.

Capítulo 5

Análisis de los resultados

Después de finalizar la fase de diseño de hardware, software y el proceso de montaje en el vehículo, llega el momento de someter al sistema de telemetría desarrollado a pruebas en un escenario real: las competiciones. Debido a diversos problemas surgidos en la electrónica del acumulador, IFS-05 no pudo participar en las pruebas dinámicas de FS Italy (12-16 de julio) y por lo tanto no se pudo poner a prueba el sistema.

No obstante, en FS Spain (6-12 de agosto), tras superar todas las inspecciones técnicas requeridas, el equipo logró un hito significativo, pues se logró competir y finalizar todas las pruebas dinámicas. Se pudieron recoger datos de la última prueba de la competición: la *Endurance*, en la que durante los 22km se monitorizó en tiempo real el comportamiento del vehículo, permitiendo tomar decisiones que garantizaron la finalización del evento.

Una de estas decisiones fue en relación al ritmo del piloto. Dado que era la primera vez que se participaba en este evento y debido la naturaleza de esta prueba de resistencia, se planteó una estrategia conservadora. Al ser una prueba en la que habría cambio de piloto, durante la primera mitad, el primer piloto mantuvo un ritmo menor que el segundo, como se puede ver en la Figura 5.1 que muestra el porcentaje del pedal del acelerador que se traduce en porcentaje de par solicitado al inversor. Cotejando este dato con el valor de tensión del bus de continua (Figura 5.3), y el valor de corriente a la salida del acumulador (Figura 5.2), se pudo comprobar que estos estaban dentro del rango normal de tensiones y corrientes y por tanto se podía aumentar el ritmo con el segundo piloto, como refleja la parte derecha de la Figura 5.1. Se puede comprobar como al aumentar el par solicitado al inversor, aumenta la caída de tensión en el bus de continua y también aumenta la corriente a la salida del acumulador. Las temperaturas del motor y del inversor (ver Figura 5.4), estuvieron dentro de sus valores de operación.

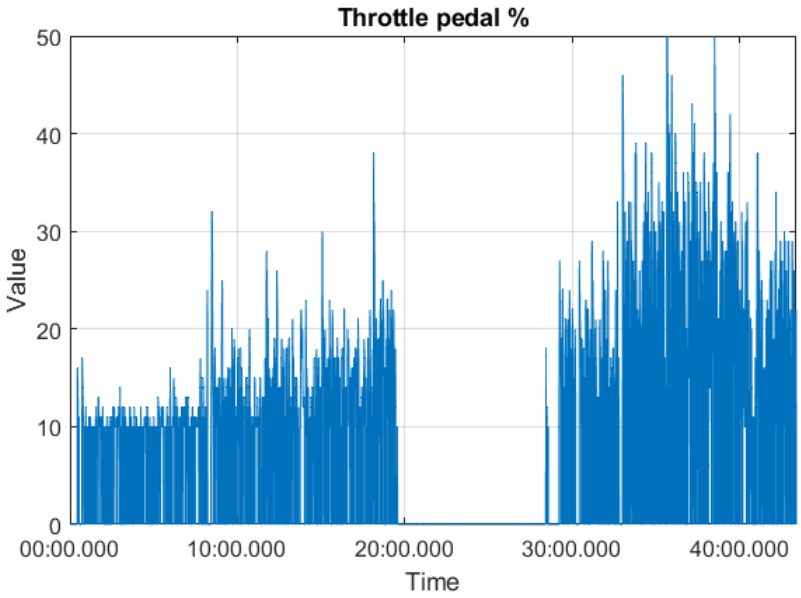


Figura 5.1: Porcentaje del pedal del acelerador - FSS Endurance

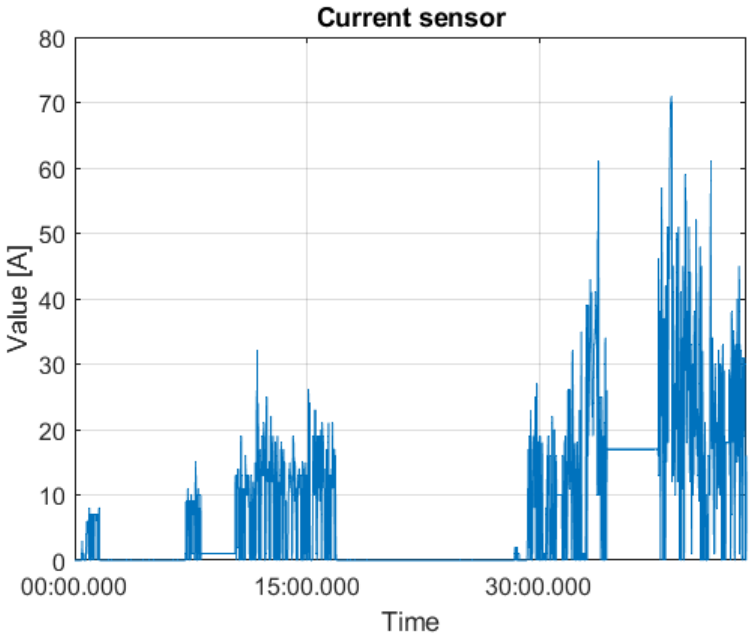


Figura 5.2: Corriente a la salida del acumulador - FSS Endurance

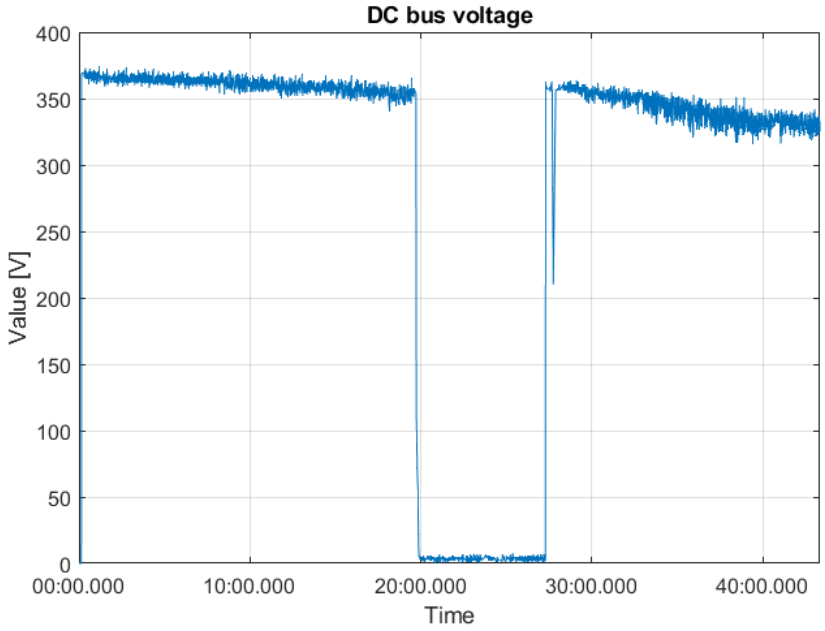


Figura 5.3: Tensión del bus de continua - FSS Endurance

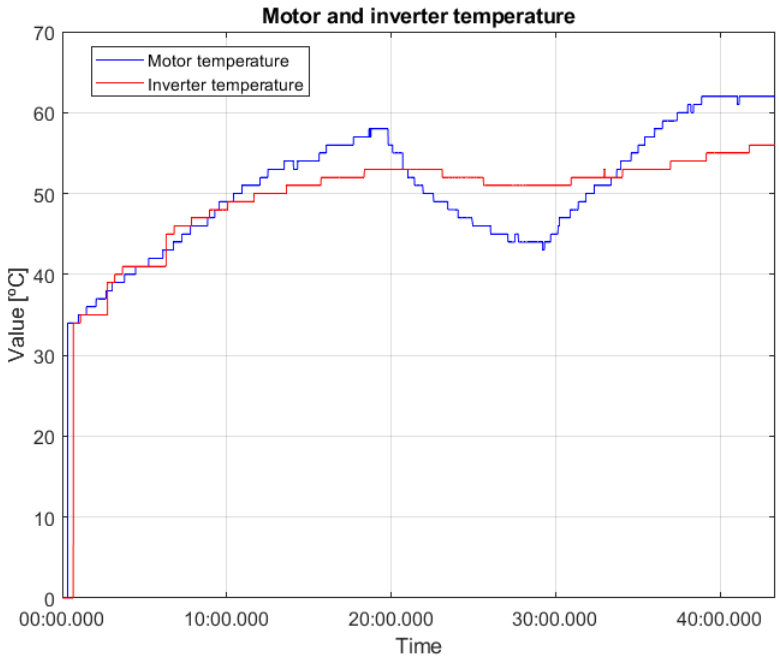


Figura 5.4: Temperatura motor e inversor - FSS Endurance

Finalmente, al contrastar los datos recopilados a través de este sistema de telemetría con los datos obtenidos mediante el sistema IFS-04 de la temporada anterior, se constata un aumento en la recepción de paquetes en la transmisión de datos. La Figura 5.5 muestra los registros de revoluciones del motor capturados por el sistema de telemetría del IFS-04, mientras que la Figura 5.6 ilustra los registros obtenidos mediante el sistema de telemetría del IFS-05 durante el mismo intervalo (3 minutos y 45 segundos). Resulta evidente que la cantidad de datos recibidos, representados por el asterisco en el sistema de la temporada anterior, es significativamente inferior en comparación con la cantidad de datos recibidos durante esta temporada mediante el nuevo sistema.

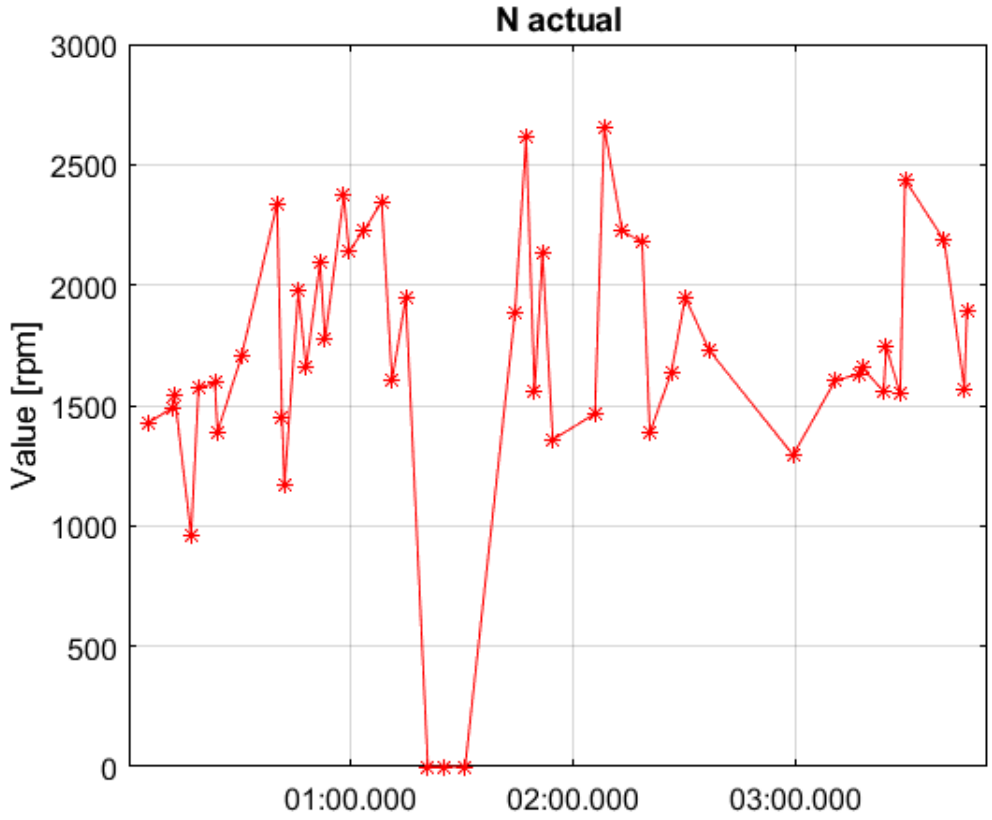


Figura 5.5: Revoluciones del motor - Telemetria IFS-04

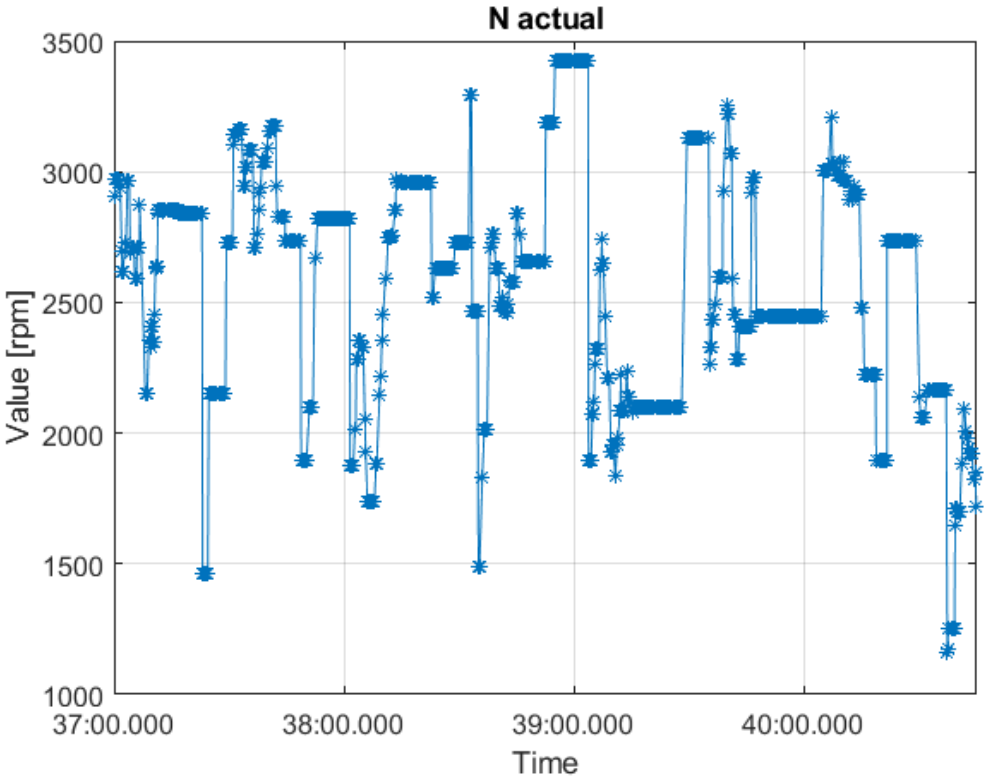


Figura 5.6: Revoluciones del motor - Telemetría IFS-05

Capítulo 6

Conclusiones

Una vez analizados los resultados en el capítulo 5, se puede concluir que el sistema diseñado cumple efectivamente con los requisitos establecidos inicialmente en la introducción de este proyecto.

Uno de los objetivos principales era diseñar un sistema de telemetría que permitiera la visualización de los datos del vehículo en tiempo real durante las pruebas dinámicas. Gracias al protocolo nRF24 seleccionado y al hardware y software implementados, se ha conseguido monitorizar de forma continua variables críticas como revoluciones del motor, temperaturas, suspensiones, etc. Esto ha posibilitado la toma de decisiones en tiempo real para garantizar el correcto funcionamiento y seguridad del vehículo.

Otro de los requisitos era incrementar la tasa de transmisión de datos con respecto al sistema previo basado en LoRa. Tal y como muestra la comparativa en el Capítulo 5, el nuevo sistema no presenta pérdidas de paquetes y la tasa de envío de datos es muy superior. pudiendo disponer de más información sobre el estado dinámico del vehículo. De cara a versiones futuras de este sistema, se incluirá la recogida y registro en un sistema de almacenamiento local, para disponer de una copia de todos los datos exenta de los problemas de las comunicaciones inalámbricas.

Por último, también se buscaba empaquetar los datos de forma sincronizada para facilitar su análisis posterior. El software desarrollado integra esta funcionalidad, permitiendo una correlación sencilla de las variables medidas y abriendo la puerta al estudio avanzado del comportamiento del vehículo.

En conclusión, el sistema de telemetría diseñado satisface los requisitos planteados inicialmente, cumpliendo los objetivos de monitorización en tiempo real, incremento de la tasa de datos y empaquetamiento sincronizado de la informa-

ción. Esto se traduce en una mejora sustancial del rendimiento y la seguridad del vehículo.

Anexo A

Diseño de PCBs

En este apartado se muestran los esquemas tanto del circuito completo como del diseño de la PCB de la unidad de telemetría (TCU) y de la unidad de control (VCU) del IFS-05.

En cuanto al circuito completo de la unidad de telemetría, se pueden destacar los siguientes elementos:

- El circuito de alimentación, que incluye el convertidor CC-CC conmutado para obtener 5V a partir de los 24V de la batería, y el regulador lineal para obtener 3.3V a partir de los 5V. Este circuito alimenta todos los componentes de la placa y los sensores externos.
- Los transceptores CAN para permitir la comunicación con el bus CAN de telemetría y recibir datos de otras unidades de control.
- El microcontrolador Teensy 4.1 que gestiona todas las tareas de lectura, procesado y envío de los datos.
- El módulo nRF24L01 para la comunicación inalámbrica con el receptor.
- Los sensores integrados como la IMU.
- Las entradas para conectar el resto de sensores externos.

En cuanto al diseño de la PCB, se trata de una placa de 4 capas con los conectores situados en los extremos para facilitar el montaje. El circuito y PCB de la unidad de control (VCU) sigue una estructura muy similar, con la diferencia de no incluir el módulo de comunicación inalámbrica ni el GPS.

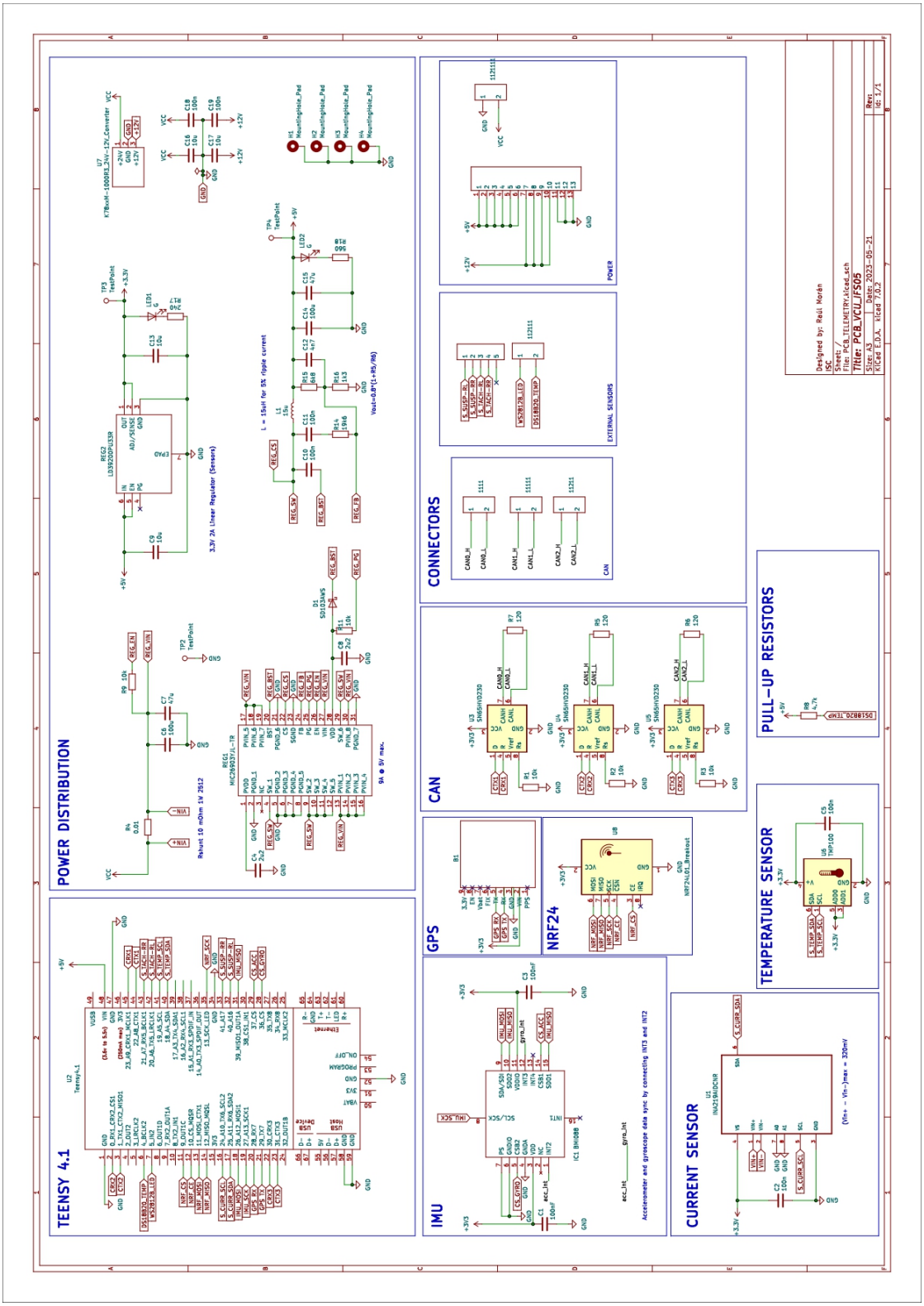


Figura A.1: Esquema circuito completo de la unidad de telemetría

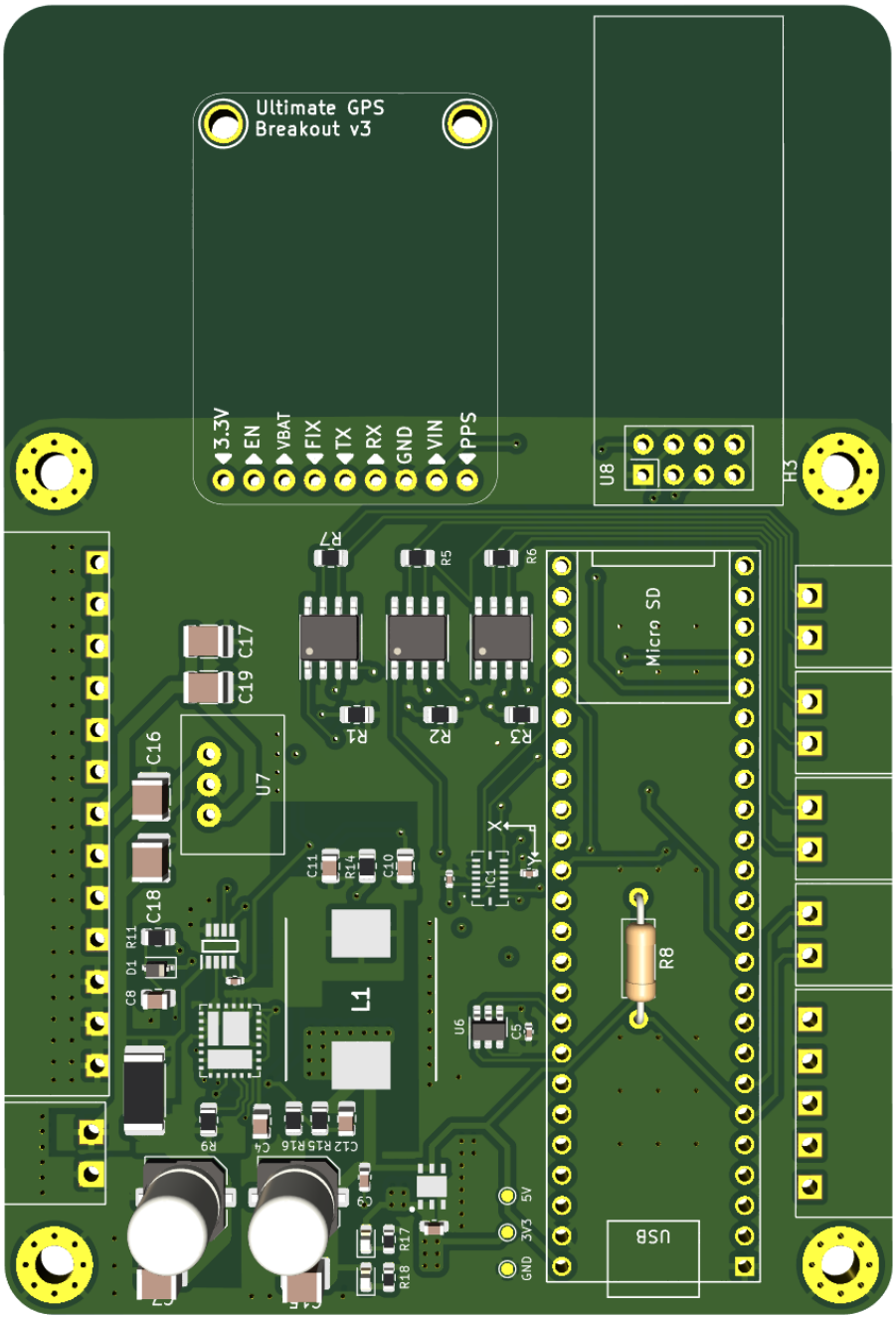


Figura A.2: Esquema general de la PCB de la unidad de telemetría

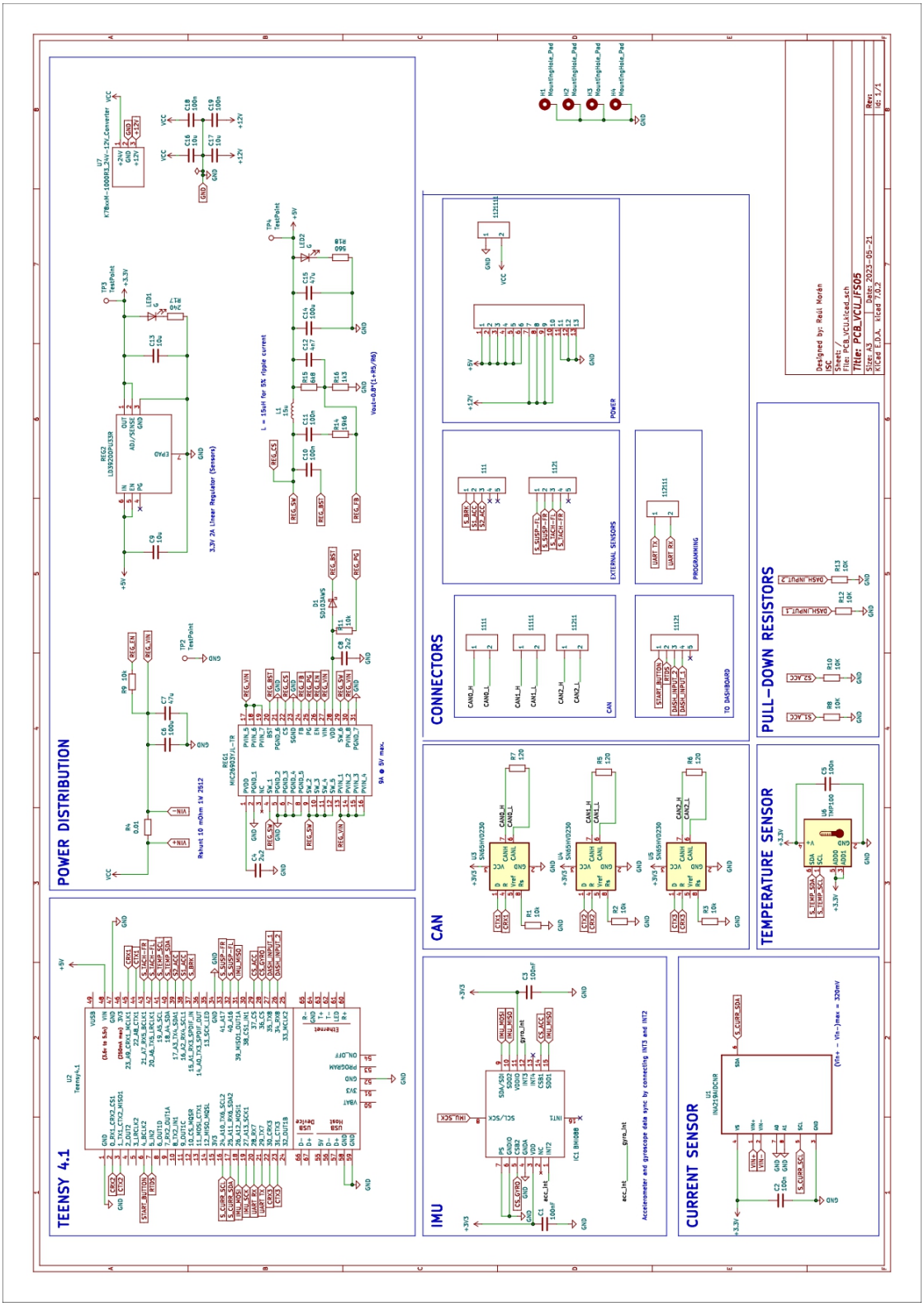


Figura A.3: Esquema circuito completo de la unidad de control

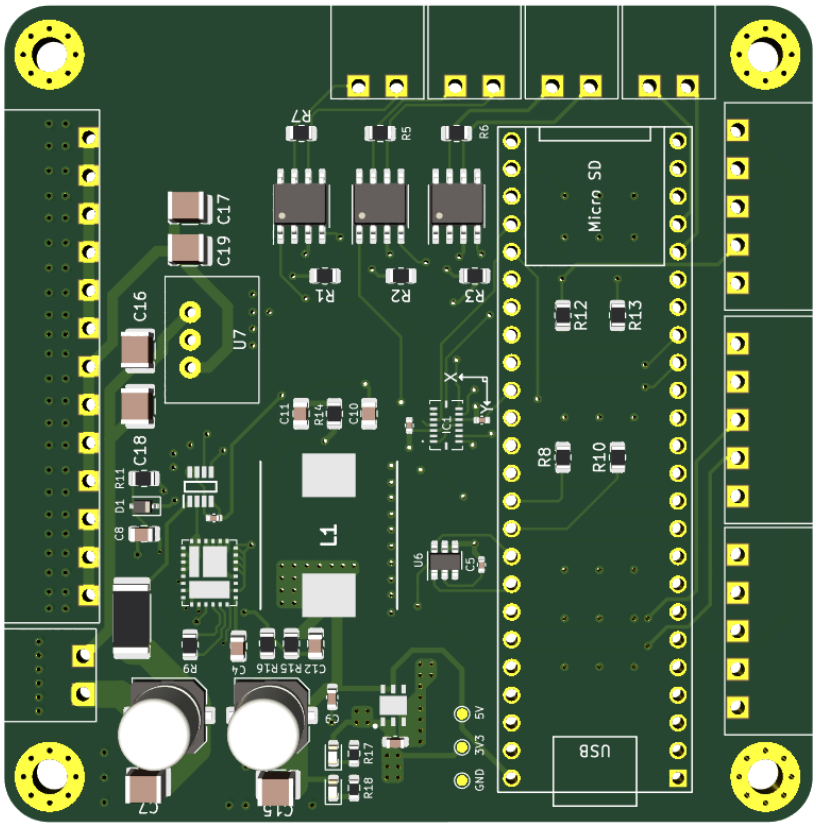


Figura A.4: Esquema general de la PCB de la unidad de control

Anexo B

Análisis económico

Se realizará un breve análisis económico del proyecto y para ello se tendrá en cuenta el gastos de fabricación de las PCBs diseñadas, así como los componentes, mano de obra y herrammientas necesarias.

B.1. Fabricación PCBs

Como el resultado de este proyecto ha sido el diseño y fabricación dos PCBs, la de la unidad de telemetría (TCU) y la unidad de control del vehículo (VCU), aunque en un pricipio solo se planteó el diseño de la primera, se incluirán ambas para el cálculo de los costes de fabricación y de los componentes.

Después del diseño en el software kiCAD, se enviaban los archivos a un fabricante ya que debido a la complejidad de los circuitos, no era posible fabricarlas con los equipos disponibles. A la hora de elegir el fabricante se barajaron tres alternativas como muestra la Tabla B.1. Para esta comparación se han empleado los precios y tipos de cambio de junio de 2023, cuando se enviaron a fabricar. En los tres fabricantes candidatos se elegirá una cantidad de 5 PCBs para contar con repuestos en caso de algún fallo durante el proceso de soldadura o durante la vida útil dentro del vehículo. Se acabó escogiendo JLCPCB debido a ser la opción más económica, suponiendo un total de 49,10€.

B.2. Gastos de personal

Para calcular los gastos de personal se han contabilizado un total de 550 horas de trabajo en la realización del proyecto repartidas entre horas de diseño y horas

Característica	JLCPCB	PCBWay	Eurocircuits
Precio por cm^2	0,0015€	0,005€	0,015€
Cantidad mínima	5 PCBs	1 PCB	1 PCB
Capas máximas	6	12	16
Precio TCU	34,60€	70,93€	160,60€
Precio VCU	14,50€	25,97€	120,45€
Precio total	49,10€	96,90€	281,05€

Tabla B.1: Comparación entre JLCPCB, PCBWay y Eurocircuits

de fabricación. Asumiendo un coste de 25€/ hora según [40] se obtiene un total de 13.750€.

B.3. Gastos de equipamiento y material

El coste de los componentes fue aproximadamente 600€, incluyendo tanto los componentes SMD como la placa de desarrollo *Teensy 4.1* (una para la TCU y otra para la VCU, más repuestos), varios módulos nRF24L01PALNA y varios módulos GPS.

Para la fabricación de las PCBs se ha empleado una plancha de dimensiones 200x200 con un coste total de 150€. También se ha empleado pasta de soldadura y estaño para soldar los componentes SMD y through-hole. La Tabla B.2 recoge los gastos de equipamiento, teniendo en cuenta una depreciación máxima anual del 10 % para este tipo de activos.

Equipamiento	Tiempo usado	Importe	Importe total
Plancha de soldadura	2 meses	150€	2,50€
Soldador de estaño	2 meses	40€	0,60€
Osciloscopio	1 mes	5.000€	41,60€
Multímetro	1 mes	60€	0.50€
		TOTAL:	45,20€

Tabla B.2: Gastos de equipamiento

B.4. Gastos totales

Una vez calculados todos los gastos, el coste total de este proyecto asciende a catorce mil cuatrocientos cuarenta y cuatro euros con treinta céntimos (14.444,30€).

Tipo de gasto	**Importe**
Gastos de fabricación	49,10€
Gastos de personal	13.750€
Gastos de material	600.00€
Gastos de equipamiento	45,20€
TOTAL:	14.444,30€

Tabla B.3: Gastos totales

Bibliografía

[1] Formula Student Germany. *Rules 2023 v1.1*. 2023. URL: https://www.formulastudent.de/fileadmin/user_upload/all/2023/rules/FS-Rules_2023_v1.1.pdf (visitado 04-07-2023).

[2] J. R. Knisley Werchan y H. A. "Telemetry: enhancing tactical network management". En: *Proceedings of TCC'94 - Tactical Communications Conference* (1994).

[3] *LoRa WAN What is it? - A technical overview of LoRa®and LoRaWAN*. Inf. téc. LoRa Alliance, 2015.

[4] Robert Bosch GmbH. *CAN Specification 2.0*. September 1991. URL: http://esd.cs.ucr.edu/webres/can20.pdf (visitado 30-08-2023).

[5] CSS Electronics. *CAN Bus Explained - A Simple Intro*. 2021. URL: https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial (visitado 28-08-2023).

[6] Jhihoon Joo, Dong Seog Han y Hong-Jong Jeong. "First Fresnel zone analysis in vehicle-to-vehicle communications". En: *2015 International Conference on Connected Vehicles and Expo (ICCVE)*. 2015, págs. 196-197. DOI: 10.1109/ICCVE.2015.18.

[7] Jcmcclurg [CC-BY-SA-3.0] via Wikimedia Commons. *FresnelSVG1*. URL: https://commons.wikimedia.org/wiki/File:FresnelSVG1.svg (visitado 30-08-2023).

[8] Jerry C. Whitaker. *The Electronics Handbook*. CRC Press, 1996, pág. 1321. ISBN: 9780849383458.

[9] Richard Johnson. *Antenna Engineering Handbook*. 2nd. New York, NY: McGraw-Hill, Inc., 1984, págs. 1-12. ISBN: 0-07-032291-0.

[10] Harvey Lehpamer. *Microwave Transmission Networks: Planning, Design and Deployment, 2nd Edition*. New York: McGraw Hill, 2010.

[11] *Cuadro Nacional de Atribución de Frecuencias*. 2023. URL: https://www.boe.es/eli/es/o/2023/06/12/etd625 (visitado 28-06-2023).

[12] *International Telecomunication Union*. 2023. URL: https://www.itu.int/en/Pages/default.aspx (visitado 28-07-2023).

[13] *Radio Regulations 2020, CHAPTER II – Frequencies, ARTICLE 5 Frequency allocations, Section IV – Table of Frequency Allocations*. International Telecommunication Union (ITU). 2020. URL: https://www.itu.int/en/ITU-R/terrestrial/RR/Pages/text/Table-of-Frequency-Allocations.aspx (visitado 28-08-2023).

[14] Maximilian Dörrbecker [CC-BY-SA-2.5] via Wikimedia Commons. *International Telecommunication Union regions with dividing lines.svg*. URL: https://commons.wikimedia.org/wiki/File:International_Telecommunication_Union_regions_with_dividing_lines.svg (visitado 30-08-2023).

[15] *IEEE 802.15 WPAN Task Group 4 (TG4), «ZigBee»*. 2003. URL: https://www.ieee802.org/15/pub/TG4.html (visitado 25-07-2023).

[16] *LoRa Alliance*. 2021. URL: https://lora-alliance.org/ (visitado 21-02-2023).

[17] *Nordic Semiconductor, «nRF24L01. Single Chip 2.4GHz Transceiver. Preliminary Production Specification v1.0*. 2006. URL: https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf (visitado 20-07-2023).

[18] *nRF24L01 Single Chip 2.4GHz Transceiver Product Specification*. 2007. URL: https://cdn.sparkfun.com/datasheets/Wireless/Nordic/nRF24L01_Product_Specification_v2_0.pdf (visitado 24-06-2023).

[19] Last Minute Engineers. *NRF24L01 Arduino Wireless Communication Tutorial*. 2018. URL: https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/ (visitado 12-04-2023).

[20] Personal Communications Programme University of Hull. *Digital Modulation and GMSK: From the Report Entitled 'Electromagnetic Compatibility Aspects of Radio-Based Mobile Telecommunication Systems'*. 2018. URL: http://www.emc.york.ac.uk/reports/linkpcp/pcp_emc.pdf (visitado 01-07-2023).

[21] https://www.flu-project.com/2019/09/teleco-in-nutshell-v84-modulacion-por.html.

[22] *Texas Instruments,«SN65HVD23x 3.3-V CAN Bus Transceivers»*. 2018. URL: https://www.ti.com/lit/ds/symlink/sn65hvd230.pdf (visitado 01-06-2023).

[23] *Teensy 4.1 (Headers)*. 2020. URL: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/DEV-16996_Web.pdf (visitado 24-05-2023).

[24] *i.MX RT1060 Crossover Processors for Consumer Products*. 2019. URL: https://www.pjrc.com/teensy/IMXRT1060CEC_rev0_1.pdf (visitado 24-05-2023).

[25] *Variohm EuroSensor, «Motorsports Linear Position Sensor - ELPM series»*. 2019. URL: https://www.cyan-rs.com/relays/download/88/197/168/2258/?file=/files/libs/2258//201907291502444266.pdf (visitado 20-06-2023).

[26] *Variohm EuroSensor,«Pressure Sensor - EPT3100»*. 2019. URL: https://www.efihardware.com/download/148/Variohm-3100-Datasheet (visitado 20-06-2023).

[27] *LEM,«AUTOMOTIVE CURRENT TRANSDUCER OPEN LOOP TECHNOLOGY»*. 2019. URL: https://www.lem.com/sites/default/files/products_datasheets/dhab_s_118.pdf (visitado 20-06-2023).

[28] *Bosch,«BMI088: 6-axis Motion Tracking for High Performance Applications»*. 2022. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi088-ds001.pdf (visitado 18-06-2023).

[29] *CD Technology,«CD-PA1616S GPS patch antenna module»*. 2022. URL: https://cdn-shop.adafruit.com/product-files/746/CD+PA1616S+Datasheet.v03.pdf (visitado 01-07-2023).

[30] Maxim Integrated. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. 2019. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20.pdf (visitado 23-07-2023).

[31] Maxim Integrated. *Overview of 1-Wire Technology and Its Use*. 2019. URL: https://www.analog.com/media/en/technical-documentation/tech-articles/guide-to-1wire-communication--maxim-integrated.pdf (visitado 23-07-2023).

[32] *Microchip,«MIC26903 - 28V, 9A Hyper Light Load Synchronous DC/DC Buck Regulator»*. 2011. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/MIC26903.pdf (visitado 01-06-2023).

[33] *Microchip,«2 A high PSRR ultra low drop linear regulator with reverse current protection»*. 2018. URL: https://www.st.com/resource/en/datasheet/ld39200.pdf (visitado 01-06-2023).

[34] Collin Kidder. $FlexCAN_T4 : FlexCAN(CAN2{,}0/CANFD) Library for Teensy 3.x and 4{,}0$. 2019. URL: https://github.com/tonton81/FlexCAN_T4 (visitado 27-02-2023).

[35] TMRh20. *RF24: OSI Layer 2 driver for nRF24L01 on Arduino Raspberry Pi/Linux Devices*. 2021. URL: https://github.com/nRF24/RF24 (visitado 27-02-2023).

[36] Thomas Fredericks. *Metro Arduino Wiring Library Repository*. 2018. URL: https://github.com/thomasfredericks/Metro-Arduino-Wiring (visitado 29-05-2023).

[37] Bolder Flight Systems. *bmi088-arduino Repository*. 2021. URL: https://github.com/bolderflight/bmi088-arduino (visitado 01-07-2023).

[38] Adafruit. *Adafruit GPS Library Repository*. 2023. URL: https://github.com/adafruit/Adafruit_GPS (visitado 30-07-2023).

[39] *BAMOCAR-PG-D3-700-400_EN*. URL: https://www.unitek-industrie-elektronik.de/images/pdf/BAMOCAR%20Digital/BAMOCAR-PG-D3-700-400%5C_EN.pdf.

[40] Instituto Nacional de Estadística (INE). *Encuesta anual de estructura salarial. Serie 2013 - 2021*. Madrid: INE. 2021.