

Desarrollo de un receptor de telemetría aeronáutica basado en Radio Definida por Software

Autores:

Nivio Guillán, Lautaro
Willig, Mauricio

Director: Dr. Ing. Alejandro Uriz
Co-Director: Ing. Juan Alberto Etcheverry



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Desarrollo de un receptor de telemetría aeronáutica basado en Radio Definida por Software

Autores:

Nivio Guillán, Lautaro
Willig, Mauricio

Director: Dr. Ing. Alejandro Uriz
Co-Director: Ing. Juan Alberto Etcheverry



Universidad Nacional de Mar del Plata
Facultad de Ingeniería
Departamento de Ingeniería Electrónica
6 de diciembre de 2022

A mis padres, *Guillermo y Liliana*.
A mi hermano, *Agustin*
A la memoria de mis abuelos, *Nicolás, Norma, Boggi y Pirucha*
A mi familia y amigos
Mauricio

A mis padres, *Carlos e Irene*.
A mis hermanos, *Santiago y Carlitos*.
A mis hermanas *Ileana y Melina*.
Lautaro

Agradecimientos

A nuestro director Alejandro Uriz, por todo el apoyo, paciencia, predisposición y compromiso. Fue fundamental en el desarrollo de este proyecto final.

A nuestro Co-Director Juan Alberto Etcheverry, por su ayuda técnica y sus conocimientos.

A nuestro gran colaborador Jorge Luis Marquez, por sus conocimientos, experiencia y equipos.

A Fernando Zagnoni y Noelia Garcia, por su acompañamiento en la redacción y formato del presente informe.

A nuestras familias y amigos, por darnos el aliento necesario cuando se presentaban dificultades.

Índice general

A. Informe final	9
A.1. Ficha del documento	10
A.2. Introducción	11
A.3. Propósito	11
A.3.1. Alcance del proyecto	11
A.3.2. Personal involucrado	12
A.3.3. Definiciones, acrónimos y abreviaturas	12
A.3.4. Referencias	13
A.3.5. Descripción del dispositivo	13
A.3.6. Contexto del proyecto	13
A.4. Anteproyecto	14
A.4.1. Requerimientos	14
A.4.2. Cronograma del proyecto	14
A.5. Proyecto	15
A.5.1. Comienzos del proyecto	15
A.5.2. ADS-B	15
A.5.2.1. Primeros pasos en la detección de ADS-B	15
A.5.2.2. Antena <i>colineal</i> 1090 MHz	17
A.5.2.3. Antena <i>spider</i> 1090 MHz	19
A.5.2.4. Filtro 1090MHz	22
A.5.2.5. Implementación en GNU Radio Companion	26
A.5.3. ACARS	27
A.5.3.1. Primeros pasos en la detección de ACARS	27
A.5.3.2. Filtro 131.725 MHz	29
A.5.4. Unificación de los dos modos de funcionamiento	31
A.5.4.1. Primera solución	31
A.5.4.2. Segunda solución	32
A.5.4.3. Tercera solución	33
A.5.4.4. LNA	34
A.5.4.5. Alimentación del LNA	36
A.5.4.6. Combinador	39
A.5.5. Solución final	40
A.5.5.1. Montaje	40
A.5.5.2. Rendimiento	42
A.5.5.3. Modificaciones a futuro	43
A.6. Conclusiones	44
B. Plan de proyecto	49
C. Especificación de requerimientos	53
C.1. Ficha del documento	54
C.2. Introducción	55
C.2.1. Propósito	55
C.2.2. Alcance	55
C.2.3. Personal involucrado	55
C.2.4. Definiciones, acrónimos y abreviaturas	56
C.2.5. Referencias	56
C.2.6. Visión general del documento	56
C.3. Descripción general	57

C.3.1. Perspectiva del producto	57
C.3.2. Funcionalidad del producto	57
C.3.3. Características de los usuarios	57
C.3.4. Restricciones	58
C.3.5. Suposiciones y dependencias	58
C.4. Requisitos específicos	59
C.4.1. Requisitos de las interfaces	59
C.4.1.1. Interfaz de usuario	59
C.4.1.2. Interfaz de hardware	59
C.4.1.3. Interfaz de software	59
C.4.1.4. Interfaz de comunicación	59
C.4.2. Requerimientos funcionales	59
C.4.3. Requerimientos No Funcionales	60
C.4.4. Requerimientos de Rendimiento	60
C.4.5. Restricciones de diseño	61
C.4.5.1. Cumplimiento de Standards	61
C.4.6. Atributos del sistema	61
C.4.6.1. Fiabilidad	61
C.4.6.2. Disponibilidad	61
C.4.6.3. Seguridad	61
C.4.6.4. Mantenibilidad	61
C.4.7. Otros Requisitos	61
C.5. Apéndice	61
D. Especificación Funcional	68
D.1. Ficha del documento	64
D.2. Introducción	65
D.2.1. Propósito	65
D.2.2. Alcance del proyecto	65
D.2.3. Personal involucrado	65
D.2.4. Definiciones, acrónimos y abreviaturas	66
D.2.5. Referencias	66
D.3. Descripción del dispositivo	67
D.3.1. Antena	67
D.3.2. LNA	67
D.3.3. Filtro 1090MHz	67
D.3.4. Filtro 131MHz	67
D.3.5. Combinador	68
D.3.6. SDR	68
D.3.7. PC	68
D.4. Especificaciones funcionales	69
D.4.1. RF01: Acondicionamiento de la señal de entrada	69
D.4.2. RF02: Identificación, decodificación y análisis de paquetes ADS-B	69
D.4.3. RF03: Identificación, decodificación y análisis de paquetes ACARS	69
D.5. Requerimientos no funcionales	70
D.5.1. RNF01: Protecciones meteorológicas	70
D.5.2. RNF02 Protecciones eléctricas	70
D.6. Requerimientos de rendimiento	71
D.6.1. RR01: Relación señal a ruido	71
D.6.2. RR02: Rango de recepción	71
E. Especificación técnica	73
E.1. Ficha del documento	73
E.2. Introducción	74
E.2.1. Propósito	74
E.2.2. Alcance del proyecto	74
E.2.3. Personal involucrado	74
E.2.4. Definiciones, acrónimos y abreviaturas	75
E.2.5. Referencias	75
E.2.6. Visión general del documento	75
E.3. Descripción del dispositivo	76

E.4. Hardware	76
E.4.1. Antena Spider 1090 MHz	76
E.4.2. Antena ACARS 131,725 Mhz	77
E.4.3. INA	78
E.4.4. Filtro pasa banda 1090 MHz	80
E.4.5. Filtro Pasa Banda centrado en 131,725 MHz	82
E.4.6. Combinador	83
E.4.7. SDR	84
E.4.8. PC	86
E.4.9. Protecciones eléctricas	86
E.4.10. Caja estanca	88
E.4.11. Interconexión de componentes de RF	88
E.5. Software	88
E.5.1. ADS-B	88
E.5.1.1. Mensaje	89
E.5.1.2. Procesamiento en GNU Radio	89
E.5.1.2.1. ADS-B Framer	90
E.5.1.2.2. ADS-B Demodulator	91
E.5.1.2.3. ADS-B Decoder	91
E.5.1.2.4. ZMQ PUB Message Sink	96
E.5.2. ACARS	97
E.5.2.1. Frame	97
E.5.2.2. Procesamiento en GNU Radio	98
E.5.2.3. Decodificación	98
F. Manual de usuario	101
E.1. Ficha del documento	102
E.2. Propósito	103
E.3. Audiencia	103
E.4. Partes del equipo	103
E.5. Instalación	104
E.6. ADS-B	104
E.6.1. Instalación de librerías	104
E.7. Inicialización detección de ADS-B	104
E.7.1. Ejemplo de la visualización en la consola	108
E.7.2. Inicialización del Web Server	108
E.7.3. Ejemplo de la visualización del Web Server	110
E.8. ACARS	111
E.8.1. Instalación de librerías	111
E.8.2. Inicialización detección de ACARS	111
G. Plan de pruebas	115
G.1. Ficha del documento	116
G.2. Introducción	117
G.2.1. Alcance	117
G.2.2. Ambientes de prueba	117
G.2.3. Instrumental, herramientas y software	117
G.2.4. Política de trabajo	117
G.2.5. Comunicación	117
G.3. Plan	117

Capítulo A

Informe final

A.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
14 de Septiembre de 2021	1.0	Escritura introducción y anteproyecto	Nivio Guillán, Lautaro Willig, Mauricio
31 de Mayo de 2022	1.1	Escritura proyecto	Nivio Guillán, Lautaro Willig, Mauricio
23 de Junio de 2022	1.2	Desarrollo conclusión	Nivio Guillán, Lautaro Willig, Mauricio
17 de Septiembre de 2022	1.3	Correcciones	Nivio Guillán, Lautaro Willig, Mauricio
25 de Septiembre de 2022	1.4	Modificación estructura del informe	Nivio Guillán, Lautaro Willig, Mauricio
30 de octubre de 2022	1.5	Correcciones	Nivio Guillán, Lautaro Willig, Mauricio
22 de noviembre de 2022	1.6	Correcciones y formato	Nivio Guillán, Lautaro Willig, Mauricio

A.2. Introducción

¿Qué es un *receptor de telemetría*? Se define como un equipo tecnológico que permite la recepción de magnitudes físicas emitidas por un transmisor y el posterior envío de información hacia el operador del sistema. En esta investigación trabajaremos con telemetría *aeronáutica*, lo que significa que recibiremos información de aeronaves como su altitud, velocidad, posición entre otros datos.

La información que se envía de dichas aeronaves son recibidas en distintos protocolos de comunicación adoptados bajo normas en el mundo de la aviación. Por protocolos nos referimos al conjunto de reglas y normas que permiten que dos o más dispositivos se comuniquen entre ellos para transmitir y recibir información. Dos de los protocolos de comunicación mas utilizados en la aviación civil son ADS-B y ACARS.

ADS-B es una tecnología de vigilancia aérea diseñada para permitir a las aeronaves transmitir su estado de vuelo periódicamente sin la necesidad de ser interrogadas. ACARS por su parte también se define como una tecnología de vigilancia para las aerolíneas comerciales y a la vez se establece como sistema de comunicación. En ambos protocolos el enlace de datos de telemetría puede ser aire-tierra, para establecer intercambio con estaciones terrestres, como también aire-aire, con la finalidad de comunicarse con otras aeronaves.

Una de las alternativas para la recepción de telemetría aeronáutica es utilizar *Radios definidas por software* o SDR por sus siglas en inglés. Los SDR son dispositivos de radiocomunicación donde varios de los componentes típicamente implementados en hardware son realizados por software, utilizando una computadora o dispositivos embebidos. La ventaja de utilizar un SDR se ve claramente reflejada en su flexibilidad y facilidad de uso, en comparación con los receptores convencionales en los cuales es necesario ajustar sus componentes de forma física, como por ejemplo la sintonía de osciladores locales, el ajuste de filtros intermedios o demoduladores.

Existen herramientas para la programación de SDR como lo es el software *GNU Radio Companion*, un entorno de desarrollo visual de código libre para el procesamiento de señales utilizando librerías. Este ofrece bloques de procesamiento de señales tales como filtros, demoduladores, PLLs, entre otros. Su implementación esta basado en los lenguajes Python y C++ y permite realizar la programación de su configuración mediante diagrama de bloques, agilizando el desarrollo y la implementación.

A lo largo del documento *Informe final* se introducirá al lector en cuestiones y terminologías básicas para el entendimiento del proyecto y se lo guiará de forma cronológica en cómo se fueron presentando y resolviendo las implicancias durante la ejecución del proyecto.

A.3. Propósito

Este documento corresponde al informe final para el proyecto titulado *Receptor de telemetría aeronáutica basado en Radio Definida por Software*. Se ha estructurado en base a la información mencionada en los documentos: Especificación de Requerimientos (ER), Especificación Funcional (EF), Especificación Técnica (ET) y Plan de pruebas (PP).

El presente documento tiene como propósito resumir el proceso de diseño y fabricación de la solución, proveyendo información de cómo funcionará, cuáles serán sus comportamientos deseados y cómo se deberá construir. Se anexarán además las especificaciones referenciadas, que contarán con la información detallada de lo resumido en este informe.

Además del aspecto técnico y constructivo, este informe apunta a mostrar el desarrollo del Trabajo Final de Carrera de Ingeniería Electrónica desde el aspecto de gestión de proyecto. Cuenta con la asistencia de la cátedra *Trabajo Final. Ing Electrónica (4D0) / Ing Computación (4F0)* y apunta a introducir a los estudiantes a la gestión de proyecto y como se lleva a cabo un desarrollo de un producto.

El documento se encuentra dirigido a los desarrolladores y solicitantes del proyecto, así como también a la comisión asignada para su evaluación.

A.3.1. Alcance del proyecto

El proyecto abarca la definición de los requerimientos, implementación y la presentación un sistema receptor de datos de telemetría aeronáutica basado en Radio Definida por Software para una estación en tierra. Los requerimientos fueron planteados en conjunto con el Laboratorio de Comunicaciones (LAC) de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Es de este laboratorio que surge el proyecto, donde se realizó la mayor parte del trabajo, haciendo uso de los recursos disponibles, bajo la guía del director del proyecto, con la asesoría del co-director y la ayuda de colaboradores.

A.3.2. Personal involucrado

Nombre	Lautaro Nivio Guillán
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	lautaroniviog@gmail.com

Nombre	Mauricio Willig
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	mauriwilly@gmail.com

Nombre	Alejandro Uriz
Rol	Director
Categoría profesional	Profesor Adjunto Exclusiva/Investigador Adjunto CONICET
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	ajuriz@fi.mdp.edu.ar

Nombre	Juan Alberto Etcheverry
Rol	Codirector
Categoría profesional	Jefe de Trabajos Prácticos Exclusiva
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	jaetcheverry@fi.mdp.edu.ar

Nombre	Jorge Luis Marquez
Rol	Colaborador
Categoría profesional	Investigador RPIDFA Categoría C1
Responsabilidad	Orientar en el diseño y construcción del proyecto
Información del contacto	jmarquez@mdp.edu.ar

A.3.3. Definiciones, acrónimos y abreviaturas

Acrónimo	Significado
ADS-B	Automatic Dependent Surveillance Broadcast
ENACOM	Ente Nacional de Comunicaciones
HTTP	Hypertext Transfer Protocol
LAN	Amplificador Lineal de bajo ruido
RF	Radiofrecuencia
SDR	Software Defined Radio
TCP	Transmission Control Protocol
TRL	Technology Readiness Level
ACARS	Aircraft Communication Addressing and Reporting System

A.3.4. Referencias

Título del documento	Referencia
ABCs of SDR: Why your next radio will be software defined	Amer Radio Relay League
Baseband Analog Circuits for Software Defined Radio	Vito Giannini - Jan Craninckx - Andrea Baschirotto
Building a Software Defined Radio	DigiKey
Software defined radio for engineers	Travis Collins - Robin Getz - Alexander Wyglinski
Telemetry Standards	IRIG STANDAR 106-19
THE 1090 MEGAHERTZ RIDDLE	Junzi Sun
GNU Radio Wiki	GNU Radio

A.3.5. Descripción del dispositivo

El dispositivo a implementar será capaz de recibir información de telemetría enviada por aeronaves comerciales mediante el protocolo ADS-B en la frecuencia de 1090 MHz y el protocolo ACARS en la frecuencia de 131,725 MHz. En la Figura A.1 se puede ver un diagrama en bloques simplificado:

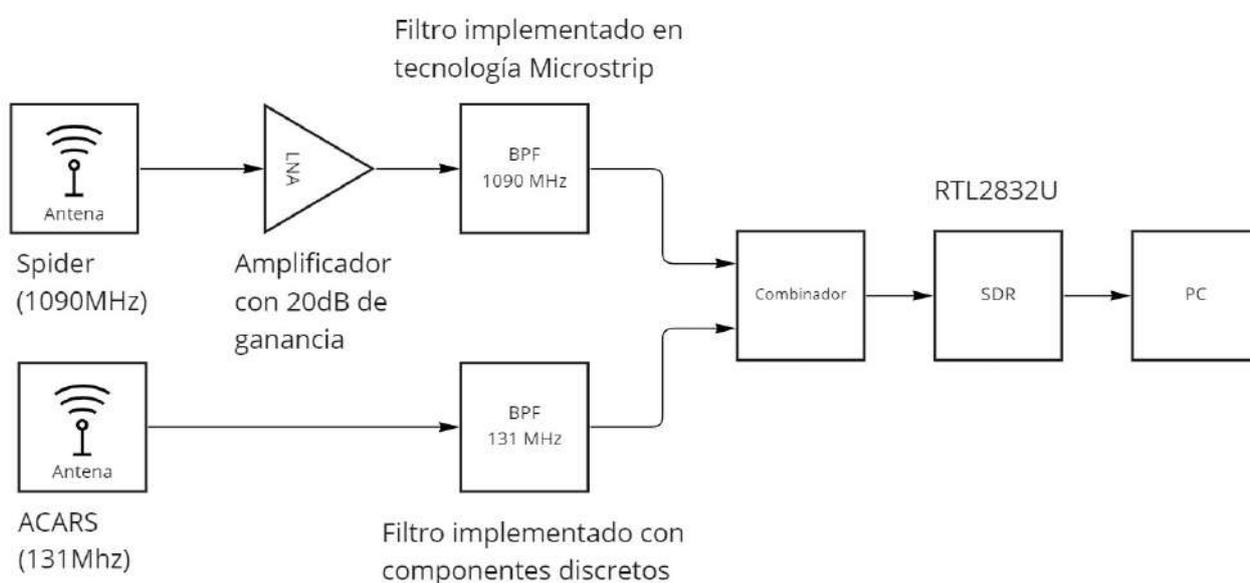


Figura A.1: Diagrama en bloques de la solución

Para su funcionamiento se diseñarán y adoptarán antenas, amplificadores y filtros característicos de implementaciones de radiofrecuencia. La adquisición de la señal se llevará a cabo mediante el uso de la tecnología de Radio Definido por Software, mientras que el procesamiento de la señal será implementado mediante software de código libre en una computadora. Esta última oficiará de interfaz gráfica: mostrará en pantalla la información pertinente a la telemetría enviada por las aeronaves.

El sistema permitirá la elección entre dos modos de operación: recepción de información mediante el protocolo ACARS y recepción de información mediante el protocolo ADS-B. El usuario es quien deberá elegir entre estos modos de funcionamiento.

A.3.6. Contexto del proyecto

El proyecto original surge de la necesidad del Laboratorio de Comunicaciones (LAC) de expandir los conocimientos en Radio definida por Software. Se plantea a los alumnos el desarrollo de un proyecto basado en SDR capaz de recibir telemetría aeronáutica. Se pretende que los alumnos se introduzcan en la tecnología y brinden la experiencia adquirida en futuros desarrollos académicos del laboratorio.

En un comienzo, se proponía el diseño de un sistema de telemetría aeronáutica capaz recibir información en el protocolo ADS-B. A partir de la iniciativa de los integrantes y con aprobación del director del proyecto, se decidió expandir el trabajo con la inclusión de otro protocolo: ACARS. De esta manera, el proyecto pasó a incluir dos tecnologías ampliamente usadas en la aviación civil.

A.4. Anteproyecto

Se propuso el diseño, desarrollo y ensayo de un dispositivo receptor de telemetría basado en Radio definida por Software capaz de recibir los protocolos ADS-B y ACARS. Se determinó la implementación mediante la interconexión de filtros, el uso de circuitos integrados y herramientas propias del laboratorio. Se optó por realizar el procesamiento de los datos mediante el uso de librerías de código libre posterior a la adquisición de la señal mediante un dispositivo SDR.

A continuación, se presentan secciones que detallan los requerimientos previos al desarrollo del proyecto y el cronograma del mismo.

A.4.1. Requerimientos

El dispositivo posee tres requerimientos funcionales: ser capaz de recibir la señal de RF enviada por distintas aeronaves (**RF01**); identificar, decodificar y analizar mensajes ADS-B (**RF02**) y mensajes ACARS (**RF03**) presentes en la señales adquiridas. Se incorporan dos requisitos no funcionales, deberá contar con protecciones meteorológicas para soportar inclemencias climáticas (**RNF01**) y protecciones eléctricas contra descargas electrostáticas (**RNF02**). De los tres primeros requerimientos funcionales, surgen dos requerimientos de rendimiento: uno que establece un mínimo valor de relación señal a ruido en 15 dB (**RR01**) y un rango de recepción de por lo menos 100 km (**RR02**) en la recepción de la señal de ambos protocolos.

A.4.2. Cronograma del proyecto

En el *Seminario de Trabajo Final* se brindaron las herramientas para ejecutar una planificación de tareas con el objetivo de realizar un diagrama de *Gantt* que abarque todo el proyecto. En el diagrama que se expone a continuación se observa la división del proyecto en cuatro etapas: 'Investigación', 'Diseño de filtros', 'Implementación en GNU Radio' y 'Diseño de PCBs'. A cada una se le estimó las tareas que implicarían junto su duración. Además, se incluyeron tres hitos, dos corresponden a revisiones y uno a la finalización del trabajo final.

El proyecto se inició en el mes de Abril de 2021. Se pueden observar dos períodos de ausencia de tareas, el primero comprendido entre los meses de Julio y Agosto de 2021 debido a que los estudiantes se vieron afectados por llamados a exámenes totalizadores. El segundo de ellos, en el mes de Enero de 2022, es producto del Receso académico de verano.

Finalmente se evidencia una excesiva duración en la escritura del informe final posterior a la finalización de la tarea de 'Corrección de fallos'. Este tiempo coincide con la inserción al mercado laboral de ambos estudiantes, lo cual incluyó oportunamente re-localización geográfica de los mismo, dificultando la dedicación en forma regular al avance del proyecto.

En la Figura [A.2](#) se plasma el diagrama de Gantt:

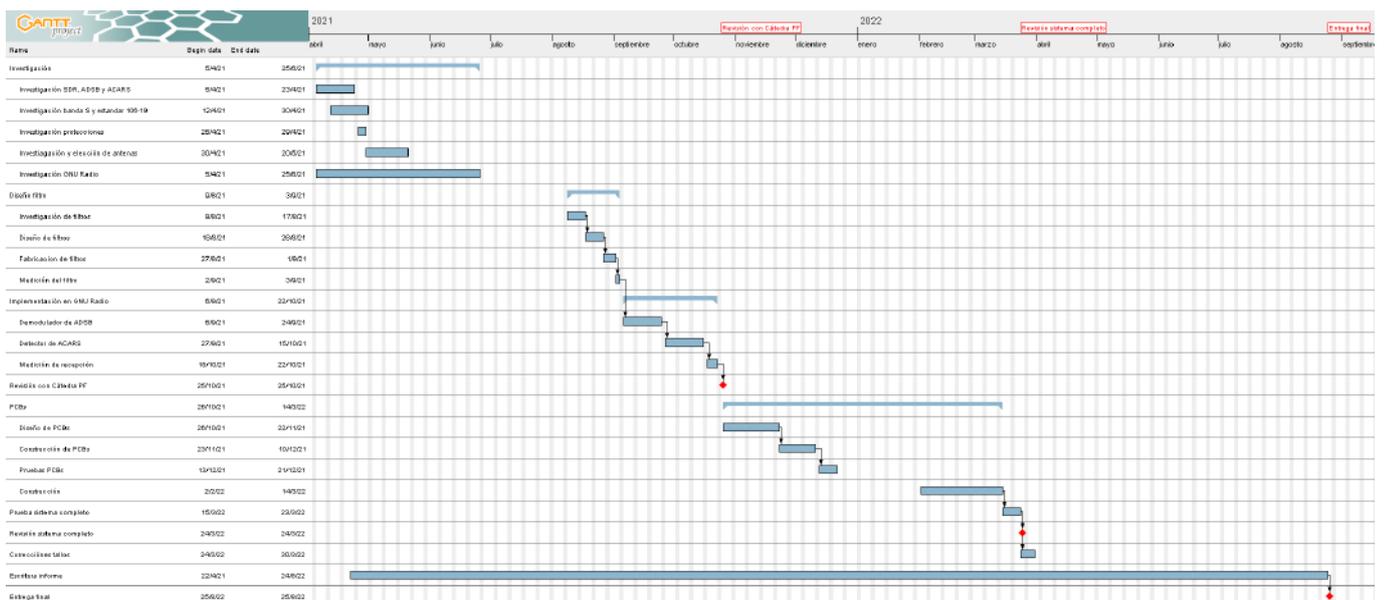


Figura A.2: Diagrama de Gantt del proyecto final

A.5. Proyecto

A.5.1. Comienzos del proyecto

Se comenzó el trabajo investigando y leyendo sobre la tecnología SDR y aplicaciones típicas. Se incursionó en el software habitual con el que se trabaja y se optó por el software GNU Radio Companion: un conjunto de herramientas de desarrollo de software gratuito y de código abierto que proporciona bloques de procesamiento de señales para implementar radios por software. Se puede usar con hardware de RF externo fácilmente disponible para crear radios definidas por software, o sin hardware en un ambiente de simulación. El hardware externo cumple la función de complementar el SDR, mejorando su rendimiento, como por ejemplo el filtrado y amplificación de señales, previamente a su entrada. Es ampliamente utilizado en entornos de investigación, industria, académicos y radio-aficionados para apoyar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real.

Con el objetivo de introducirse con el entorno de programación y el hardware disponible, se utilizó un RTL-SDR, logrando implementar un receptor de radio FM mediante software, como práctica inicial para comprender el principio de funcionamiento del GNU Radio Companion y el SDR. El diagrama en bloques generado se puede ver representado en la Figura [A.3](#).

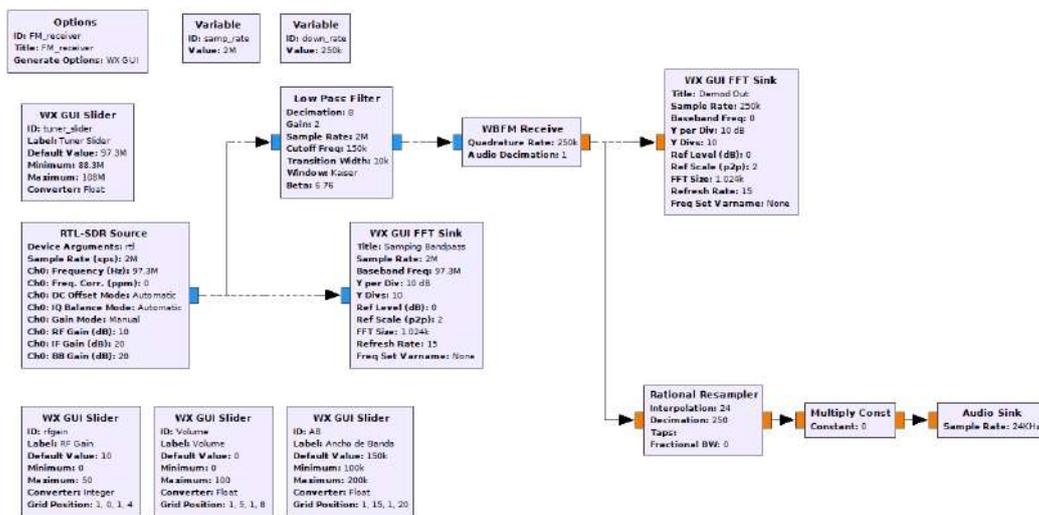


Figura A.3: Flowgraph de un receptor FM en GNU radio

La implementación fue exitosa, se logró recepcionar distintas emisoras de radios FM comerciales y brindó una primera experiencia en la utilización del SDR.

A.5.2. ADS-B

ADS-B, por las siglas en inglés *Automatic Dependent Surveillance Broadcast*, es una tecnología de vigilancia área diseñada para permitir a las aeronaves transmitir su estado de vuelo periódicamente sin la necesidad de ser interrogadas. La palabra automática refiere al hecho de que no es necesario ninguna acción por parte del controlador o piloto para su funcionamiento. Dependiente, por su lado, hace referencia a que esta tecnología extrae datos de otros sistemas, como el de geolocalización, del cual obtiene datos de posición y altura. Por último, difusión, dado que se emite a todos los usuarios que cuenten con dispositivos adecuados para recibir la señal, sin ningún tipo de encriptación. [6](#)

El tipo de enlace de datos de ADS-B son del tipo aire-tierra o aire-aire: la información puede ser recibida por las estaciones terrestres de control de tráfico aéreo como también puede ser recibida por otras aeronaves para proporcionar conocimiento de la situación y evitar colisiones. Dependiendo del tipo de mensaje, se suelen emitir datos de dirección, posición, velocidad, altitud, número de vuelo, identificación de la aeronave, estado de operación, entre otros parámetros.

A.5.2.1. Primeros pasos en la detección de ADS-B

Una vez adquirido el conocimiento del principio de funcionamiento del protocolo ADS-B y sus características técnicas detalladas en el apartado [E.5.2](#) de la ET, se definió que el dispositivo RTL-SDR usado hasta el momento, sería el indicado para continuar con el proyecto. Esta decisión se efectúa en base a evaluar que el SDR escogido

es apto para el rango frecuencias en que se emite el protocolo, cumple con el ancho de banda y alcanza la tasa de transmisión de datos necesaria. Se detallan sus parámetros y limitaciones en el apartado [E.4.7](#) de la ET.

Debido a la nula experiencia previa en la utilización de software de decodificación de ADS-B, se comenzó por utilizar una herramienta probada por la comunidad de radio aficionados, de fácil uso para el usuario. Es así que la primera recepción de los datos se realizó con el software RTL1090 [\[2\]](#), un ejecutable desarrollado para el sistema operativo Windows, el cual colecta y decodifica los mensajes recibidos por el SDR. En la Figura [A.5](#) se observa, como una vez que se ejecutó la herramienta, detectó el dispositivo RTL-SDR y ajustó de manera automática la frecuencia de sintonización, frecuencia de muestreo y ganancia.



```
OPEN          RTL1090 - (c) jetvision.de - B:161
1090.000 MHz  STOP

Started...
1 RTLSDR device(s) found.
Device: Generic RTL2832U OEM
TCP server port opened: 31001
UDP receiver port opened: 31002
Port 30003 type TCP server opened: 31004
HTTP server port opened: 31008
UDP target is: 127.0.0.1:31012
Device opened: "170101296"
** Manufacturer: Realtek
** Product:      RTL2838UHIDIR
** Serial:      00000001
Tuner type: "R820T"
RTL Xtal Freq: "28800000 Hz"
TUNER Xtal Freq: "28800000 Hz"
Gains: 0,9,14,27,37,77,87,125,144,157,166,197,207,229,254,
Cannot find gain. Gain adjusted to: 22.9 dB
Gain: 22.9 dB
Sample rate: 2000000 S/s
Tuner gain set to AUTO
RTL AGC set ON
Freq correction: 0 ppm
Freq set: "1090000000 Hz"
Buffer cleared

List  Table  Stats  I/SI  Scope  Plug-ins  RTL1090 homepage
>10  >20  >40  >80  >120  >180      UDP  BS  TCP  HTTP
32 ms  0/sec  THR: -78db [9]  Port:31001  A/C: 0  R820T-00000001
```

Figura A.4: Captura de iniciación del software RTL1090

En esta primera aproximación, se empleó el dispositivo RTL-SDR con un ordenador portátil con el sistema operativo Windows, el mencionado software RTL1090 y una antena comercial de la cual no se tenían datos técnicos precisos. La antena, con conexión SMA, fue extraída de un equipo con tecnología *GSM* de cuádruple banda, que aseguraba a priori, cobertura en la frecuencia de 1090 MHz. Se dispusieron estos elementos en una localización cercana al Aeropuerto Internacional Astor Piazzolla (distancia menor a los 2 km de distancia) con el objetivo de lograr línea de visión con las aeronaves ya sea durante el despegue o aterrizaje de las mismas.



Figura A.5: Instrumental utilizado en conjunto con RTL1090

Como resultado, se decodificaron 44 mensajes durante el despegue de la aeronave con numero de identificación ICAO E03341 como se muestra en la Figura A.11. La aeronave recepcionada pertenece a la empresa Aerolíneas Argentinas y cuenta con la matricula LV-CMA [3].



Figura A.6: Decodificación de 44 mensajes ADB-S mediante RTL1090

En esta oportunidad únicamente se logró obtener el numero de identificación, sin poder conocer datos de posición, velocidad o estado, entre otros. Las estadísticas evidenciaron numerosos paquetes dañados. La utilización de una antena sin conocer sus datos de fabricación, como rango de recepción o la ganancia en la frecuencia de 1090 MHz fueron las razones por las cuales se atribuyó esta mala recepción de la señal.

A.5.2.2. Antena *colineal* 1090 MHz

El primer desafío que surgió fue la fabricación de una antena sintonizada para una frecuencia de 1090MHz, capaz de ser utilizada para recibir la información del protocolo ADS-B. Debido a la escasa necesidad de materiales y sencilla construcción, se optó por un diseño de antena de $\frac{\lambda}{2}$ [5] comúnmente conocida como *colineal* como se muestra en la Figura A.7, tal de lograr una primera aproximación.

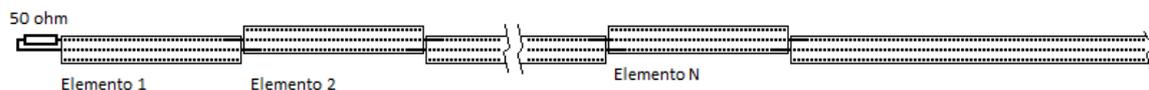


Figura A.7: Diagrama de construcción de una antena *colineal*

El diseño adoptado consistió en 8 dipolos de media longitud de onda, dispuesto de forma vertical. Para su fabricación, se utilizaron 8 segmentos de cable Coaxial RG 213 el cual posee una impedancia característica de 50 Ohms. Con el fin de calcular la longitud de los segmentos que oficiaron de dipolos, se calculó en primer instancia la longitud de onda λ :

$$\lambda = \frac{\text{Velocidad de la luz}}{\text{Frecuencia}}$$

$$\lambda = \frac{299792000\text{m/s}}{1090000000\text{Hz}}$$

$$\lambda = 0,275\text{m} = 275\text{mm}$$

Para el calculo de longitud, se consideró el factor de velocidad del cable Coaxial, que para el caso del Coaxial RG 213 es de 0.66:

$$\text{Longitud de segmentos} = \lambda \cdot \text{Factor de media onda} \cdot \text{Factor de Velocidad}$$

$$\text{Longitud de segmentos} = 0,275\text{m} \cdot \frac{1}{2} \cdot 0,66 = 0,09075\text{m} = 90,75\text{mm}$$

La unión de cada segmento, o dipolo, se realizó como se muestra en la Figura A.8. Se interconectó el conductor central del cable Coaxial al conductor externo (o malla) del contiguo elemento y viceversa. Se añadió cinta eléctrica aislante de PVC blanca de manera de aislar los conductores entre sí. En uno de los extremos, se soldó un conector SMA-macho que permitió la interconexión con el RTL-SDR.

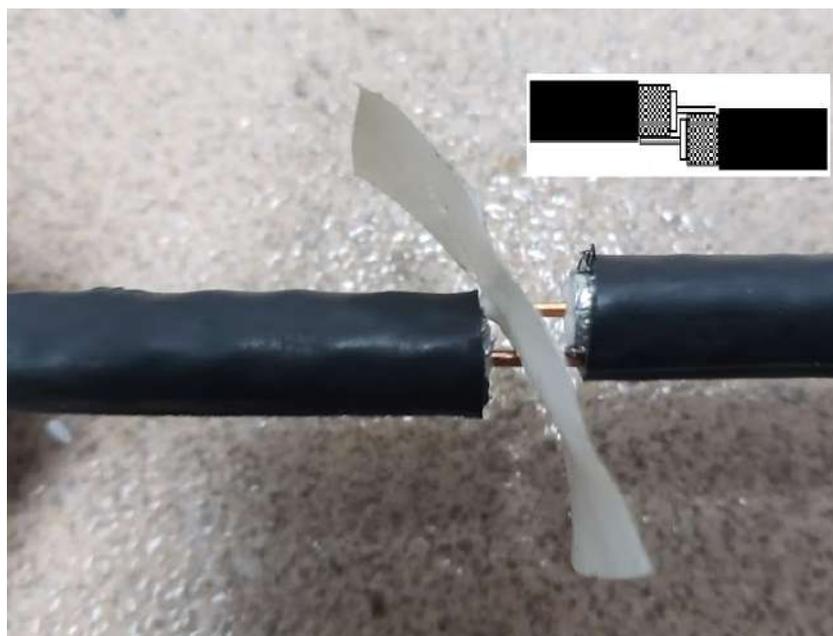


Figura A.8: Foto de la unión de segmentos en la antena *colineal* construida

Finalmente, la antena construida resultó como se expone en la Figura A.26



Figura A.9: Foto de la antena *colineal* finalizada

Se repitió el procedimiento detallado en la sección [A.5.2.1](#) con el objetivo de tener noción del comportamiento de la antena *colineal* recientemente fabricada. En dicha instancia se utilizó el siguiente banco de prueba:

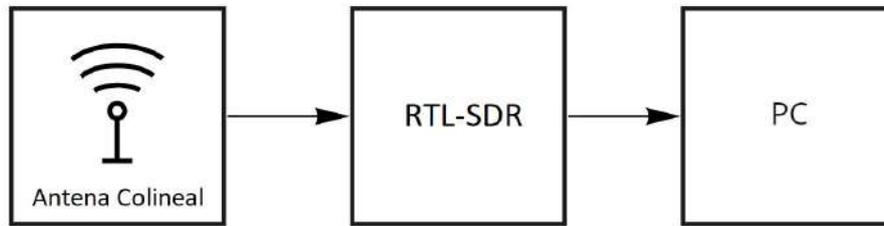


Figura A.10: Esquema del banco de pruebas utilizado con la antena *colineal*



Figura A.11: Decodificación de 221 mensajes ADB-S mediante RTL1090

A.5.2.3. Antena *spider* 1090 MHz

Una segunda antena se construyó, luego de la primera aproximación. En este caso se optó por diseñar y construir una antena $\frac{\lambda}{4}$ de carácter omnidireccional, popularmente conocida bajo el nombre de *spider*.

Los cálculos para su diseño y materiales utilizados se formalizaron en el apartado [E.4.1](#) de la ET.

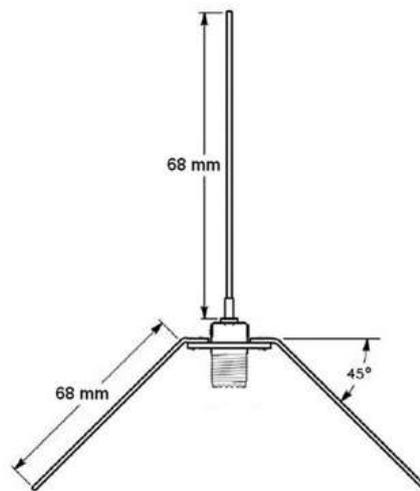


Figura A.12: Diagrama de construcción de antena *spider* para 1090MHz

Se procedió a la fabricación de la antena, obteniéndose como resultado lo expuesto en la Figura [A.13](#)

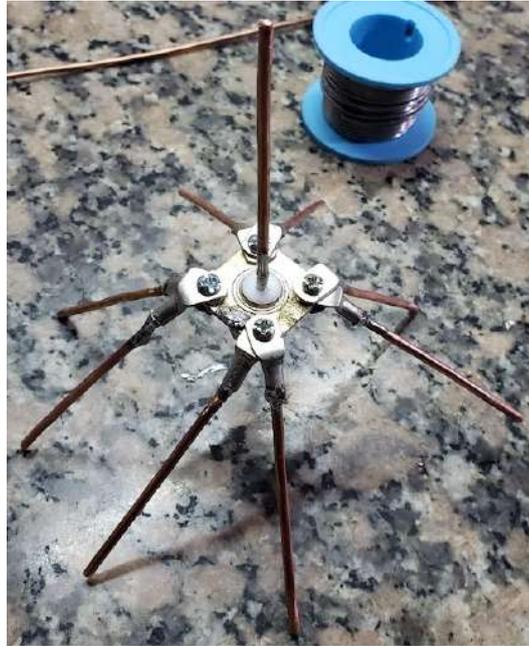


Figura A.13: Antena *spider* construida,

La etapa de calibración de la antena se llevó a cabo en el *Laboratorio de Comunicaciones* de la *U.N.M.D.P.* Para dicha tarea, se hizo uso del analizador de espectros *Hewlett Packard 8594E* en conjunto con el acoplador bidireccional *Hewlett Packard 778D* como se dispone en la Figura [A.15](#). El objetivo consistió en centrar la frecuencia de recepción de la antena en 1090MHz y lograr buenos parámetros de resonancia y selectividad. El procedimiento consistió en medir la respuesta de la antena y en consecuencia limar o recortar los 8 elementos conductores para ajustar la frecuencia de operación. Además, se fue modificando el ángulo tanto del conductor central como el del plano de masa de manera de lograr una respuesta de antena óptima.

Se realizó un barrido de frecuencias en el analizador de espectro hasta 2.9 GHz, en la que se identificó un claro mínimo en la respuesta en frecuencia. Mediante la utilización del *marker* se registró que el mínimo se ubicaba cercano a 1090 MHz, cumpliendo con lo esperado. En la Figura [A.14](#) se observa dicha respuesta en el analizador de espectros luego de la calibración.

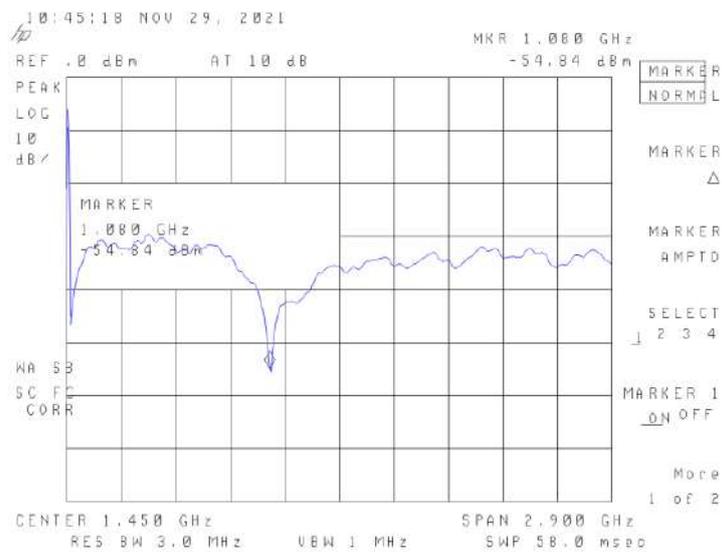


Figura A.14: Captura de la respuesta de la antena *spider* en el analizador de espectros



Figura A.15: Banco de medición para la antena *spider*. Analizador de espectros *Hewlett Packard 8594E* en conjunto con el acoplador bidireccional *Hewlett Packard 778D*.

En la Figura [A.16](#) se observa parte del proceso de calibración que logró la adecuada sintonización demostrada en la respuesta en frecuencia. En el mismo, se modificaron tanto la longitud de los elementos como el ángulo de estos respecto al plano de masa, factor clave para mantener la adaptación y por consecuencia la sintonización de la antena.

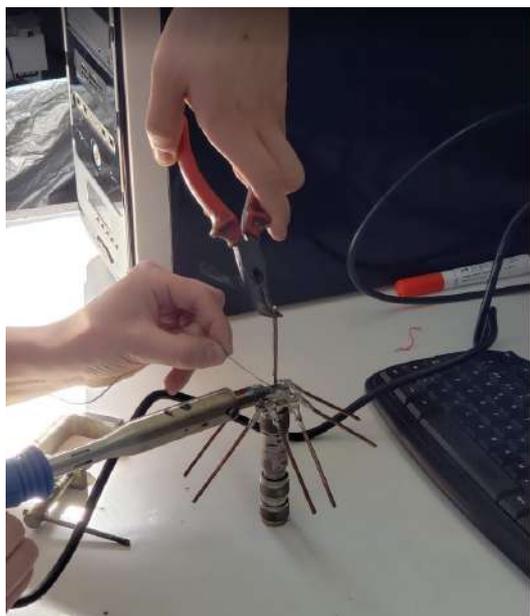


Figura A.16: Captura del proceso de calibración de la antena *spider*.

A.5.2.4. Filtro 1090MHz

Con el fin de limitar en banda la señal de ADS-B se optó por el diseño de un filtro microstrip. Se estudió la alternativa de realizar un filtro del tipo LC, con componentes discretos, pero el valor de estos para trabajar en el orden de 1 GHz no era físicamente realizables con la tecnología a disposición en el laboratorio. En consecuencia se diseñó un filtro microstrip Hairpin de orden 5.

Luego de diseñar el filtro con las características deseadas y medir la constante dieléctrica ϵ_r y la altura h del sustrato de la placa de cobre de material epoxi, se obtuvo el diseño de la Figura A.17. Las características técnicas se evidencian en el apartado E.4.4 de la ET.



Figura A.17: Layout del filtro 1090MHz en Ansoft Designer Student Version

La fabricación al igual que el filtro de 131MHz se realizó con el *método de transferencia térmica* (detallado en el apartado E.4.4 de la ET). Si bien este método no es óptimo para la fabricación de componentes que requieren una precisión muy alta como lo sería un filtro microstrip, se realizó con la precaución de en la etapa de diseño mantener una resolución de por lo menos 30 mils. También se consideró la posibilidad de hacer una calibración fina posteriormente.

El proceso de calibración consistió en conectar el filtro al analizador de espectros e ir removiendo cuidadosamente parte del cobre para aumentar la frecuencia central de la banda de paso, ya que esta había quedado a una frecuencia menor. Recortar la longitud de las pistas de cobre aumenta la frecuencia central dado que las mismas son inversamente proporcionales a longitud de onda. En la Figura A.18 se ve como se fue removiendo parte del conductor.

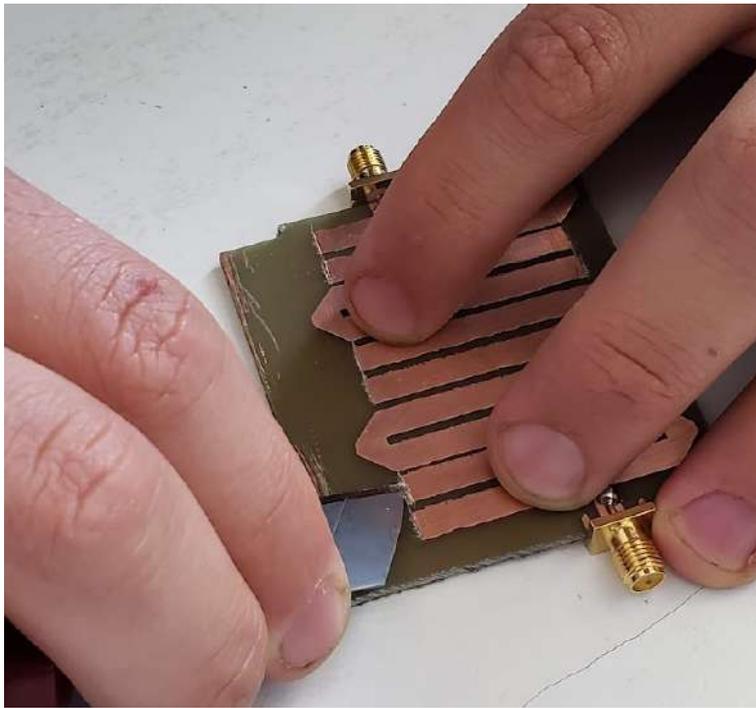


Figura A.18: Foto del proceso de calibración del filtro para 1090MHz

El resultado final fue el que se muestra en la Figura [A.19](#)

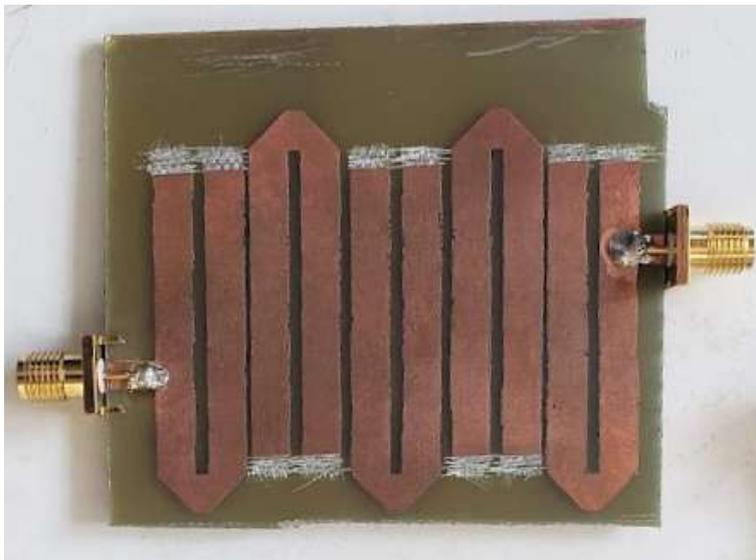


Figura A.19: Foto del filtro 1090 MHz calibrado

A continuación, se muestra en la Figura [A.20](#) del banco de medición que se utilizó para la prueba.

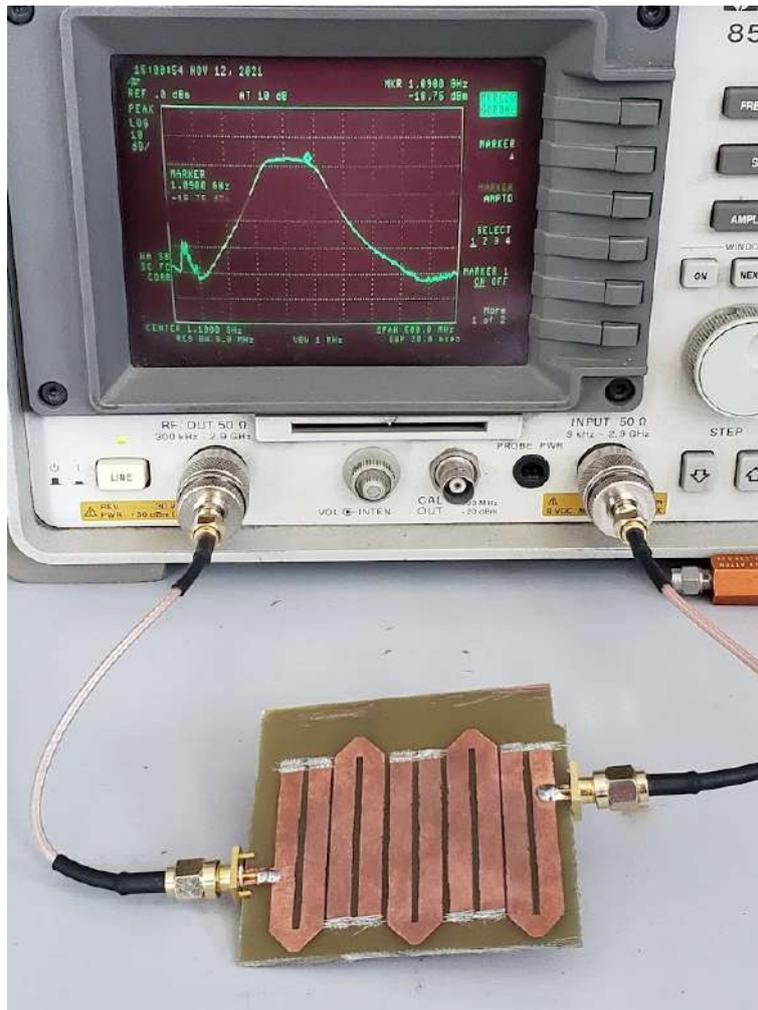


Figura A.20: Captura del banco de medición para la calibración del filtro ADS-B

El desplazamiento en frecuencia es evidente en las siguientes imágenes, en ambas puntero se encuentra en la frecuencia de 1090MHz. En la Figura [A.21](#) el filtro se encuentra originalmente como se fabricó.

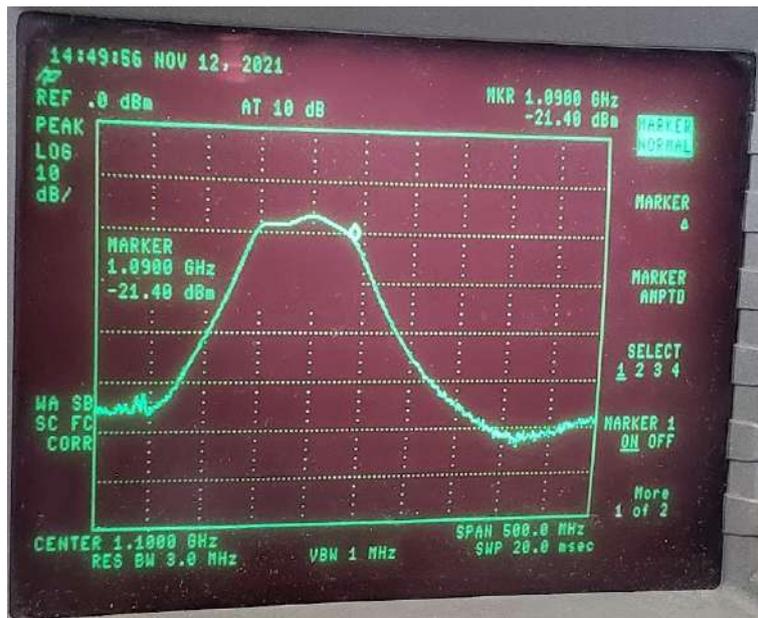


Figura A.21: Foto del analizador de espectros en el intermedio del proceso de calibración

En la Figura [A.22](#) se ve el resultado de remover parte del cobre perteneciente al diseño del filtro.

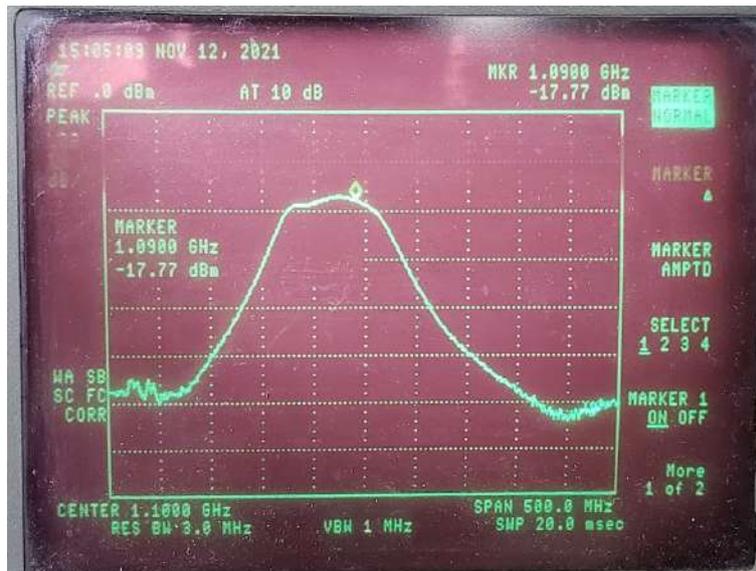


Figura A.22: Foto del analizador de espectros con el filtro de ADS-B calibrado

El resultado final se traduce en una mejora en relación a las pérdidas en la banda de paso. En la Figura [A.23](#) se muestra una captura del analizador de espectros haciendo un full span del filtro microstrip.

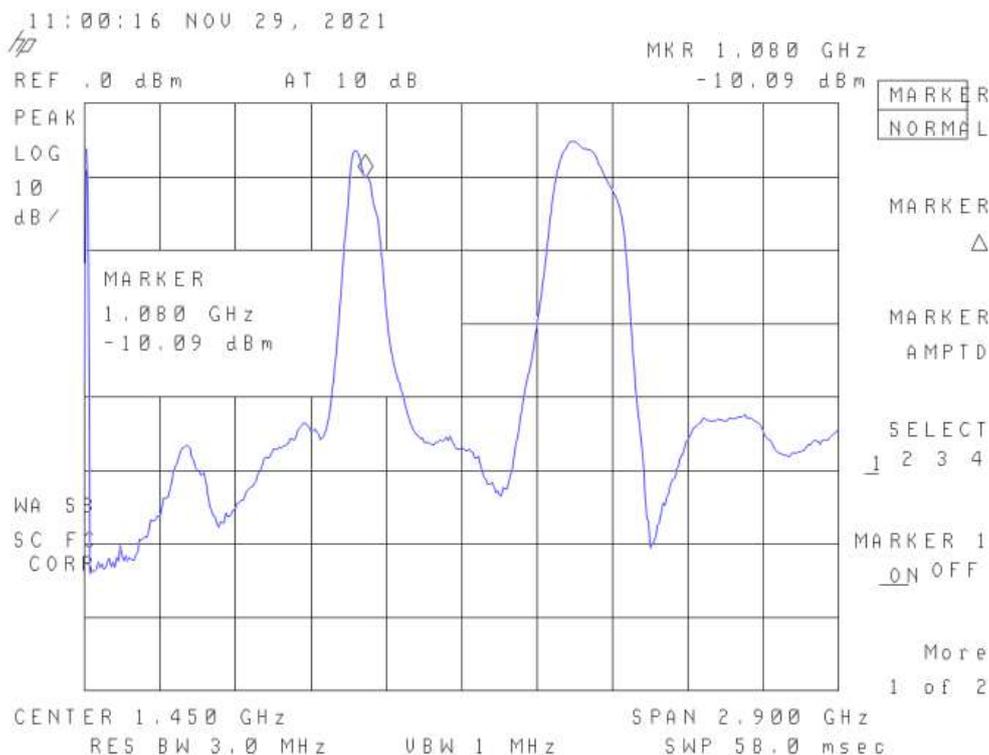


Figura A.23: Captura del analizador de espectros en modo full span del filtro ADS-B

Cabe destacar que al ser un filtro microstrip Hairpin se levantan los múltiplos de la frecuencia fundamental que se encuentra la banda de paso. Por lo tanto, como se puede apreciar en la Figura [A.23](#) se ve como se levanta una segunda banda de paso en 2,2GHz, lo que se encuentra muy próximo a la banda de 2,4GHz. Siendo esta una banda libre, es muy utilizada por equipos comerciales, lo cual implica una fuente de posible ruido, como pueden ser las señales de Wi-Fi. Se trabajó con la premisa de que posteriormente estas señales son eliminadas por el filtrado interno del SDR, que posee un pasa bajos con frecuencia de corte en 1.7GHz según su fabricante.

A.5.2.5. Implementación en GNU Radio Companion

GNU Radio es una herramienta de desarrollo libre y de código abierto la cual provee bloques de procesamiento de señal para la implementación de radios definidas por software. Es utilizada con hardware de RF de bajo costo o sin hardware en un ambiente propio de simulación. [4] A nivel de usuario la estructura de GNU Radio se basa en Python, desde la ejecución de los bloques hasta la configuración de los mismos. Los bloques comprenden funciones como decodificadores, filtros, PLLs hasta sistemas compactos ya realizados como de-moduladores para televisión digital terrestres, de-modulación de GSM o corrección de errores. [1]

La interfaz de usuario de GNU Radio consiste en el *flowgraph*, que representa las conexiones de los bloques por la que fluyen los datos. El concepto de *flowgraph* es un grafo no cíclico con uno o mas bloques fuente, los cuales introducen las muestras; uno o mas bloques de salida, que exportan muestras fuera del *flowgraph*; y bloques de procesamiento de señal entre estos.

El proceso de de-modulación, de-codificación, almacenaje y muestra de los paquetes recibidos de ADS-B se implementó mediante el ya introducido software GNU Radio. Se expone el *flowgraph* de la Figura A.24 que se adoptó para el protocolo ADS-B. Se hace un análisis extensivo del funcionamiento en Python de cada uno de los bloques involucrados en el apartado E.5.1.2 de la ET.

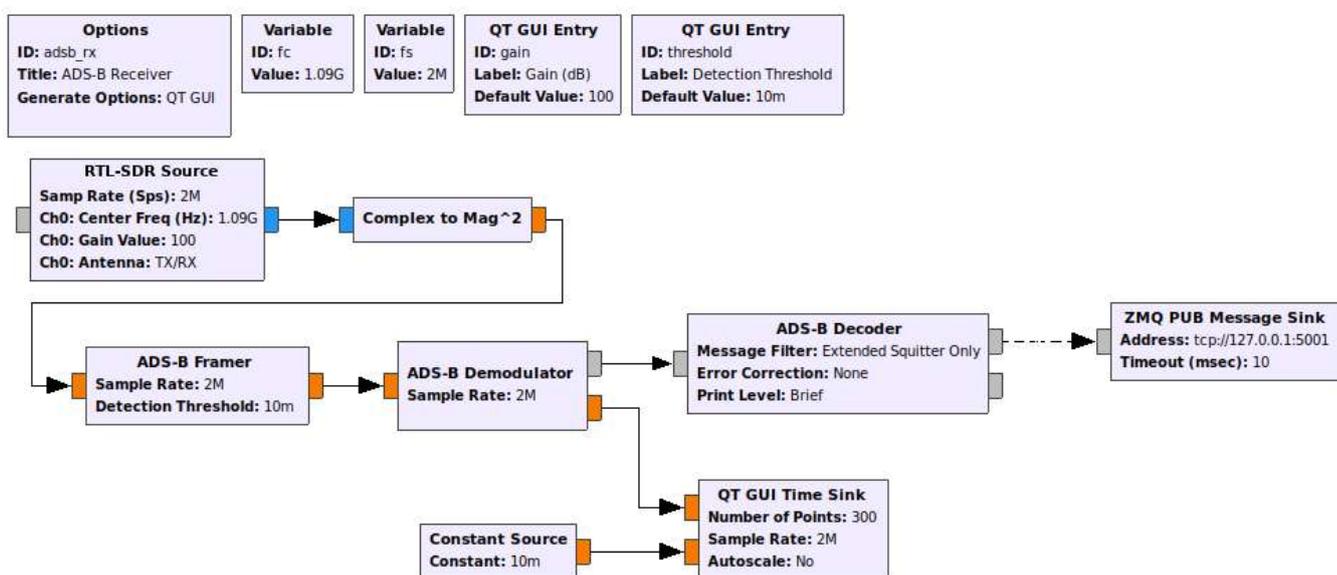


Figura A.24: Flowgraph ADS-B en GNU Radio

La obtención de muestras de la señal se obtuvo a partir del bloque *RTL-SDR Source*, el cual actuó de interfaz entre GNU Radio y el SDR. En su configuración, se ajustaron parámetros como la frecuencia de sintonización en 1090 MHz (*Ch0: Center Freq Hz*), la frecuencia de muestreo en 2 M muestras/seg (*Samp Rate Sps*) y ganancia en 100 (*Ch0: Gain Value*). Dichos valores son enviados al dispositivo, el cual internamente ajusta los parámetros para lograr la sintonización adecuada.

Una vez obtenidas las muestras, se realizó una conversión del mediante el bloque *Complex to Mag*. Dicho bloque calcula la magnitud en modulo de las muestras complejas obtenidas previamente y las envía a los siguientes bloques.

Los adyacentes bloques de procesamiento de señal: *ADS-B Framer*, *ADS-B Demodulator* y *ADS-B Decoder* oficiaron en las etapas de separación de frames, de-demodulación y de-codificación de las muestras, respectivamente.

La visualización de la posición de las aeronaves en tiempo real fue posible gracias a la vinculación que establece el bloque *ZMQ PUB Message Sink* con la integración de un servidor web, accesible mediante un navegador en la dirección *http://localhost:5000*. El servidor web, implementado en Python, recibió instante a instante la posición de las aeronaves mediante coordenadas de latitud y longitud; y las gráfico en un mapa 2D.

En la Figura A.25 se puede observar como se ve en pantalla la recepción de dos aeronaves. Una con un recorrido en naranja dirigiéndose al norte y otra con recorrido en rojo, dirigiéndose al oeste. Se seleccionó la aeronave que circulaba hacia el oeste, mostrando así la información recibida por el sistema de recepción ADS-B. En ella se incluye el ICAO, el distintivo de llamada, fecha y hora, altura, velocidad vertical, velocidad respecto de tierra, dirección, latitud y longitud.

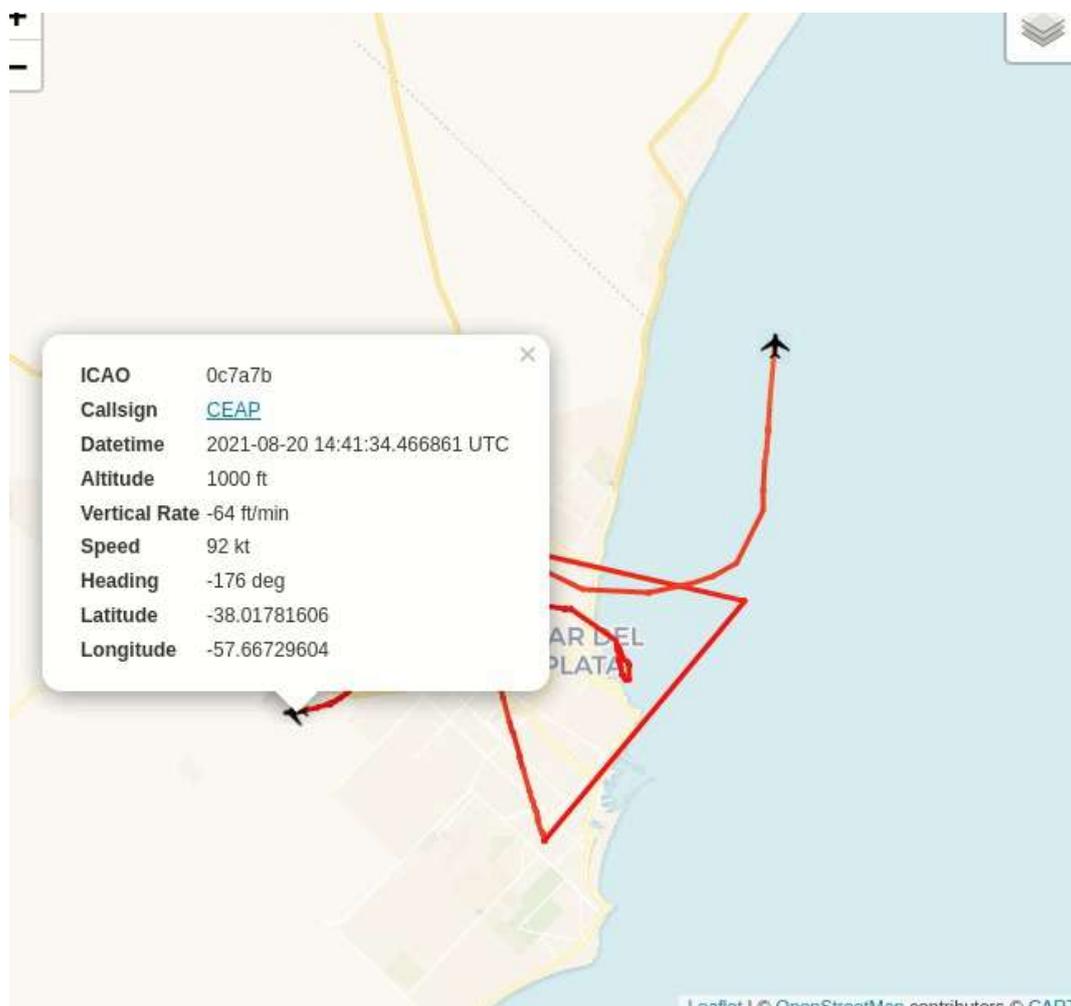


Figura A.25: Interfaz de visualización

A.5.3. ACARS

Aircraft Communication Addressing and Reporting System o ACARS es una infraestructura de comunicación aérea de datos entre una aeronave y una estación terrestre. Es utilizada como sistema de comunicación y de vigilancia por radio utilizada por las aerolíneas comerciales. Permite el control automático del estado del avión en vuelo, envío de telemetría hacia el centro de mantenimiento de la aerolínea propietaria de la aeronave, como así también, el encaminamiento de comunicaciones operativas y logísticas. Los mensajes del protocolo ACARS que se transmiten mediante VHF utilizan el rango de frecuencias de 118 MHz a 137 MHz; en Argentina y países de la región, se utiliza la frecuencia de 131,725 MHz. Se introducen las características técnicas más importantes del protocolo en la sección [E.5.2](#) de la ET.

A.5.3.1. Primeros pasos en la detección de ACARS

Se propuso realizar una primera aproximación a la recepción del protocolo. Con dicho fin, se optó por la utilización de una antena comercial bi-banda que disponía el laboratorio, detallada en la sección [E.4.2](#) de la ET. Esta antena originalmente se empleaba para cubrir la banda de 144 MHz con una ganancia de al menos 2.15dBi (según fabricante). Dada la cercanía con la frecuencia de interés en 131,725 MHz, se escoge como una alternativa viable.

En cuanto a la adquisición y procesamiento de la señal se utilizó el mismo esquema que para el caso de ADS-B, se empleó el RTL-SDR en conjunto con el procesamiento de una PC con GNU Radio Companion como se expone en la Figura [A.26](#):

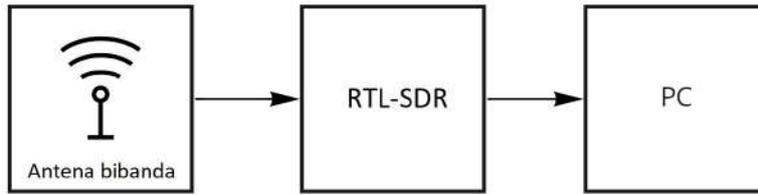


Figura A.26: Esquema del banco de pruebas utilizado con la antena *bibanda*

El *flowgraph* que se utilizó para decodificar los mensajes del protocolo ACARS se hace presente en la Figura A.27. Se exponen los aspectos técnicos más importantes sobre los bloques involucrados en el apartado E.5.2.2 de la ET.

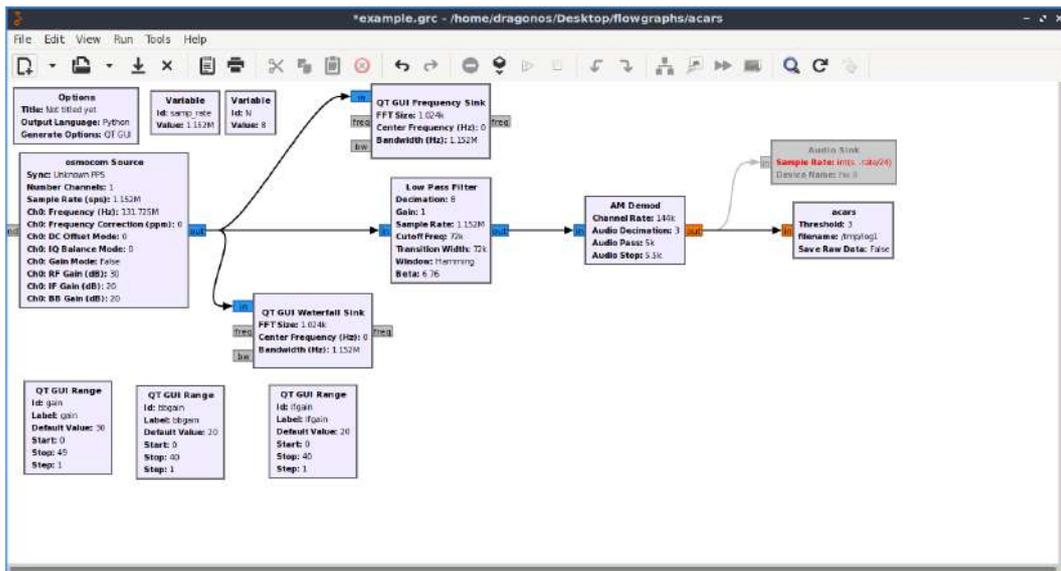


Figura A.27: *Flowgraph* recepción de ACARS en GNU Radio 3.7

Se logró decodificar parcialmente el contenido de al menos 10 mensajes. La totalidad de los datos recibidos fueron durante el aterrizaje en el Aeropuerto Internacional Astor Piazzolla de la aeronave con ICAO AR1604 y matrícula LV-CHS.

```

netbook@netbook-A0756: ~/acarsdec-master
File Edit Tabs Help
No: M68A
Reassembly: skipped
#2 Too many parity errors
#2 parity error(s): 3
#2 crc error
#2 errors fixed
-----
[#2 (F:131.725 L:-32 E:3) 24/09/2021 23:22:31.632 -----
Mode : 2 Label : 00 Id : 4 Nak
Aircraft reg: LV-CHS Flight id: AR1604
No: M69A
Reassembly: skipped
#2 parity error(s): 1
#2 crc error
#2 errors fixed
-----
[#2 (F:131.725 L:-31 E:1) 24/09/2021 23:22:49.815 -----
Mode : 2 Label : 00 Id : 4 Nak
Aircraft reg: LV-CHS Flight id: AR1604
No: M69A
Reassembly: skipped
-----
[#2 (F:131.725 L:-30 E:0) 24/09/2021 23:23:08.086 -----
Mode : 2 Label : 00 Id : 5 Nak
Aircraft reg: LV-CHS Flight id: AR1604
No: M70A
Reassembly: skipped
-----
[#2 (F:131.725 L:-28 E:0) 24/09/2021 23:23:19.960 -----
Mode : 2 Label : 00 Id : 5 Nak
Aircraft reg: LV-CHS Flight id: AR1604
No: M70A
Reassembly: skipped
SSS
  
```

Figura A.28: Captura de la recepción de ACARS

Los errores en la decodificación se atribuyeron una mala calidad en la recepción de la señal, y en consecuencia

se planteó la necesidad de limitar en banda la misma mediante el uso de un filtro.

A.5.3.2. Filtro 131.725 MHz

Para la fabricación del filtro para ACARS, se optó por un diseño de un filtro LC con componentes SMD. La justificación de esta elección es la facilidad que presenta fabricar estos tipos de filtros a la frecuencia que vamos a trabajar. En primera instancia se simuló mediante el software *Ansoft Designer Student Version* el diseño del filtro y su esperada respuesta en frecuencia. Se puede encontrar con mas detalle en la Especificación Técnica. El resultado final del diseño que muestra sus componentes es el de la Figura [A.29](#) y su layout en la Figura [A.30](#).

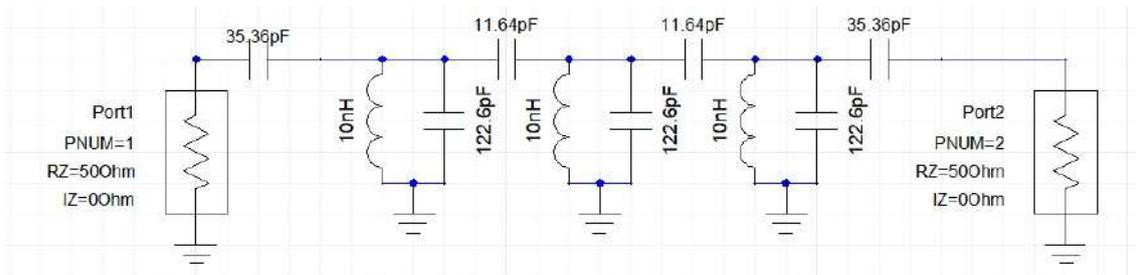


Figura A.29: Diagrama circuital del filtro de 131.725 MHz

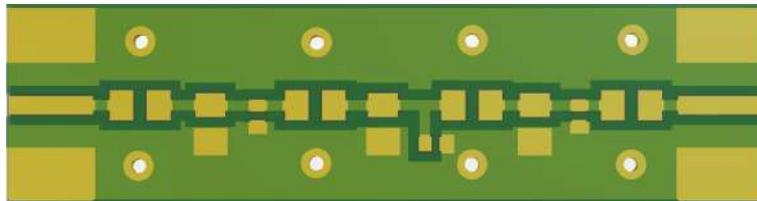


Figura A.30: Layout del filtro de 131.725 MHz

Para su fabricación se optó por el *método de transferencia térmica* (detallado en el apartado [E.4.4](#) de la ET) debido a su rapidez y facilidad para la fabricación. Se imprimió 4 veces el diseño del PCB, se verificó mediante un microscopio la integridad de las pistas de cobre y se eligió el mejor de estos. Los resultados se pueden observar en la Figura [A.31](#).

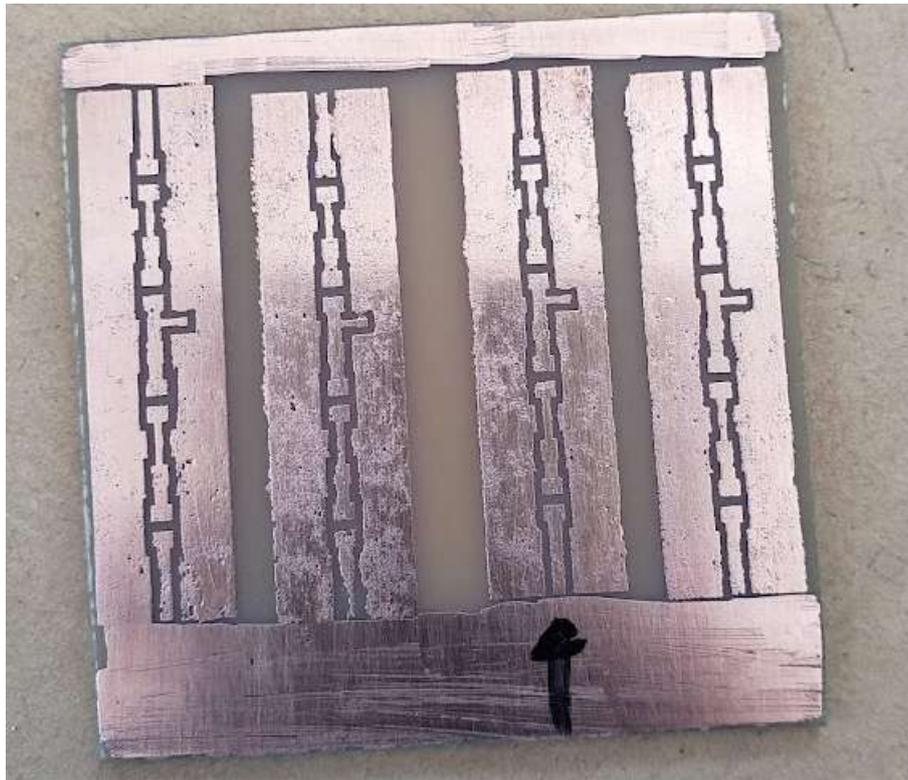


Figura A.31: Foto de las 4 impresiones del PCB para el filtro de 131.725 MHz

El resultado final luego de soldar los componentes capacitivos e inductivos SMD y los conectores SMA se muestra en la Figura [A.32](#).



Figura A.32: Foto del filtro de 131.725 MHz construido y soldado

Finalmente se comprobó el funcionamiento del filtro. A tener en cuenta que el banco de medición tiene un atenuador de 20 dB, por lo tanto la atenuación en la banda de paso del filtro es inferior a 6 dB como se puede apreciar en la Figura [A.33](#).

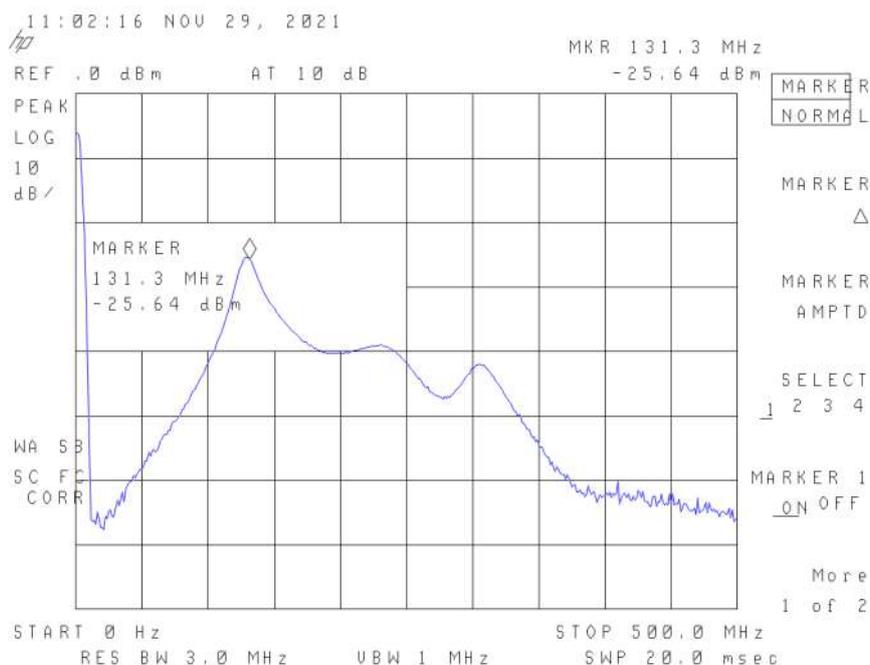


Figura A.33: Captura de la respuesta del filtro de 131.725 MHz en el analizador de espectros

Posteriormente se comprobó la atenuación a las frecuencias de las estaciones de FM que pueden ser una fuente importante de ruido. Estas son atenuadas mas de 20 dB para las estaciones superiores como las de 108 MHz (las más críticas por su cercanía a la banda de interés) y mas de 60 dB para las estaciones inferiores, cercanas a 88 MHz.

A.5.4. Unificación de los dos modos de funcionamiento

El siguiente desafío importante fue la necesidad de unir los dos sistemas de detección en un solo SDR. Dado que la mayoría de los SDR y en especial el RTL-SDR, cuentan con una sola entrada de RF mediante un conector SMA, existía la necesidad de unir dos canales de recepción en uno.

Se propusieron distintos diagramas en bloque para resolver este problema. Las consideraciones que se tuvieron en cuenta al momento de diseñar la solución fueron:

- La necesidad de amplificar la señal de entrada de ADS-B
- La opción de amplificar la señal de entrada de ACARS en caso de ser necesario
- La necesidad de filtrar las señales de entrada para disminuir el piso de ruido y mejorar la SNR
- Preferencia por una solución de forma pasiva (no requerir una acción de control para la selección del canal)
- Mantener la alimentación de sistema mediante un único cable USB

A.5.4.1. Primera solución

La primera solución que es propuso fue la siguiente:

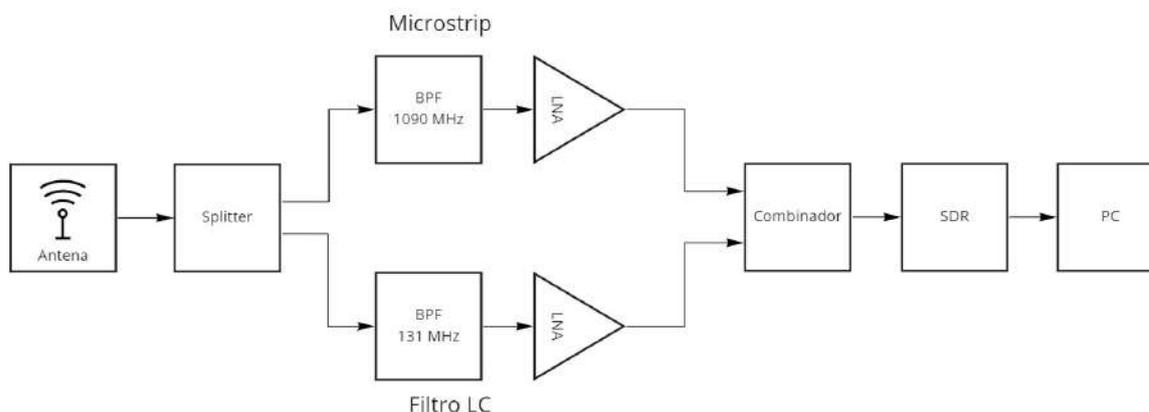


Figura A.34: Diagrama en bloque de la primera solución propuesta

En principio se propuso utilizar una sola antena que fuera capaz de sintonizar ambas frecuencias, 1317,25 MHz y 1090 MHz. Luego, mediante la utilización de un splitter, realizar un filtrado que no degrade significativamente la señal en cada banda de paso, eliminando la mayor parte del espectro no deseado y así reducir el piso de ruido. Para esto, se consideró necesario amplificar ambas señales utilizando de un LNA en cada una de las bandas. Finalmente, mediante un combinador, unificar el espectro nuevamente e introducir la señal en el SDR para su procesamiento.

El primer inconveniente que estuvo presente al momento de evaluar esta configuración fue la imposibilidad de fabricar con los elementos a disposición, una antena que pueda cubrir las frecuencias de 131,725 MHz y 1090 MHz de forma efectiva. Se investigó su fabricación y se probaron distintas antenas comerciales dispuestas en el laboratorio, de las cuales ninguna arrojó resultados positivos. Se hizo uso de un banco de medición similar al propuesto en la Figura A.15 para la medición de la antena Spider, mediante el uso del analizador de espectros *Hewlett Packard 8594E* en conjunto con el acoplador bidireccional *Hewlett Packard 778D*. Las antenas comerciales que presentaban una característica aceptable para la frecuencia de 1090 MHz, rechazaban de manera casi completa la señal en torno a los 131 MHz, y viceversa. Por este motivo, se decidió continuar con el estudio de otras alternativas.

A.5.4.2. Segunda solución

La segunda solución propuesta se observa en el diagrama en bloques de la Figura A.35.

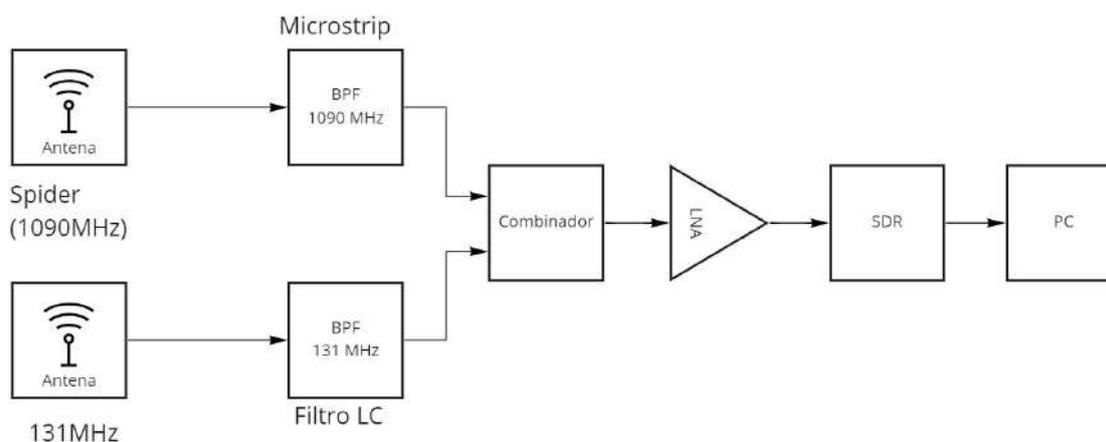


Figura A.35: Diagrama en bloque de la segunda solución propuesta

En vista de la imposibilidad de utilizar una única antena se recurrió al uso de dos antenas, una para 1090 MHz de fabricación propia (Spider, detallada en el apartado E.4.1 de la ET) y el uso de una antena bi-banda con un buen desempeño en 131 MHz (apartado E.4.2 de la ET). Se mantuvo el concepto de la primera solución: aplicar el filtrado correspondiente en cada canal mediante filtros pasa-banda y unificar mediante un combinador. El espectro resultante es posteriormente amplificado por un único LNA y la señal es procesada finalmente por el SDR.

Las ventajas de este diseño respecto al anterior fue la posibilidad del uso de las antenas optimizadas para cada banda, se eliminó la necesidad de utilizar un splitter, el cual después se comprobó que reducía por lo menos en 3dB las potencia por cada vía. Además, requería la utilización de un único LNA, posibilitando un menor consumo del dispositivo haciendo posible la alimentación del integrado mediante un único cable USB.

El problema que presenta esta configuración al momento de testear el funcionamiento, es que la degradación de la señal de 1090 MHz fue lo suficientemente grande como para quedar por debajo del piso de ruido, lo que resultó en que sea imposible decodificar la información recibida. Esto se debió a que la señal al ser de tan baja potencia y alta frecuencia, el filtrado inicial en 1090 MHz (que la banda de paso atenúa levemente), la degradación extra que aportaba el combinador y la pérdida de inserción de los conectores, terminaba degradando considerablemente la SNR. Por este motivo, se propuso una nueva modificación a fin de sortear este inconveniente.

A.5.4.3. Tercera solución

La tercera solución propuesta se encuentra representada por su diagrama en bloques en la Figura [A.36](#)

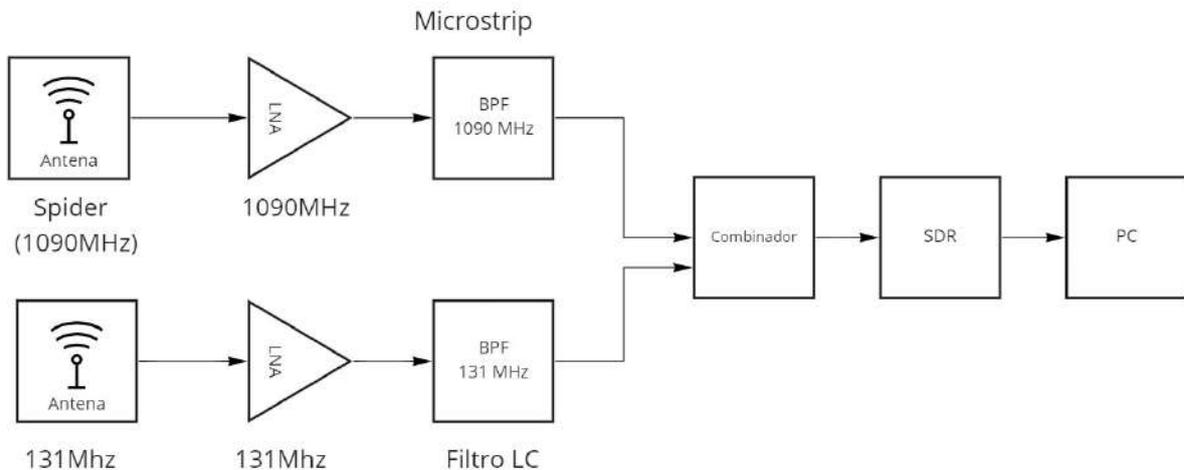


Figura A.36: Diagrama en bloque de la tercera solución propuesta

Para la tercera propuesta se resolvió el problema presentado en la segunda solución, con la amplificación de las señales de entrada y después su posterior filtrado. Esta inversión de los componentes generó excelentes resultados. Se logró recibir los paquetes de información a distancias superiores a los 100km como estaba especificado en la Especificación de Requerimientos y con una SNR superior a los 15 dB.

En esta etapa se cuestionó el uso del amplificador abocado a la señal de 131,725 MHz, dada la frecuencia de operación de la misma y la implicancia de la atenuación del espacio libre. Se propuso analizar esta atenuación en cada una de las frecuencias:

Teniendo en cuenta que la fórmula de atenuación del espacio libre es la siguiente:

$$At = \left(\frac{4\pi Rf}{c} \right)^2$$

Se calculó la atenuación correspondiente a cada frecuencia a un rango de 100 km tal que:

$$At_{131MHz} = 10 \log \left(\frac{4\pi Rf}{c} \right)^2 = 10 \log \left(\frac{4\pi \cdot 100000 \cdot 131000000}{3 \cdot 10^8} \right)^2 = 114,78dB$$

$$At_{1090MHz} = 10 \log \left(\frac{4\pi Rf}{c} \right)^2 = 10 \log \left(\frac{4\pi \cdot 100000 \cdot 1090000000}{3 \cdot 10^8} \right)^2 = 133,19dB$$

Se observa una diferencia de 18,41 dB de atenuación entre las señales, lo cual se traduce en términos de potencia, una diferencia cercana a las 100 veces.

A partir de lo expuesto anteriormente y luego de múltiples ensayos se identificó que el rendimiento extra que generaba tener un LNA para la frecuencia de 131,725 MHz no justificaba su implementación. La ganancia en la distancia de recepción no era significativa dado que sin el LNA ya superaba el requerimiento de 100km.

En consecuencia se optó por remover el amplificador de esta rama, reduciendo así los costos y manteniendo el rendimiento.

La solución final con la que se procedió a fabricar el dispositivo se encuentra representado en la Figura [A.37](#)

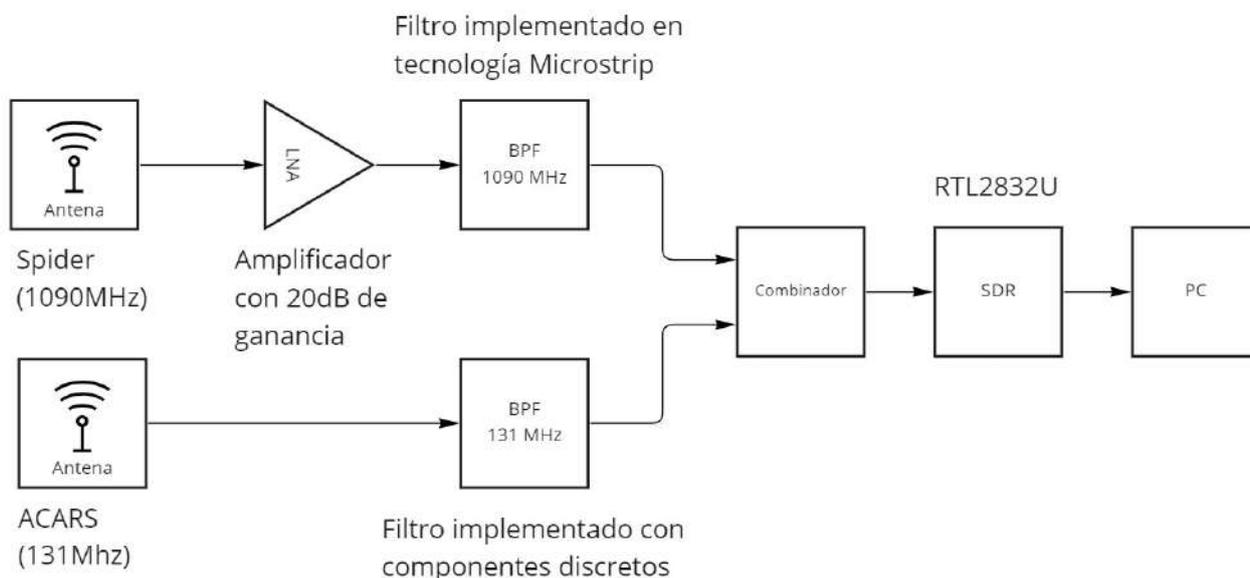


Figura A.37: Diagrama en bloque de la solución final

A.5.4.4. LNA

Como se mencionó en la el apartado de la ET, el LNA que se utilizó fue el correspondiente a la Figura [A.38](#)

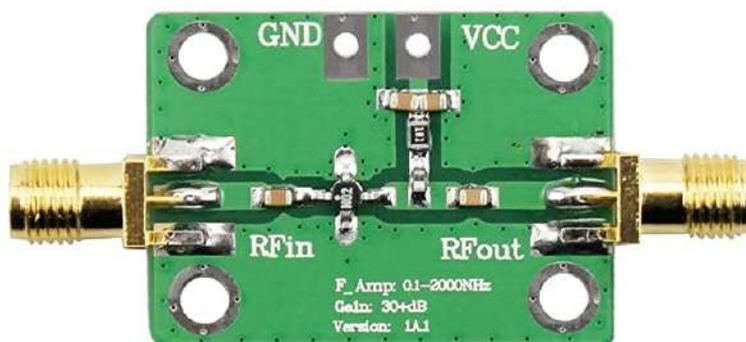


Figura A.38: Foto del LNA utilizado

Este amplificador se disponía en el laboratorio dado que había sido utilizado en otro proyecto y se garantizaba su correcto funcionamiento. En un principio se había estudiado el diseño de un LNA a partir de el integrado comercial MAX2640 el cual según su fabricante se especializa en el rango de 300-1500 MHz. La idea del diseño paso a segundo plano y se decidió conservar el LNA que ya disponía el laboratorio. Esta decisión se basó en que el resultado final que se podía obtener con una fabricación propia en el laboratorio, no iba a presentar tan buena Figura de ruido como el que ya estaba disponible.

Se procedió a medir la ganancia del LNA con el banco de medición de la Figura [A.39](#), que incluía un atenuador *Hewlett-Packard 8496B* y el analizador de espectros *Hewlett Packard 8594E*.



Figura A.39: Foto del banco de medición utilizado para medir la ganancia del LNA

Se obtuvo el resultado de la Figura A.40, el cual si bien estaba por debajo de lo especificado por el fabricante, igualmente presentaba una ganancia del cercana a los 20dB en 1090MHz, lo cual era suficiente para nuestra aplicación.

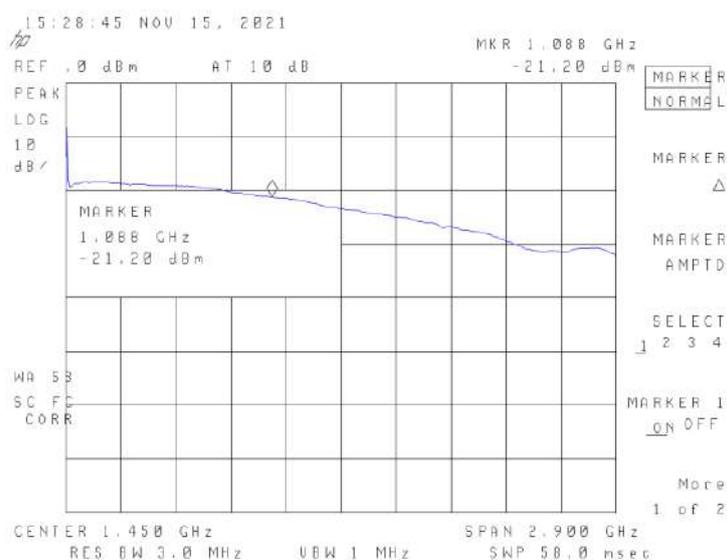


Figura A.40: Captura de la ganancia del LNA. Ganancia en pantalla -21,20 dBm con atenuador de entrada en 10 dB y atenuador externo en 30 dB

La ganancia medida fue de 18,8 dBm, dado que a la entrada del analizador de espectros el atenuador estaba fijado en 30 dB, mientras que el atenuador de entrada propio analizador estaba fijado en 10 dB. Esto corresponde con una alimentación de 9 V. Se podría aumentar la ganancia, aumentando la tensión de alimentación, pero concluimos que no era necesario, debido a que las pruebas realizadas con esta configuración ya sobrepasaban los requerimientos.

Se midió el consumo del LNA conectando en serie un multímetro a la alimentación, en un modo normal de funcionamiento. El consumo era un aspecto crítico debido a que posteriormente se quería alimentar todo el sistema con la potencia entregada por un USB 2.0. Se hará una descripción detallada más adelante en este informe. La Figura A.41 muestra el resultado de la prueba.

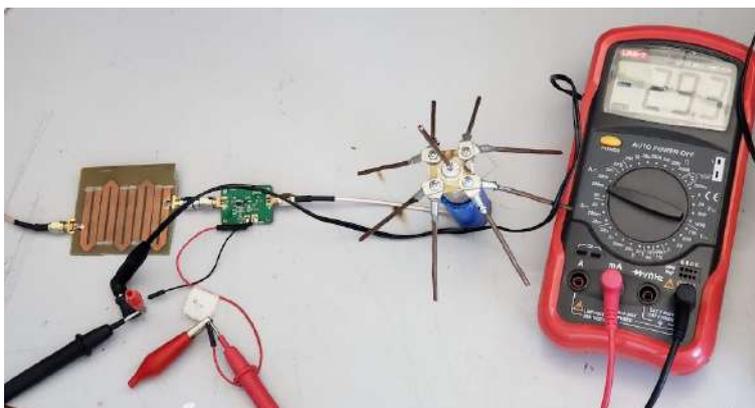


Figura A.41: Foto del banco de medición para el consumo del LNA

Otro aspecto importante a destacar era la necesidad de fabricar un blindaje electromagnético para el LNA. El funcionamiento de este es reducir el ruido que entra al dispositivo, lo que se traduce en una mejor SNR y menor piso de ruido a la entrada de nuestro sistema. Se fabricó a partir de una lámina de cobre de 0,1 mm y se lo conectó a masa. En el interior cuenta con un marco de PVC de alto impacto, para evitar cortos entre el blindaje y el LNA, además de dar rigidez y estructura para un manejo más cómodo. En la Figura [A.42](#) se puede observar el blindaje instalado.



Figura A.42: Foto del Shield instalado

Se verificó continuidad tanto con la alimentación como con la entrada de RF para evitar corto circuito que generen daños en el amplificador.

A.5.4.5. Alimentación del LNA

Como fue descrito en la Especificación Técnica, uno de los desafíos importantes que se presentaron fue la alimentación del LNA. En un principio para las pruebas se alimentaba con una fuente switching externa de 9V, pero como se quería generar una solución unificada por un solo cable, era prioridad alimentar todo el sistema (SDR + LNA) solo con un cable USB, el cual además transmitiría los datos a la PC.

Se midió el consumo del LNA en 30mA a 9V (270mW), lo cual era crítico ya que el USB 2.0 con el que se quería alimentar el sistema puede entregar un máximo de 500mA a 5V (2,5W). Dado que el RTL-SDR consume 300mA a 5v (1,5W), eso nos dejaba con menos de 200mA a 5V (1W) disponible para alimentar el LNA y un BOOST converter (cuyo consumo rondaba los 150mW) para elevar la tensión de 5V a 9V.

Una vez establecido que el USB 2.0 suministraba la suficiente potencia para alimentar todo el sistema, se fabricó un primer diseño para derivar la energía por un lado (para alimentar LNA + convertidor BOOST) y por el otro los pares de datos (para la conexión con el SDR) . Lo primero que se probó fue en una placa experimental soldar dos conectores USB, uno macho y otro hembra para conectar un extensor USB y el RTL-SDR en el otro extremo. Se quería derivar en paralelo las pistas de energía para alimentar el BOOST converter. En la Figura A.43 se muestra una imagen del primer prototipo.



Figura A.43: Foto de la primera solución propuesta para la derivación de energía y datos por USB

La verificación del funcionamiento fue un éxito en cuanto a la capacidad de alimentar el convertidor BOOST derivando en paralelo la alimentación del cable USB. El problema se presentó cuando, además de la alimentación se probó la capacidad del USB de transmitir la información del SDR a la PC: fue imposible transmitir datos. La primera hipótesis que existió fue que era un problema de conexión en los datos de los conectores USB, por lo tanto se probó de eliminar los conectores. Se utilizó un nuevo cable, se cortó a la mitad, se separaron los cables y se soldaron nuevamente lo más próximos entre sí en una placa experimental, como se muestra en la Figura A.44.

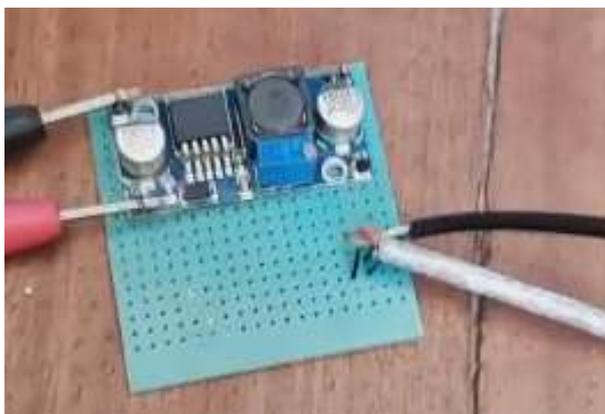


Figura A.44: Foto de la segunda solución propuesta

Esta nueva configuración ahora permitía alimentar exitosamente el convertidor BOOST y transmitir datos a través del USB. El problema se presentó cuando se analizó el rendimiento. Se vio enormemente deteriorada la SNR y sobre todo el piso de ruido. Era mucho mayor el ruido, comparado con un cable que no se había alterado. Por lo tanto se descartó esta opción, no porque fallara completamente, si no por el pobre rendimiento que presentaba la conexión.

Por último se procedió tal como se describe en el apartado [E.4.3](#) de la ET, se removió con cuidado el aislante de un nuevo cable USB, se desplazó la malla metálica que recubre los cables y se identificaron los cables tal como lo muestra la Figura [A.45](#).



Figura A.45: Foto de un cable USB sin la aislación

Posteriormente se removió la aislación de los cables de alimentación, con cuidado de no cortarlos y se soldó en paralelo con forma de "T" los cables para derivar la alimentación al convertidor, tal cual como se ejemplifica en la Figura [A.46](#).

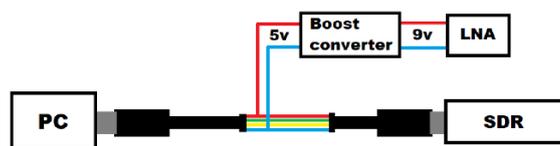


Figura A.46: Diagrama de la derivación de energía del cable USB

Finalmente, se volvió a colocar la malla metálica al rededor de los cables y se selló con cinta aisladora. Siendo el resultado final el de la Figura [A.47](#)

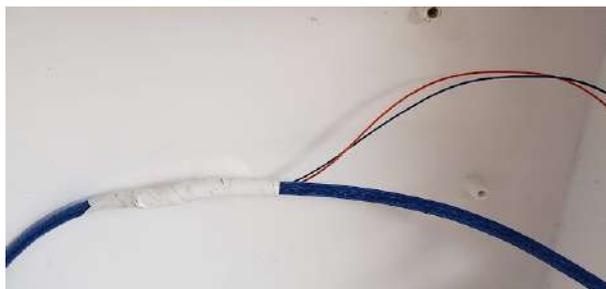


Figura A.47: Resultado final del cable con dos conductores derivados para la alimentación del LNA

Al momento de probar el cable no se pudo observar diferencia alguna en el rendimiento, comparado con un cable USB sin adulterar. El piso de ruido en ambos casos era prácticamente el mismo, por lo que se concluyó que la derivación de energía con este método fue exitoso. Además, se midió la tensión de alimentación del convertidor y cumplía con los requerimientos.

A.5.4.6. Combinador

Para el combinador se analizó y se estudió la posibilidad de usar una configuración Wilkinson, pero debido a que la diferencia de frecuencias entre una banda y otra era casi de una década, dificultaba el diseño de este tipo de combinadores. En consecuencia, se optó por diseñar y fabricar un combinador resistivo puro adaptado para 50 Ohms, que duplique el espectro en 2. El proceso de diseño se detalla en la sección [E.4.6](#) de la ET, y el diseño final es el que se expone en la Figura [A.48](#).

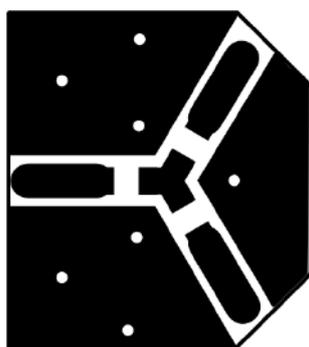


Figura A.48: Diseño final del combinador resistivo adaptado a 50 Ohms.

Para la fabricación nuevamente se optó por el *método de transferencia térmica*. Luego se soldaron 3 resistencias SMD en paralelo de 50 Ohms en cada una de las pistas. Finalmente, se verificó que la resistencia de cada rama sea de $\frac{Z_0}{3} = \frac{50}{3} = 16,66$ Ohms, verificando una correcta soldadura, como se puede apreciar en la Figura [A.49](#).



Figura A.49: Foto de la medición de la resistencia de una las ramas del combinador

Por ultimo, se soldaron los conectores SMA y el resultado final fue el de la Figura [A.50](#)



Figura A.50: Foto del combinador resistivo construido y soldado

A.5.5. Solución final

A.5.5.1. Montaje

Como se describió en el apartado [E.4.10](#) de la ET se utilizó una caja estanca de dimensiones 420mm x 268mm x 98mm. El material de la caja es polipropileno blanco con retardador de llama, cuenta con protección

UV (solar) y mejorador de impacto. Todo el sistema que se encuentra adentro está protegido en caso de lluvias dado que según el fabricante posee certificación IP 65. dentro está protegido en caso de lluvias dado que según el fabricante posee certificación IP 65. Se realizó una perforación y se instaló un conector para sacar el cable USB hacia el exterior para que pueda ser conectado a la PC. Fue necesario sellar la salida con un tapón de goma para evitar la entrada de agua. Al igual que la salida para el cable USB 2.0 , se requirió sellar la salida de las antenas para evitar las filtraciones de agua. A continuación, en la Figura [A.51](#) se puede ver el resultado final.



Figura A.51: Foto de la salida del cable al exterior de la caja

Finalmente el resultado del dispositivo completo se puede observar en la Figura [A.52](#) sin la tapa frontal y en la Figura [E.2](#) armado completamente.

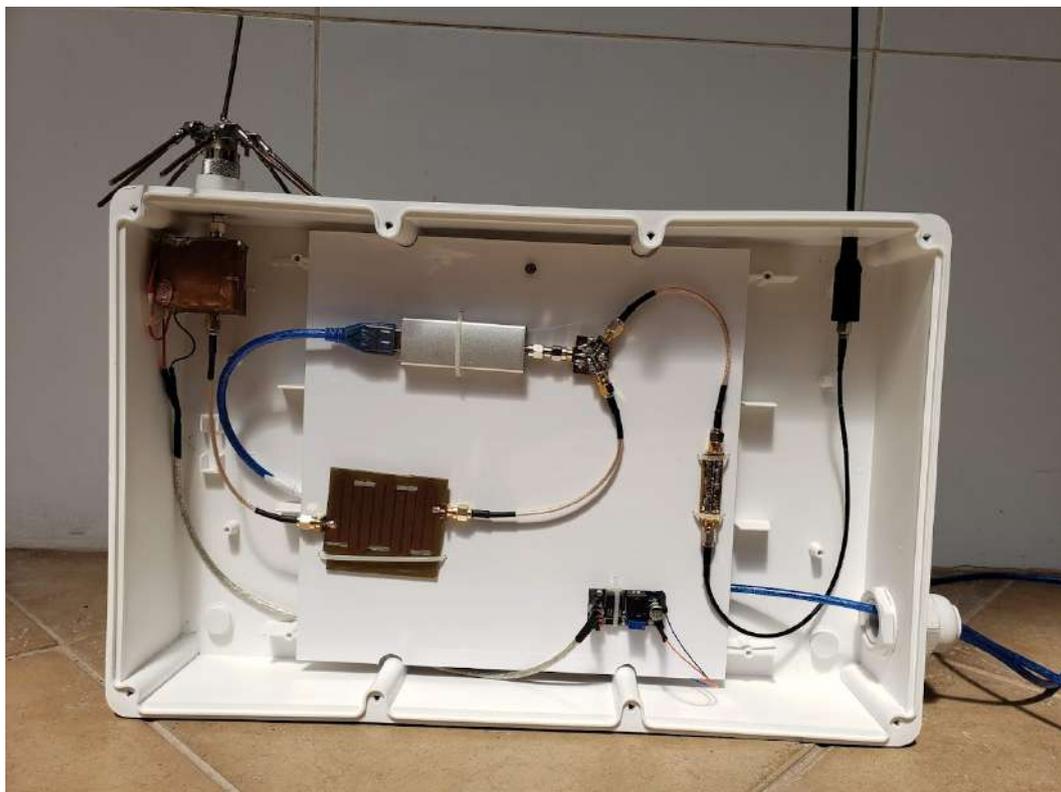


Figura A.52: Foto frontal del equipo final sin la tapa frontal



Figura A.53: Foto frontal del equipo final con la tapa frontal

Se probaron múltiples disposiciones de los elementos dentro de la caja y se concluyó que la que se ve en la Figura [A.52](#) era la más elegante y funcional que se podía realizar con los elementos a disposición.

Se trató de mantener los elementos lo más alejados posibles entre sí para evitar interferencias y para mejorar la presentación. Se pasaron los cables por detrás del PVC de alto impacto que soporta los componentes principales. Se realizaron las perforaciones para las antenas de forma precisa con un mini-torno, para evitar huecos donde pueda filtrarse agua, y se sellaron con silicona para reforzar la protección.

A.5.5.2. Rendimiento

Una vez dispuestos los elementos en la caja estanca en su disposición final, se realizaron nuevas mediciones para corroborar el cumplimiento de los requisitos de rendimiento: se estableció un mínimo valor de relación señal a ruido en 15 dB (RR01) y un rango de recepción de por lo menos 100 km (RR02) en la recepción de la señal de ambos protocolos.

En el caso de ADS-B, se lograron obtener mensajes de la aeronave con ICAO *AR1921*, en las coordenadas Lat -37.261 Lon -59.262 que corresponden a las cercanías de la ciudad de Tandil, en la Provincia de Buenos Aires. Con la ayuda del uso de la herramienta *Google Maps*, se midió la distancia entre dichas coordenadas y las que se encontraba el receptor, obteniéndose una distancia de 176 km como se observa en la Figura [A.54](#).

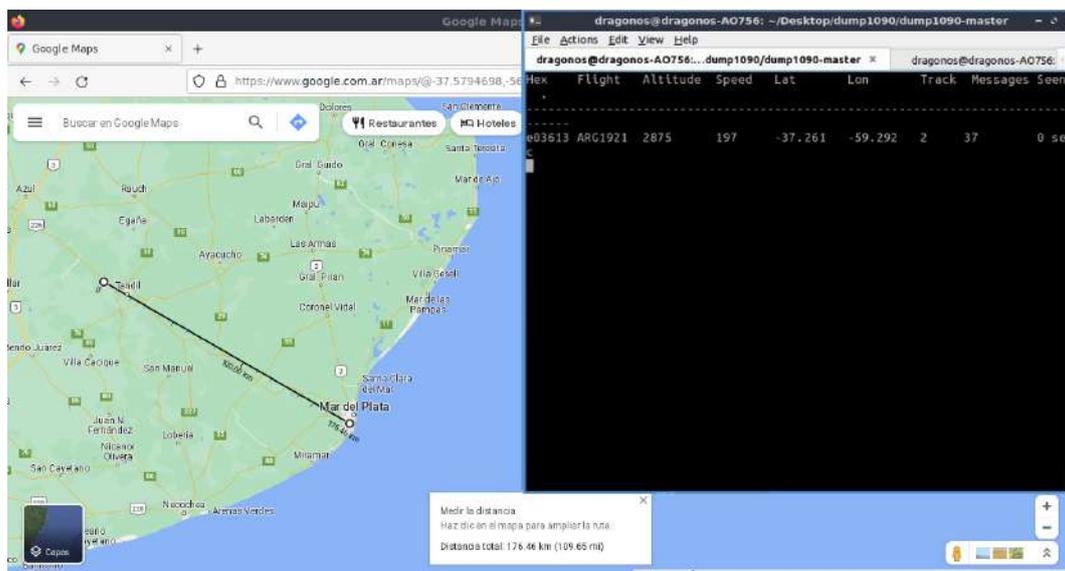


Figura A.54: Recepción de la aeronave ICAO *AR1921* en las inmediaciones de Tandil, Provincia de Buenos Aires.

Para el caso de ACARS, la distancia de recepción fue superada cuando se lograron decodificar mensajes emitidos por la aeronave matrícula *LV-FUC* con ICAO *AR1883* en el cual anuncia su posición mediante las coordenadas *S 38.470 W 59.725* en el cuerpo del mensaje. Nuevamente, utilizando *Google Maps* se midió la distancia entre dichas coordenadas y las que se encontraba el receptor, obteniéndose una distancia de 197 km como se observa en la Figura [A.55](#).

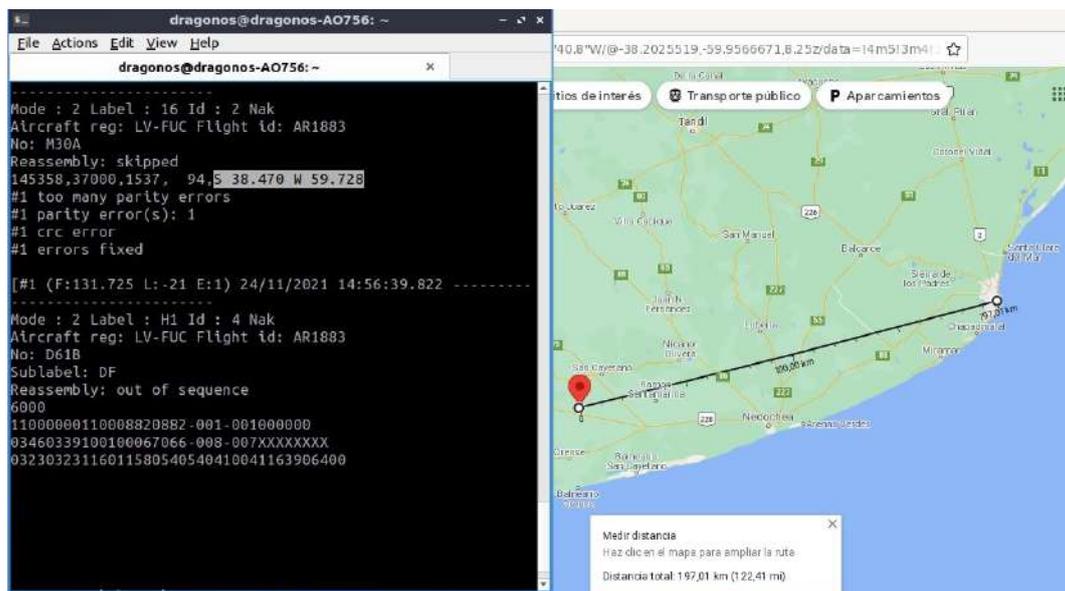


Figura A.55: Recepción de la aeronave matrícula *LV-FUC* con ICAO *AR1883* en las inmediaciones de San Cayetano, Provincia de Buenos Aires.

A.5.5.3. Modificaciones a futuro

Hay consideraciones que hay que tener en cuenta que todavía no fueron abordadas en este informe final. Una de ellas son las posibles mejoras que se le puede realizar al equipo si se contara con los suficientes recursos. En primera medida se podría agregar otro LNA en la rama de ACARS, esto aumentaría el rango de recepción. Otra alternativa para mejorar la recepción en 131,725 MHz es construir una antena especialmente diseñada para esa frecuencia, en vez de utilizar una comercial. Esto implica un consumo de tiempo en investigación y tiempo considerablemente mayor que simplemente utilizar una comercial, pero mejoraría el rango de recepción.

Se tiene que tener en cuenta que cuanto mayor sea la altura a la que se encuentre el dispositivo, mayor será la distancia máxima a la que pueda recibir, debido a que mejora la línea de visión y se mitiga el efecto que

genera la curvatura de la tierra. Además para mejorar el rendimiento es necesario que se encuentren la menor cantidad de objetos posibles entre medio del enlace entre la aeronave y el dispositivo.

Para mejorar el alcance de ADS-B se puede reducir incluso más el camino del LNA hasta la entrada del SDR. Como la frecuencia de trabajo esta en el orden del GHz las pérdidas son considerablemente mayores que las de 131MHz, por lo tanto si se utilizarán conectores rectos entre el LNA, filtro y combinador, se aumentaría el rango de detección dado que se reducen las pérdidas.

También una de las mejoras que se pueden agregar es cambiar el PVC de alto impacto, por una placa de cobre o aluminio, conectado a masa, que funcione como shield para reducir los posibles interferencias. Tener en cuenta que no se puede colocar una placa metálica en la tapa frontal de la caja, porque alteraría la respuesta del filtro de ADS-B, ya que está diseñado como microstrip (plano de masa inferior y vacío en la cara superior) y un conductor por delante de este lo transformaría en un stripline (guía de onda entre 2 planos de masa)

Otro punto importante a tener en cuenta es el montaje fijo en el exterior. En caso de querer tener el dispositivo de forma permanente conectado, será necesario instalar en la parte posterior o inferior del equipo un soporte para montar en un perfil estructural, el cual proporcionará la altura deseada. Este deberá estar sellado para evitar el ingreso de agua a la caja. Se recomienda sellar con silicona adecuada.

En caso de que sea necesario tener una mayor distancia entre el equipo y la PC, se puede utilizar un extensor activo USB para enviar la señal casi 100 metros más. En caso que se desee montar en el exterior de forma permanente se pueden remover los tornillos que mantienen cerrada la caja, e introducir el extensor activo adentro de esta para evitar que se arruine en la intemperie. No se recomienda utilizar un extensor pasivo dado que la distancia máxima que soporta el protocolo USB ronda los 2 metros, de los cuales ya son utilizados 1.8 metros en el equipo original.

Respecto a las protecciones eléctricas el fabricante del SDR asegura que el dispositivo posee un diodo ESD que protege de descargas electromagnéticas, además de un BAV99 que recorta señales excesivamente fuertes y previenen el daño del equipo. Igualmente este recomienda tener instalado en las proximidades del dispositivo un para rayos, cumpliendo los requisitos necesarios para su correcto funcionamiento. Se recomienda desconectar las antenas en caso de tormenta cuando el equipo no se encuentre en uso para evitar daños. Como seguridad adicional se estudió el uso de un tubo de descarga de gas o GDT, para proteger el equipo en caso de rayos. Este no se incluyó en el desarrollo y construcción por falta de presupuesto, pero la instalación es relativamente simple. Es necesario conectarlo en serie a la antena y conectado a tierra para descargar el exceso de energía en caso de la caída de un rayo al dispositivo. Se describe el GDT en el apartado [E.4.9](#) de la ET.

A.6. Conclusiones

El trabajo final de carrera transcurrió en un periodo de aproximadamente año y medio, durante el cual tuvimos la singular oportunidad de desarrollar un proyecto de ingeniería a elección, de forma independiente con el asesoramiento y ayuda de nuestros directores tanto como de la cátedra de Proyecto Final. Estos últimos asistieron con la gestión de proyecto y documentación del proceso, lo cual fue nuevo para nosotros dado que nunca habíamos hecho gestión o planificación de un proyecto de similar magnitud previamente. Inicialmente costó entender la necesidad de realizar los documentos pedidos (Especificación de Requerimientos, Funcional, Técnica y Plan de Pruebas) y de documentar el proceso completo en una bitácora, pero a medida que avanzaba el proyecto se hacía evidente la necesidad de tener un plan estratégico para afrontar el desafío propuesto.

Durante el desarrollo del dispositivo, fuimos ensayando soluciones de menor a mayor complejidad. En un principio, tratamos de recepcionar ambos protocolos simplemente utilizando herramientas de software ya probadas y mediante bancos de prueba simples. Una vez que se obtenían primeros resultados satisfactorios, tratábamos de mejorar la calidad de estos mediante soluciones que requerían mayor desarrollo. Luego de iterar sobre distintas configuraciones, buscando la mas adecuada, cumplimos con los requerimientos y superamos las expectativas originales.

La antena *colineal*, realizada como una primera aproximación, no tuvo el comportamiento esperado durante la recepción, aunque mediciones posteriores en el laboratorio evidenciaron una adecuada sintonización. Es posible que durante las pruebas se hayan desconectado los sucesivos dipolos, modificando así la respuesta de la misma. El segundo diseño, presentado como antena *spider*, resultó efectiva en el diseño y construcción, posibilitando la recepción de la señal. Resaltamos el aspecto clave que resultó durante las distintas pruebas la importancia de que la misma cuente suficiente altura y línea de visión frente a las aeronaves.

Respecto a las etapas de filtrado, el filtro microstrip luego de ser calibrado presentó una atenuación aceptable en la banda de paso con la desventaja de tener múltiplos de la frecuencia fundamental por consecuencia del diseño utilizado. En nuestra aplicación esto no presentó inconvenientes dado que, como se analizó previamente, el circuito de entrada del SDR posee un filtro que atenúa las frecuencias mencionadas. El filtro desarrollado con componentes SMD para ACARS, cumplió con lo esperado y al igual que el filtro de microstrip, la atenuación en la banda fue aceptable. Además, cumplió con el objetivo de rechazar fuertemente las frecuencias cercanas utilizadas por radios FM.

Concluimos que es acertado el uso de SDR para la recepción de datos de telemetría, cuenta con ancho de banda suficiente, frecuencia de muestreo y rango de sintonización adecuado para la recepción de ambos protocolos. Destacamos la ventaja de ser utilizado en combinación con *GNU Radio* generando gran versatilidad para el procesamiento de señales y posible vinculación con otras aplicaciones, como en nuestro caso, un navegador web.

Desde el punto de vista del trabajo final como instrumento de formación en la carrera de Ingeniería Electrónica de la Universidad Nacional de Mar del Plata, podemos concluir que es un excelente medio para dar un cierre a la carrera y englobar conceptos que fueron desarrollándose a lo largo de los años de estudio. El hecho de tener la libertad de orientar el trabajo hacia el camino que quiere el estudiante es motivador para el desarrollo de un dispositivo propio. Se debe destacar el beneficios de tener asignado tutores con experiencia y amplio conocimiento técnico proporciona un ambiente en el cual se puede desarrollar el trabajo de forma fluida permitiendo un avance continuo aun así en momentos de dificultad técnica y en algunas ocasiones de baja motivación. Otra oportunidad grande que brinda esta modalidad fue el tener el acompañamiento de la Cátedra de Trabajo final, para guiar a los estudiantes en una modalidad de trabajo estructurada y símil al mundo profesional en el que nos fuimos desenvolviendo a la par del desarrollo.

A lo largo del proyecto nos capacitamos en múltiples ramas de la electrónica, entre ellas, el diseño de circuitos de RF, antenas, el procesamiento digital de señales, programación, diseño de filtros en microstrip y coplanares, entre otras tantas, que si bien algunas no son rigurosamente parte del resultado final, fue necesario adentrarse para tener contexto suficiente para desarrollar de forma satisfactoria el trabajo.

Aunque el resultado final fue exitoso, el camino no estuvo corto de dificultades. A lo largo de los 16 meses en que se desarrollaron las actividades se presentaron múltiples problemas que tuvimos que enfrentar. El principal de ellos fue la pandemia de Covid-19. El hecho de que se encontraban limitados los vuelos nos perjudicó enormemente al momento de realizar pruebas, dado que al haber un único vuelo diario hacia la ciudad de Mar del Plata, nos daba una marginal ventana para realizar las pruebas. En el principio del desarrollo estas arrojaban resultados no satisfactorio y era necesario esperar 24 horas para la siguiente. Al margen de las dificultades técnicas y tecnológicas que afrontamos a lo largo de las actividades, se presentó una pérdida de motivación, que fue un factor que no se tuvo en cuenta al comienzo. Agradecemos haber contado con el total apoyo de nuestros directores, amigos y familiares para afrontar este tipo de dificultades que son difíciles de detectar, pero generan un enorme impacto en el rendimiento. Cronológicamente, el último y mayor desafío que tuvimos que enfrentar fue la culminación del informe final. Se terminó la mayor parte de la escritura en un contexto en el cual los dos integrantes comenzaron a trabajar tiempo completo, fuera de su ciudad natal. La adaptación al nuevo estilo de vida impacto fuertemente en la disponibilidad de tiempo para culminar con la escritura.

Por último, queremos agradecer enormemente a nuestros directores: Alejandro Uriz, Juan Alberto Etcheverry, Jorge Luis Marquez; nuestros amigos, compañeros de la carrera y familiares por su incondicional apoyo a lo largo de todos estos años, en el desarrollo del trabajo final y en el día a día.

Bibliografía

- [1] Kevin Herney Rodríguez Gómez. “Diseño e implementación de sistema de recepción para comunicaciones aéreas ACARS”. En: (2017).
- [2] Jetvision. *RTL1090*. URL: <https://rtl1090.com/>. (accessed: 05.04.2020).
- [3] Thomas Noack. *Planespotters*. URL: <https://www.planespotters.net/hex/E03341>. (accessed: 21.09.2022).
- [4] GNU Radio project. *GNU Radio*. URL: <https://www.gnuradio.org/about/>. (accessed: 01.10.2022).
- [5] Frank E. Sloan. *Module 11—Microwave Principles*. US NAVAL EDUCATION, TRAINING PROFESSIONAL DEVELOPMENT y TECHNOLOGY CENTER, 2021, págs. 15-16. ISBN: 978-94-6366-402-8. URL: <http://www.tscm.com/NEETS-v11-Microwave.pdf>.
- [6] Junzi Sun. *The 1090 Megahertz Riddle*. TU Delft OPEN Publishing, 2021, págs. 15-16. ISBN: 978-94-6366-402-8. URL: https://mode-s.org/decode/book-the_1090mhz_riddle-junzi_sun.pdf.

Capítulo B

Plan de proyecto

Introducción:

Título propuesto: Receptor de telemetría aeronáutica basado en Radio Definida por Software

Estudiantes: Nivio Guillán, Lautaro. Legajo 11449
Willig, Mauricio. Legajo 11664

Directores: Dr. Ing. Alejandro Uriz
Ing. Juan Alberto Etcheverry

Lugar de desarrollo: Laboratorio de Comunicaciones - FI UNMDP

El siguiente documento correspondiente al Plan de Proyecto se compone de las siguientes partes:

- El cronograma propuesto.
- Estimación de riesgos, junto con un plan de respuesta para resolver rápidamente los inconvenientes que puedan surgir.

La finalidad de este documento es complementar el plan de requerimientos, que junto con el estudio preliminar cumplen los requisitos necesarios para presentar formalmente el plan de trabajo final en la Secretaría del Departamento de Electrónica y Computación de Facultad de Ingeniería.

Cronograma:

Una vez que comenzó la cursada del Seminario de Taller para el Diseño de Soluciones Tecnológicas, se realizó la planificación de tareas y estimación de tiempos de manera formal mediante un diagrama de Gantt.

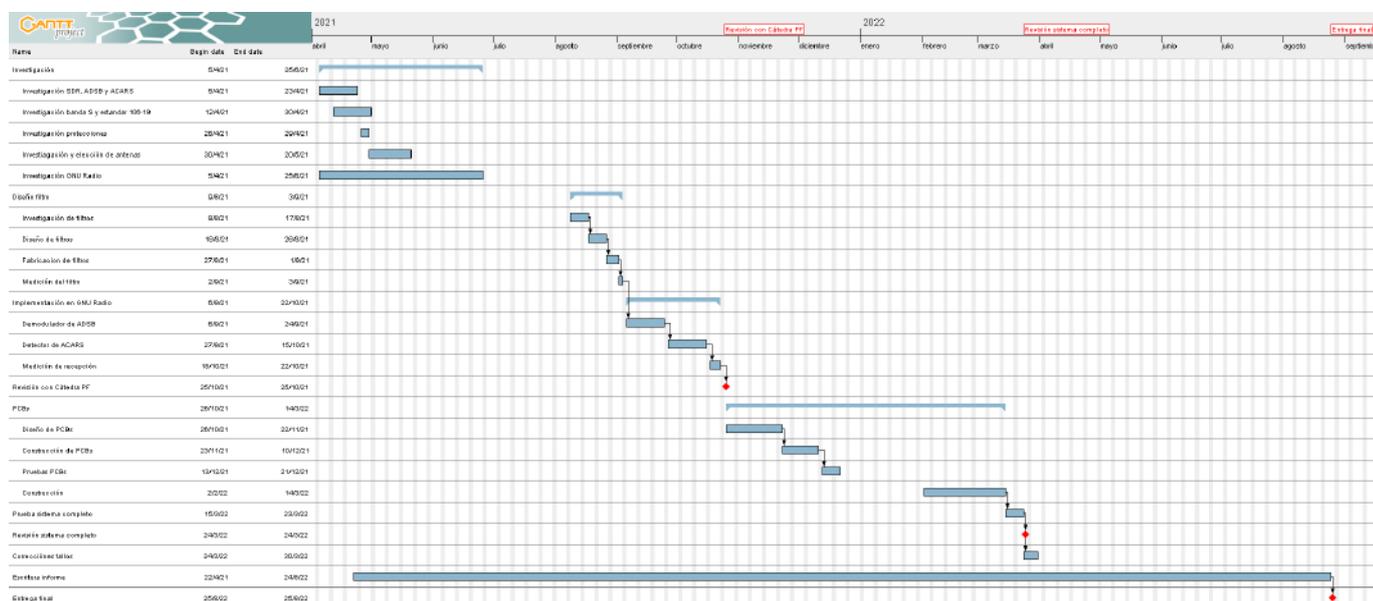
El diagrama será actualizado en caso de presentar inconvenientes, pero está destinado un intervalo de tiempo al final de proyecto para solucionar cualquier tipo de problema que se presente.

Se planea en rojo la revisión en cada etapa crucial del proyecto con los directores en el Laboratorio de Comunicaciones y con los docentes de la Cátedra de Trabajo Final oportunamente.

En cuanto a la división del trabajo, ambos autores del proyecto final cumplimos las mismas tareas al mismo tiempo. La decisión de realizar las tareas a la par se tomó como consecuencia de que este proyecto involucra el aprendizaje de técnicas de diseño de PCB y conocimiento en lenguajes de programación, los cuales no son adquiridos en profundidad durante el cursado de la carrera de Ingeniería Electrónica. Es por ello, que dada la complejidad de dichos temas, se planteó un plan de trabajo en conjunto con el fin de reducir la dificultad a la hora de adquirir nuevos conocimientos.

En caso de posibles cambios en cronograma se plantea verificar semanalmente el progreso del proyecto e ir actualizando y modificando como sea necesario el cronograma. Se plantea el uso de un registro en forma de foto en caso de que se altere el cronograma.

Al finalizar la etapa de revisión se contempló un periodo de aproximadamente 2 meses para corregir cualquier desperfecto o etapa que no funcione correctamente. En paralelo en ese periodo se continuará con la etapa final de la escritura del informe.



Riesgos:

Riesgo	Probabilidad de aparición	Severidad	Plan de respuesta
Restricción de vuelos en Gral Pueyrredon	Baja	Alta	Movilidad geográfica del sistema, en zonas con rutas aéreas activas.
Incremento en las restricciones por COVID-19	Media	Media	Plantear un modo de trabajo remoto con visitas puntuales al laboratorio de acuerdo al protocolo vigente.
Demora en la entrega del PCB	Media	Baja	Se plantea la impresión del PCB con proveedores Nacionales, en lugar de Internacionales (China).
Interferencia entre cableado de alimentación y datos.	Media	Alta	Aislar de forma física los cables por distintos conductos.

Capítulo C

Especificación de requerimientos

C.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
21 de mayo de 2021	1.0	Versión inicial	Nivio Guillán, Lautaro Willig, Mauricio
16 de junio de 2021	1.1	Aclaraciones	Nivio Guillán, Lautaro Willig, Mauricio
12 de Agosto de 2021	1.2	Correcciones	Nivio Guillán, Lautaro Willig, Mauricio
18 de Agosto de 2021	1.3	Modificaciones	Nivio Guillán, Lautaro Willig, Mauricio
3 de Octubre de 2021	1.4	Modificaciones ACARS	Nivio Guillán, Lautaro Willig, Mauricio
1 de Diciembre de 2021	1.5	Modificación diagrama en bloque	Nivio Guillán, Lautaro Willig, Mauricio
22 de Febrero de 2022	1.6	Correcciones	Nivio Guillán, Lautaro Willig, Mauricio

C.2. Introducción

C.2.1. Propósito

Este documento tiene como finalidad definir y describir los requerimientos funcionales y no funcionales del Receptor de telemetría aeronáutica basado en Radio Definida por Software. Junto con el plan de proyecto y el informe final conforman una parte de la documentación requerida para lograr la aprobación del trabajo final. Está dirigido a la cátedra de Trabajo Final. Ing Electrónica (4D0) / Ing Computación (4F0) y a todos los involucrados en el desarrollo del proyecto.

C.2.2. Alcance

En el marco de este trabajo final de grado se plantea diseñar y construir un sistema receptor de telemetría aeronáutica basado en Radio Definida por Software para una estación de tierra.

C.2.3. Personal involucrado

Nombre	Lautaro Nivio Guillán
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	lautaroniviog@gmail.com

Nombre	Mauricio Willig
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	mauriwilly@gmail.com

Nombre	Alejandro Uriz
Rol	Director
Categoría profesional	Investigador Categoría V
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	ajuriz@fi.mdp.edu.ar

Nombre	Juan Alberto Etcheverry
Rol	Codirector
Categoría profesional	Ingeniero Electrónico
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	jaetcheverry@fi.mdp.edu.ar

C.2.4. Definiciones, acrónimos y abreviaturas

Acrónimo	Significado
ADS-B	Automatic Dependent Surveillance Broadcast
ENACOM	Ente Nacional de Comunicaciones
HTTP	Hypertext Transfer Protocol
LAN	Amplificador Lineal de bajo ruido
RF	Radiofrecuencia
SDR	Software Defined Radio
TCP	Transmission Control Protocol
TRL	Technology Readiness Level
ACARS	Aircraft Communication Addressing and Reporting System

C.2.5. Referencias

Título del documento	Referencia
ABCs of SDR: Why your next radio will be software defined	Amer Radio Relay League
Baseband Analog Circuits for Software Defined Radio	Vito Giannini - Jan Craninckx - Andrea Baschiroto
Building a Software Defined Radio	DigiKey
Software defined radio for engineers	Travis Collins - Robin Getz - Alexander Wyglinski
Telemetry Standards	IRIG STANDAR 106-19
THE 1090 MEGAHERTZ RIDDLE	Junzi Sun
GNU Radio Wiki	GNU Radio

C.2.6. Visión general del documento

El documento consta de cuatro secciones, y cada una posee subsecciones en las cuales se detalla el contenido.

La primera se utiliza para registrar las versiones, qué fue modificado y quién realizó dicho cambio.

En la segunda sección se detalla una introducción y proporciona una visión general de los recursos del sistema.

En la tercera se realiza una descripción general del sistema sin incluir detalles de forma exhaustiva. Su objetivo es dar a conocer las principales funciones que éste debe realizar, los datos asociados, factores, restricciones, supuestos y dependencias que afectan al desarrollo.

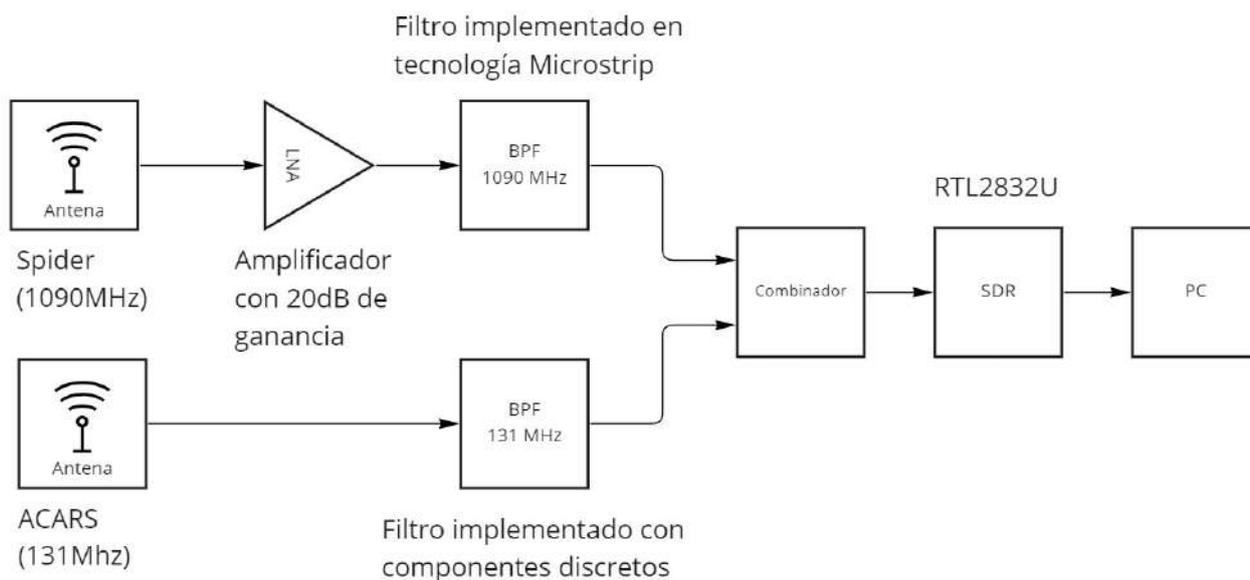
En la cuarta sección se detalla los requisitos que debe satisfacer el sistema.

C.3. Descripción general

C.3.1. Perspectiva del producto

El sistema debe ser diseñado para la recepción de señales de telemetría aeronáutica. Se plantea que el mismo sea capaz de recibir información en el estándar ADS-B y ACARS. La arquitectura sobre la que se debe implementar el desarrollo está basada en un sistema RTL-SDR, una radio definida por software o SDR. Este nuevo paradigma permite que un sistema sea capaz de operar como transmisor y/o receptor en diversas configuraciones, dependiendo de su firmware, el cual puede ser fácilmente reprogramado.

Se presenta un diagrama en bloques de alto nivel:



C.3.2. Funcionalidad del producto

El sistema que se detalla a continuación debe poder recepcionar, decodificar y almacenar paquetes del protocolo ADS-B. Además, debe contar con la capacidad de detectar mensajes del protocolo ACARS en la banda de 131MHz.

C.3.3. Características de los usuarios

Los usuarios pueden ser estudiantes que quieran trabajar para mejorar el sistema o radioaficionados. Se plantea como usuarios finales a:

- Operadores de la Estación Satelital de Mar Chiquita
- Investigadores del Laboratorio de Comunicaciones de la UNMDP.
- Radioaficionados.
- Estudiantes de telecomunicaciones.
- Aficionados a la aviación.
- Operadores de comunicaciones de las fuerzas armadas.

C.3.4. Restricciones

Restricción	Aplica	Explicación
Regulación	SI	Es necesario regirse por leyes y normas emitidas por el ENACOM y estándares IRIG.
Limitaciones de hardware	SI	Se conocen limitaciones propias del SDR, como son la velocidad de procesamiento, frecuencia de muestreo y sensibilidad del instrumento.
Interfaces con otras aplicaciones	SI	Requiere interactuar con un servidor WEB para ayudar a la visualización de los datos.
Operación en paralelo	NO	No requiere operaciones en paralelo.
Lenguaje alto nivel	SI	Utiliza Python y C++, dentro del entorno de desarrollo de GNU Radio Companion.
Auditoría	NO	No requiere ser auditable.
Protocolos	SI	Se requiere el manejo de protocolos TCP y HTTP.
Seguridad	NO	No requiere validación de usuario.

C.3.5. Suposiciones y dependencias

El sistema debe contar con una antena que es parte de la etapa de recepción, la cual debe ser montada a la intemperie y en una posición elevada lejos de objetos que puedan generar interferencias, para mejorar la línea de visión con los dispositivos que ofician de transmisores. Dependiendo del lugar de emplazamiento de la misma, se deben tener en cuenta consideraciones en cuanto a protecciones contra condiciones climáticas severas que pueden afectar su integridad.

C.4. Requisitos específicos

C.4.1. Requisitos de las interfaces

C.4.1.1. Interfaz de usuario

La interfaz de usuario constará de dos alternativas. La primera, posibilitará el uso sistema mediante software GNU Radio Companion, el permitirá ajustes en ciertos parámetros del funcionamiento. La segunda opción, utilizará un servidor WEB, el cual podrá ser accedido mediante un navegador de internet. Este ultimo solo oficiará como una alternativa de visualización de los datos obtenidos, sin posibilidad de ejercer cambios sobre los parametros del sistema.

Además se contará con la posibilidad de utilizar la consola del sistema operativo si el usuario así lo desea.

C.4.1.2. Interfaz de hardware

Será necesario disponer de una computadora que cuente con:

- Adaptadores de Red.
- Procesador 2GHz o superior.
- Memoria mínima de 5GB.
- Puerto USB 2.0 o superior.
- Mouse.
- Teclado.

C.4.1.3. Interfaz de software

- Sistema operativo: Linux Ubuntu 18.04
- Software: GNU Radio Companion 3.7.11
- Explorador: Mozilla Firefox, Google Chrome u Opera.

C.4.1.4. Interfaz de comunicación

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos USB 2.0 y protocolo TCP/IP para la funcionalidad del servidor web.

C.4.2. Requerimientos funcionales

Identificación de requerimiento	RF01
Nombre del requerimiento	Acondicionamiento de la señal de entrada
Descripción del requerimiento	El sistema debe ser capaz de recibir la señal de RF enviada por distintas aeronaves. Se pretende cumplir con dicho requerimiento, mediante el diseño de amplificadores y filtros de entrada adecuados. Se plantea trabajar con señales de 1GHz y 130MHz.
Prioridad del requerimiento	Alta

Identificación de requerimiento	RF02
Nombre del requerimiento	Identificación, decodificación y análisis de paquetes ADS-B
Descripción del requerimiento	Debe ser capaz de recibir los mensajes enviados por aeronaves con protocolo ADS-B. Se deberá identificar el tipo de mensaje perteneciente al protocolo, ejecutar código corrección de errores e interpretar los datos obtenidos.
Prioridad del requerimiento	Alta

Identificación de requerimiento	RF03
Nombre del requerimiento	Identificación, decodificación y análisis de paquetes ACARS
Descripción del requerimiento	Debe ser capaz de recibir los mensajes enviados por aeronaves con protocolo ADS-B. Se deberá identificar el tipo de mensaje perteneciente al protocolo, ejecutar código corrección de errores y exponer los datos obtenidos.
Prioridad del requerimiento	Alta

C.4.3. Requerimientos No Funcionales

Identificación de requerimiento	RNF01
Nombre del requerimiento	Protecciones meteorológicas
Descripción del requerimiento	Acondicionar el sistema correctamente para soportar inclemencias climáticas como lluvias, vientos y descargas eléctricas.
Prioridad del requerimiento	Media

Identificación de requerimiento	RNF02
Nombre del requerimiento	Protecciones eléctricas
Descripción del requerimiento	Acondicionar el sistema con protecciones eléctricas.
Prioridad del requerimiento	Alta

C.4.4. Requerimientos de Rendimiento

Identificación de requerimiento	RR01
Nombre del requerimiento	Relación señal a ruido
Descripción del requerimiento	Reducir el nivel de ruido con el uso de filtros, para obtener menores errores en la decodificación de paquetes ADS-B. Se espera una relación señal a ruido de por lo menos 15dB.
Prioridad del requerimiento	Alta

Identificación de requerimiento	RR02
Nombre del requerimiento	Rango de recepción
Descripción del requerimiento	Tener un rango de recepción considerable en comparación a equipos comerciales, mayor a 100 KM para la recepción de paquetes ADS-B y ACARS. Se puede ajustar este rango utilizando distintos tipos de antena y amplificadores.
Prioridad del requerimiento	Alta

C.4.5. Restricciones de diseño

C.4.5.1. Cumplimiento de Standards

Se debe cumplir con el IRIG STANDARD 106-19: TELEMETRY STANDARDS en al menos 2 de sus apartados, en las bandas de frecuencia anteriormente propuestas.

C.4.6. Atributos del sistema

C.4.6.1. Fiabilidad

Se plantea lograr un grado de avance TRL nivel 4, el cual consiste en tener: Validación de componentes y/o placas de prueba en entornos de laboratorio. Se pretende alcanzar un sistema confiable dentro de ese nivel.

C.4.6.2. Disponibilidad

Debe estar disponible las 24 horas del día, los 7 días de la semana.

C.4.6.3. Seguridad

Para el acceso WEB, no hay requisitos de seguridad, el acceso es libre, cualquier usuario que posea la dirección del mismo, está autorizado a usarlo. En tanto el uso mediante el software GNU Radio Companion, recae la totalidad de la seguridad en el control de accesos y usuarios propio del sistema operativo Linux.

C.4.6.4. Mantenibilidad

En caso de ser necesaria una actualización de software se puede resolver fácilmente cargando el nuevo código en el GNU Radio, a través de la computadora en la cual es sistema es comandado. En caso de tener que actualizar el hardware se realiza simplemente intercambiando los componentes necesarios, para este paso no requiere de un ajuste en el software.

C.4.7. Otros Requisitos

No aplica.

C.5. Apéndice

No aplica.

Capítulo D

Especificación Funcional

D.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
14 de Septiembre de 2021	1.0	Versión inicial	Nivio Guillán, Lautaro Willig, Mauricio
3 de Octubre de 2021	1.1	Modificaciones ACARS	Nivio Guillán, Lautaro Willig, Mauricio

D.2. Introducción

Este documento corresponde a la Especificación Funcional del Receptor de telemetría aeronáutica basado en Radio Definida por Software. Esta especificación se ha estructurado basándose en la información mencionada en el documento de Especificación de Requerimientos.

D.2.1. Propósito

El presente documento tiene como propósito proveer información detallada de cómo funcionará el prototipo y cuáles serán sus comportamientos deseados, con base en los requerimientos anteriormente definidos en la Especificación de Requerimientos.

Está dirigido a:

- Los desarrolladores del instrumento, quienes lo construirán.
- Los directores y solicitantes del proyecto, así como a quienes lo evaluarán, con el fin de que corroboren su funcionamiento.

D.2.2. Alcance del proyecto

En el marco de este trabajo final de grado se plantea diseñar y construir un sistema receptor de datos de telemetría aeronáutica basado en Radio Definida por Software para una estación de tierra.

Deberá tener la capacidad de recibir y demodular mensajes enviados por aeronaves en el protocolo ADS-B y ACARS, en las bandas de 1090 MHz y 131 MHz respectivamente.

D.2.3. Personal involucrado

Nombre	Lautaro Nivio Guillán
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	lautaroniviog@gmail.com

Nombre	Mauricio Willig
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	mauriwilly@gmail.com

Nombre	Alejandro Uriz
Rol	Director
Categoría profesional	Investigador Categoría V
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	ajuriz@fi.mdp.edu.ar

Nombre	Juan Alberto Etcheverry
Rol	Codirector
Categoría profesional	Ingeniero Electrónico
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	jaetcheverry@fi.mdp.edu.ar

D.2.4. Definiciones, acrónimos y abreviaturas

Acrónimo	Significado
ADS-B	Automatic Dependent Surveillance Broadcast
ENACOM	Ente Nacional de Comunicaciones
HTTP	Hypertext Transfer Protocol
LAC	Laboratorio de Comunicaciones - FI UNMDP
LNA	Amplificador Lineal de bajo ruido
RF	Radiofrecuencia
SDR	Software Defined Radio
TCP	Transmission Control Protocol
TRL	Technology Readiness Level
ACARS	Aircraft Communication Addressing and Reporting System

D.2.5. Referencias

Título del documento	Referencia
ABCs of SDR: Why your next radio will be software defined	Amer Radio Relay League
Baseband Analog Circuits for Software Defined Radio	Vito Giannini - Jan Craninckx - Andrea Baschirotto
Building a Software Defined Radio	DigiKey
Software defined radio for engineers	Travis Collins - Robin Getz - Alexander Wyglinski
Telemetry Standards	IRIG STANDAR 106-19
THE 1090 MEGAHERTZ RIDDLE	Junzi Sun
GNU Radio Wiki	GNU Radio

D.3. Descripción del dispositivo

El dispositivo será capaz de recibir información de telemetría enviada por aeronaves con el protocolo ADS-B en la banda de 1090MHz y el protocolo ACARS en 131 MHz.

El dispositivo puede verse simplificado en el siguiente diagrama:

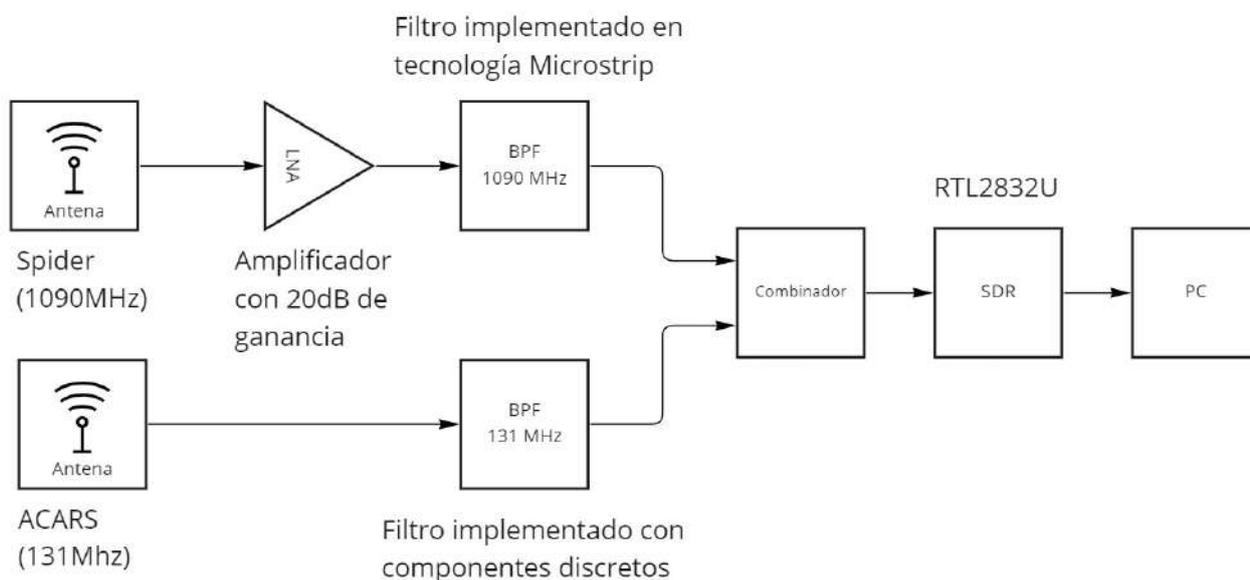


Figura D.1: Diagrama en bloques de la solución

El sistema deberá estar conectado a una PC mediante un cable USB y usando el *GNU Radio* como interfaz gráfica, mostrará en pantalla la información pertinente a la telemetría enviada por las aeronaves.

El sistema permitirá la elección entre dos modos de operación: recepción de información mediante el protocolo ACARS y recepción de información mediante el protocolo ADS-B. El usuario es quien deberá elegir entre estos modos de funcionamiento.

A continuación se describen los bloques pertinentes al diagrama mostrado anteriormente.

D.3.1. Antena

El dispositivo contará con dos antenas, estas tendrán que ser capaces de recibir información en las dos bandas de interés centradas en 1090 MHz y 131 MHz respectivamente.

Para 1090MHz se utilizará una antena spider, construida y sintonizada específicamente para la longitud de onda de la misma. Para el protocolo ACARS se utilizará una antena bi-banda, con una de sus bandas centrada en 131MHz.

D.3.2. LNA

LNA se denomina al amplificador de bajo ruido, cuya función será amplificar la señal de entrada que recibe la antena. Su principal objetivo será compensar las pérdidas producto de la etapa de filtrado posterior, en la banda de 1090MHz. Se utilizará 1 amplificador y se plantea una ganancia neta de 20dB en la banda de interés.

D.3.3. Filtro 1090MHz

La función del filtro de 1090 MHz es reducir el nivel de ruido y dejar pasar la señal relevante. Se utilizará un filtro con una pérdida de inserción lo menor posible en la banda de paso, al rededor de 6 db.

La tecnología que se utilizará será un filtro microstrip hairpin de orden 5.

D.3.4. Filtro 131MHz

La función del filtro de 131 MHz es reducir el nivel de ruido y una fuerte atenuación en frecuencias inferiores para rechazar las estaciones de FM que se encuentran cerca de la banda de interés.

La tecnología que se utilizará será un filtro con componentes discretos tales como inductores y condensadores SMD.

D.3.5. Combinador

La función de este bloque es unir el espectro resultante del filtrado en cada banda de frecuencias que se realizó anteriormente y generar un único camino el cual ingresa al SDR con todo el espectro unificado.

Se utilizará un combinador resistivo para garantizar el funcionamiento a lo largo de todo el rango de frecuencias necesario.

D.3.6. SDR

Se utilizará el dispositivo RTL-SDR, un Software Defined Radio (SDR) capaz de recibir señales de radio de 1MHz hasta 1.7GHz. En este caso se limitará su uso a recibir las señales en las frecuencias detalladas anteriormente para luego ser enviadas al ordenador para su procesamiento. Dicho dispositivo posee mezcladores, filtros, detectores que en conjunto con las etapas mencionadas colaborarán en adecuar la señal para su procesamiento.

D.3.7. PC

La PC cumple la función de procesar la información enviada por el SDR mediante el software GNU RADIO 3.7. Esta herramienta de desarrollo libre y abierta provee bloques de procesamiento de señal para implementar sistemas de radio definida por software. El sistema operativo a implementar en el ordenador será Ubuntu 18.04.

D.4. Especificaciones funcionales

D.4.1. RF01: Acondicionamiento de la señal de entrada

El sistema deberá ser capaz de recibir la señal de RF enviada por distintas aeronaves. Se pretende cumplir con dicho requerimiento mediante la etapa de filtrado y amplificación detalladas anteriormente.

D.4.2. RF02: Identificación, decodificación y análisis de paquetes ADS-B

Debe ser capaz de recibir los mensajes enviados por aeronaves con protocolo ADS-B. Se deberá identificar el tipo de mensaje perteneciente al protocolo, ejecutar código corrección de errores e interpretar los datos obtenidos

D.4.3. RF03: Identificación, decodificación y análisis de paquetes ACARS

Debe ser capaz de recibir los mensajes enviados por aeronaves con protocolo ADS-B. Se deberá identificar el tipo de mensaje perteneciente al protocolo, ejecutar código corrección de errores y exponer los datos obtenidos.

D.5. Requerimientos no funcionales

D.5.1. RNF01: Protecciones meteorológicas

El sistema deberá estar protegido de las inclemencias del tiempo, por lo que el amplificador, filtro y SDR, deberán estar protegidos en una caja estanca en el exterior para que no se filtre agua en caso de lluvias.

D.5.2. RNF02 Protecciones eléctricas

El sistema tiene que estar protegido eléctricamente, para evitar que se dañen los componentes o PC en caso de existir algún imprevisto. Para ello, el sistema cuenta internamente con un diodo ESD y será necesario la instalación de un para rayos adecuado en las proximidades de la antena.

D.6. Requerimientos de rendimiento

D.6.1. RR01: Relación señal a ruido

El sistema para detecte exitosamente un paquete de ADS-B tiene que tener por lo menos una SNR de 15dB o superior, lo cual se logra con el uso del LNA y el filtro en su respectiva banda.

D.6.2. RR02: Rango de recepción

El rango de recepción tiene que ser por lo menos de 100km, lo cual se puede lograr con el uso de una antena centrada en la banda de interés y un LNA para aumentar la señal que entra al sistema.

Capítulo E

Especificación técnica

E.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
14 de Septiembre de 2021	1.0	Versión inicial	Nivio Guillán, Lautaro Willig, Mauricio
31 de Mayo de 2022	1.1	Desarrollo apartado de Software	Nivio Guillán, Lautaro Willig, Mauricio
23 de Junio de 2022	1.2	Correcciones apartado de Hardware	Nivio Guillán, Lautaro Willig, Mauricio
16 de Julio de 2022	1.3	Agregado de especificaciones técnicas componentes	Nivio Guillán, Lautaro Willig, Mauricio
30 de octubre de 2022	1.4	Correcciones	Nivio Guillán, Lautaro Willig, Mauricio

E.2. Introducción

Este documento corresponde a la Especificación Funcional del Receptor de telemetría aeronáutica basado en Radio Definida por Software. Esta especificación se ha estructurado basándose en la información mencionada en el documento de Especificación de Requerimientos [1].

E.2.1. Propósito

El presente documento tiene como propósito proveer información detallada de cómo funcionará el prototipo y cuáles serán sus comportamientos deseados, en base a los requerimientos anteriormente definidos en la Especificación de Requerimientos.

Está dirigido a:

- Los desarrolladores del instrumento, quienes lo elaborarán.
- Los directores y solicitantes del proyecto, así como a quienes lo evaluarán, con el fin de que corroborar su funcionamiento.

E.2.2. Alcance del proyecto

En el marco de este trabajo final de grado se plantea diseñar y construir un sistema receptor de datos de telemetría aeronáutica basado en Radio Definida por Software para una estación de tierra.

Deberá tener la capacidad de recibir información de los protocolos ADS-B y ACARS en sus respectivas bandas de frecuencia.

E.2.3. Personal involucrado

Nombre	Lautaro Nivio Guillán
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	lautaroniviog@gmail.com

Nombre	Mauricio Willig
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidad	Desarrollo y diseño
Información del contacto	mauriwilly@gmail.com

Nombre	Alejandro Uriz
Rol	Director
Categoría profesional	Profesor Adjunto Exclusiva / Investigador Adjunto CONICET
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	ajuriz@fi.mdp.edu.ar

Nombre	Juan Alberto Etcheverry
Rol	Codirector
Categoría profesional	Jefe de Trabajos Prácticos Exclusiva
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información del contacto	jaetcheverry@fi.mdp.edu.ar

E.2.4. Definiciones, acrónimos y abreviaturas

Acrónimo	Significado
ADS-B	Automatic Dependent Surveillance Broadcast
ENACOM	Ente Nacional de Comunicaciones
HTTP	Hypertext Transfer Protocol
LAN	Amplificador Lineal de bajo ruido
RF	Radiofrecuencia
SDR	Software Defined Radio
TCP	Transmission Control Protocol
TRL	Technology Readiness Level
ACARS	Aircraft Communication Addressing and Reporting System
PMT	Polymorphic Types
PPM	Modulación por posición de pulso

E.2.5. Referencias

Título del documento	Autor
ABCs of SDR: Why your next radio will be software defined	Amer Radio Relay League
Baseband Analog Circuits for Software Defined Radio	Vito Giannini - Jan Craninckx - Andrea Baschirotto
Building a Software Defined Radio	DigiKey
Software defined radio for engineers	Travis Collins - Robin Getz - Alexander Wyglinski
Telemetry Standards	IRIG STANDAR 106-19
THE 1090 MEGAHERTZ RIDDLE	Junzi Sun
GNU Radio Wiki	GNU Radio

E.2.6. Visión general del documento

Esta especificación técnica estará organizada en distintas etapas de diseño especificando como se procede para la fabricación del dispositivo.

E.3. Descripción del dispositivo

El dispositivo será capaz de recibir información de telemetría enviada por aeronaves con el protocolo ADS-B en la banda de 1090MHz y el protocolo ACARS en 131,725 MHz.

El dispositivo puede verse simplificado en el siguiente diagrama:

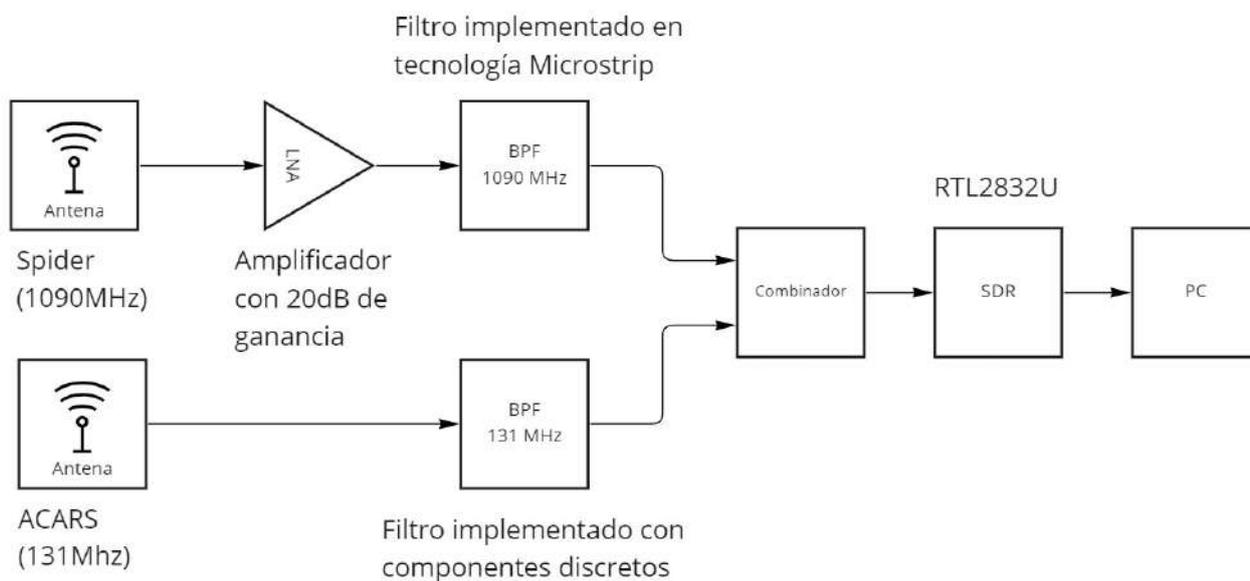


Figura E.1: Diagrama en bloque de la solución

El sistema deberá estar conectado a una PC mediante cable USB 2.0 y se ejecutará el software *GNU Radio 3.7*. Este software hará uso de distintas librerías y módulos, procesará las muestras adquiridas por el SDR y finalmente mostrará en pantalla la información recibida mediante telemetrías.

GNU Radio 3.7 permitirá la elección entre dos modos de operación: recepción de información mediante el protocolo ACARS y recepción de información mediante el protocolo ADS-B. El usuario es quien deberá elegir entre estos modos de funcionamiento. Cada uno de los modos se ejecutará mediante un Flowgraph distinto.

E.4. Hardware

A continuación se describe cada componente necesario para la construcción del dispositivo y como se debe proceder para la fabricación o instalación en caso de que sea necesario.

E.4.1. Antena Spider 1090 MHz

Se optará por el uso de una antena del tipo *Spider* con una frecuencia central próxima a 1090 MHz. Para su construcción se seguirá el siguiente modelo:

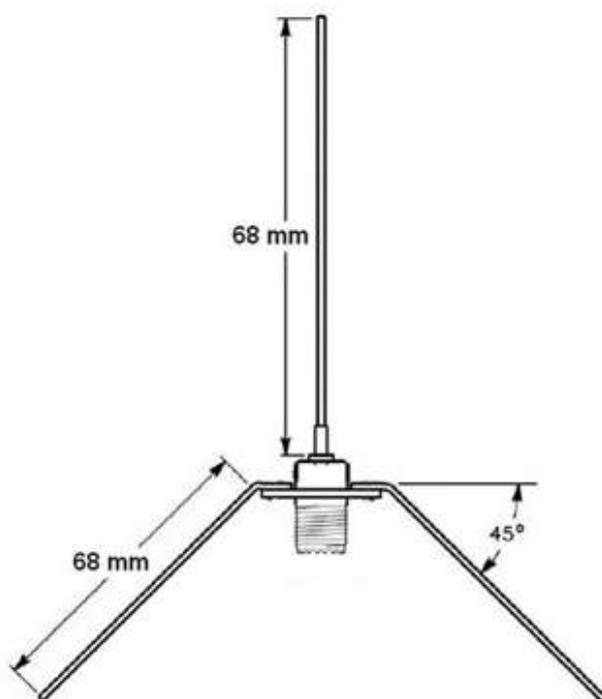


Figura E.2: Diagrama del diseño de la antena Spider para 1090 MHz

La antena Spider utiliza como base a un conector SO-239. En el conector central se soldará el conductor interno de cobre de un cable RG-8 con una longitud de 68.75mm. Esta longitud se obtiene de utilizar una antena de cuarto de onda tal que:

$$\text{Longitud de onda} = \frac{\text{Velocidad de la luz}}{\text{Frecuencia}}$$

$$\text{Longitud de onda} = \frac{299792000\text{m/s}}{1090000000\text{Hz}}$$

$$\text{Longitud de onda} = 0,275\text{m} = 275\text{mm}$$

$$\text{Longitud de cuarto de onda} = \frac{275\text{mm}}{4} = 68,75\text{mm}$$

Al igual que el conductor central, el plano de masa que constará de 8 conductores también tendrán una longitud de 68,75mm. Estos deberán estar a un ángulo de 45° por debajo de la horizontal e igualmente espaciados entre sí. Posteriormente, se realizará una calibración fina con el analizador de espectro verificando que la antena se encuentre centrada en la frecuencia deseada de 1090 MHz. Al momento de calibrar hay que tener en cuenta el efecto de borde (consecuencia de los cortes uniformes en el cobre) de los conductores, obteniéndose una longitud menor a la calculada teóricamente.

Los conductores para el plano de masa serán soldados al conector SO-239 y tendrá en cuenta la longitud desde la base.

Una consideración a tener en cuenta al momento de construir la antena es evitar que se caliente en exceso el conector SO-239 dado que puede modificarse la aislación interna y en consecuencia la impedancia perdiendo la característica de 50 Ohms.

E.4.2. Antena ACARS 131,725 Mhz

Se seleccionará una antena bi-banda comercial Nagoya NA-771 que será puesta a disposición por parte del personal del LAC, análoga a la antena de la siguiente imagen:



Figura E.3: Imagen de la antena para ACARS

La antena presenta una impedancia de 50 Ohms con un conector SMA hembra en el extremo inferior y una altura total de 39cm. Según su especificación, la misma fue diseñada para cubrir las bandas de 144 MHz y 433 MHz con una ganancia de al menos 2.15dBi. Si bien la misma no fue diseñada exactamente para la frecuencia de 131,725 MHz, es lo suficientemente cercana para funcionar correctamente. Esta última premisa deberá ser demostrada con posteriores mediciones.

Se resumen las características en la siguiente tabla:

Parámetro	Valor
Rango de frecuencias	136 - 174 MHz / 400 - 520 MHz
Ganancia	2.15 dBi @ 144 MHz / 3.0 dBi @ 430 MHz
Impedancia típica	50 Ohms
Longitud	38,15 mm

E.4.3. LNA

El Low Noise Amplifier (LNA) funcionará en un rango de 0,1 MHz a 2 GHz con una ganancia máxima de 30dB. La función de este será amplificar la señal de baja potencia que entra en la banda de 1090 MHz perteneciente al protocolo de ADSB. Se utilizará un módulo amplificador como el siguiente:

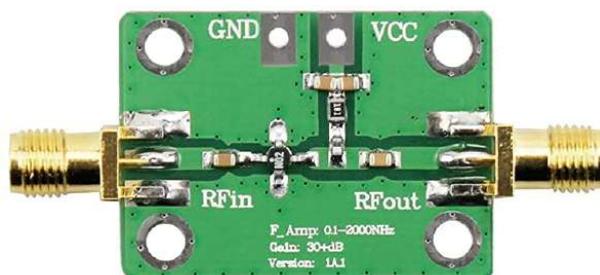


Figura E.4: Imagen del LNA

Se resumen las especificaciones en la siguiente tabla:

Parámetro	Valor
Rango de frecuencia de trabajo	0.1 - 2000 MHz
Rango tensión de alimentación	6V - 12V
Impedancia típica	50 Ohms
Dimensiones	25mm x 52mm x 5mm

El fabricante provee la siguiente curva de Ganancia vs Frecuencia en la siguiente imagen, que posteriormente será verificada con el analizador de espectros:

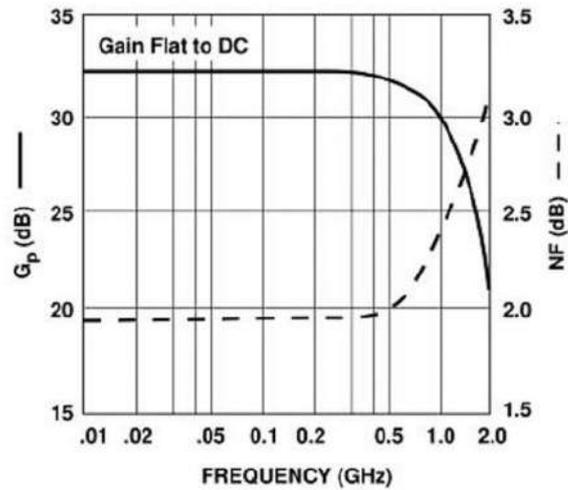


Figura E.5: Respuesta de ganancia vs frecuencia del LNA provista por el fabricante

El LNA requiere de una alimentación de 9V y 30mA, los cuales serán suministrados por un Boost Converter como el de la imagen siguiente:

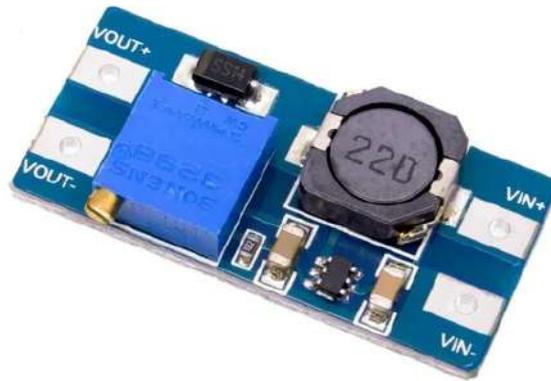


Figura E.6: Imagen del Boost converter

Se incluyen las especificaciones en la siguiente tabla:

Parámetro	Valor
Rango tensión de entrada	2V - 24V
Rango tensión de salida	5V - 28V
Corriente máxima de salida	2A
Eficiencia	93 %
Dimensiones	36mm x 17mm x 14mm

El convertidor DC-DC recibirá la alimentación de una derivación en paralelo hecha del extensor USB que conectará el SDR a la PC. Con el fin de lograr la alimentación a partir del cable USB sin perjudicar la señal que se transmite por el par de datos, se deberá quitarse el recubrimiento en un extremo y desplazar la malla metálica que cumple la función de aislar el par interno. Luego, se identificará el par de alimentación y nuevamente se deberá remover el aislante sin dañar el cobre, para luego soldar en paralelos los cables que van a energizar el convertidor DC-DC. La conexión de alimentación se hará de acuerdo al siguiente diagrama:

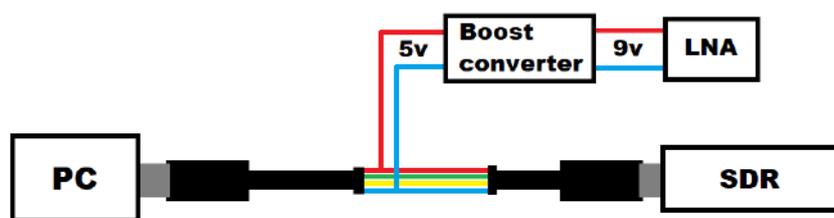


Figura E.7: Diagrama de derivación de energía para el LNA

El LNA deberá contar con un shield fabricado a partir de laminas de cobre, el cual será conectado a masa, para protegerlo de interferencias evitando que se degrade la señal que ingresa al sistema. Este tendrá forma de caja en el cual se introducirá completamente el LNA. Se ejemplifica con la siguiente imagen:



Figura E.8: Ejemplo genérico de un shield para amplificadores de RF

E.4.4. Filtro pasa banda 1090 MHz

Para el filtro pasa bandas centrado en 1090 MHz se utilizará un diseño hecho en tecnología Microstrip. El filtro será un 'Hairpin' de orden 5, centrado en 1090 MHz con un ancho de banda de 130 MHz. Para su fabricación habrá que medir previamente el ϵ_r (constante dieléctrica del sustrato) de la placa de cobre a utilizar. Se colocarán todos los valores necesarios en el Ansoft Designer Student Version, el cual se utilizara para diseñar y simular su comportamiento.

De las simulaciones se puede obtener las siguientes características:

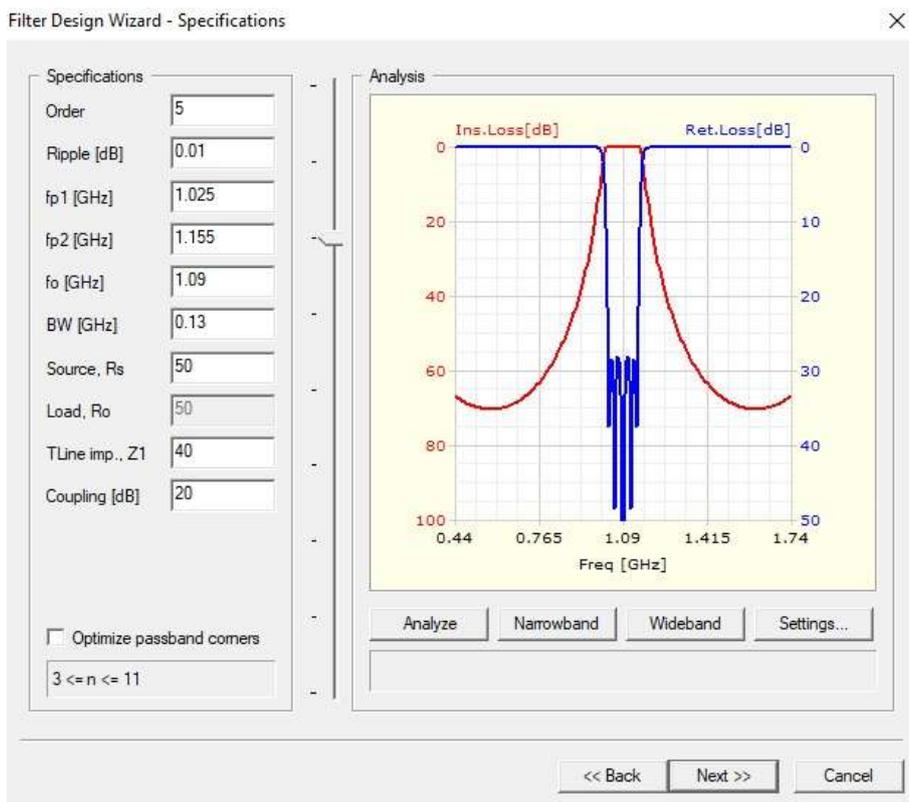


Figura E.9: Captura del diseño del filtro en Microstrip para 1090 MHz

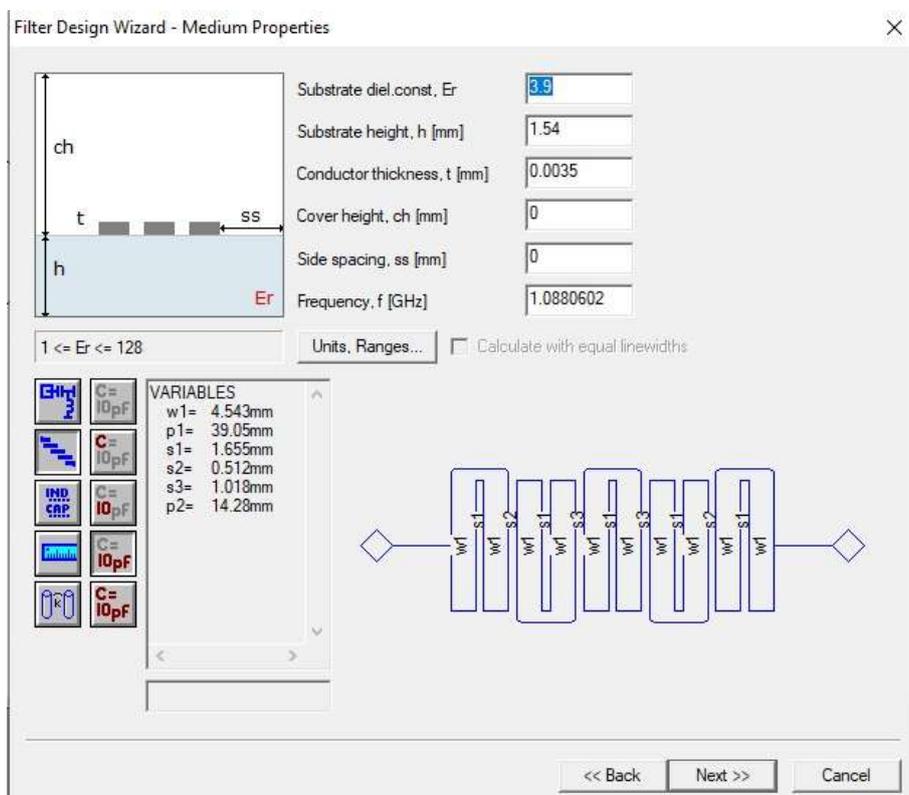


Figura E.10: Captura de la elección de parámetros para el diseño del filtro Microstrip

En el proceso de diseño se tiene que tener en cuenta que la máxima resolución con la que se puede fabricar el filtro es de 10 mils lo que es equivalente a 0,254 mm. Se recomienda que la mínima separación entre pistas, sea por lo menos el doble de la resolución, es decir, mayor a 20 mils.

En el proceso de fabricación se utilizará la *técnica de transferencia térmica* que consiste en transferir el tóner

impreso en papel al cobre, aplicando calor. Se imprime el diseño en un papel satinado mediante una impresora láser y se transfiere por calor. Una vez transferido al cobre, se pasa a la etapa de revisión con un microscopio para verificar que esté en buenas condiciones. En caso que no lo esté, se deberá rellenar los sectores faltantes de tinta con un marcador o micro fibra permanente. Finalmente, se deposita la placa en Cloruro Férrico (FeCl_3) durante 30 minutos o hasta que el exceso de cobre haya desaparecido completamente.

El filtro será similar al siguiente:



Figura E.11: Ejemplo de un filtro microstrip pasa banda de orden 4 para ADS-B

E.4.5. Filtro Pasa Banda centrado en 131,725 MHz

Para el filtro pasa banda centrado en 131,725 MHz se utilizarán componentes discretos para la fabricación de un filtro LC de orden 3. El diseño resultó de simulación con el Ansoft Designer Student Version y es el siguiente:

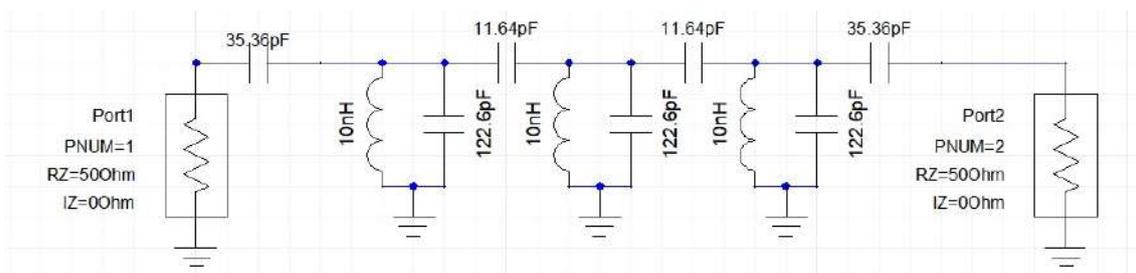


Figura E.12: Diagrama circuital para el filtro LC de 131,725 MHz

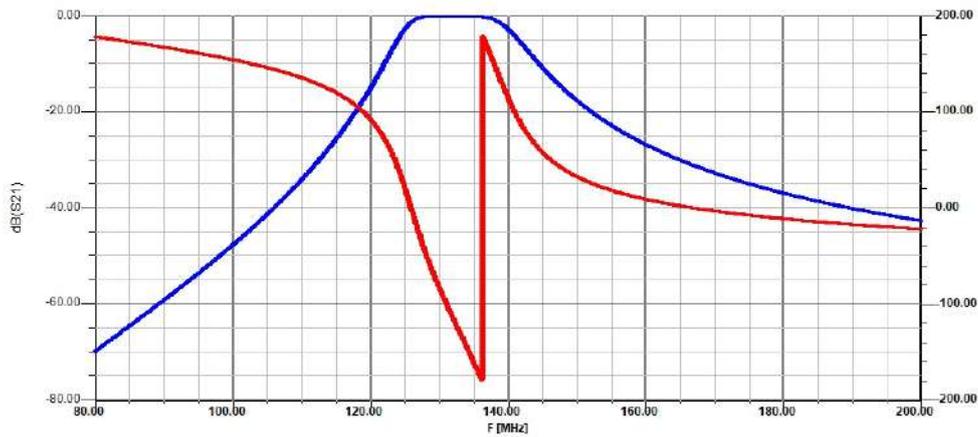


Figura E.13: Simulación de la respuesta del filtro para 131,725 MHz

Debido a que no existen algunos de los valores comerciales de capacidad requeridos por el diseño, se utilizarán los valores más próximos siendo: 12pF, 33pF y 120pF. Todos los condensadores serán del tipo SMD 0402 y los inductores del tipo SMD 0201.

El diseño del PCB será coplanar para tener un mejor control de impedancia. Se diseñará como guía de onda de 50 ohms, para evitar desadaptaciones.

El diseño final será el de la figura [E.14](#).

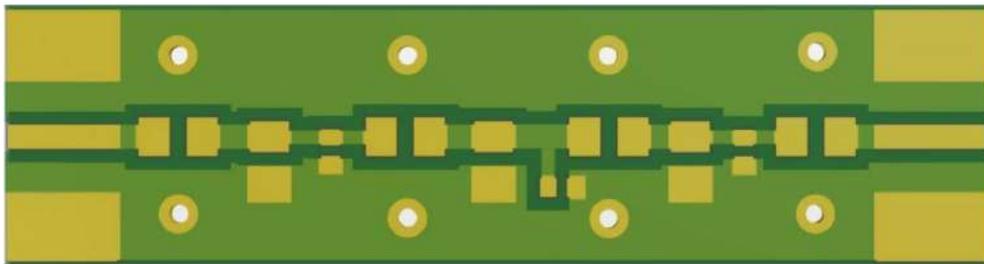


Figura E.14: Simulación del PCB del filtro co-planar para ACARS

E.4.6. Combinador

El combinador será uno resistivo puro simétrico de 50 Ohms. La configuración será del tipo 'estrella' con una resistencia de 16,66 Ohms en cada 'rama' circuital, que se logrará con el uso de 3 resistencias de 50 Ohms en paralelo apiladas o el equivalente paralelo que logre la resistencia deseada. Nuevamente, para un mejor control de impedancia se utilizará un diseño coplanar. El diseño circuital será el siguiente:

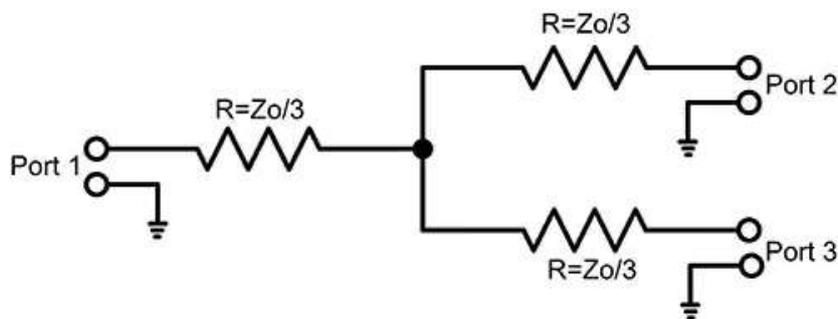


Figura E.15: Imagen de la configuración estrella del combinador

En el diseño en PCB se buscará que cada resistencia de 16,66 Ohms se encuentre a 120° entre sí. Luego, en cada uno de los extremos de las resistencias se soldará un conector SMA del tipo hembra.

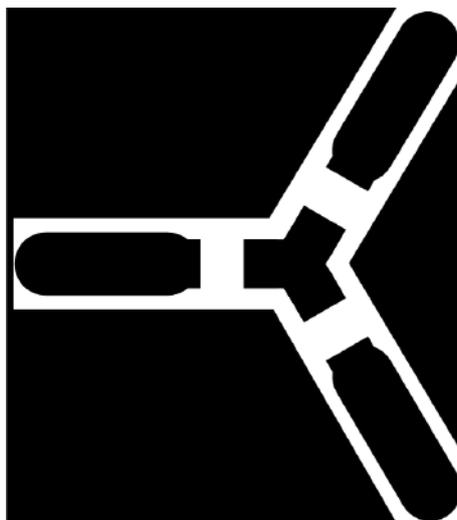


Figura E.16: Captura del diseño del PCB para el combinador

E.4.7. SDR

El SDR a utilizar será el RTL-SDR, el cual incorpora un de-modulador REALTEK RTL2832U. El integrado propietario de REALTEK permite la transferencia de muestras mediante la representación I/Q, característica de los SDR. Además, incluye un sintonizador Rafael Micro R820T que brinda una sintonización estable en el rango de frecuencias de 24 - 1766 MHz.



Figura E.17: Imagen del RTL-SDR

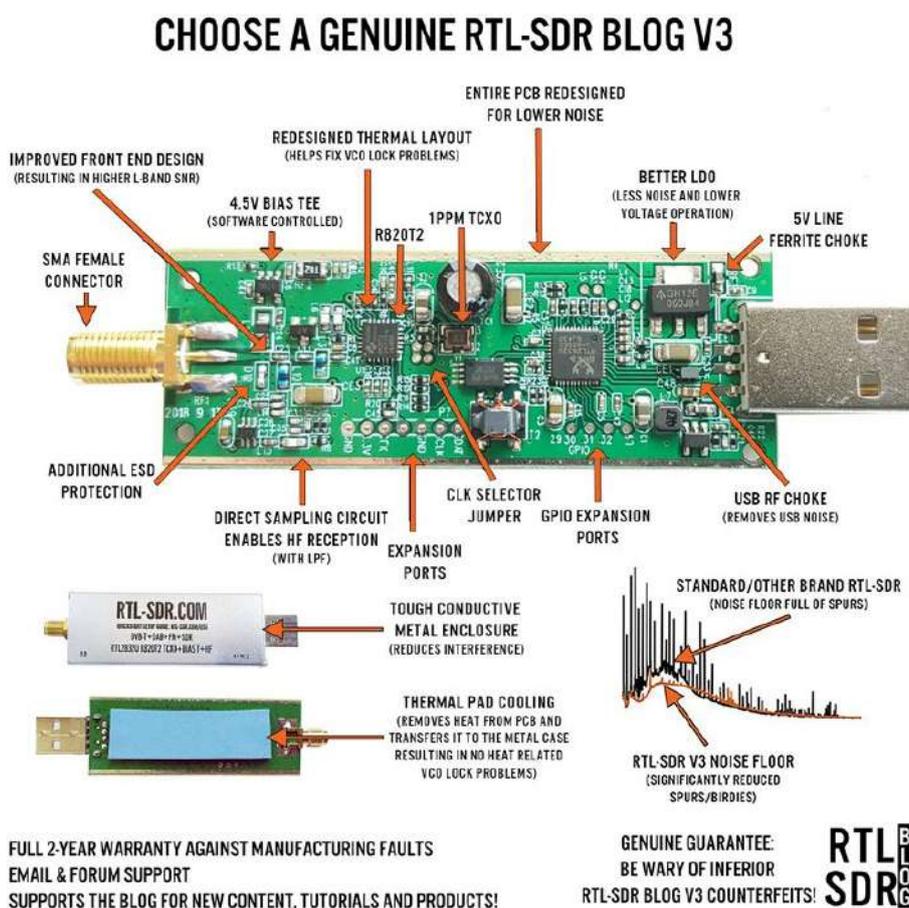


Figura E.18: Imagen de los principales componentes del RTL-SDR.

[Fuente.](#)

Las características técnicas relevantes a este proyecto, se incluyen en la siguiente tabla:

Parámetro	Valor
Frecuencia de muestreo máxima (estable)	<2.56 MS/s
ADC	8 bits (RTL2832U)
Rango de Frecuencia	24 MHz – 1766 MHz
Impedancia de entrada típica	50 Ohms
Consumo de corriente típico	270 – 280 mA
Dimensiones	69 mm x 27 mm x 13 mm

La hoja de datos puede ser consultada en esta [dirección web](#).

La conexión simplemente requiere un cable USB 2.0 en el cual un extremo se conectará al SDR y el otro a la PC. Además, para lograr la interconexión entre el SDR y el combinador, se utilizará un conector SMA-macho a SMA-macho tal como aparece en la figura [E.19](#).



Figura E.19: Imagen de un conector recto SMA-macho a SMA-macho

El fabricante anuncia bajas pérdidas y una impedancia típica de 50 Ohms.

E.4.8. PC

En complemento de los requisitos de hardware mínimos previamente mencionados en la especificación de requerimientos, también será necesario instalar el sistema operativo DragonOS en su versión LTS, en el ordenador.

DragonOS esta basado en Linux Ubuntu y cuenta con herramientas y paquetes pre-instalados relacionados a Radio Definida por Software. El mismo es de licencia de código abierto y esta permitida su distribución e instalación. Mas información sobre el proyecto DrangonOS puede ser consultada en esta [dirección web](#). Es importante mencionar, que para la utilización del servidor web el cual permite la visualización de aeronaves en una interfaz gráfica será necesario conexión a Internet.

E.4.9. Protecciones eléctricas

El RTL-SDR R820T2 RTL2832U cuenta en su entrada con un diodo BAV99 empleado como diodo ESD (electro-static discharges), el cual protege la entrada de señales que excedan la capacidad del SDR. Su funcionamiento básico consiste en recortar la señal excedente a la entrada. Dado que este diodo no es una verdadera protección ESD, en la versión RTL-SDR R820T2 RTL2832U (la que se utilizará) se incluyo un verdadero diodo ESD además del BAV99, con el fin de reforzar la protección ('Additional ESD protection' en Figura N° [E.18](#)). El fabricante no especifica que el modelo del diodo ESD que utiliza.

Adicionalmente, se incluirá en serie con la antena una protección GDT (Gas Discharge Tube) el cual protegerá al dispositivo en caso de sufrir una descarga eléctrica.



Figura E.20: Imagen de un GDT con conector SMA

El GDT está formado por electrodos envueltos en un tubo sellado de cerámica que contiene gas inerte. Bajo condiciones normales, la tensión operación no alcanza la tensión de descarga y el GDT mantiene su estado de alta impedancia. Sin embargo, cuando la sobretensión alcanza la tensión de encendido del GDT, la alta energía generada por la sobretensión provoca que el gas de relleno comience la descarga. En este momento, el GDT se convierte en cortocircuito, lo que lleva la corriente de sobrecarga a tierra y protege el equipamiento. Tan pronto como la sobretensión desaparece, el GDT regresa al estado de alto aislamiento.

La instalación consiste en conectar un extremo del conector SMA a la antena y el otro al LNA. Además cuenta con un tercer conector, en su chasis, el cual es derivado a tierra para la descarga.

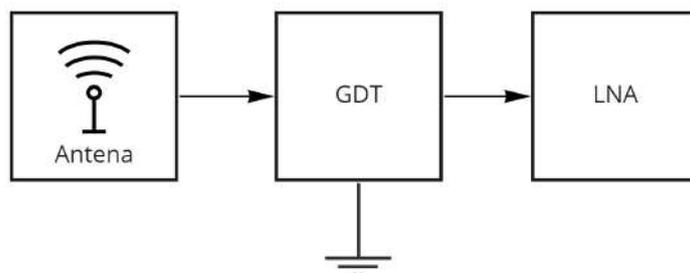


Figura E.21: Diagrama en bloques de la protección

Se utilizara un Come-buy SMA-JKY-6GN, el cual proporciona una protección con un rango de funcionamiento de DC hasta 6GHz, con una perdida de inserción menor a 0,4dB y una impedancia de 50 Ohms. Se resumen las características en la siguiente tabla:

Parámetro	Valor
Rango de frecuencia	DC - 6 GHz
Máxima tensión de operación (DC)	90 V
Pérdida de inserción	<0.4 dB
Impedancia de entrada típica	50 Ohms
Durabilidad	>500 ciclos
Dimensiones	17 mm x 12 mm x 3 mm

En la Figura E.22 se deja un ejemplo de conexión.



Figura E.22: Ejemplo de conexión GDT

E.4.10. Caja estanca

Para contener todo el dispositivo se utilizará una caja de paso estanca con cuerpo y tapa en Polipropileno blanco con retardador de llama, protección UV (solar) y mejorador de impacto.

Las dimensiones de esta serán de 420mm x 268mm x 98mm. Se deberá perforar los agujeros necesarios para la entrada de las antenas. Estos deberán contar con una separación mayor a 300mm para reducir la interferencias entre ellas. Se agregará un agujero extra al lateral derecho para la salida del extensor USB que se conectará a la PC. La caja a utilizar será la siguiente:



Figura E.23: Imagen de la caja estanca necesaria

Se incluye la ficha técnica:

Parámetro	Valor
Grado de protección	IP 65
Resistencia al calor / Deformación de bola	70 °C / <1.4 mm
Resistencia al envejecimiento	70°C / 168 hs
Rigidez Dieléctrica	5250 V / 1 min
Rango de tension	230/400 V
Dimensiones	420mm x 268mm x 98mm

E.4.11. Interconexión de componentes de RF

La conexión entre la antena de 1090 MHz y el LNA utilizará un conector recto SMA macho-macho, al igual que la conexión previamente mencionada en el apartado 4.7 entre el SDR y el combinador. Todos los componentes de RF restantes, serán interconectados mediante cables Rg58 con conectores SMA macho-macho.

Se tomará la consideración de que la longitud de los cables de RF siempre tiene que ser la menor posible para reducir las pérdidas, en especial, a los abocados a la señal de mayor frecuencia que corresponde a la de ADS-B en 1090 MHz.

E.5. Software

E.5.1. ADS-B

La señal del protocolo ADS-B posee las siguientes características:

Parámetro	Valor
Frecuencia de recepción	1090 MHz
Ancho de Banda	≈2 MHz
Modulación	PPM
Tasa de transmisión de datos	1 Mbit/s

E.5.1.1. Mensaje

El mensaje comienza con preámbulo fijo de 8 μs y continua con bloque de información de 112 μs ; cada 1 μs , un bit es transmitido. La estructura puede verse ejemplificada en la figura E.24

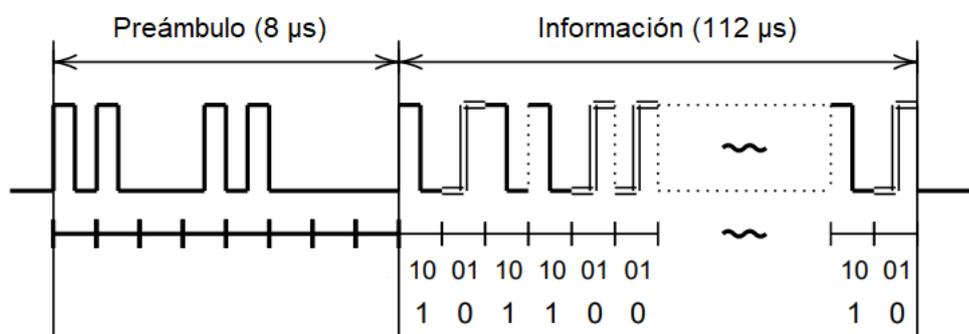


Figura E.24: Estructura de mensaje ADS-B

El patrón del preámbulo consiste en 4 pulsos con una duración de 0,5 μs por pulso y el espaciado (respecto al primer pulso) es de 1, 3,5 y 4,5 μs respectivamente.

Los datos contenidos en el bloque de Información son modulados mediante modulación por posición de pulso (PPM). El bit '1' es representado por un pulso en alto de 0,5 μs seguido de un pulso en bajo de 0,5 μs . El bit '0' es al inverso del bit '1', se representa con un pulso en bajo de 0,5 μs seguido de un pulso en alto de 0,5 μs . El campo de Información del mensaje ADS-B posee 112 bits y contiene 5 partes principales:

DF (5 bits)	CA (3 bits)	ICAO (24 bits)	ME (56 bits)	PI (24 bits)
-------------	-------------	----------------	--------------	--------------

Mediante el campo 'Donwlink Format' se identifica el tipo de mensaje de mensaje ADS-B. En el caso de aviación civil el 'Donwlink Format' toma el valor 17, el cual corresponde en binario al número 10001 para los primeros 5 bits. Entre los bits 6 a 8 se detalla 'Transponder capability' que indica la capacidad del transpondedor. A continuación, se incluye el código de indentificación de 24 bits del transpondedor conocido como ICAO. Los últimos dos segmentos pertenecen al mensaje útil 'Message, extender squitter' de 56 bits y al código de paridad de 24 bits 'Parity ID'. Se resume en la siguiente tabla:

Bit	Número de Bits	Abreviación	Nombre del campo
1-5	5	DF	Donwlink Format
6-8	3	CA	Transponder capability
9-32	24	ICAO	ICAO aircraft address
33-88	56	ME	Message, extended squitter
(33-37)	(5)	(TC)	(Type code)
89-112	24	PI	Parity/Interrogator ID

El segundo campo del mensaje ADS-B consiste de 3 bits que indican el nivel del transpondedor. El 'Capacity Value' está comprendido en la escala decimal del 1 al 7 y su función es definir el sub-tipo de mensaje a recibir.

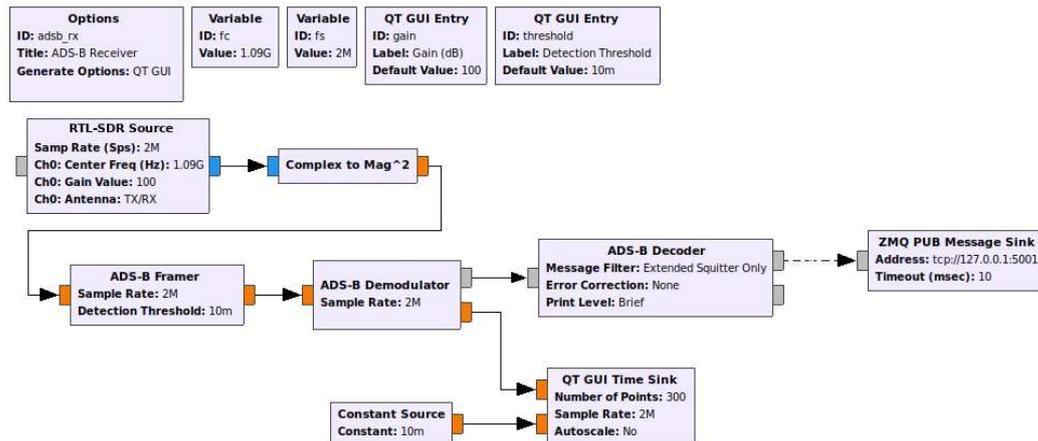
En cada uno de los mensajes ADS-B, la aeronave emisora puede ser identificada mediante código 'ICAO' asignado al transpondedor acorde a las regulaciones existentes. El identificador 'ICAO' se encuentra ubicado entre los bits 9 y 32. Un único código 'ICAO' está asignado a cada equipo emisor por lo que sirve de identificador para cada aeronave.

Es posible conocer qué tipo de información está contenida en el mensaje ADS-B observando el campo 'Type Code' que se encuentra entre los bits 33 a 37 (primeros 5 bits del segmento 'ME'). Se presenta la siguiente tabla a fin de demostrar que tipo de información posee cada uno de los 'Type Code':

E.5.1.2. Procesamiento en GNU Radio

Se utiliza el siguiente esquema de la figura E.25 para lograr el procesamiento de la señal ADS-B en GNU Radio 3.7:

Type Code	Contenido del frame
1-4	Identificación de la Aeronave
5-8	Posición en la superficie
9-18	Posición y altura (barométrica) en vuelo
19	Velocidad relativa de vuelo
20-22	Posición y altura (GNSS) en vuelo
23-27	Reservado
28	Estado de la aeronave
29	Objetivo y estado de la aeronave
31	Estado de la operación de la aeronave


 Figura E.25: *Flowgraph* recepción de ADS-B en GNU Radio 3.7

La obtención de muestras de la señal se obtiene a partir del bloque 'RTL-SDR Source', el cual actúa de interfaz entre GNU Radio y el SDR. Se ajustan parámetros como la frecuencia de sintonización, el ancho de banda y ganancia. Dichos valores son enviados al dispositivo, el cual internamente ajusta los parámetros para lograr la sintonización adecuada.

Una vez obtenidas las muestras, se realiza una conversión del mediante el bloque 'Complex to Mag'. Dicho bloque calcula la magnitud en modulo de las muestras complejas obtenidas previamente y la almacena en una variable del tipo punto flotante ('float'). Realiza el siguiente cálculo:

$$\text{Salida} = \text{Re}(in)^2 + \text{Im}(in)^2$$

donde 'Re(in)' es la parte real e 'Im(in)' es la parte imaginaria del dato de entrada y 'Salida' es el valor real flotante.

E.5.1.2.1. ADS-B Framer

El reconocimiento de la trama ADS-B es realizada por el bloque 'ADS-B Framer'. El objetivo del mismo es detectar los flancos de subida y de bajada de los pulsos obtenidos en las muestras de la señal. Luego de comparar contra el preámbulo conocido de los mensajes ADS-B, puede identificar si se esta en presencia de una trama valida de datos.

El código genera un vector de 1 y 0, cuando el valor de la muestra supera el valor de 'threshold' y cuando no es alcanzado, respectivamente. Este vector resulta en un vector de pulsos, simbolizadlos por los valores '1'. Con objetivo de encontrar las transiciones de dichos pulsos, se resta cada índice del vector con su predecesor. De esta manera se obtiene la cantidad de flancos ascendentes y descendentes y su índice en el vector de pulsos de la señal PPM. Posteriormente, compara cuantos pulsos del vector definido anteriormente coinciden con los del preámbulo de una trama válida (y conocida) de pulsos de ADS-B. Si coinciden en su totalidad, se esta en presencia de una trama de ADS-B y se calcula la relación señal a ruido. Finalmente, se etiqueta el inicio de la trama y se envía al bloque 'ADS-B Demodulator' para lograr la demodulación.

E.5.1.2.2. ADS-B Demodulator

El bloque 'ADS-B Demodulator' tiene como objetivo formar la trama binaria a partir de tener el preámbulo de la señal ADS-B identificado. Una vez demodulada la trama binaria utiliza la librería PMT propia de GNU Radio. Esta librería es utilizada para realizar la transmisión y recepción de datos entre bloques con las respectivas etiquetas para comenzar el proceso de decodificación. Un tipo especial del objeto PMT, es PDU, que consiste de un diccionario y un vector uniforme. El diccionario permite guardar pares de llave/valores que describen el tipo de dato que transportan.

Posee dos puertos de salida, la mencionada salida PDU que interconecta con el bloque 'ADS-B Decoder' y la salida alternativa del tipo flotante real. Esta última se conecta al bloque "QT GUI Time Sink" donde permite visualizar la señal binaria luego de la demodulación. En esta ventana, es posible fijar (y observar gráficamente) el valor del anteriormente mencionado 'Threshold' así como la ganancia del SDR en escala de dB.

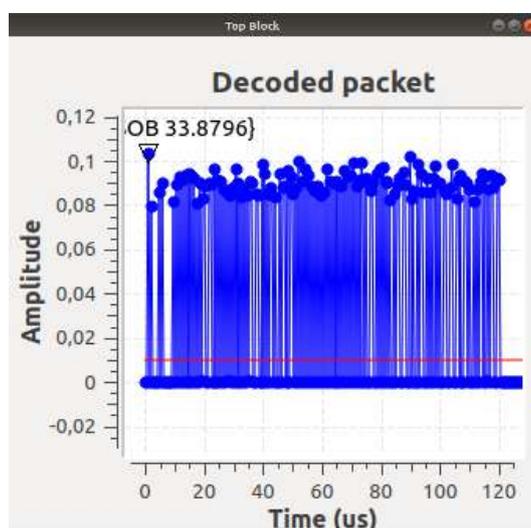


Figura E.26: QT GUI Time Sink

E.5.1.2.3. ADS-B Decoder

La etapa final del procesamiento de la señal, culmina en el bloque 'ADS-B Decoder'. El proceso de decodificación obtiene datos como latitud, longitud, velocidad, dirección, altitud y matrícula de la aeronave emisora de la trama ADS-B.

El bloque desarrollado en Python permite setear opciones en GNU Radio para decodificación como los son: 'Message Filter', 'Error Correction' y 'Print Level'. La primera opción permite elegir entre decodificar únicamente los mensajes ADS-B con la opción 'Extended Squitter Only' (DF=17 correspondiente a aviación civil) o todo tipo de mensajes mediante 'All messages'. A su vez posibilita elegir si se quiere ejecutar un código de corrección de errores luego de que la paridad de la trama ADS-B recibida sea incorrecta, como también el nivel de detalle en la decodificación con la opción 'Print level'. En su versión más completa 'Verbose' además de evidenciar las variables mencionadas en el párrafo anterior, también detalla el nivel de SNR, el resultado de la verificación del CRC e información propia del mensaje ADS-B recibido: 'Type code (TC)', 'Capability (CA)', 'Downlink Format (DF)' y 'Surveillance Status (SS)'.

El siguiente diagrama de flujo permite comprender el funcionamiento del bloque decodificador, así como los distintos módulos y funciones que permitirán la correcta decodificación de la señal binaria.

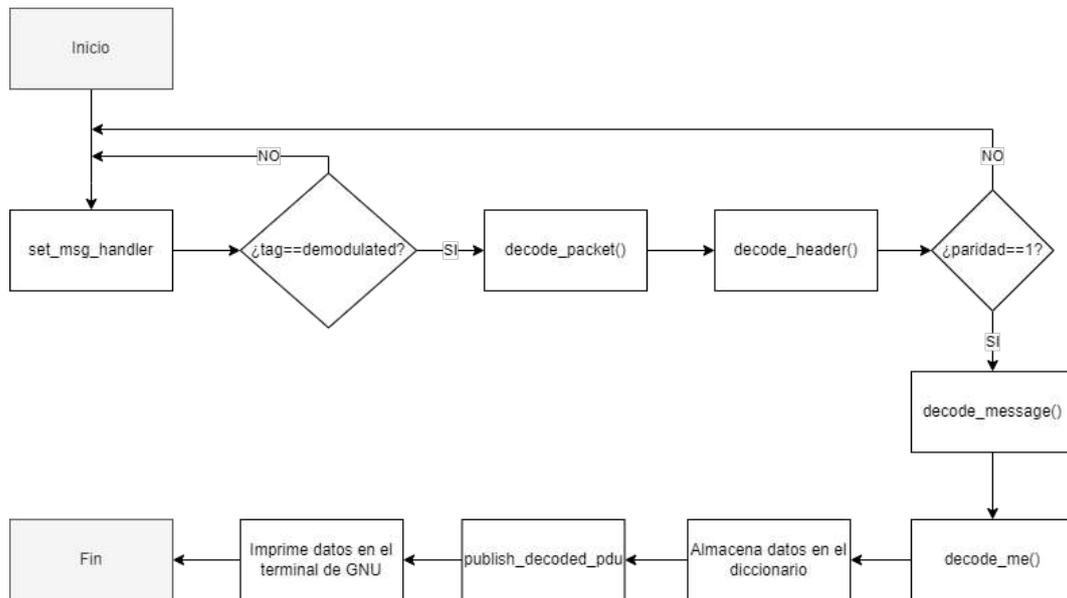


Figura E.27: Diagrama de flujo del bloque ADS-B Decoder

Se evidencian los principales aspectos del código desarrollado en Python siguiendo el diagrama de flujo y se exponen a continuación.

Inicialmente se define una clase (class) la cual cuenta con métodos (def) que desarrollan funciones a lo largo del proceso de decodificación. Previamente a la definición de los métodos, se realiza la inicialización de los valores iniciales de las variables (init). Además, se declaran las salidas y entradas, como así las variables de la interfaz gráfica de GNU Radio, las cuales vinculan el código en Python con la plataforma GNU Radio.

```

def __init__(self, msg_filter, error_corr, print_level):

    gr.sync_block.__init__(self, name="ADS-B Decoder", in_sig=None, out_sig=None)

    self.msg_filter = msg_filter
    self.error_corr = error_corr
    self.print_level = print_level

    # Initialize plane dictionary
    self.plane_dict = dict([])

    # Reset packet values
    self.reset()

    self.message_port_register_in(pmt.to_pmt("demodulated"))
    self.message_port_register_out(pmt.to_pmt("decoded"))
    self.message_port_register_out(pmt.to_pmt("unknown"))
    self.set_msg_handler(pmt.to_pmt("demodulated"), self.decode_packet)
  
```

Figura E.28: Definición clase decoder() e inicialización de variables

Se crea un diccionario y se lo asigna al objeto 'plane_dict()' en el cual se almacena a lo largo de la ejecución del bloque valores obtenidos de los mensajes decodificados. Dicha estructura permite almacenar un conjunto de valores asociados a una clave, en este caso la clave será el ICAO propio de cada aeronave. Una vez terminada la decodificación del mensaje, consultando el diccionario se puede visualizar los valores obtenidos.

La librería PMT utilizada a la salida del bloque 'ADS-B Demodulator' permite la comunicación con el bloque de decodificación, etiquetando los puertos de salida y de entrada de dichos bloques. Mediante la función 'message_port_register.in()' y 'message_port_register.out()' se asignan las etiquetas de 'demodulated', 'decoded'

o ‘uknknown’ dependiendo si los datos están a la entrada del bloque, a la salida o si no pudieron ser decodificados respectivamente.

La función ‘set_msg_handler’ es la que da inicio al proceso de decodificación, su función es llamar al módulo ‘decode_packet()’ cada vez que en la entrada de datos del bloque reconozca una secuencia de datos con el tag ‘demodulated’.

```
def decode_packet(self, pdu):
    # Reset decoder values before decoding next burst
    self.reset()

    # Grab packet PDU data
    meta = pmt.to_python(pmt.car(pdu))
    vector = pmt.to_python(pmt.cdr(pdu))
    self.timestamp = meta["timestamp"]
    self.datetime = datetime.datetime.utcfromtimestamp(self.timestamp)
    .strftime("%Y-%m-%d %H:%M:%S.%f UTC")
    self.snr = meta["snr"]
    self.bits = vector

    # Decode the header (common) part of the packet
    self.decode_header()

    parity_passed = self.check_parity()

    if parity_passed == 0:
        parity_passed = self.correct_errors()

    if parity_passed == 1:
        self.decode_header() # do this again to reparse the header should it have fixes
        # If parity check passes, then decode the message contents
        self.decode_message()
```

Figura E.29: Definición módulo ‘decode_packet()’

El módulo ‘decode_packet()’ tiene como entrada el paquete PDU de la señal previamente demodulada; la convierte a bits y la almacena en la variable ‘vector’ que luego es asignada a la variable ‘bits’. Lo mismo realiza con la SNR y la marca de tiempo, asignándolas a las variables ‘snr’ y ‘timestamp’.

En este momento la información demodulada anteriormente, se encuentra en formato de bits en la variable ‘bits’. Los módulos ‘decode_header()’, ‘decode_message()’ y ‘decode_me()’ definidos a continuación decodifican de manera segmentada el mensaje y lo someten a una verificación de errores mediante ‘check_parity()’.

```
def decode_header(self):
    # Downlink Format, 5 bits
    self.df = self.bin2dec(self.bits[0:0+5])

    if self.msg_filter == "All Messages" or (self.msg_filter == "Extended Squitter
    Only" and self.df in [17,18,19]):
        logging.info("-----")
        self.log("info", "Datetime", self.datetime)
        self.log("info", "SNR", "{:1.2f} dB".format(self.snr))
        self.log("info", "Downlink Format (DF)", self.df, DF_STR_LUT[self.df])
```

Figura E.30: Definición módulo ‘decode_header()’

El modulo ‘decode_header()’ realiza la decodificación de la cabecera de la trama ADS-B, convirtiendo los 5 primeros bits de ‘bits’ a decimal. En dichas posiciones se encuentra el ‘Downlink Format (DF)’, que para el

caso de ADS-B el DF toma el valor '17'. En consecuencia, en esta temprana etapa se pueden descartar mensajes dependiendo si el valor DF coincide con o no con el esperado, únicamente inspeccionado la cabecera de la trama.

Luego de identificar el tipo de mensaje, se somete a la verificación de paridad para garantizar la integridad del mismo. La trama ADS-B posee el campo 'Parity/Interrogator ID' ubicado en los últimos 24 bits. El módulo 'check_parity()' realiza la conversión a decimal y comprueba mediante redundancia cíclica que el mensaje no contenga errores. En caso de resultar verdadera esta premisa, devuelve un valor '1', caso contrario devuelve '0' no permitiendo continuar con la decodificación (a menos que se ejecute un código de corrección de errores).

Una vez haber verificado que el mensaje no posee errores, se prosigue con la conversión de los siguientes segmentos de la trama. El módulo 'decode_message()' obtiene la capacidad de la estación de transmisión 'Capability transponder (CA)' ubicado en los bits 5 a 8 del vector 'bits' haciendo la conversión a decimal. Con los bits ubicados en las posiciones 8 a 32 primero los convierte a decimal y luego a hexadecimal, como consecuencia obtiene la sigla de identificación de la aeronave 'ICAO'.

```
def decode_message(self):
    if self.msg_filter == "All Messages" or self.msg_filter == "Extended
    quitter Only":
        # ADS-B Extended Squitter
        if self.df == 17:
            # Capability, 3 bits
            ca = self.bin2dec(self.bits[5:5+3])
            self.log("info", "Capability (CA)", ca, subvalue=CA_STR_LUT[ca])

            # Address Announced (ICAO Address) 24 bits
            self.aa_bits = self.bits[8:8+24]
            self.aa = self.bin2dec(self.aa_bits)
            self.aa_str = "{:06x}".format(self.aa)
            self.log("info", "Address Announced (AA)", self.aa_str)
            self.log("info", "Callsign", self.plane_dict.get(self.aa_str, {})
            .get("callsign", ""))

            # All CA types contain ADS-B messages
            self.decode_me()
```

Figura E.31: Definición módulo 'decode_message()'

Como paso final en la conversión luego de haber obtenido el ICAO, se decodifican los 56 bits correspondientes al campo 'ME' donde se encuentran datos como latitud, longitud, velocidad, dirección, altitud. El módulo 'decode_me()' en una primera instancia convierte los primeros 5 bits, correspondientes a las posiciones 32 a 37, en el cual se evidencia el 'Type Code (TC)'. Contar con este valor se conoce a priori el tipo de información contenida en el mensaje.

```
# Message Extended Squitter, 56 bits
def decode_me(self):
    # Type Code, 5 bits
    tc = self.bin2dec(self.bits[32:32+5])
    self.log("info", "Type Code (TC)", tc, TC_STR_LUT[tc])
```

Figura E.32: Definición módulo 'decode_me()'

En el rango de tipo de mensajes 'TC=9' hasta 'TC=19', por ejemplo, contienen información sobre la latitud, longitud y altura.

```
### Airborne Position (Baro Altitude) ###
elif tc in range(9,19):
    # Surveillance Status, 2 bits
    ss = self.bin2dec(self.bits[37:37+2])
```

```

# NIC Supplement-B, 1 bit
nic_sb = self.bits[39]

# Altitude, 12 bits
alt_bits = self.bits[40:40+12]

# Time, 1 bit
time_bit = self.bits[52]

# CPR Odd/Even Frame Flag, 1 bit
frame_bit = self.bits[53]

# Latitude in CPR Format, 17 bits
lat_cpr = self.bin2dec(self.bits[54:54+17])

# Longitude in CPR Format, 17 bits
lon_cpr = self.bin2dec(self.bits[71:71+17])

# Update planes dictionary
self.update_plane(self.aa_str)
assert(frame_bit >= 0 and frame_bit <= 1)
self.plane_dict[self.aa_str]["cpr"][frame_bit] = (lat_cpr, lon_cpr,
int(time.time()))

(lat, lon) = self.calculate_lat_lon(self.
plane_dict[self.aa_str]["cpr"])
alt = self.decode_ac12(self.bits[40:40+12])

# TODO: Temporary hack to make sure bad lat/lons don't get published
if (lat - self.plane_dict[self.aa_str]["latitude"]) < 0.1 and
(lat - self.plane_dict[self.aa_str]["latitude"]) < 0.1:
    valid_lat_lon = True

if valid_lat_lon:
    self.publish_decoded_pdu(self.aa_str)

```

Figura E.33: Definición módulo ‘decode_me()’ para TC=9 a TC=19

Realiza la conversión de los respectivos bits y almacena los datos en el diccionario ‘plane_dict’ mediante el uso de la función ‘update_planes()’. Cada vez que se finaliza la decodificación de un paquete ADS-B dicha función almacena los valores con el ICAO asociado como clave. Se verifica que la longitud y latitud sean correctas para luego llamar al módulo ‘publish_decoded_pdu()’ que ejecuta la transmisión de los datos PDU al puerto de salida. El módulo consulta el diccionario ‘plane_dict’ y transmite los datos almacenados al puerto de salida del bloque. Nuevamente hace uso de las librerías PMT para lograr la vinculación con el bloque final del Flowgraph en GNU Radio: ‘ZMQ PUB Message Sink’

```

def publish_decoded_pdu(self, aa_str):
    decoded = self.plane_dict[aa_str].copy()
    decoded.pop("last_seen", None)
    decoded.pop("cpr", None)
    decoded["timestamp"] = self.timestamp
    decoded["datetime"] = self.datetime
    decoded["icao"] = aa_str
    decoded["df"] = self.df
    decoded["snr"] = self.snr

    meta = pmt.to_pmt(decoded)

```

```

vector = pmt.to_pmt(self.bits)
pdu = pmt.cons(meta, vector)
self.message_port_pub(pmt.to_pmt("decoded"), pdu)

```

Figura E.34: Definición módulo ‘publish_decoded_pdu()’

Por último, el módulo ‘print_planes()’ es el encargado de imprimir por el terminal de consola de GNU Radio. Itera con un ciclo del tipo ‘for’, imprimiendo por consola todos los valores que tiene asociados a cada uno de los ICAO almacenados en el diccionario ‘plane_dict’, es decir, todos los datos decodificados asociados a cada aeronave.

```

def print_planes(self):
    index = 0
    for icao in self.plane_dict:
        last_seen = datetime.datetime.utcnow().timestamp(self.timestamp)
        .strftime("%H:%M:%S")

        if self.plane_dict[icao]["callsign"] is not None:
            callsign = "{:8s}".format(self.plane_dict[icao]["callsign"])
        else:
            callsign = " "*8

        if np.isnan(self.plane_dict[icao]["altitude"]) == False:
            altitude = "{:5.0f}".format(self.plane_dict[icao]["altitude"])
        else:
            altitude = " "*5

        if np.isnan(self.plane_dict[icao]["vertical_rate"]) == False:
            vertical_rate = "{:5.0f}".format(self.plane_dict[icao]["vertical_rate"])
        else:
            vertical_rate = " "*5

        if np.isnan(self.plane_dict[icao]["speed"]) == False:
            speed = "{:5.0f}".format(self.plane_dict[icao]["speed"])
        else:
            speed = " "*5

        if np.isnan(self.plane_dict[icao]["heading"]) == False:
            heading = "{:5.0f}".format(self.plane_dict[icao]["heading"])
        else:
            heading = " "*5

        if np.isnan(self.plane_dict[icao]["latitude"]) == False:
            latitude = "{:11.7f}".format(self.plane_dict[icao]["latitude"])
        else:
            latitude = " "*11

        if np.isnan(self.plane_dict[icao]["longitude"]) == False:
            longitude = "{:11.7f}".format(self.plane_dict[icao]["longitude"])
        else:
            longitude = " "*11

```

Figura E.35: Definición módulo ‘print_planes()’

E.5.1.2.4. ZMQ PUB Message Sink

Con la inclusión de este bloque culmina el Flowgraph de GNU Radio. ‘ZMQ PUB Message Sink’ permite el envío de datos mediante protocolo TCP desde GNU Radio hacia el webserver para una visualización en tiempo

real de la posición de las aeronaves. En las opciones de configuración se declara el 'Address' que incluye la dirección de IP y el puerto en el cual el webservice recibe los datos decodificados por los anteriores bloques.

E.5.2. ACARS

Los mensajes del protocolo ACARS que se transmiten mediante VHF utilizan el rango de frecuencias de 118 MHz a 137 MHz; en Argentina y países de la región, se utiliza la frecuencia de 131,725 MHz. El protocolo se emite con una tasa de bit de 2400bps y el ancho de banda que se asigna para la transmisión es de aproximadamente 5 KHz.

ACARS adopta un esquema de modulación mediante MSK-AM. La modulación MSK (un caso particular de la modulación FSK) codifica mediante dos frecuencias distintas los estados del bit '1' o '0' de modo que las transiciones del estado del bit a otro son codificadas, en vez de codificar directamente el valor del bit. Mediante la señal en 1200 Hz se codifica que el estado, ya sea "1" o "0" debe ser cambiado respecto al anterior valor, mientras que mediante la señal de 2400 Hz que el estado del bit debe permanecer constante.

En resumen, la señal del protocolo ACARS posee las siguientes características:

Parámetro	Valor
Frecuencia de recepción	131,725 MHz (en Argentina)
Ancho de Banda	≈ 5 kHz
Modulación	MSK-AM
Tasa de transmisión de datos	2400 bit/s

E.5.2.1. Frame

Existen dos tipos de enlaces de mensajes en ACARS, enlaces de subida (uplink) donde la información tiene como origen una estación en tierra y destino la aeronave, y viceversa mediante enlaces de bajada (downlink). Un ejemplo de la cabecera del mensaje del segundo caso se expone en la figura. Cada carácter se representa mediante 8 bits.

SOH	Mode	Address	TAK	Label	DBI	STX	MSN	Flight ID	Text
1 caráct.	1 caráct.	7 caract.	1 caráct.	2 caract.	1 caráct.	1 caráct.	4 caract.	6 caract.	210 caract.

Suffix	BSC
1 caráct.	1 caráct.

- SOH: Establece el inicio de la trama del mensaje.
- Mode: Define el modo de la estación receptora en tierra, se dividen en Categoría A y B.
- Address: Aloja la identificación de la aeronave emisora del mensaje.
- TAK (Technical Acknowledgment): Notifica el estado del anterior mensaje. Mediante caracteres notifica una recepción positiva (ACK) o solicita la retransmisión (NACK).
- Label: Utilizado para describir el tipo de mensaje (uplink o downlink).
- DBI (Downlink Block Identifier): Se utiliza tanto para volver a solicitar una petición que no fue respondida o realizar una retransmisión luego de recibir un NACK.
- STX: marca el fin del preámbulo.
- MSN: Es el número de identificación del mensaje.
- Flight ID: Consiste en un campo de 6 caracteres, de los cuales 2 identifican la aerolínea y los restantes 4 el número de vuelo.
- Text: Cuerpo del mensaje, puede ser libre o pre-establecido y como máximo 210 caracteres. Junto con los campos MSN y Flight ID totalizan un mensaje de 220 caracteres.
- Suffix: determina el fin del mensaje, en caso de que combinen más de una cabeceras para transmisiones de mensajes.
- BSC: Consiste en 16 caracteres que posibilitan la detección de errores.

E.5.2.2. Procesamiento en GNU Radio

Se utiliza el siguiente esquema para lograr el procesamiento de la señal ADS-B en GNU Radio 3.7:

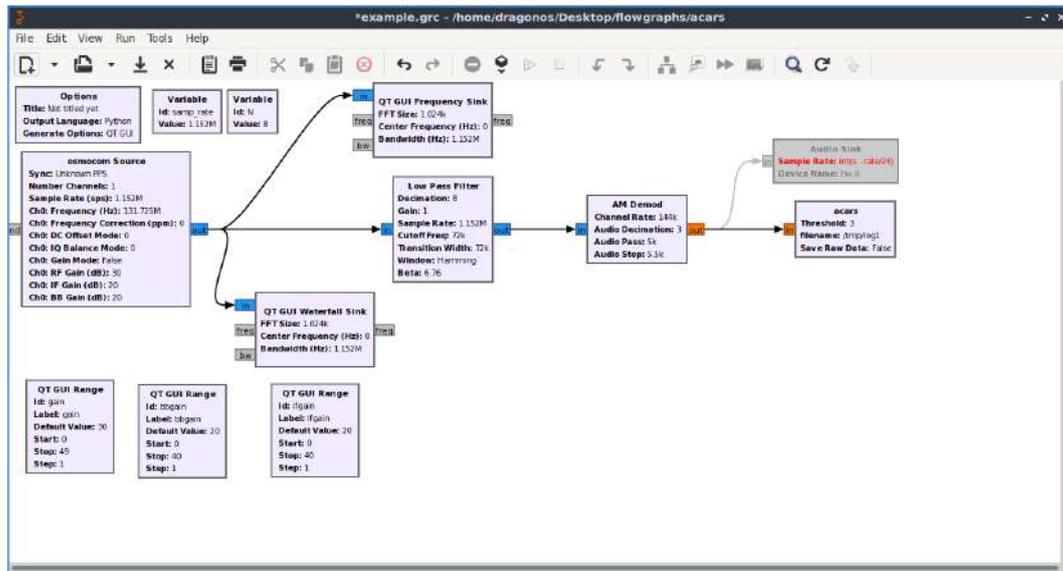


Figura E.36: *Flowgraph* recepción de ACARS en GNU Radio 3.7

‘osmocom Source’ establece la conexión del SDR a GNU Radio y logra adaptar la señal recibida por el dispositivo a partir de muestrearla a una frecuencia de 1,152 MHz. Dentro de los parámetros más distintivos de este bloque se ajusta además de la mencionada frecuencia de muestreo (Sample Rate), la frecuencia de sintonía (Ch0: Frequency Hz) que en el caso de ACARS es de 131,725 MHz, la ganancia de RF (Ch0: RF Gain) y la ganancia de frecuencia intermedia (Ch0: IF Gain).

Una vez adquirida la señal, es necesario limitar el ancho de banda mediante la utilización de un filtro paso bajos. Para ello se utiliza el bloque ‘Low pass filter’. Se definen parámetros como al frecuencia de muestreo (coincidente con la del bloque previo), la frecuencia de corte (Cutoff freq) , el ancho de banda de transición (Transition Width), la ganancia (Gain) y el tipo de ventana a usar para el muestreo (Window).

Además se fija una relación de muestreo, la cual decima la cantidad de muestras mediante un factor de 6. En el subsiguiente bloque que realiza la demodulación de la señal AM, nuevamente se decima por un factor de 4, de manera que la frecuencia de muestreo a la salida de este sea de 48 kHz de la siguiente manera:

$$\text{Frecuencia previa decodificación} = \text{Frecuencia muestreo} * \frac{1}{D_1} * \frac{1}{D_2}$$

$$\text{Frecuencia previa decodificación} = 1,152\text{MHz} * \frac{1}{6} * \frac{1}{4}$$

$$\text{Frecuencia previa decodificación} = 48\text{KHz}$$

De esta manera a la salida del bloque demodulador se tiene una señal de 48 KHz, la cual resulta compatible con el bloque “decoder” y además, con las placas de sonido comerciales, posibilitando la reproducción mediante altavoces de tonos sonoros cuando se registra actividad en la señal.

E.5.2.3. Decodificación

Se utiliza la librería gr-acars la cual es una portación de un algoritmo en GNU/Octave que permite decodificar el protocolo ACARS. Con el objetivo de ser compatible con GNU Radio, el código fue rescrito en C y a partir de el se genero el bloque “decoder”. Se sumarizan las partes mas significativas del código en GNU/Octave a fin de entender como se lleva a cabo la decodificación.

En principio, se identifica el comienzo de la trama del mensaje ACARS, el cual consta de 16 bytes que conforman una señal oscilatoria de 2400 Hz. Dicha oscilaciones, determinaran el inicio y fin de la trama adquirida.

En consecuencia que el estado de los bits de la trama, estén codificados en una señal de mitad de periodo de la señal adquirida, y el otro estado en una señal de igual periodo a la señal de audio permite identificar el bit codificado mediante propiedades de la correlación cruzada. A partir de tener valor codificados a una frecuencia de $f_m = 1200$ Hz y $2f_m = 2400$ Hz, la convolución de la muestras tomadas de la señal con la función $\sin(2\pi f_m t)$ genera un valor media nulo cuando el valor del segmento bajo decodificación no tiene el valor adecuado y $1/2$ si

el valor del bit es acertado. Análogamente, la convolución con la función $\sin(\pi f_m t)$ arroja los mismo resultados, valor nulo cuando no es el estado del bit codificado, y $1/2$ cuando es correcto.

$$-\int_0^1 \sin(2\pi t) \sin(\pi t) dt \propto \int_0^1 (\cos(3\pi t) - \cos(\pi t)) dt = \frac{\sin(3\pi) - \sin(0) - (\sin(\pi) - \sin(0))}{3\pi - \pi} = 0 \quad (\text{E.1})$$

$$-\int_0^1 \sin(2\pi t) \sin(2\pi t) dt = 1/2 \times \int_0^1 (\cos(4\pi t) - \cos(0)) dt = 1/2 \times (\sin(4\pi) - \sin(0) + 1) = 1/2 \quad (\text{E.2})$$

$$-\int_0^1 \sin(\pi t) \sin(\pi t) dt = 1/2 \times \int_0^1 (\cos(2\pi t) - \cos(0)) dt = 1/2 \times (\sin(4\pi) - \sin(0) + 1) = 1/2 \quad (\text{E.3})$$

Por lo tanto, una vez identificado el principio de la trama (a partir de la sincronización con la señal de 2400 Hz) la decodificación es simplemente el producto de la de realizar la convolución con $\sin(2\pi f_m t)$ y $\sin(\pi f_m t)$ con las muestra de la señal adquiridas. Dado que la secuencia de bit decodificadas representa las transiciones de un estado a otro y no el valor del bit en si, resta obtener la cadena de bits completa, que luego es expresada en Hexadecimal para su final identificación mediante caracteres ASCII.

Una consecuencia directa de codificar las transiciones de bits y no su estado, es que al cometerse un error en la detección convierte al mensaje en erróneo a partir de donde se comete el error. ACARS posee identificador de paridad en sus mensajes pero no cuenta con la posibilidad de corregir errores. Es por esto que durante la recepción del protocolo es probable poder identificar la matricula y el número de vuelo (primeros campos del mensaje) pero no el mensaje completo.

Capítulo F

Manual de usuario

F.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
30 de octubre de 2022	1.0	Versión inicial	Nivio Guillán, Lautaro Willig, Mauricio

F.2. Propósito

El propósito del siguiente documento es indicarle al usuario las partes del equipo, la instalación y el uso del mismo.

F.3. Audiencia

Va dirigido a todos los usuarios que vayan a hacer uso del dispositivo.

F.4. Partes del equipo

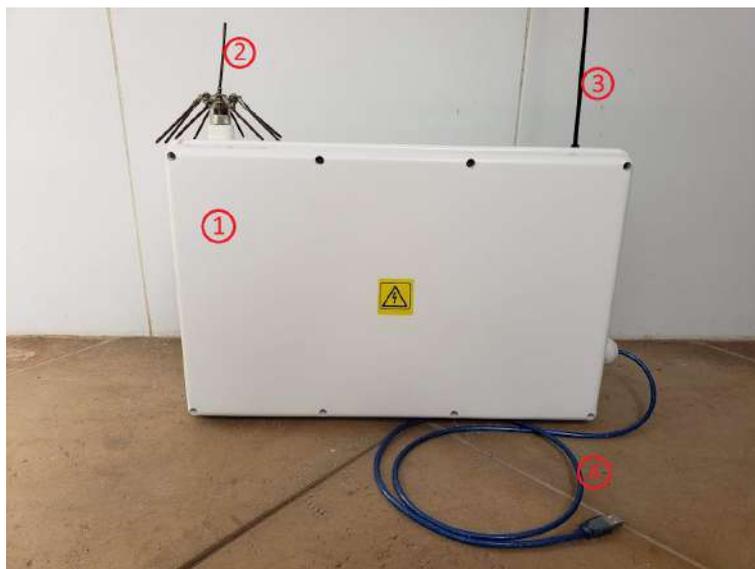


Figura F.1: Foto frontal del equipo

- 1: Caja
- 2: Antena ADS-B 1090 MHz
- 3: Antena ACARS 131.725 MHz
- 4: Cable USB 2.0

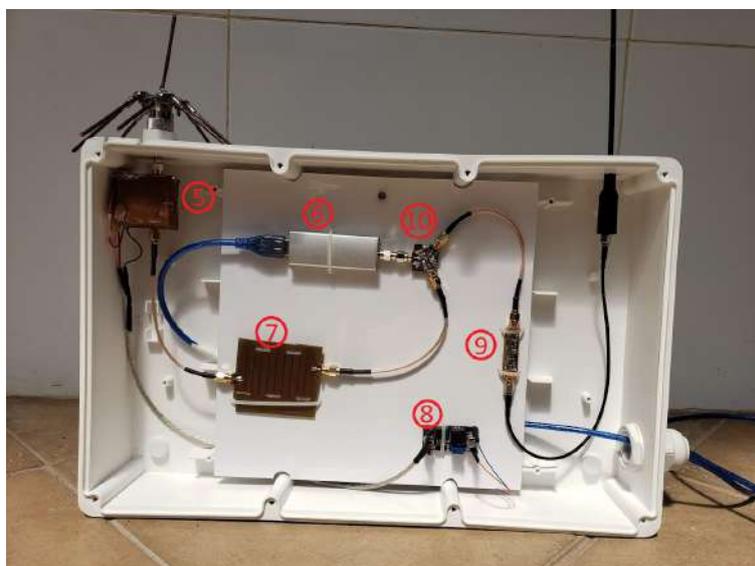


Figura F.2: Foto interna del equipo

- 5: LNA
- 6: SDR RTL
- 7: Filtro ADS-B 1090 MHz
- 8: Convertidor Boost
- 9: Filtro ACARS 131.725 MHz
- 10: Combinador

F.5. Instalación

La instalación del hardware consta únicamente de colocar el equipo en una zona despejada de obstáculos, alejado de otros equipos de radiofrecuencia y sin grandes superficies metálicas próximas a las antenas. Se aconseja buscar la locación con la mayor altura posible respecto al nivel del mar y con la mayor línea de visión hacia el cielo. Finalmente, conectar el cable USB 2.0 a una PC con el sistema operativo *DragonOS*.

La imagen de instalación del sistema *DragonOS* puede ser descargada en el siguiente [repositorio](#). Para su instalación se recomienda seguir los pasos detallados en el video tutorial creado por el autor, disponible en el siguiente [enlace](#).

F.6. ADS-B

F.6.1. Instalación de librerías

La librería que posibilita la detección de ADS-B se puede descargar en el siguiente [repositorio](#).

Los pasos a seguir para la instalación son los siguientes:

1. Abrir la terminal del sistema
2. Clonar el repositorio con el comando: `git clone https://github.com/mhostetter/gr-adsb`
3. Acceder al directorio donde se encuentran los archivos para la instalación, con el comando: `cd gr-adsb/`
4. Crear el sub-directorio *build* con el comando: `mkdir build`
5. Acceder al sub-directorio *build* con el comando: `cd build`
6. Introducir el comando: `cmake ../`
7. Introducir el comando: `make`
8. Introducir el comando: `sudo make install`
9. Introducir el comando: `sudo ldconfig`

F.7. Inicialización detección de ADS-B

1. En primera instancia se encontrará con la pantalla de inicio del DragonOS

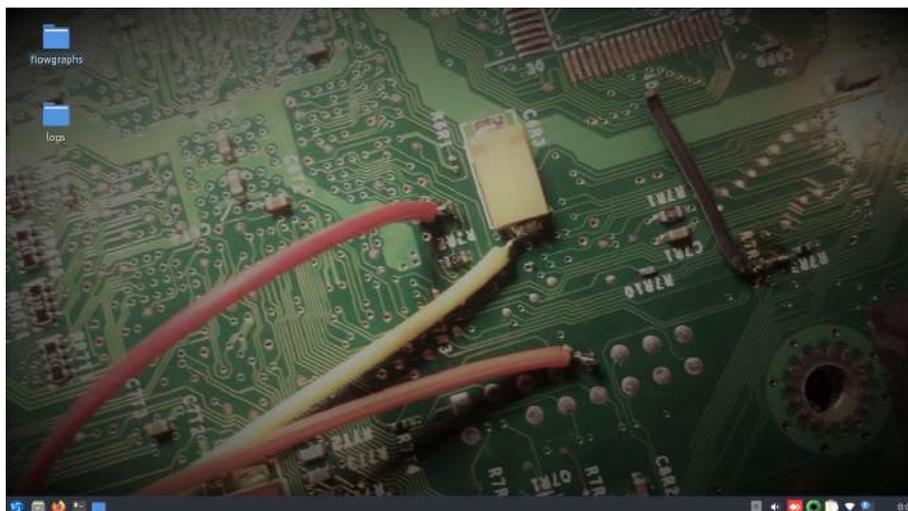


Figura F.3: Pantalla de inicio del DragonOS

2. Ir a la carpeta *adsb* que se encuentra dentro del directorio *flowgraphs* en el escritorio.

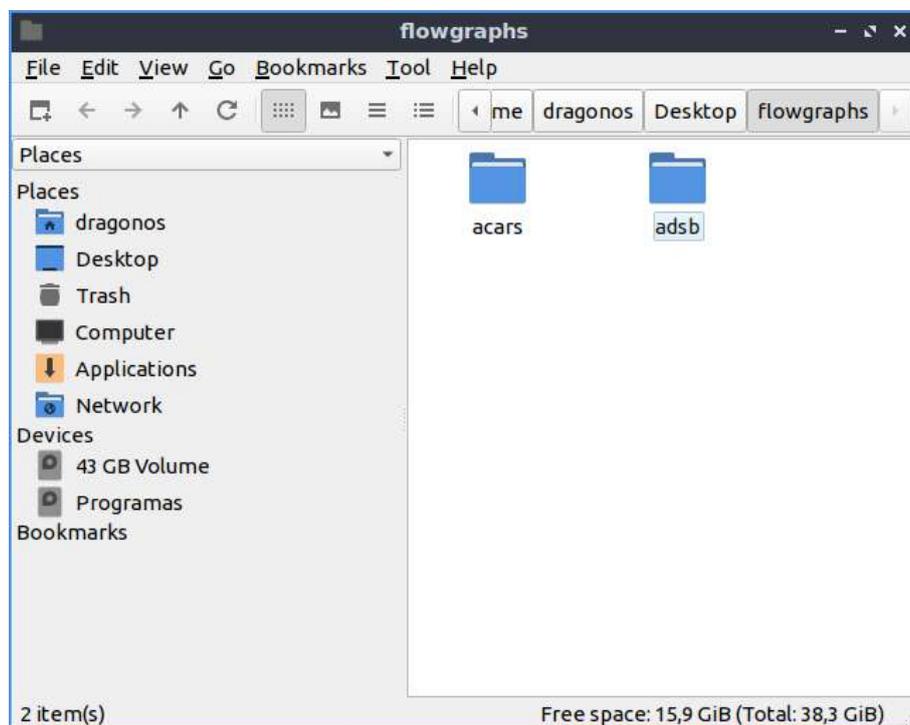


Figura F.4: Directorio flowgraphs

3. Abrir la terminal en la carpeta *adsb*

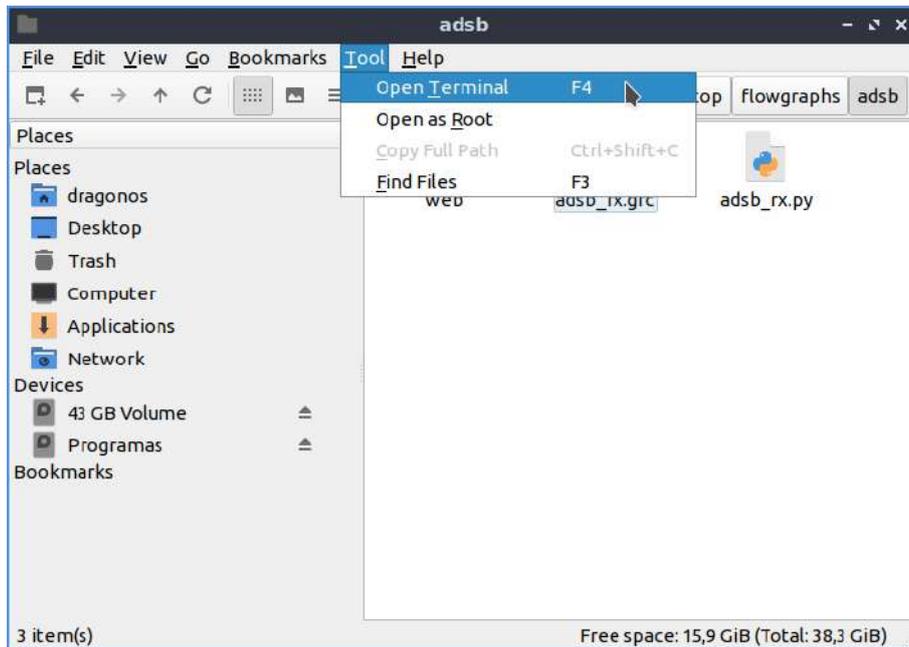


Figura F.5: Selección de terminal

4. Inicializar el programa de detección de ADS-B dentro de la terminal del sistema con el siguiente comando:
gnuradio-companion adsb_rx.grc

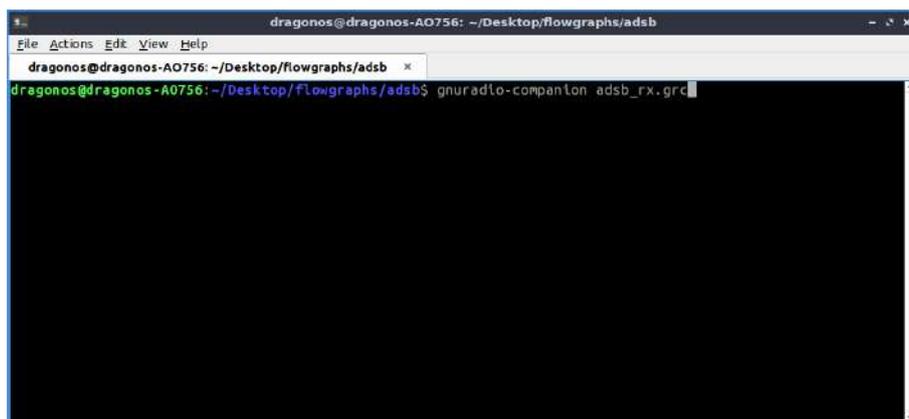


Figura F.6: Pantalla de la terminal

5. En la figura [F.18](#) se ve el GNU Radio Companion con el diagrama en bloques para la detección de ADS-B.

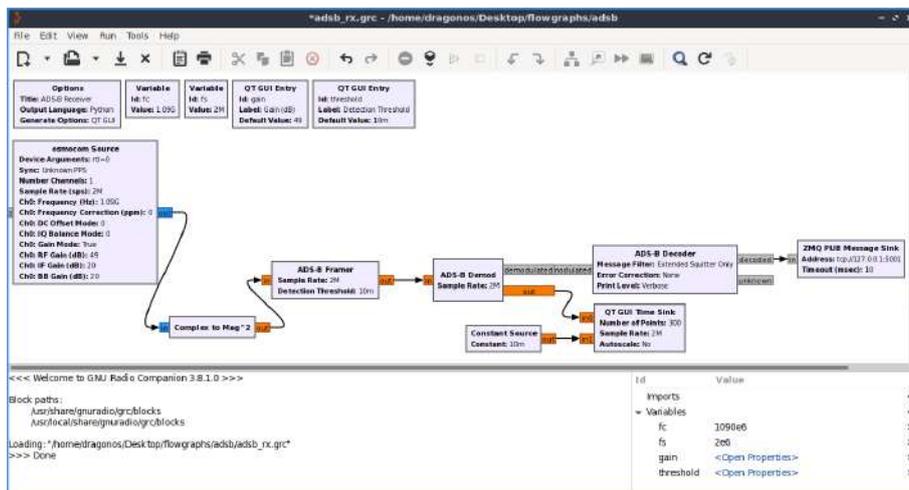


Figura F.7: Flowgraph de detección ADS-B en GNU radio

6. Para ejecutar el programa hacer click en *Execute the flow graph*.

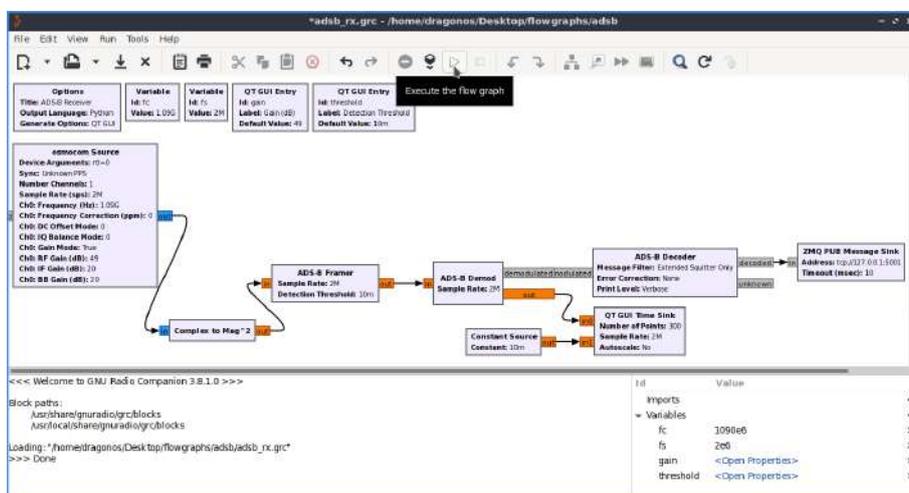


Figura F.8: Ejecución flow-graph detección ADS-B en GNU Radio

F.7.1. Ejemplo de la visualización en la consola

```
[INFO] -----  
[INFO] Datetime: 2019-07-31 00:43:30.944816 UTC  
[INFO] SNR: 19.90 dB  
[INFO] Downlink Format (DF): 0 Short Air-Air Surveillance (ACAS)  
[INFO] CRC: Passed Recognized AA from AP  
[INFO] Address Announced (AA): ac53a4  
[INFO] Callsign: EDV5271  
[INFO] Vertical Status (VS): 0 In Air  
[INFO] Reply Information (RI): 3 Reserved for ACAS  
[INFO] Altitude: 7025 ft  
[INFO] Crosslink Capability (CC): Does Support Crosslink Capability  
[INFO] -----  
[INFO] Datetime: 2019-07-31 00:43:32.114965 UTC  
[INFO] SNR: 21.85 dB  
[INFO] Downlink Format (DF): 4 Surveillance Altitude Reply  
[INFO] CRC: Passed Recognized AA from AP  
[INFO] Address Announced (AA): ac53a4  
[INFO] Callsign: EDV5271  
[INFO] Flight Status (FS): 0 No Alert, No SPI, In Air  
[INFO] Downlink Request (DR): 0 No Downlink Request  
[INFO] IIS: 0  
[INFO] IDS: 0 No Information  
[INFO] Altitude: 7075 ft  
[INFO] -----  
[INFO] Datetime: 2019-07-31 00:43:36.695273 UTC  
[INFO] SNR: 22.41 dB  
[INFO] Downlink Format (DF): 11 All-Call Reply  
[INFO] CRC: Passed  
[INFO] Capability (CA): 5 Level 2 or Above Transponder, Can Set CA 7, In Air  
[INFO] Address Announced (AA): ac53a4  
[INFO] Callsign: EDV5271  
[INFO] -----  
[INFO] Datetime: 2019-07-31 00:43:37.784807 UTC  
[INFO] SNR: 21.87 dB  
[INFO] Downlink Format (DF): 17 Extended Squitter  
[INFO] CRC: Passed  
[INFO] Capability (CA): 5 Level 2 or Above Transponder, Can Set CA 7, In Air  
[INFO] Address Announced (AA): ac53a4  
[INFO] Callsign: EDV5271  
[INFO] Type Code (TC): 19 Airborne Velocity  
[INFO] Subtype (ST): 1 Ground Velocity  
[INFO] Intent Change (IC): 1 No Change in Intent  
[INFO] Speed: 267 kt  
[INFO] Heading: 173 deg (W)  
[INFO] Climb: 2816 ft/min  
[INFO] Climb Source: 0 Geometric Source (GNSS or INS)
```

F.7.2. Inicialización del Web Server

Una vez conectado el dispositivo en una PC con el sistema operativo DragonOS es necesario iniciar el web server de la siguiente forma:

1. Abrir la terminal del sistema.

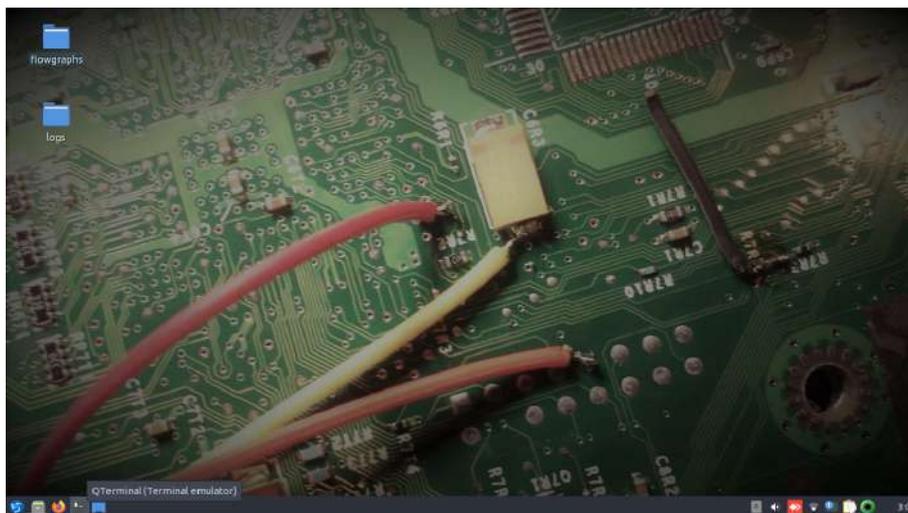


Figura F.9: Apertura de la terminal del sistema DragonOS

2. Introducir el comando: `cd /usr/src/gr-adsb/web`



Figura F.10: Localización de la terminal en el directorio del Webserver

3. Introducir el comando: `python3 ./webserver.py`

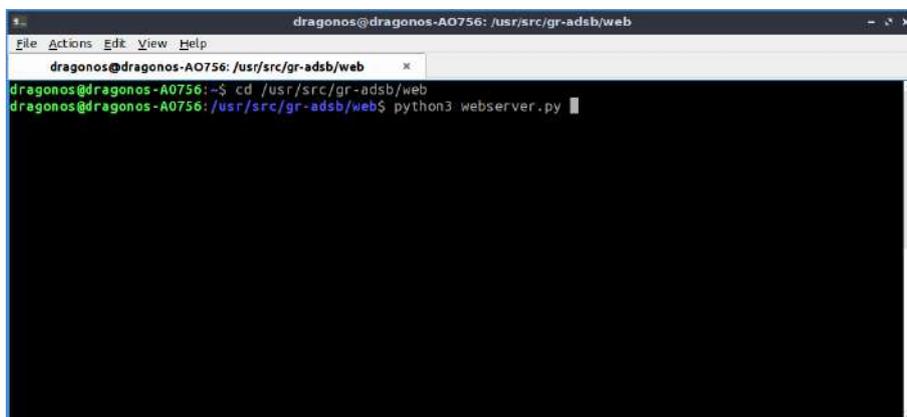


Figura F.11: Ejecución del Webserver

4. Abrir el navegador web integrado al sistema e introducir en la barra de navegación el siguiente enlace:
`https://localhost:5000`

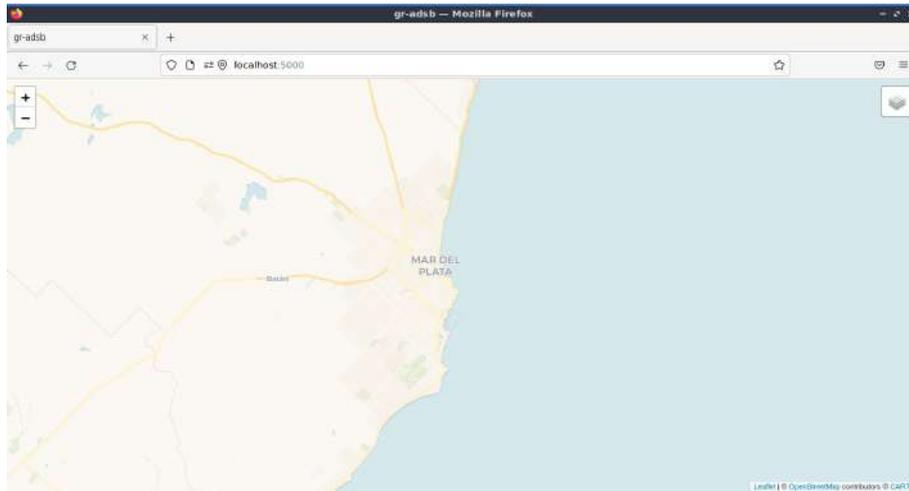


Figura F.12: Visualización del Webservice como cliente desde el navegador web del sistema

F.7.3. Ejemplo de la visualización del Web Server

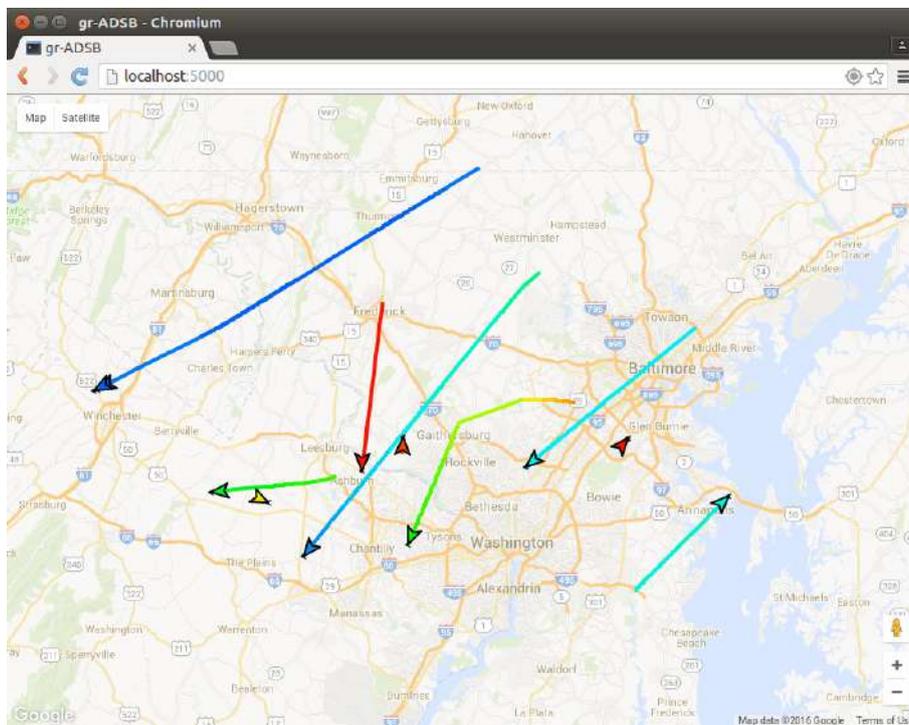


Figura F.13: Ejemplo de la visualización del web server

F.8. ACARS

F.8.1. Instalación de librerías

La librería que posibilita la detección de ACARS se puede descargar en el siguiente [link](#).
Los pasos a seguir para la instalación son los siguientes:

1. Abrir la terminal del sistema
2. Acceder al directorio donde se encuentran los archivos para la instalación con el comando: `cd 3.8ng`
3. Crear el sub-directorio `build` con el comando: `mkdir build`
4. Acceder al sub-directorio `build` con el comando: `cd build`
5. Introducir el comando: `cmake -DCMAKE_INSTALL_PREFIX : PATH = /usr..`
6. Introducir el comando: `make -j4`
7. Introducir el comando: `sudo make install`

F.8.2. Inicialización detección de ACARS

1. En primera instancia se encontrará con la pantalla de inicio del DragonOS.

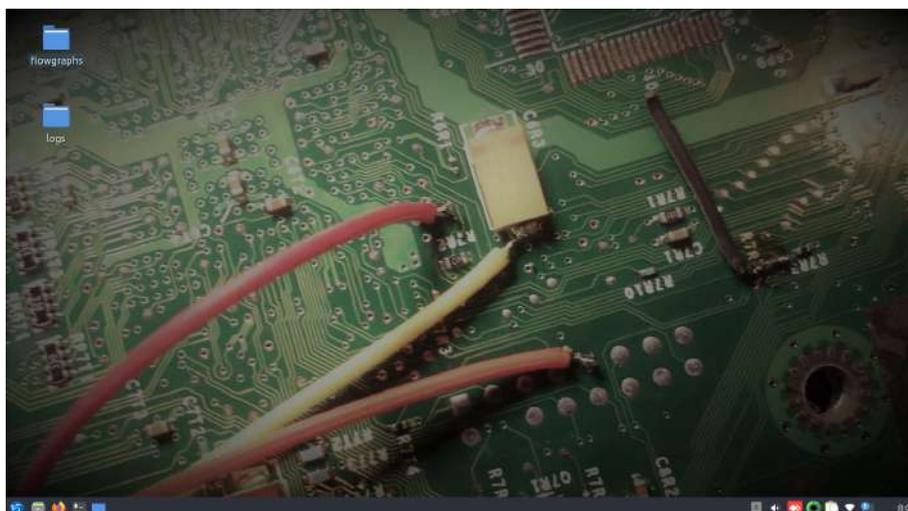


Figura F.14: Pantalla de inicio del DragonOS

2. Ir a la carpeta `acars` que se encuentra dentro del directorio `flowgraphs` en el escritorio.

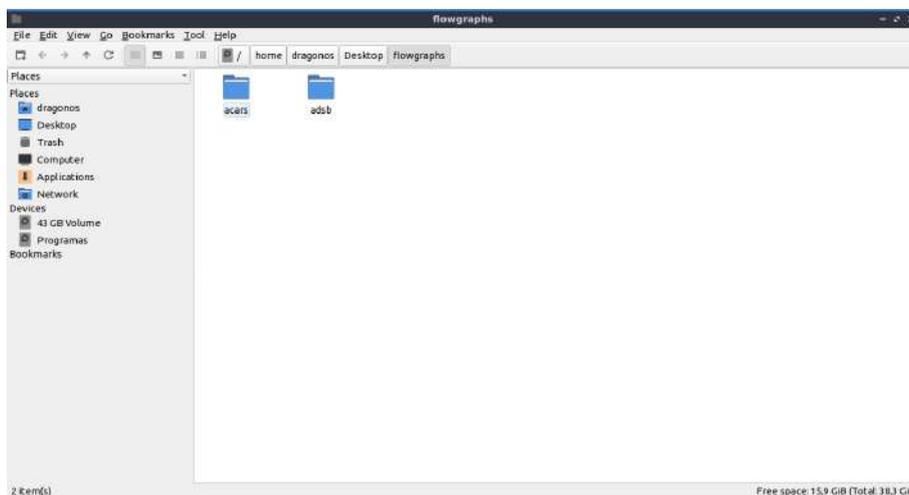


Figura F.15: Directorio flowgraphs

3. Abrir la terminal en la carpeta *acars*

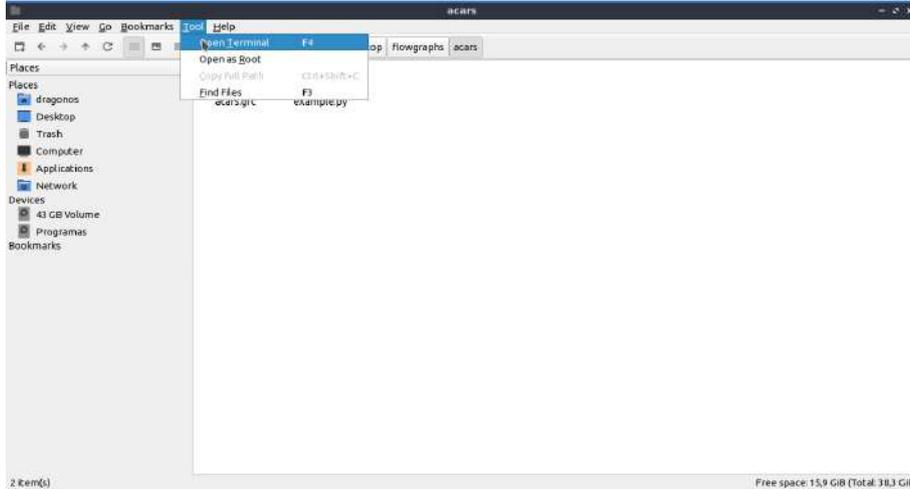


Figura F.16: Selección de terminal

4. Inicializar el programa de detección de ACARS dentro de la terminal del sistema con el siguiente comando: *gnuradio-companion acars.grc*

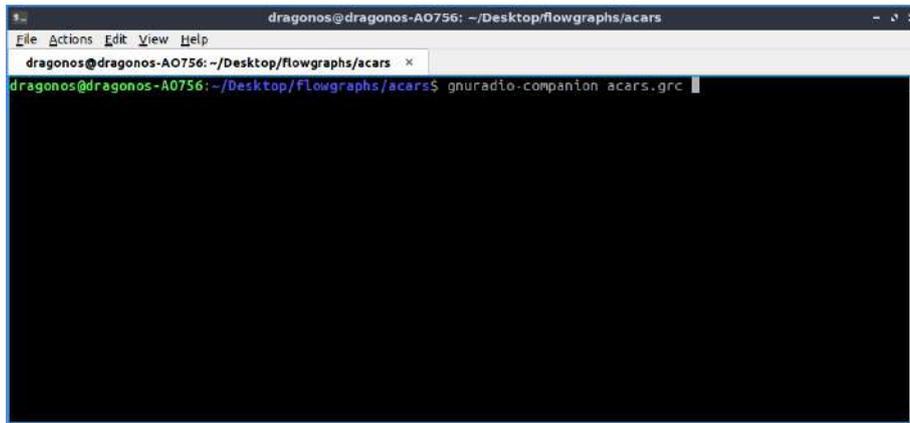


Figura F.17: Pantalla de la terminal

5. En la figura [F.18](#) se ve el GNU radio companion con el diagrama en bloques para la detección de ACARS.

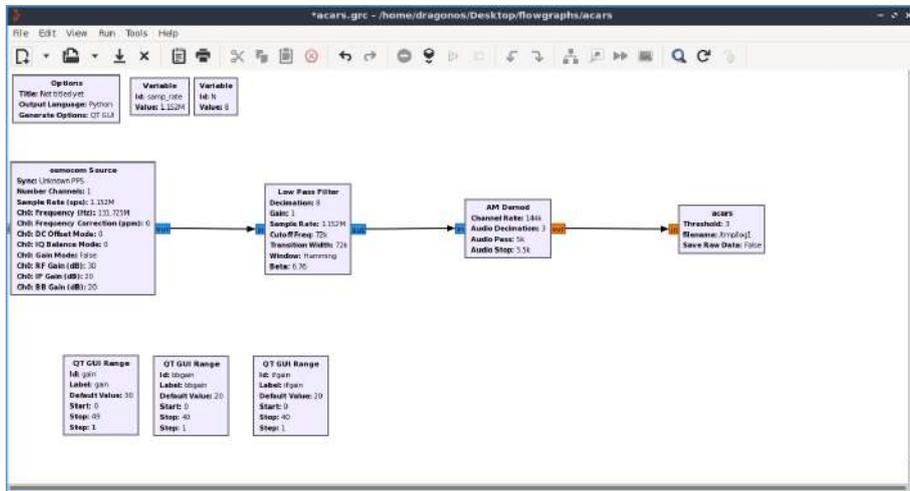


Figura F.18: Flowgraph de detección ACARS en GNU radio

6. Para ejecutar el programa hacer click en *Execute the flow graph*.

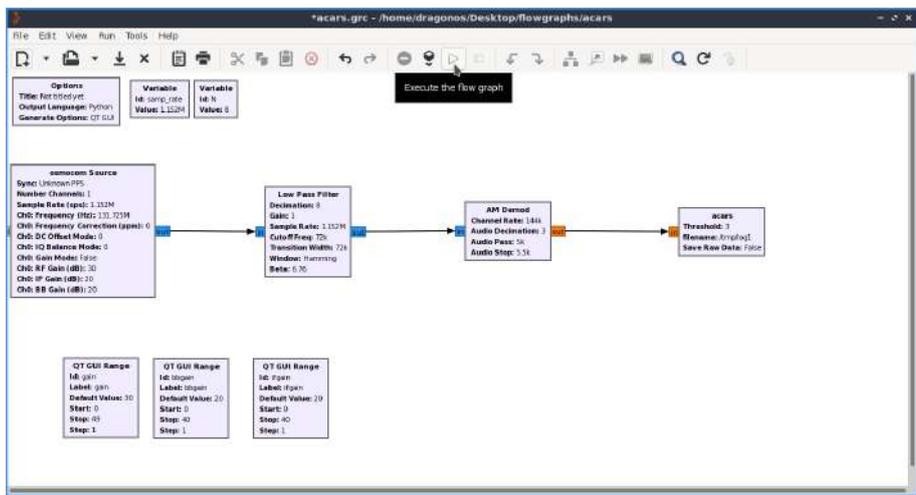


Figura F.19: Ejecución flowgraph detección ACARS en GNU Radio

Capítulo G

Plan de pruebas

G.1. Ficha del documento

Fecha	Versión	Descripción	Autor/es
22 de Febrero de 2022	1.0	Redacción	Nivio Guillán, Lautaro Willig, Mauricio

G.2. Introducción

Este documento tiene como finalidad actuar de guía para gestionar las distintas pruebas que se realizarán a lo largo del proyecto Receptor de telemetría aeronáutica basado en Radio Definida por Software. Está dirigido a la cátedra de Trabajo Final Ing. Electrónica (4D0) / Ing. Computación (4F0) y a todos los involucrados en el desarrollo del proyecto.

G.2.1. Alcance

El plan de pruebas detallado a continuación contiene pruebas unitarias sobre de los distintos componentes del proyecto, pruebas integrales a partir de vincular dos o más componentes y pruebas de homologación con el fin de validar los requerimientos funcionales planteados.

G.2.2. Ambientes de prueba

Se utilizarán 6 ambientes de prueba para cumplir con los 13 casos de prueba planteados:

- SDR + PC
- LNA
- FILTRO ADS-B
- FILTRO ACARS
- COMBINADOR + FILTRO ADS-B + FILTRO ACARS
- ANTENA ADS-B + FILTRO ADS-B + LNA + ANTENA ACARS + FILTRO ACARS + COMBINADOR + SDR + PC (SISTEMA COMPLETO)

G.2.3. Instrumental, herramientas y software

Se utilizarán los elementos listados a continuación:

- PC
- Analizador de espectro Hewlett-Packard 8594e
- Acoplador direccional Hewlett-Packard 778D
- Atenuador Hewlett-Packard 8496B
- Multímetro UNI-T ut61d
- Google Maps, rtl_test y GNU Radio 3.8

G.2.4. Política de trabajo

Se plantea la realización de la totalidad de los casos de prueba en el ámbito del Laboratorio de Comunicaciones de la Facultad de Ingeniería de la UNMDP. Ambos estudiantes realizarán en conjunto las pruebas detalladas, bajo la supervisión del uso del instrumental por parte del personal del Laboratorio.

G.2.5. Comunicación

El reporte de resultados e incidencias se realizará principalmente con los directores Dr. Ing. Alejandro José Uriz e Ing. J. A. Etcheverry. Además, se tendrá una fluida comunicación con el resto de los investigadores del LAC.

G.3. Plan

Ambientes	ID	Prueba	Tipo de Prueba	Procedimiento	Instrumental	Resultado Esperado	Fecha de Prueba	Iteración	Responsable	Resultado Obtenido	Observaciones
SDR + PC	1	Verificación de la frecuencia de sampleo del SDR	Integral	Conectar el SDR a la PC. Ejecutar el script rti_test provisto por el fabricante. Ajustar los parametros de la frecuencia de sampleo en 2 Ghz.	Script rti_test	Se espera no evidenciar la pérdida de paquetes en la comunicación	10/10/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
LNA	2	Verificación de la ganancia del amplificador en la banda de interés de 1090MHz	Unitaria	Conectar el LNA, alimentado con una fuente de 9 Volts, al analizador de espectro con un atenuador. Setear el atenuador en la máxima atenuación posible e ir reduciéndola progresivamente hasta encontrar un punto de referencia para medir la ganancia. En la prueba se toma -10 dBm. Remover el amplificador dejando conectado el atenuador y calcular la diferencia. Dicha diferencia es la ganancia aportada por el amplificador.	Analizador de espectros, atenuador y cables rg58 con conectores SMA	Se espera una ganancia de 30 dB a una frecuencia de 1090 MHz	11/15/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo una ganancia de 18.8 dB	La ganancia obtenida fue menor que la especificada por el fabricante del amplificador
	3	Consumo de corriente del LNA	Unitaria	Conectar en serie a la alimentación del amplificador un multímetro en modo amperímetro y registrar el valor de corriente indicado por el instrumento. Realizar la prueba con el amplificador en funcionamiento, alimentado con una fuente de 9 Volts, y estando conectado a una carga de 50 Ohms en su extremo.	Multímetro y carga de 50 Ohms SMA	Se espera un consumo menor a 30 mA	12/13/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	Se obtuvo un consumo de 29.3 mA
Filtro ADSB	4	Verificación de la frecuencia central de la banda de paso	Unitaria	Conectar el filtro microstrip al analizador de espectros y configurar el equipo para visualizar la banda de paso en 1090 MHz	Analizador de espectros	Se espera que el filtro esté centrado en 1090 MHz	10/28/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
	5	Verificar la pérdida en la banda de paso	Unitaria	Conectar el filtro microstrip al analizador de espectros y configurar el equipo para visualizar la banda de paso en 1090 MHz	Analizador de espectros	Se espera encontrar una pérdida del orden de 6 dB	10/28/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo un resultado aproximado al esperado	Se registró una pérdida de 7 dB aproximadamente
Filtro ACARS	6	Verificación de la frecuencia central de la banda de paso	Unitaria	Conectar el filtro ACARS al analizador de espectros y configurar el equipo para visualizar la banda de paso en 131 MHz	Analizador de espectros	Se espera encontrar que el filtro se encuentre centrado en 131 MHz	11/29/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
	7	Verificar la pérdida en la banda de paso	Unitaria	Conectar el filtro ACARS al analizador de espectros y configurar el equipo para visualizar la banda de paso en 131 MHz	Analizador de espectros	Se esperará encontrar una pérdida del orden de 3 dB	11/29/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
Combinador	8	Verificar la pérdida del combinador	Unitaria	Conectar el combinador al analizador de espectro. Una de las entradas se conecta al analizador y la restante a una carga de 50 Ohm.	Analizador de espectros y carga de 50 Ohm SMA	Se esperará encontrar una pérdida del orden de 3 dB	11/29/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
Combinador + Filtro ADS-B + Filtro ACARS	9	Verificación de la frecuencia central de la banda de paso (ADS-B)	Integral	Conectar el filtro de ADS-B en cascada con el combinador al analizador de espectro. La entrada restante del combinador se coloca una carga de 50 Ohm. Configurar el equipo para realizar un barrido en frecuencia y observar la respuesta obtenida.	Analizador de espectros y carga de 50 Ohm SMA	Se esperará que el filtro en cascada con el combinador estén centrados en 1090 MHz	11/29/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	Producto de la existencia de múltiplos en la frecuencia de la banda de paso, se observa un bajo rechazo cercano a los 2000 MHz
	10	Verificación de la frecuencia central de la banda de paso (ACARS)	Integral	Conectar el filtro de ACARS en cascada con el combinador al analizador de espectro. La entrada restante del combinador se coloca una carga de 50 Ohm. Configurar el equipo para realizar un barrido en frecuencia y observar la respuesta obtenida.	Analizador de espectros y carga de 50 Ohm SMA	Se espera que el filtro en cascada con el combinador estén centrados en 131 MHz	11/29/2021	1	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-
Sistema completo	11	Registrar la máxima distancia de recepción en ADS-B	Homologación	Utilizando el sistema en modo ADS-B registrar el máximo rango de recepción de mensajes provenientes de aeronaves. Realizar la medición teniendo en cuenta las coordenadas incluidas en los mensajes decodificados, utilizando la herramienta web Google Maps.	Software online Google Maps	Se espera un rango de recepción mayor a 100 km	11/24/2021	5	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	El rango alcanzó los 140 km en línea recta
	12	Registrar la máxima distancia de recepción en ACARS	Homologación	Utilizando el sistema en modo ADS-B registrar el máximo rango de recepción de mensajes provenientes de aeronaves. Realizar la medición teniendo en cuenta las coordenadas incluidas en los mensajes decodificados, utilizando la herramienta web Google Maps.	Software online Google Maps	Se espera un rango de recepción mayor a 100 km	11/24/2021	5	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	El rango alcanzó los 180 km en línea recta
	13	Registrar la relación de señal a ruido en la recepción de paquetes ADS-B	Homologación	Utilizar el sistema en modo ADS-B y registrar la relación señal a ruido de señales emitidas por aeronaves dentro de un rango menor a 150 Km en línea recta	Indicador SNR GNU RADIO	Se espera una relación señal a ruido mayor a 15 dB	11/24/2021	5	Mauricio Willig y Lautaro Nivio	Se obtuvo el resultado esperado	-