

Universidad Nacional de Mar del Plata

Facultad de Ingeniería

Proyecto Final de Grado

Transmisor de Identificación Automática para uso Naval (TIAN)

Informe Final



Autor: Gorak, Roberto Sergio

Director: Dr. Ing. Alejandro Uriz

Co Director: Ing. Brian Gluzman

Co Directora: MSc. Mónica Liberatori



RINFI es desarrollado por 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/).

Universidad Nacional de Mar del Plata

Facultad de Ingeniería

Proyecto Final de Grado

Transmisor de Identificación Automática para uso Naval (TIAN)

Informe Final



Autor: Gorak, Roberto Sergio

Director: Dr. Ing. Alejandro Uriz

Co Director: Ing. Brian Gluzman

Co Directora: MSc. Mónica Liberatori

Agradecimientos

A mi madre y hermanas, Inés, Susana y Cecilia, por su apoyo incondicional a lo largo de todos los años que duró mi paso por la UNMdP.

A los directores del proyecto, Dr. Ing. Alejandro Uriz, Ing. Brian Gluzman y MSc. Mónica Liberatori, cuya experiencia y paciencia fueron la guía para completar este desarrollo.

A la cátedra Seminario de Trabajo Final, Ing. Gustavo Uicich, Noelia Garcia e Ing. Fernando Zagnoni, quienes aportaron no sólo herramientas de gestión sino además una mirada alternativa acerca del proyecto y la manera de abordarlo.

A mis amigos y compañeros de carrera con quienes compartí las vicisitudes de la vida universitaria.

Índice

Capítulo 1: Introducción.....	8
1.1 Acerca de AIS.....	8
1.2 Propósito del Documento.....	8
1.3 Alcance del Proyecto.....	9
1.4 Personal Involucrado.....	9
1.5 Siglas y Definiciones.....	10
Capítulo 2: Anteproyecto.....	12
2.1 Propuesta del Proyecto.....	12
2.2 Requerimientos.....	12
RF01: Capacidad de programación.....	12
RF02: Capacidad de modificación.....	12
RF03: Código de identificación único.....	12
RF04: Datos de posición en tiempo real.....	13
RF05: Potencia de Transmisión.....	13
RF06: Tensión de Alimentación.....	13
2.3 Plan de Proyecto.....	13
Capítulo 3: Proyecto.....	15
3.1 Descripción General.....	15
3.2 Diagrama en Bloques.....	16
3.3 Adquisición de Datos.....	16
3.3.1 Módulo CAN Bus.....	16
3.3.2 Módulo Microcontrolador.....	18
3.3.3 Módulo NMEA2000.....	19
3.3.4 Módulo UART-USB.....	22
3.4 Computadora.....	23
3.4.1 Interfaz de Usuario.....	23
3.4.2 Comandos AT.....	23
3.4.3 GNU Radio.....	24
3.5 Transmisión.....	26
3.5.1 Adalm Pluto SDR.....	26
3.5.2 Acceso al Medio.....	27
3.5.3 Amplificador 2W.....	27

Capítulo 4: Montaje.....	30
4.1 Adquisidor de Datos.....	30
4.2 Módulo UART-USB.....	32
4.3 Gabinete.....	34
Capítulo 5: Plan de Pruebas.....	36
5.1 Pruebas Unitarias.....	36
5.2 Banco de Pruebas.....	36
5.3 Simulador de Tráfico de Red.....	37
5.4 Receptor.....	40
5.4.1 RTL-SDR.....	40
5.4.2 SDRAngel.....	41
5.4.3 OpenCPN.....	42
5.5 Prueba Integral.....	44
Capítulo 6: Conclusiones.....	48
6.1 Acerca del dispositivo.....	48
6.2 Acerca del proceso.....	49
Apéndice A: Amplificador RF.....	51
A.1 MOSFET de Potencia para RF: RD15HVF1.....	51
A.2 Circuito de Prueba a 175 MHz.....	51
Apéndice B: Estudio Preliminar.....	52
B.1 Acerca del Equipo.....	52
Integrante.....	52
Director.....	52
Co Directora.....	52
Co Director.....	52
B.2 Acerca del Proyecto Final.....	52
Descripción del proyecto.....	52
Beneficiario / Cliente.....	53
Nombre del producto resultante.....	53
Diagrama en bloques.....	53
Grado de Avance.....	53
Apéndice C: Plan de Proyecto.....	54
C.1 Acerca del Plan.....	54
Introducción.....	54
Software.....	54
C.2 Diagrama de Gantt.....	55

C.3 Riesgos.....	56
Apéndice D: Especificación de Requerimientos.....	57
D1. Introducción.....	57
D1.1 Propósito.....	57
D1.2 Alcance.....	57
D1.3 Personal involucrado.....	58
D1.4 Definiciones.....	59
D2. Descripción General.....	60
D2.1 Perspectiva del Producto.....	60
D2.2 Funcionalidad.....	60
D2.3 Características de los usuarios.....	61
D2.4 Restricciones.....	62
D2.5 Suposiciones y dependencias.....	62
D3. Requisitos Específicos.....	63
D3.1 Requisitos comunes de las interfaces.....	63
D3.2 Requisitos Funcionales.....	63
D3.3 Requisitos no Funcionales.....	64
Apéndice E: Especificación Funcional.....	65
E1. Introducción.....	65
E1.1. Propósito del documento.....	65
E1.2. Alcance del Proyecto.....	66
E1.3. Alcance del Documento.....	66
E1.4. Documentos Relacionados.....	66
E1.5. Siglas y Definiciones.....	67
E1.6. Suposiciones y Riesgos.....	67
E2. Descripción General.....	68
E2.1. Diagrama en Bloques.....	68
E2.2. Personal Involucrado.....	69
E2.3. Dependencias y Cambios.....	69
E3. Especificaciones Funcionales.....	70
E3.1. Capacidad de Programación.....	70
E3.2. Capacidad de Modificación.....	70
E3.3. Código de Identificación único.....	70
E3.4. Datos de posición en tiempo real.....	71
E3.5. Alimentación y Potencia.....	71
E4. Configuración del Sistema.....	71
E5. Otros requerimientos.....	72

E6. Requerimientos para reportes.....	72
E7. Requerimientos de integración.....	72
E8. Requerimientos de Conversión/Migración.....	72
E9. Referencias.....	72
E10. Incidencias por resolver.....	72
Apéndice F: Especificación Técnica.....	73
F1. Introducción.....	73
F1.1. Propósito del documento.....	73
F1.2. Alcance del Proyecto.....	73
F1.3. Personal Involucrado.....	73
F1.4. Siglas y Definiciones.....	74
F1.5. Suposiciones y Riesgos.....	75
F2. Descripción General.....	76
F2.1. Diagrama en Bloques.....	76
F3. Descripción de los Módulos.....	77
F3.1. Módulo CAN Bus.....	77
F3.1.1 MCP2551.....	77
F3.1.2 MCP2515.....	78
F3.2 Módulo Microcontrolador.....	79
F3.2.1 Módulo NMEA2000.....	80
F3.3 Módulo UART-USB.....	82
F3.4 Computadora.....	83
F3.4.1 Interfaz de Usuario.....	83
F3.4.2 Comandos AT.....	84
F3.4.3 GNU Radio.....	84
F3.5 Transmisión.....	85
F3.5.1 Adalm Pluto SDR.....	85
F3.5.2 Acceso al Medio.....	86
F3.5.3 Amplificador 2W.....	87
F4. Referencias.....	89
Apéndice G: Plan de Pruebas.....	90
G1. Acerca del Equipo.....	90
Integrante.....	90
Director.....	90
Co Director.....	90
Co Directora.....	90
G2. Acerca del Proyecto Final.....	90

G3. Plan de Pruebas.....	91
Alcance.....	91
Ambientes de Prueba.....	91
Recursos, Herramientas e Instrumentos.....	92
Políticas de Trabajo.....	92
Estrategia de Comunicación.....	92
G4. Banco de Pruebas.....	92
Objetivo.....	92
Diagrama en Bloques.....	93
G5. Casos de Prueba.....	94
Referencias.....	100

Capítulo 1: Introducción

1.1 Acerca de AIS

El Sistema de Identificación Automática, AIS por sus siglas en inglés, es un estándar técnico marítimo desarrollado por la Organización Marítima Internacional (IMO) [1].

AIS es una sofisticada tecnología de radiofrecuencia que combina la implementación de GPS, VHF y tecnologías de procesamiento de datos para establecer el intercambio de información relevante en un formato estrictamente definido entre distintas entidades marítimas. El intercambio puede ser tan simple como datos de posición, rumbo, velocidad y credenciales de identificación o bien más sofisticado entre estaciones costeras y dispositivos instalados en boyas flotantes [2].

AIS provee una forma dinámica y sencilla de identificación para embarcaciones en todo el mundo. El sistema permite obtener en tiempo real y de forma remota los datos más relevantes de dichas embarcaciones.

El sistema requiere un receptor compatible que pueda recibir y decodificar la información generada por las embarcaciones, estos datos suelen ser presentados en forma gráfica en algún dispositivo tipo plotter. Se recomienda su uso como complemento de un sistema de radar. Las embarcaciones utilizan un transmisor capaz de enviar la información requerida además de proveer a la tripulación una forma de modificar dicha información en caso de ser necesario.

1.2 Propósito del Documento

El presente documento corresponde al informe final para el proyecto “Transmisor de Identificación Automática para uso Naval (TIAN)” y tiene como propósito proveer información resumida acerca del proceso de diseño, fabricación y pruebas. Dicha información abarca detalles de su funcionamiento y construcción. Con el objetivo de proveer información más específica, se anexan los documentos confeccionados durante el desarrollo del proyecto para las distintas especificaciones y etapas.

El documento se ha estructurado en base a la información detallada en los documentos “Especificación de Requerimientos Ver. 0.4”, “Especificación Funcional Ver. 0.3”, “Especificación Técnica Ver. 0.1” y “Plan de Pruebas Ver. 0.4”.

Este documento se encuentra dirigido a los desarrolladores del proyecto, la mesa examinadora para su evaluación, profesores, estudiantes y personal técnico que desee conocer las soluciones implementadas para el desarrollo del dispositivo.

1.3 Alcance del Proyecto

El proyecto abarca la definición de los requerimientos, diseño, implementación y la presentación final del dispositivo. Los requerimientos fueron planteados en conjunto con los directores que pertenecen al Laboratorio de Comunicaciones (LAC) de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Es desde este laboratorio que surge la propuesta para el proyecto y donde se realizó la mayor parte del trabajo.

El proyecto fue evaluado según la escala TRL (Technology Readiness Level), en dicha escala se lo considera en el nivel TRL = 4. Los bloques básicos se integran para comprobar que las piezas funcionan como un sistema. Los ensayos incluyen la integración de hardware específico en un laboratorio, pruebas con una variedad de simuladores y pruebas a pequeña escala en condiciones de cargas reales [3].

1.4 Personal Involucrado

Nombre: **Roberto S. Gorak**

Rol: Diseñador, programador, tester.

Categoría: Estudiante

Tareas: Diseño, cálculo, montaje y testeo del sistema.

Contacto: gorakroberto@gmail.com

Nombre: **Dr. Ing. Alejandro José Uriz**

Rol: Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: ajuriz@fi.mdp.edu.ar

Nombre: **Ing. Brian Gluzman**

Rol: Co Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: briangluzman@mdp.edu.ar

Nombre: **MSc. Mónica Liberatori**

Rol: Co Directora de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: mlibera@fi.mdp.edu.ar

1.5 Siglas y Definiciones

Sigla	Significado	Descripción
TIAN	Transmisor de Identificación Automática para uso Naval	Dispositivo electrónico transmisor compatible con sistema AIS.
AIS	Sistema de Identificación Automática	Sistema de identificación utilizado en embarcaciones que provee variada información.
TRL	Technology Readiness Level	Escala de niveles que permite conocer el grado de madurez de un proyecto tecnológico.
RF	RadioFrecuencia	Medio por el cual se realiza la comunicación entre los dispositivos.
IMO	Organización Marítima Internacional	Organización que dicta las normas internacionales para la navegación.
ENACOM	Ente Nacional de Comunicaciones	Ente Nacional que regula las telecomunicaciones en Argentina.
SDR	Radio Definida por Software (Software Defined Radio)	Sistema de radiocomunicaciones donde varios de los componentes son implementados en software.
LAC	Laboratorio de Comunicaciones	Laboratorio de Comunicaciones de la Facultad de Ingeniería.
NMEA2000 N2K	--	Protocolo de comunicación para dispositivos electrónicos y sensores en embarcaciones.
PGN	Parameter Group Number	Identificador único de mensajes.
SPI	Interfaz de Periféricos Serie (Serial Peripheral Interface)	Bus de comunicaciones serie multipropósito.

Sigla	Significado	Descripción
SOG	Velocidad sobre fondo (Speed over Ground)	Velocidad de la embarcación con referencia en el fondo.
COG	Rumbo sobre tierra (Course over Ground)	Dirección a la que apunta la proa de la embarcación.
MMSI	Identificación del sistema móvil marítimo (Maritime Mobile Service Identity)	Serie de dígitos que identifica inequívocamente a cada estación del servicio móvil marítimo.
NRZI	No Retorno a Cero Invertido (Non Return to Zero Inverted)	Código binario utilizado en la transmisión.
GMSK	Modulación por desplazamiento mínimo gaussiano	Esquema de modulación digital por desplazamiento de frecuencia de fase continua.
TDMA	Multiplexado por división de tiempo	Esquema de acceso al medio.
SOTDMA	Multiplexado por división de tiempo con auto-organización	Esquema de acceso al medio utilizado por dispositivos AIS clase A.
CSTDMA	Multiplexado por división de tiempo con sensado de portadora	Esquema de acceso al medio utilizado por dispositivos AIS clase B.
MSG XX	Mensaje XX	Forma abreviada de mencionar los distintos mensajes AIS.

Capítulo 2: Anteproyecto

2.1 Propuesta del Proyecto

Se propuso el diseño, desarrollo, implementación y pruebas de un dispositivo transmisor compatible con el sistema AIS de clase B y con capacidad de interconectividad con redes NMEA 2000. Se determinó su implementación con microcontroladores, circuitos integrados comerciales y herramientas propias del laboratorio. Para su desarrollo se optó por la utilización de librerías de código abierto para la adquisición y procesamiento de los datos de la embarcación. Para la transmisión por RF se optó por el uso combinado de GNU Radio y un dispositivo SDR.

Se definió el nombre TIAN para el instrumento a desarrollar y se diseñó el logotipo que se observa en la portada del documento.

2.2 Requerimientos

A continuación, se detallan los requerimientos planteados para el proyecto.

Los requisitos RF01, RF02 y RF03 provienen de la interacción del usuario con el dispositivo. La tripulación debe tener la capacidad de configurar TIAN con los datos correspondientes a la embarcación.

RF01: Capacidad de programación.

El sistema debe proveer al usuario una interfaz que le permita establecer los datos estáticos de la embarcación.

RF02: Capacidad de modificación.

El sistema debe proveer al usuario una interfaz que le permita modificar los datos estáticos de la embarcación en caso de ser necesario.

RF03: Código de identificación único.

La trama generada por el programa debe incluir la identificación provista por el ente encargado a tal fin, de modo que el sistema permita identificar unívocamente a qué embarcación pertenece.

Los requisitos RF04 y RF05 derivan del ITU-R M.1371-4 Annex 7 [4] que dicta las normas de transmisión para los dispositivos compatibles con AIS Clase B con tecnología CSTDMA.

RF04: Datos de posición en tiempo real.

La trama generada por el programa debe ser compatible con los mensajes establecidos por IMO para informar la posición mediante AIS. La frecuencia de actualización de los datos debe seguir la norma establecida.

RF05: Potencia de Transmisión.

El rango de potencia permitido para la transmisión de señal de identificación es de 2W para la frecuencia de trabajo y la clase del dispositivo.

Finalmente el requisito RF06 parte del tipo de alimentación que el bus N2K de las embarcaciones provee a los dispositivos y sensores conectados.

RF06: Tensión de Alimentación.

El sistema deberá utilizar como fuente de tensión un máximo de 24 volts de corriente continua.

2.3 Plan de Proyecto

Durante el Seminario de Trabajo Final se redactaron una serie de documentos, entre ellos el Plan de Proyecto. Éste detalla las actividades y el tiempo de desarrollo que cada una de éstas requiere. Para facilitar su manejo y visualización se utiliza el software libre GanttProject [5]. Se genera simultáneamente una lista de tareas y el gráfico correspondiente a la línea de tiempo.

Resulta importante aclarar que la confección del plan de proyecto se realizó de forma tentativa inicialmente y se fue corrigiendo conforme las etapas se fueron completando. La principal causa de incertidumbre a la hora de establecer los tiempos para cada tarea se deben a la actividad laboral que desarrollo, su carga horaria y las variaciones relacionadas con la actividad en sí misma. Se puede observar también ítems marcados como problemas de salud que aportaron sendos retrasos a la hora de completar algunas tareas. La Figura 2.1 muestra el diagrama de Gantt para el plan de proyecto. Se pueden observar 9 etapas que representan los distintos estadios de desarrollo de la totalidad del proyecto.

Las primeras dos etapas corresponden al estudio previo y a la presentación y aprobación del proyecto por parte de las autoridades de la Facultad de Ingeniería. Las siguientes tres etapas corresponden al diseño y desarrollo del dispositivo. Luego se presentan las dos etapas correspondientes a las pruebas y finalmente la etapa de Documentación. La etapa final, que abarca toda la línea de tiempo del proyecto, representa una tarea sugerida por la cátedra del Seminario de Trabajo Final, se creó una Bitácora de Proyecto que permitiera llevar un recuento de los sucesos que se fueran dando durante todo el proceso. Dicha Bitácora resultó de vital importancia a la hora de redactar este documento.

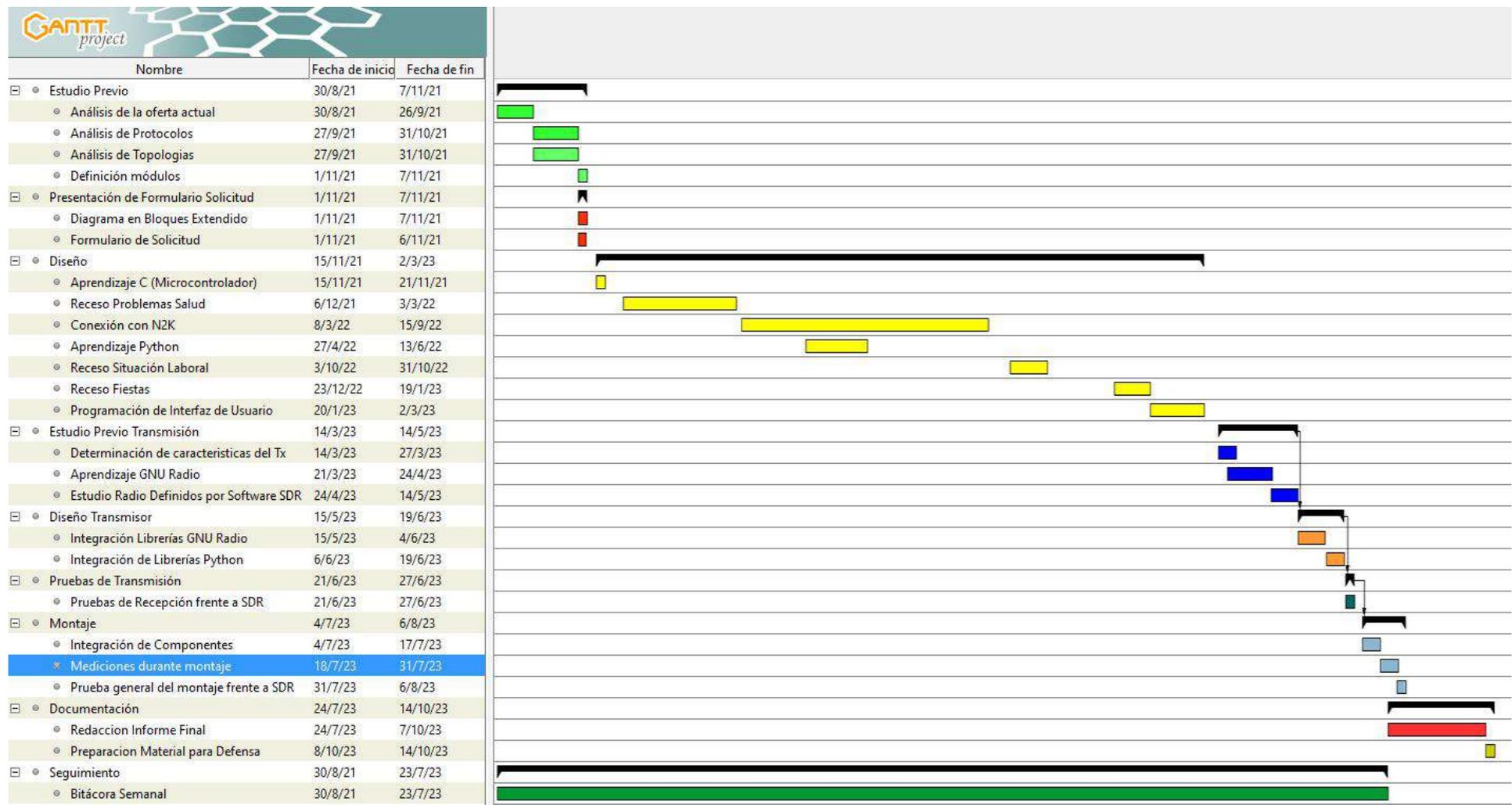


Figura 2.1 - Diagrama de Gantt. Lista de Tareas y Línea de Tiempo

Capítulo 3: Proyecto

3.1 Descripción General

Se propone una solución basada en los dispositivos comerciales estudiados. Dicha solución cuenta con tres etapas principales que cumplen los Requerimientos Funcionales establecidos en el capítulo anterior.

La primera etapa consta de un dispositivo de hardware basado en microcontroladores que se conecta con la red de la embarcación. Este Adquisidor de Datos recibe todos los paquetes disponibles y filtra aquellos que no son de utilidad para la generación de los mensajes AIS.

La segunda etapa del proyecto consiste en un programa de computadora que se comunica por puerto USB con el Adquisidor de Datos para obtener la información dinámica de la embarcación. También genera la interfaz de usuario que permite la carga de la información estática y muestra algunos datos de la transmisión AIS. Además, se encarga de controlar la frecuencia de transmisión de los mensajes interactuando con la tercera etapa sin intervención del usuario. Esta etapa es implementada en lenguaje Python.

La tercera y última etapa es la que controla la transmisión de los mensajes AIS. Ésta se implementa utilizando un SDR modelo Adalm-Pluto de la empresa Analog Devices. El control del SDR se realiza mediante el software GNU Radio, Python y librerías compatibles.

3.2 Diagrama en Bloques

La Figura 3.1 muestra el diagrama en bloques correspondiente a la solución propuesta.

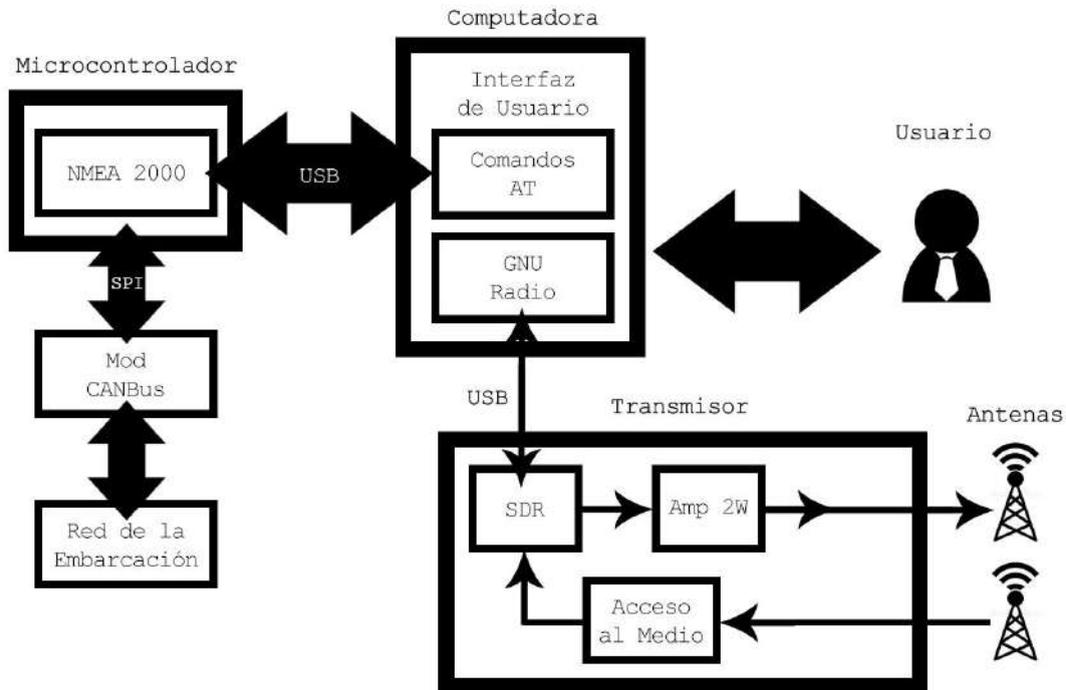


Figura 3.1 - Diagrama en Bloques

3.3 Adquisición de Datos

3.3.1 Módulo CAN Bus

La red utilizada en las embarcaciones se conoce como NMEA 2000. La norma que la define abarca desde la capa física hasta el protocolo de alto nivel. La capa física de la red es compatible con las redes tipo CAN Bus y el protocolo de alto nivel es una versión de SAE J1939 modificada para el entorno marítimo [6].

El módulo CAN Bus del proyecto cumple las funciones de capa física del dispositivo, es decir, permite la conexión física a la red de la embarcación. La implementación del bloque se logra con el uso de dos circuitos integrados comerciales destinados a tal fin.

El circuito integrado que se conecta físicamente con la red de la embarcación es el MCP2551 [7]. Este circuito integrado es un transceptor CAN de alta velocidad de la empresa Microchip. Su función es servir como interfaz entre la red física y el controlador CAN adaptando los niveles y protegiendo al resto de la circuitería de posibles picos o señales espurias que se puedan generar en la red. Este circuito integrado no requiere más configuración que los componentes que se le conectan.

El siguiente eslabón en la capa física es el circuito integrado MCP2515 [8] también de la empresa Microchip. El circuito integrado en cuestión es un Controlador CAN autónomo con interfaz SPI. Su función es el control de los mensajes CAN que se reciben o se desean transmitir. Los pines de transmisión de datos del MCP2515 se conectan directamente a los del MCP2551. El bus SPI se conecta al bus SPI del microcontrolador que funciona como Adquisidor de Datos. Para su correcto funcionamiento el controlador requiere además un oscilador de cristal con sus respectivos capacitores y una resistencia.

A diferencia del transceptor, el controlador requiere una rutina de configuración que se ejecuta enviando comandos por el bus SPI. Esta rutina es ejecutada por el microcontrolador a través de la librería “NMEA2000 library for C++” que agrupa a su vez tres librerías que trabajan en conjunto. Para este caso, la librería específica encargada de controlar y configurar el MCP2515 es CAN Bus Shield de SeeedStudio [9].

Una vez conectados ambos circuitos integrados con sus respectivos componentes, y configurados cuando corresponde, el módulo funciona como un dispositivo de comunicación CAN. El transceptor provee los terminales CAN_H y CAN_L que le permiten conectar directamente con la red. Por otro lado, se conecta con el controlador CAN mediante los pines RX y TX, de mismo nombre en ambos circuitos integrados, generando un bus de datos compatible con éste último. El controlador se encarga de realizar todas las tareas relacionadas con el envío y recepción de los mensajes CAN. Las características del bus son cargadas como parte de la configuración inicial que realiza el microcontrolador. Todos los mensajes CAN que se encuentran en el bus de la red, son recibidos por el controlador y enviados por el bus SPI al microcontrolador. En este nivel no se realiza ningún procesamiento de información que no sea el manejo de mensajes CAN, es decir, ése es el tipo de mensajes que recibe el microcontrolador.

La Figura 3.2 muestra el circuito utilizado para este módulo.

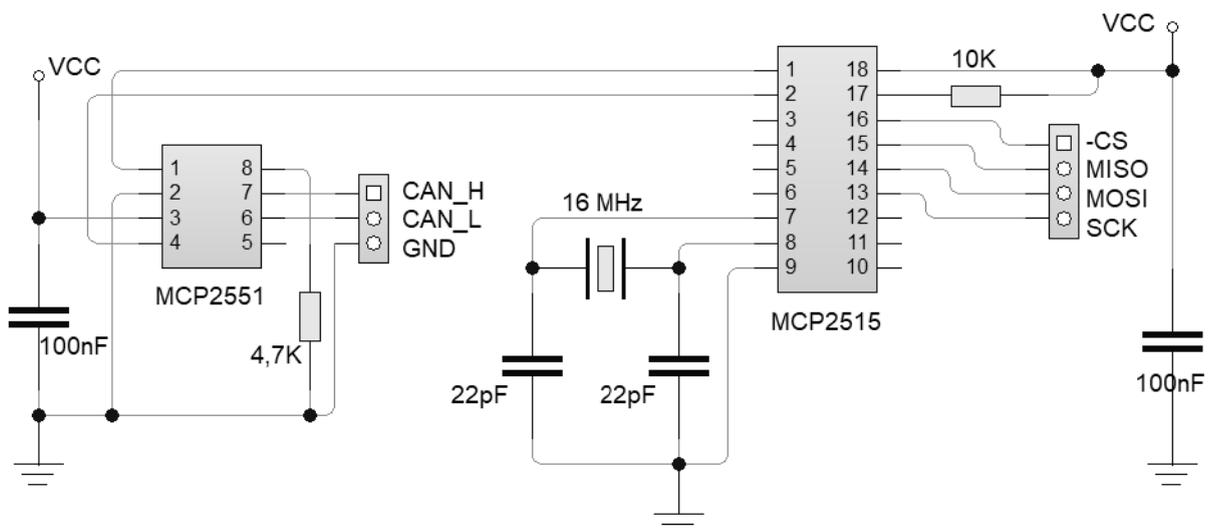


Figura 3.2 - Circuito Módulo CAN

3.3.2 Módulo Microcontrolador

El módulo CAN establece la conexión física del dispositivo, el resto de las funciones relacionadas a la conectividad quedan a cargo del microcontrolador y la librería NMEA2000 previamente mencionada.

El microcontrolador elegido para el desarrollo es el ATMEGA328P. Dicho dispositivo es un microcontrolador AVR de 8-bits con 32 kBytes de memoria flash programable fabricado por la empresa Atmel [10]. Su función es procesar la información de los mensajes que el módulo CAN recibe. Luego de ser procesada, la información es guardada en la memoria propia del dispositivo y enviada por puerto serie a demanda al programa que presenta la interfaz de usuario en la computadora.

Para su correcto funcionamiento, el ATMEGA328P requiere algunos componentes circuitales que lo configuran. En este caso se recurre a la configuración mínima para el funcionamiento dado que no se hace uso extensivo de las entradas y salidas generales. La comunicación con el MCP2515 se realiza mediante el bus SPI. Así mismo, la comunicación con el programa de la interfaz de usuario se realiza mediante el puerto serie del microcontrolador y un circuito de adaptación UART-USB que permite crear un puerto serie virtual en la computadora desde un puerto USB.

La Figura 3.3 muestra el circuito utilizado para este módulo.

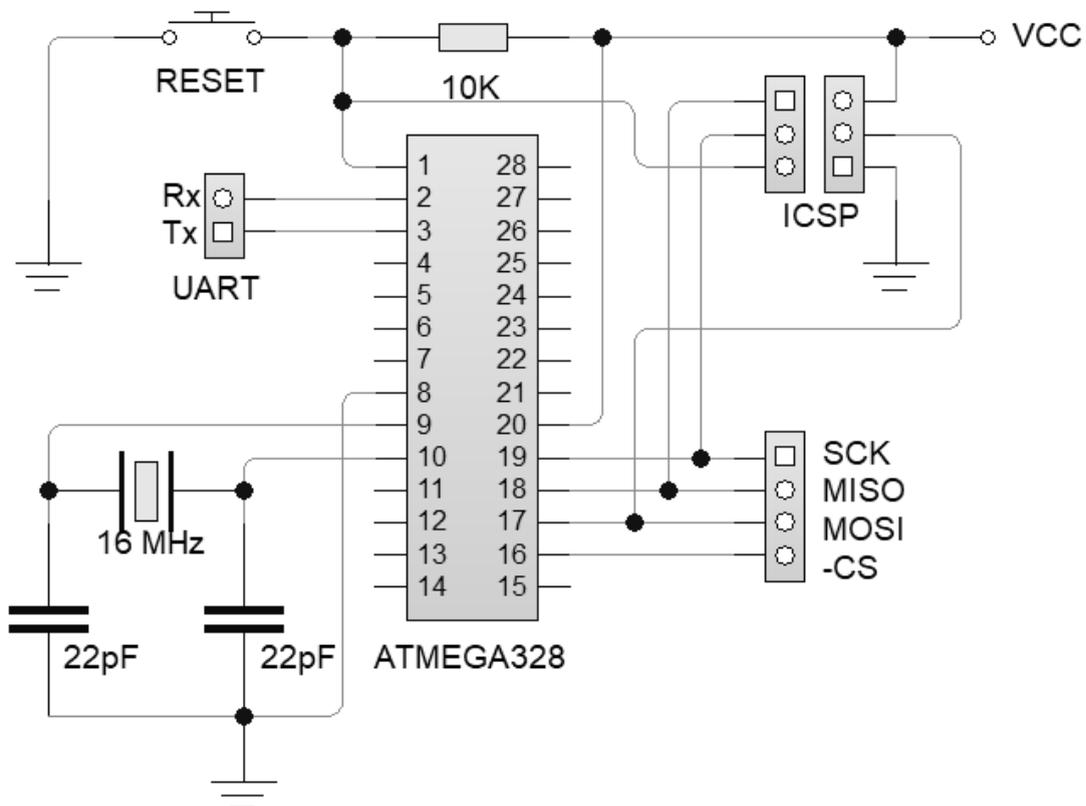


Figura 3.3 - Circuito ATMEGA328P

3.3.3 Módulo NMEA2000

El componente de software de este módulo corresponde a la librería “NMEA2000 library for C++” programada por Timo Lappalainen y distribuida bajo la licencia del MIT. La librería es compatible con una gran cantidad de dispositivos en la medida que estos puedan ser programados en C++ [11]. En este caso se utiliza la interfaz de usuario de Arduino para generar un enlace entre la librería y el microcontrolador ATMEGA328P.

Como se había adelantado, la librería está en realidad dividida en tres librerías individuales. La primera de ellas, que ya fue mencionada en el módulo CAN Bus, es CAN Bus Shield de SeeedStudio y se encarga de controlar la comunicación con el bus CAN.

La segunda librería desempaqueta los mensajes CAN para obtener su payload, la cual es el mensaje N2K que contiene la información que debe procesar el microcontrolador en última instancia. Esta librería es necesaria al utilizar como controlador CAN el circuito integrado MCP2515. El nombre de la librería es NMEA2000_MCP programada por Timo Lappalainen.

La tercera y última librería, de nombre NMEA2000 y programada por Timo Lappalainen, provee herramientas que permiten manipular los mensajes de N2K como una estructura de datos dentro del programa general. Provee clases y métodos para procesar los mensajes N2K, permitiendo que de forma sencilla se obtengan los atributos de los objetos de la clase de mensajes. Cada tipo de mensaje tiene su propia clase, lo que permite procesarlos de forma individual discriminando según su PGN. Para ello, la librería cuenta con una base de datos de dichos identificadores y su estructura interna. Este software está en parte basado en el proyecto CANBoat que aplicando ingeniería inversa consiguió decodificar la base de datos de los PGN.

La librería tiene otras funciones que para este módulo no son de interés pero que resultan inmensamente útiles a la hora de generar un banco de pruebas para realizar la puesta en marcha del dispositivo completo. En secciones futuras de este documento se explicarán algunas de esas funciones y cómo se utilizan en la fase de pruebas en la forma de un Simulador de Tráfico de Red.

Debido a las limitaciones de memoria del microcontrolador ATMEGA328P algunas opciones de la librería fueron deshabilitadas. Sin embargo, los dispositivos que utilicen esta librería en su totalidad, serán capaces de superar las pruebas de certificación de NMEA2000. Según su autor, existen dispositivos comerciales certificados que ya utilizan este software.

La Figura 3.4 muestra un diagrama en bloques de la estructura de las librerías mencionadas.

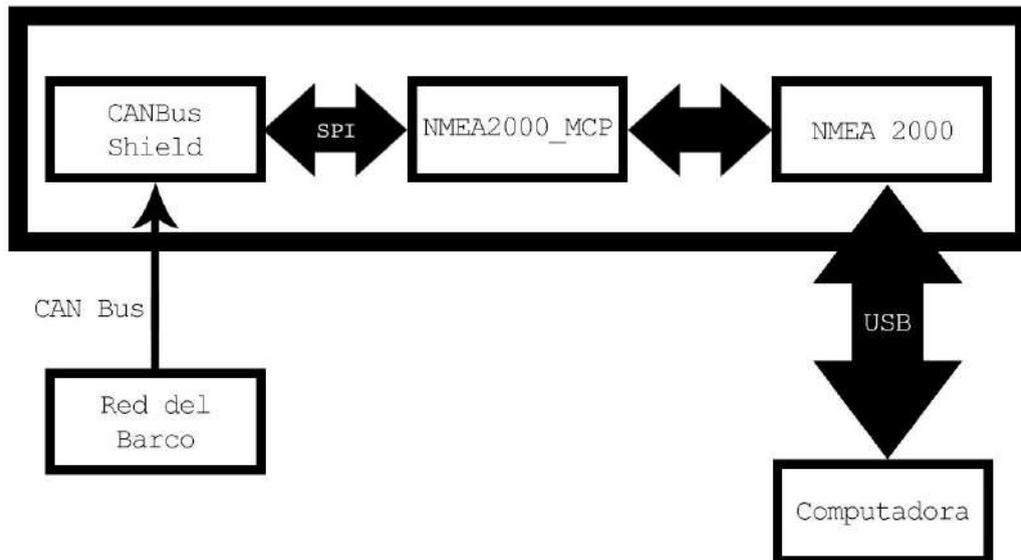


Figura 3.4 - Estructura NMEA2000 library for C++

El módulo Microcontrolador y NMEA2000 funcionan como un único bloque Adquisidor de Datos. Los datos de la red CAN llegan al microcontrolador por el bus SPI desde el controlador CAN. La librería procesa los datos hasta obtener un mensaje N2K sobre el cual se aplica un filtro según su PGN. El filtro se ejecuta en el microcontrolador comparando el mensaje recibido contra una lista definida de mensajes útiles. Si el mensaje es aceptado, una función destinada a procesar este tipo de mensajes es llamada para obtener los datos del mensaje N2K. Los datos obtenidos son guardados en la memoria del microcontrolador a la espera de un nuevo dato que lo actualice o una demanda de datos recibida en forma de comando AT en el puerto serie. En caso de que el mensaje N2K no fuera de utilidad, simplemente se ignora sin realizar ningún procesamiento.

La Figura 3.5 muestra el diagrama de flujo del filtrado de PGN realizado en el Adquisidor de Datos que fue explicado en el párrafo anterior.

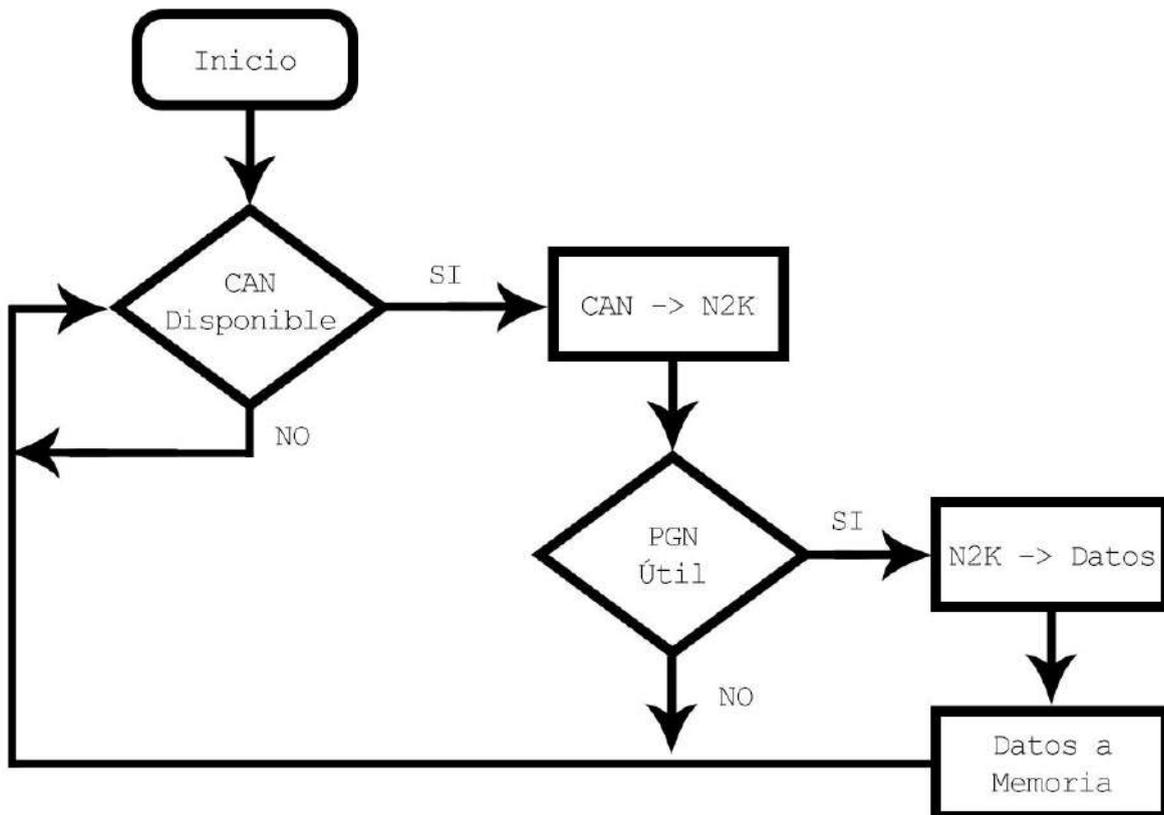


Figura 3.5 - Diagrama de Flujo del filtrado de PGN

Frente a la llegada de un comando AT en el puerto serie, se dispara una rutina que verifica el llamado y envía los datos guardados. La Figura 3.6 muestra el diagrama de flujo de la rutina de interrupción para el evento de recepción de un comando AT que requiera el envío de los datos almacenados en memoria.

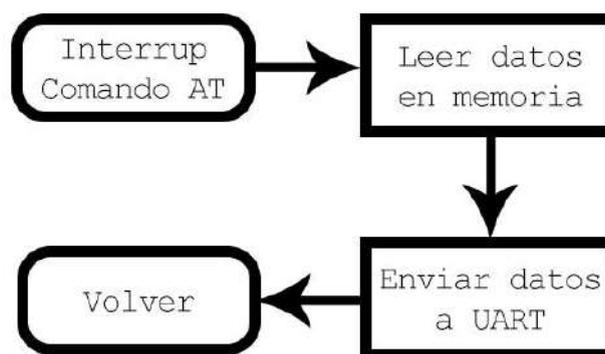


Figura 3.6 - Diagrama de Flujo de la rutina de interrupción por comando AT

3.3.4 Módulo UART-USB

El microcontrolador ATMEGA328P no posee un controlador USB de forma nativa. Para conectar el Adquisidor de Datos mediante puerto USB se utiliza un módulo comercial basado en el adaptador PL2303HX de la empresa Prolific Technology Inc [12]. Este circuito integrado es un controlador puente de USB a serie que provee una solución conveniente para conectar un dispositivo serie asincrónico con capacidad full-duplex a cualquier dispositivo con puertos USB. El driver provisto por el fabricante permite simular un puerto serie COM virtual en la mayoría de los sistemas operativos permitiendo que aplicaciones basadas en dichos puertos sean fácilmente migradas para utilizar puertos USB. La Figura 3.7 muestra una foto del módulo.



Figura 3.7 - Módulo UART-USB basado en PL2303HX

El módulo está integrado en una pequeña placa que contiene los componentes para configurarlo, un conector USB A macho y un puerto de 5 pines compatibles con UART. Para este proyecto, se removieron dichos conectores y se soldó el módulo a una placa que agrega además, optoacopladores para aislar eléctricamente al Adquisidor de Datos de la computadora.

Los optoacopladores utilizados son los 4N26 de la empresa Vishay. El encapsulado contiene un acoplador de fototransistor de canal único estándar. Cada optoacoplador consiste de un LED infrarrojo de arseniuro de galio y un fototransistor NPN de silicio [13]. La Figura 3.8 muestra el empaquetado y el circuito interno de optoacoplador.

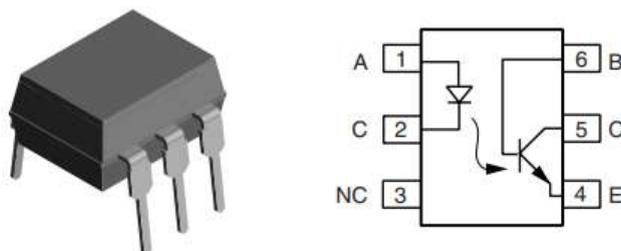


Figura 3.8 - Optoacoplador 4N26

Se fabricó una placa de circuito impreso y se montó un circuito con dos optoacopladores que aíslan las alimentaciones y las líneas de datos de la conexión serie, es decir un optoacoplador para la línea Rx y otro para la línea Tx. Se agregaron resistencias de polarización para los optoacopladores, pines para la conexión con la placa del Adquisidor de Datos y un puerto USB B hembra. La Figura 3.9 muestra el circuito implementado para el módulo completo.

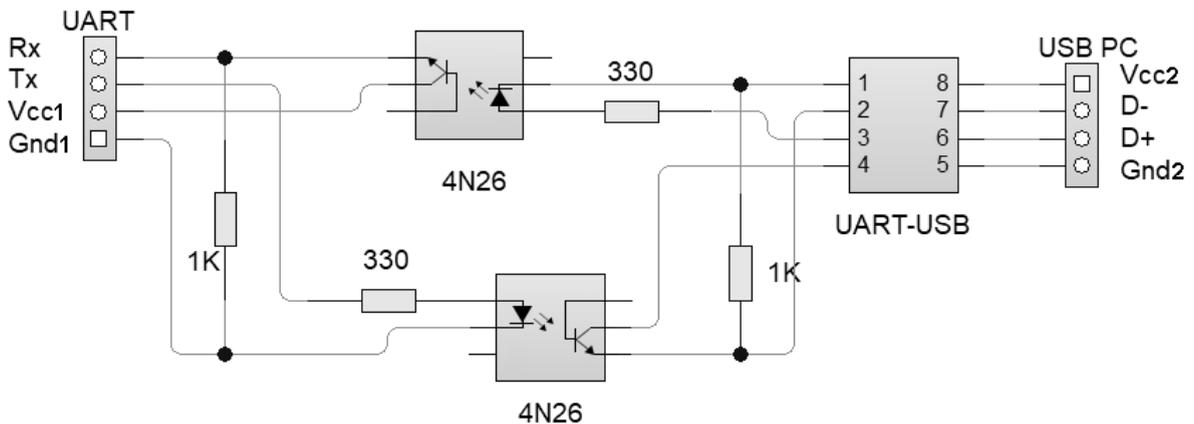


Figura 3.9 - Circuito completo del Módulo UART-USB con optoacopladores.

3.4 Computadora

3.4.1 Interfaz de Usuario

La interfaz de usuario del proyecto tiene como objetivo cumplir con los requisitos RF01, RF02 y RF03. El programa brinda una solución visual e interactiva a la carga de información necesaria por parte del usuario. Dicha información será clasificada como estática dado que está relacionada con características físicas de la embarcación y con datos de navegación que no deberían cambiar, en circunstancias normales, una vez iniciado el recorrido.

Además de estas funciones visuales, el programa contiene los módulos que controlan el Adquisidor de Datos y el SDR para la transmisión.

3.4.2 Comandos AT

Este módulo se encarga de realizar el pedido de datos actualizados al Adquisidor de Datos. Mediante el puerto serie envía un comando cuando requiere actualizar la información.

Los mensajes AIS son enviados con una frecuencia relativamente baja, siendo el mensaje 18 el más enviado con un intervalo de 30 segundos de repetición cuando la

embarcación se desplaza a más de 2 nudos. Debido a esto, carece de sentido tener una conexión serie que envíe datos continuamente, por ello se establece una frecuencia de actualización de datos de 1 Hz, es decir se envía el pedido de actualización una vez por segundo.

3.4.3 GNU Radio

GNU Radio es un conjunto de herramientas de desarrollo gratuito y de código abierto que provee bloques de procesamiento de señales para implementar con sistemas de radio mediante software [14].

El módulo en cuestión permite controlar mediante software el funcionamiento del SDR Pluto que provee las capacidades de transmisión y recepción de RF. Gracias a sus características de código abierto, GNU Radio cuenta con una gran cantidad de módulos y librerías que facilitan la creación de sistemas complejos que bajo otro entorno requieren un mayor trabajo de programación. Para este proyecto se utiliza GNU Radio v3.8 corriendo en un sistema Linux Ubuntu 22.04 LTS.

Para generar el sistema que transmite los mensajes AIS utilizando el SDR Pluto el entorno requiere la instalación de dos librerías.

La primera de ellas se llama AISTX, programada por el usuario zwh2 en GitHub [15]. Este desarrollo proviene de una serie de forks creados a partir de la librería original, gr-aistx de TrendMicro, que buscan actualizarla para nuevas versiones de GNU Radio.

La librería provee una serie de herramientas que facilitan la creación de los mensajes AIS. Agrega a GNU Radio un bloque llamado “AIS Frame Builder”, el cual toma un dato binario y le aplica las características determinadas por la norma para su transmisión.

Incluye una herramienta Encoder, en forma de script standalone en Python, que permite generar un dato binario a partir de información provista mediante línea de comandos. El dato obtenido como salida es el que debe utilizarse como entrada para el AIS Frame Builder.

La segunda librería simplemente genera una serie de bloques que permiten configurar algunos SDR comerciales de la empresa Analog Devices para ser utilizados en el diseño de sistemas con GNU Radio. En este caso, se utilizarán los bloques PlutoSDR Sink y en menor medida el PlutoSDR Source.

En los siguientes párrafos se explicará el funcionamiento del programa sin las funciones relacionadas con la transmisión RF. Esas funciones serán explicadas más adelante junto con los bloques que le corresponden.

Cuando el programa es ejecutado, lee un archivo de configuración. Este archivo guarda dos grupos de datos estáticos. El primer grupo contiene los datos de identificación de la embarcación: Nombre, MMSI, Tipo de Embarcación/Carga y Callsign. El segundo grupo contiene las medidas físicas de la embarcación. Este grupo se compone por 4 medidas en metros que representan la longitud, ancho y posición de referencia del dispositivo AIS. Para conocer en detalle acerca de estos valores referirse al documento ITU-R M.1371-4 Annex 7, página 108.

Los datos de este archivo son mostrados en la sección “Vessel Information” de la interfaz. Esta sección también provee un botón que permite modificar estos datos.

Luego de esta inicialización, el programa envía cada 1 segundo un comando AT al Adquisidor de Datos y éste devuelve en el puerto serie los últimos datos guardados. Los datos recibidos son Latitud, Longitud, SOG, COG y Heading y son mostrados al usuario en la sección “Vessel Readings”.

Según la norma, los datos de posición deben ser obtenidos de un sensor GPS que pertenezca al dispositivo AIS y sea independiente del GPS de la embarcación. Por razones económicas dicho sensor no fue implementado en el sistema y a cambio se obtienen esos datos del sensor de la embarcación. Un beneficio extra de esta decisión será explorado en la fase de pruebas del sistema.

La Figura 3.10 muestra una captura de pantalla de la interfaz de usuario con la ventana para edición de datos abierta.

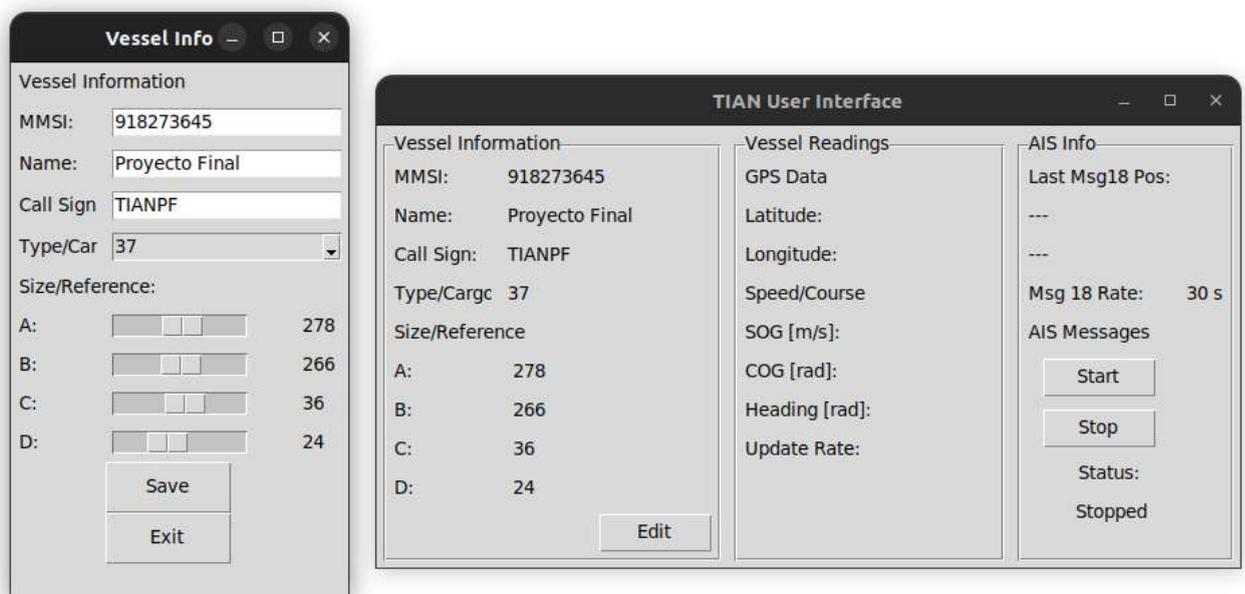


Figura 3.10 - Interfaz de Usuario

3.5 Transmisión

3.5.1 Adalm Pluto SDR

Los bloques precedentes se encargan de obtener y procesar la información de la embarcación para generar los mensajes AIS a transmitir. Esta etapa se encarga de la transmisión en sí misma mediante la implementación conjunta de GNU Radio y un dispositivo SDR.

Adalm Pluto SDR es una herramienta fácil de usar fabricada por Analog Devices que se puede utilizar para introducir nociones básicas acerca de SDR, radiofrecuencia y comunicaciones así como temas más avanzados en el campo de la ingeniería ya sea en forma guiada o autodidacta [16].

Basado en el transceptor de RF AD9363, el dispositivo ofrece un canal de recepción y un canal de transmisión que pueden ser operados en modo full-duplex con la capacidad de generar o medir señales analógicas de RF desde 325 a 3800 MHz, hasta un máximo de 61.44 Mega Samples por segundo con un ancho de banda de 20 MHz. Mediante la modificación del firmware, el rango de frecuencias puede ser extendido para funcionar desde 70 MHz a 6000 MHz. El dispositivo es alimentado mediante puerto USB y es compatible con los sistemas operativos Windows, OSX y Linux. La Figura 3.11 muestra una foto del dispositivo.



Figura 3.11 - Adalm Pluto SDR

3.5.2 Acceso al Medio

La normativa establece que el sistema AIS debe permitir la coexistencia de distintos dispositivos transmitiendo al mismo tiempo, para ello se implementa un esquema TDMA. En este esquema, cada minuto es dividido en 2250 slots de tiempo que son utilizados de a uno por dispositivo. Es decir, cada dispositivo debe utilizar solo los slots que estén disponibles o reservados.

Para verificar la disponibilidad de slots, existen dos métodos de acceso al medio. Los dispositivos clase A utilizan SOTDMA que les permite reservar slots así como saber qué slots están ya reservados. Debido a esta mecánica de reservas, estos dispositivos pueden incluso transmitir mensajes AIS que abarquen una longitud de múltiples slots.

Los dispositivos clase B utilizan CSTDMA como esquema de acceso al medio. Este método sensa el estado del canal para determinar si es posible realizar la transmisión. El dispositivo mide la potencia en el canal al inicio de cada slot y la compara con un umbral definido a partir del piso de ruido para determinar si está en uso o disponible. En caso de estar disponible, el mensaje es enviado. Debido a esta política de sensar slot por slot, este esquema sólo permite enviar mensajes que ocupen un único slot. En caso de que el canal esté en uso, la transmisión del mensaje se pospone.

Este bloque no existe físicamente en el dispositivo. Su funcionamiento es definido e implementado exclusivamente por software y se compone de una parte realizada mediante GNU Radio y otra programada directamente con la interfaz de usuario.

3.5.3 Amplificador 2W

La normativa establece una potencia de transmisión máxima para los transmisores AIS según su clase. Para un transmisor clase B la potencia permitida es de 2 watts. Debido a las limitaciones legales que establece el ENACOM, no está permitido transmitir en las frecuencias reservadas para AIS con esa potencia sin tener los permisos necesarios [17]. Por este motivo el amplificador no será implementado pero se incluye una propuesta de diseño como apéndice a este documento.

Cuando se inicia la transmisión, el sistema debe realizar un proceso de calibración. Dicho proceso tiene como objetivo establecer el piso de ruido del medio para poder determinar si los slots están disponibles.

El proceso de calibración dura un minuto completo. El dispositivo toma una muestra de potencia en el canal cada 4 segundos para un total de 15 muestras. Se identifica la de menor magnitud y se define como valor de piso de ruido. Se define, el umbral de sensado de canal como piso de ruido más 10dB. Contra este valor se comparará la potencia medida a la hora de enviar mensajes.

El dispositivo debe enviar tres tipos de mensajes AIS distintos, estos mensajes se detallan a continuación:

Mensaje 18: Informe de Posición de Equipo Clase B Estándar.

Mensaje 24A: Informe de Datos Estáticos Parte A.

Mensaje 24B: Informe de Datos Estáticos Parte B.

La norma establece la frecuencia de envío de cada uno de estos mensajes. El mensaje 18 debe ser enviado cada 3 minutos si la embarcación se desplaza con SOG menor a 2 nudos, en caso contrario, el mensaje debe ser enviado cada 30 segundos. El mensaje 24A debe ser enviado cada 6 minutos, y dentro del próximo minuto, debe ser enviado el mensaje 24B. La frecuencia de estos mensajes no varía por factores externos.

Los intervalos de tiempo son controlados por el programa en Python que genera la interfaz de usuario. El programa corre un temporizador de 30 segundos, cada iteración de tiempo se intenta enviar un nuevo mensaje AIS según la velocidad de desplazamiento y el tiempo que haya pasado desde los mensajes anteriores.

Como primer paso, el programa consulta una bandera de velocidad que permite determinar la velocidad de desplazamiento de la embarcación. Si SOG es mayor a 2 nudos, el programa inicia la rutina de envío para el MSG18. Si SOG es menor a 2 nudos, el programa realiza una comparación contra un timer global que permite identificar si han pasado 3 minutos desde el último envío de MSG18. Luego, se realiza la misma comprobación contra el timer global para MSG24A y MSG24B para sus respectivos tiempos de envío. Finalmente se programa un nuevo llamado a envíos en 30 segundos.

En caso de que las condiciones se cumplan, el programa activará una rutina de envío de mensajes. Esta rutina es la misma para todos los mensajes sin importar el tipo de mensaje o su frecuencia. La rutina inicia con la creación del mensaje a ser enviado mediante el script Encoder.py provisto por la librería AISTX previamente mencionada. Los datos obtenidos del Adquisidor de Datos son ingresados al script mediante línea de comandos. El script devuelve una cadena con valores binarios que representan la payload del mensaje a enviar.

Obtenida la payload del mensaje, inicia el proceso de transmisión. Para ello la norma establece un procedimiento específico para los dispositivos clase B. Como primer paso se definen 10 posibles slots candidatos para la transmisión. Estos candidatos son slots elegidos al azar de los posibles 2250 en que se divide el minuto. Con los candidatos definidos, el sistema espera a la llegada de un candidato y mide la potencia en el canal. Si la potencia supera el umbral establecido, canal en uso, se aplaza el envío para el próximo candidato disponible. Esto se repite hasta que el mensaje es enviado o no hay más candidatos disponibles, en cuyo caso el mensaje no se envía.

Cuando un slot disponible es encontrado, se inicia la transmisión del mensaje. Para completar esta tarea, el programa llama a un script de Python basado en un diagrama de GNU Radio.

La norma establece para la transmisión de los mensajes AIS la codificación binaria NRZI y modulación GMSK con B=0.4. Estas características se logran por etapas utilizando un diagrama de GNU Radio que utiliza el bloque AIS Frame Builder de la librería AISTX y el bloque PlutoSDR Sink de la librería de Analog Devices. La Figura 3.12 muestra el diagrama de GNU Radio.

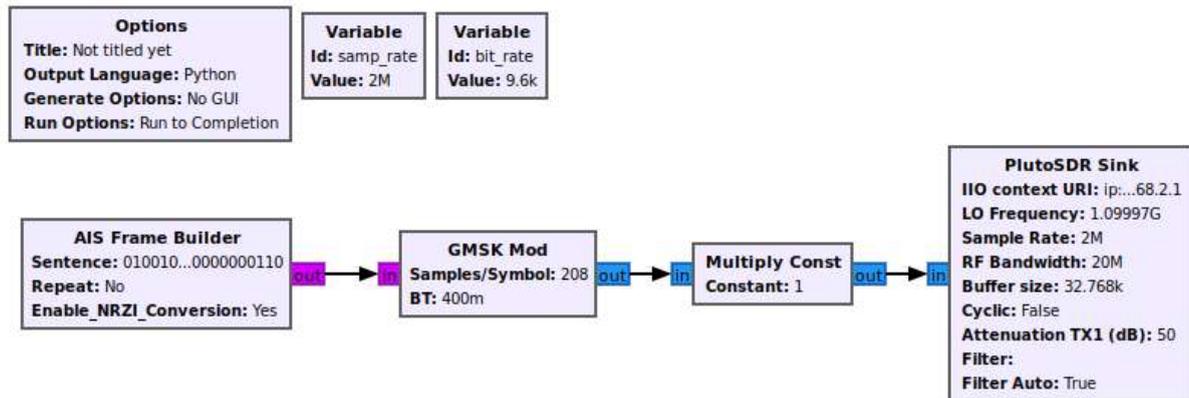


Figura 3.12 - Diagrama del Transmisor en GNU Radio

A partir de este diagrama, se exporta el código Python que será llamado como script desde el programa principal para transmitir el mensaje. Este código tiene algunas características especiales agregadas luego de ser exportado. La ejecución del script no muestra al usuario ninguna salida o interfaz gráfica. Los parámetros requeridos para su funcionamiento son enviados por consola, para ello se programó manualmente las funciones que procesan dichos parámetros y los ingresan en los bloques de GNU Radio.

Capítulo 4: Montaje

Para el montaje del sistema se diseñaron e implementaron dos placas en circuito impreso. El diseño de las placas fue realizado con el software PCB Wizard [18]. Todas las placas fueron fabricadas de forma manual y con herramientas e insumos disponibles en el LAC.

4.1 Adquisidor de Datos

El diseño de la placa del Adquisidor de Datos se realizó teniendo como objetivo colocarla en un pequeño gabinete con una cantidad mínima de conectores que permitan su funcionamiento.

La placa provee contactos para la conexión de los puertos necesarios: alimentación, bus N2K y puerto USB. Además, se colocaron contactos para funciones especiales como encendido y apagado, botón de reseteo y capacidad de programación on-board mediante conector ICSP. La placa elegida para la implementación es una placa simple faz fenólica. La Figura 4.1 muestra el diseño final en el software.

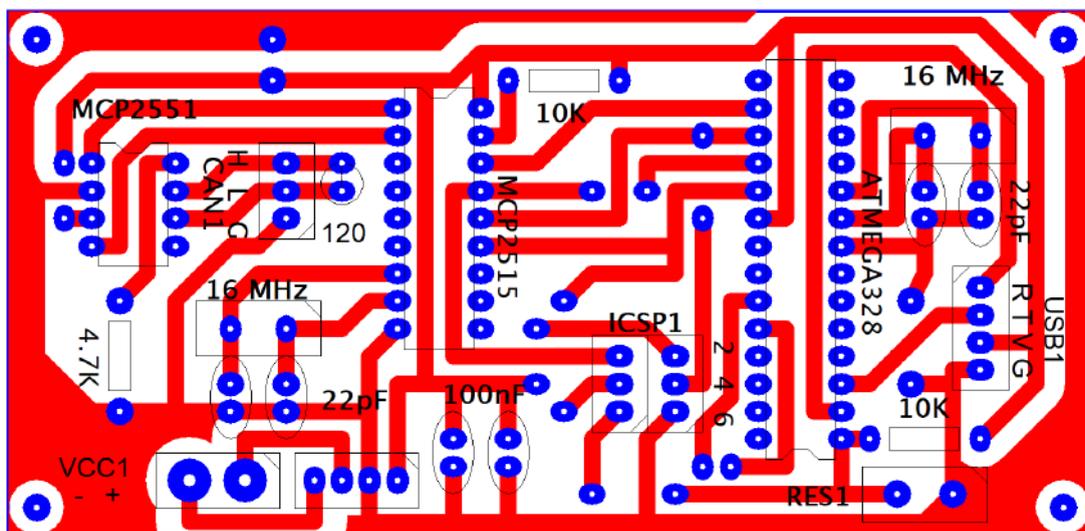


Figura 4.1 - Diseño Placa Adquisidor de Datos

Originalmente se planeó hacer una placa de prueba para comprobar el correcto funcionamiento y luego enviar a fabricar la placa en un taller especializado. Sin embargo, al ver los buenos resultados obtenidos con el proceso de fabricación, se decidió realizar todas las placas de esta manera. La Figura 4.2 muestra la placa fabricada.

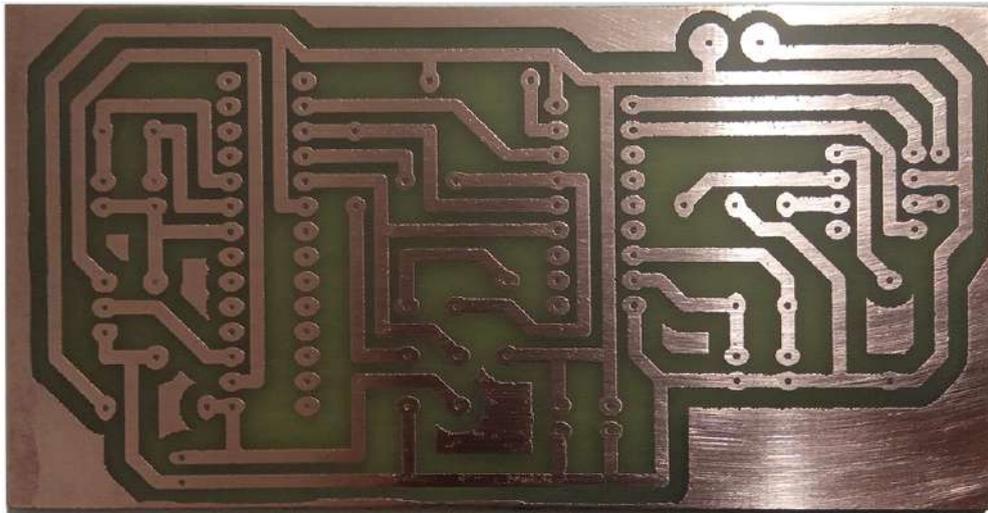


Figura 4.2 - Placa de prueba Adquisidor de Datos

Una vez probado el circuito en la placa de prueba, se acomodaron algunos componentes dando por terminado el diseño. Se fabricaron tres placas durante el mismo proceso, dos placas de Adquisidor de Datos y una para el módulo UART-USB.

Se hicieron las perforaciones necesarias con el taladro de banco del LAC y se procedió a soldar los componentes. Para los circuitos integrados se utilizaron zócalos en caso que se requiera alguna corrección. Para los contactos de los puertos se colocaron postes para mayor facilidad en caso de necesitar conectar o desconectar los respectivos cables. Las Figuras 4.3 y 4.4 muestran el resultado de la placa con los componentes soldados.



Figura 4.3 - Placa Adquisidor de Datos Componentes

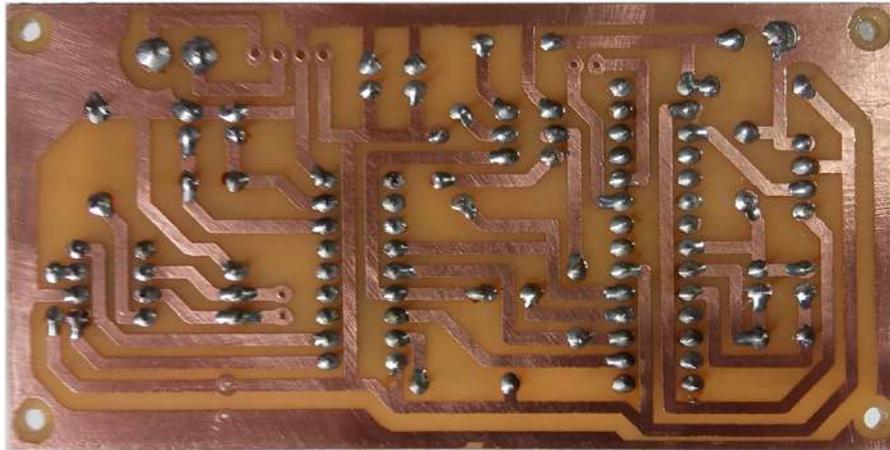


Figura 4.4 - Placa Adquisidor de Datos Soldadura

4.2 Módulo UART-USB

Como se mencionó en el capítulo anterior, el objetivo de implementar este módulo en una placa separada es agregar una etapa de protección al agregar optoacopladores en las líneas de datos de UART.

En este caso, la placa se diseñó considerando la eliminación de los puertos que incluía el módulo original. Estos puertos fueron reemplazados en el diseño con postes en la ubicación correcta, lo que permitió fijarlo a la placa. Se incluyó además, zócalos para los optoacopladores, sus resistencias de polarización, una serie de postes que funcionan como puerto de entrada para el bus UART proveniente del Adquisidor de Datos y un puerto USB tipo B hembra que provee la conexión con la computadora y eventualmente el programa en Python. Nuevamente, se utilizó una placa fenólica simple faz para el circuito impreso. La Figura 4.5 muestra el diseño final para este módulo.

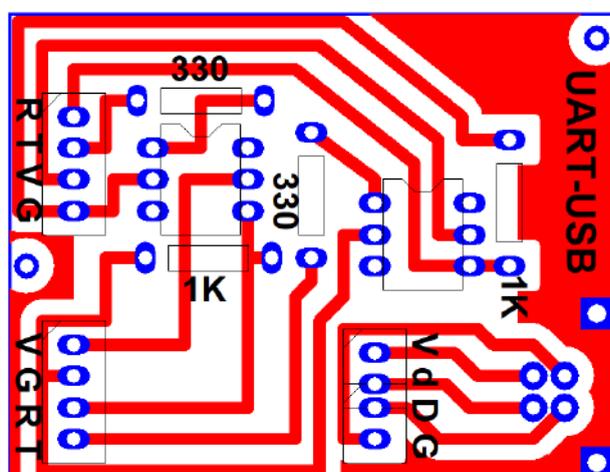


Figura 4.5 - Diseño Placa Módulo UART-USB

El proceso de fabricación fue el mismo que para la placa anterior. Una vez realizadas las perforaciones para los componentes se procedió a soldarlos y comprobar su funcionamiento. Las siguientes figuras muestran la placa antes de perforar y el resultado final con los componentes soldados.

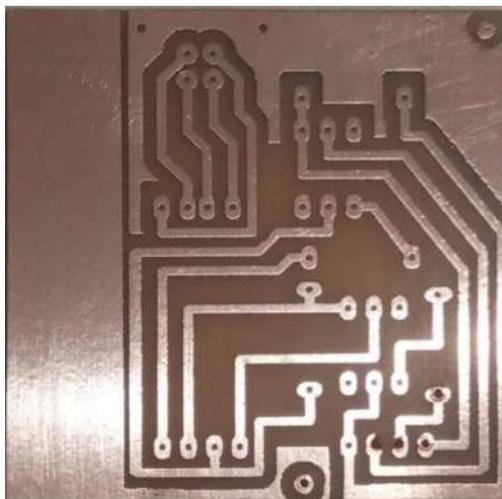


Figura 4.6 - Placa Módulo UART-USB



Figura 4.7 - Placa Módulo UART-USB Componentes

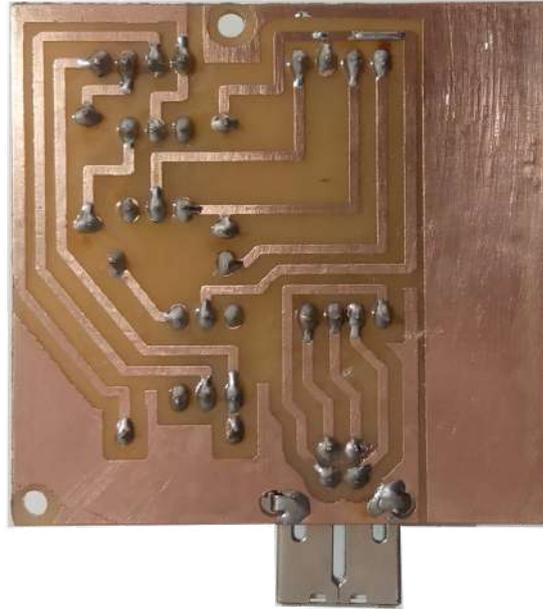


Figura 4.8 Placa Módulo UART-USB Soldadura

4.3 Gabinete

Finalmente, con ambas placas soldadas y funcionando se procedió a colocarlas en un gabinete plástico adaptado para alojarlas. El gabinete en cuestión permite además, colocar los distintos tipos de conectores y botones necesarios para el funcionamiento del dispositivo.

En el frente se colocaron el interruptor de Encendido y Apagado, el pulsador para el Reset, el conector USB B Hembra para la conexión con la computadora y el puerto DB9 para el bus N2K.

En la parte posterior del gabinete se encuentra un conector para la fuente de alimentación.

Las Figuras 4.9 y 4.10 muestran el gabinete finalizado tanto en el exterior como en el interior respectivamente.



Figura 4.9 - Adquisidor de Datos Exterior

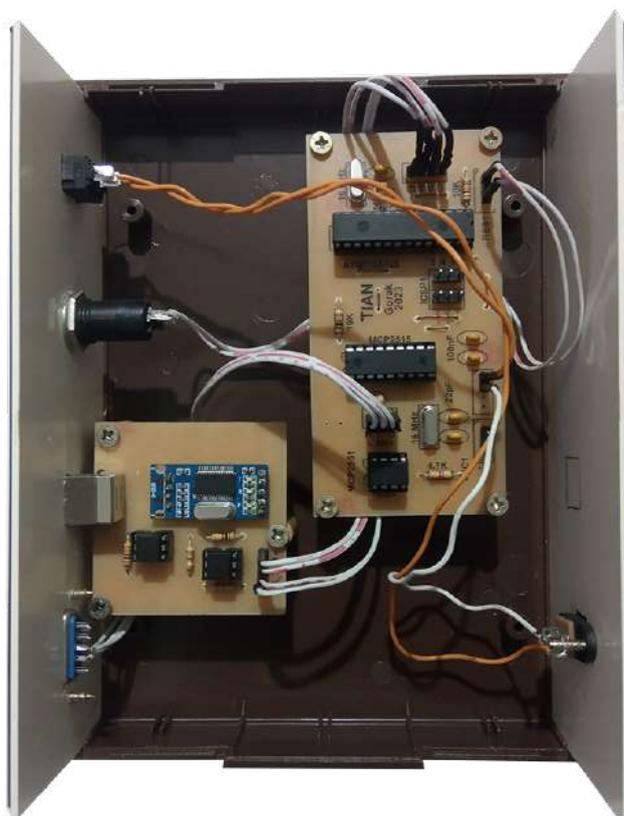


Figura 4.10 - Adquisidor de Datos Interior

Capítulo 5: Plan de Pruebas

En este capítulo se presenta un resumen de las pruebas más significativas realizadas en el marco del proyecto. Durante el desarrollo, se realizaron distintas pruebas para confirmar el funcionamiento de los módulos individualmente. Finalmente, se montó un banco de pruebas que permite apreciar el funcionamiento completo del sistema desde la Adquisición de Datos hasta la transmisión por RF de los mensajes AIS. Se adjunta al documento, como apéndice, el detalle completo del Plan de Pruebas.

5.1 Pruebas Unitarias

Durante las etapas de diseño se realizaron pruebas para comprobar que los módulos funcionaran como se esperaba previo al montaje del sistema.

En ese sentido se hicieron pruebas de comunicación en el bus CAN y con las librerías NMEA2000 para confirmar la capacidad de comunicación del Adquisidor de Datos. Se realizaron pruebas de comunicación serie entre el microcontrolador y el programa de Interfaz de Usuario, lo que confirmó además, el funcionamiento del módulo UART-USB.

Finalmente, se probaron las librerías de GNU Radio mediante la transmisión de datos estáticos con PlutoSDR. Llegado a este punto, solo quedó por realizar las pruebas finales sobre el sistema completo funcionando.

Para un mayor detalle de las pruebas unitarias y sus resultados, se recomienda leer la sección Casos de Prueba del apéndice Plan de Pruebas.

5.2 Banco de Pruebas

El banco de pruebas consta de tres bloques principales. El primer bloque es el Simulador de Tráfico de Red, encargado de generar paquetes en el bus N2K. El segundo bloque es el dispositivo TIAN en sí mismo y el tercer bloque es una computadora con un RTL-SDR y los softwares SDRAngel y OpenCPN que funciona como dispositivo receptor. La Figura 5.1 muestra el diagrama en bloques del banco completo.

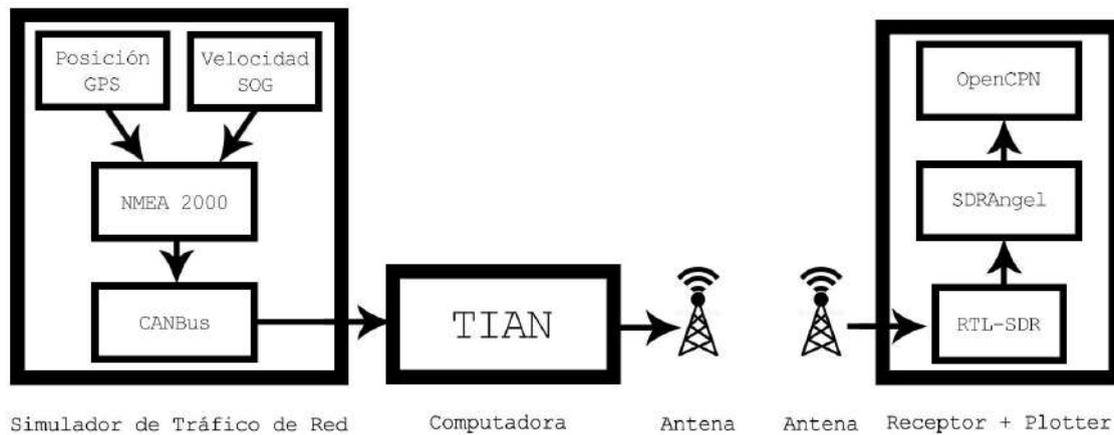


Figura 5.1 - Banco de Pruebas Diagrama en bloques

5.3 Simulador de Tráfico de Red

Durante las etapas iniciales de desarrollo, se hicieron pruebas básicas de comunicación. Debido a la necesidad de confirmar el correcto funcionamiento de la librería NMEA 2000, se generó un pequeño programa que permitía el envío de paquetes N2K al bus CAN. Al mismo tiempo, los paquetes se envían por puerto serie a una PC. Luego de implementar dicha comunicación exitosamente, se decidió modificar este programa para utilizarlo en el banco de pruebas como un dispositivo capaz de generar y enviar paquetes al bus, simulando de esta forma el tráfico de la red N2K de una embarcación real.

El Simulador cumple la función de enviar tramas N2K al Adquisidor de Datos como si se tratara de una embarcación real con mensajes que se programaron directamente en el microcontrolador.

Los mensajes programados se dividen en dos categorías para la realización de pruebas. Por un lado se envían los mensajes que TIAN necesita para transmitir por RF la información de AIS. La otra categoría son mensajes de descarte o relleno. Los mensajes de descarte no tienen utilidad para el sistema creado y serán filtrados por el Adquisidor de Datos. El simulador cuenta con dos entradas en forma de potenciómetros que permiten cambiar en tiempo real los valores de algunos de los datos generados por la embarcación simulada.

Como se mencionó anteriormente, la norma dicta que los dispositivos AIS utilicen un sensor GPS propio y no el que provee la embarcación. Debido a su costo, este módulo no fue implementado en el sistema, por ello la información de posicionamiento se genera en el Simulador de Tráfico de Red. Este contratiempo genera a su vez la oportunidad de controlar la información GPS desde el programa cargado al microcontrolador, para ello se utilizan potenciómetros conectados a las entradas

analógicas del ATMEGA328P. Para este banco se asocian estas entradas con la posición GPS y la velocidad SOG de la embarcación.

Se eligieron estas dos entradas como variables por su importancia a la hora de generar cambios notables en los mensajes AIS transmitidos. La posibilidad de variar la velocidad de desplazamiento de la embarcación se traducirá en una variación en la frecuencia de transmisión del MSG18 como se explicó en el capítulo 3. Por otro lado, modificar manualmente las coordenadas GPS que recibe el Adquisidor de Datos nos permitirá ver “desplazamientos” en el bloque receptor del banco de pruebas.

Para la construcción del Simulador se utilizó una de las dos placas fabricadas para el Adquisidor de Datos ya que las placas de ambos dispositivos son exactamente iguales. En el caso del simulador, el bus para el módulo UART-USB no tiene uso y queda libre en caso de requerir alguna actualización del programa. En cambio, los dos contactos a las entradas analógicas del microcontrolador, que en el Adquisidor de Datos no tienen uso, ahora son las entradas a las que se conectan los potenciómetros que controlan el GPS y SOG. El resto de los conectores tiene la misma utilidad en ambas placas: Encendido, Reset y bus N2K.

Una vez soldados los componentes a la placa, se preparó un gabinete que pudiera contenerla juntos con las entradas y salidas ya mencionadas. En este caso se utilizó un gabinete metálico al que se le realizaron las perforaciones necesarias para acomodar la placa, los potenciómetros y los conectores. Se acondicionó el frente del gabinete con leyendas para cada elemento disponible. Las Figuras 5.2 y 5.3 muestran el gabinete del Simulador ya ensamblado.



Figura 5.2 - Simulador de Tráfico de Red Exterior

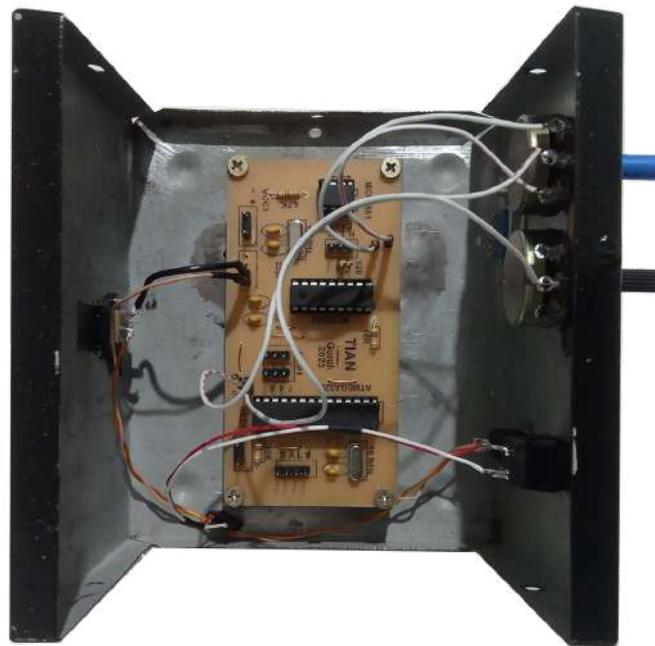


Figura 5.3 - Simulador de Tráfico de Red Interior

Para comprobar la correcta creación y envío de mensajes se conectó el Simulador, mediante el adaptador UART-USB, a una computadora corriendo el software Actisense NMEA Reader [19]. Este software de la empresa Actisense permite escanear el tráfico de una red NMEA 2000 con un adaptador USB. En la Figura 5.4 se observan los paquetes que genera el Simulador y recibe el programa.

Line	PGN	SRC	DST	Name	Time	Interval	Data	
1	130312	22	255	Temperature - DEPRECATED	16:19:08:505	2.50	01 01 04 B3	NMEA 2000 PGN: 130312 (0x1FD08) Name: Temperature - DEPRECATED Source = 22, Destination = 255 Priority = 5, Length = 8 Number Of Fields = 6 Field 1: Sequence ID = 1 Field 2: Temperature Instance = 1 (Generic Instance value ranging over 8-bit range -) Field 3: Temperature Source = 4 (Main Cabin Temperature) Field 4: Actual Temperature = 273.15 Kelvin Field 5: Set Temperature = Data not available Field 6: Reserved field
2	130311	22	255	Environmental Parameters...	16:19:08:505	2.50	01 C4 B3 6	
3	130310	22	255	Environmental Parameters...	16:19:08:506	2.50	01 C1 70 F	
4	129025	22	255	Position, Rapid Update	16:19:08:518	2.50	00 C5 53 E5	
5	129026	22	255	COG & SOG, Rapid Update	16:19:08:546	2.50	01 FC 5C 3	
6	127290	22	255	Vessel Heading	16:19:08:572	2.50	01 AD 1E F	

Figura 5.4 - Paquetes recibidos por Actisense NMEA Reader

5.4 Receptor

Una vez conectado el Simulador de Tráfico de Red al dispositivo TIAN, se puede observar los datos provenientes de la embarcación así como los datos AIS enviados en su interfaz. Sin embargo, para corroborar que los mensajes AIS son efectivamente enviados se requiere un dispositivo que sea capaz de recibir y decodificar esos mensajes.

Con ese objetivo se montó un sistema que utiliza un RTL-SDR conectado a una computadora corriendo los programas SDRAngel y OpenCPN simultáneamente.

5.4.1 RTL-SDR

El primer bloque del receptor es un SDR basado en el chip RTL2832U [20]. El RTL-SDR Blog V3 es un dispositivo RTL-SDR mejorado. Estos dispositivos fueron originalmente diseñados para la recepción de televisión de alta definición, pero hackers de hardware encontraron que podían ser utilizados como SDR de propósitos generales. Los dispositivos estándar funcionaban correctamente para la recepción de televisión pero no eran suficiente para usuarios más experimentados en SDR. El RTL-SDR Blog V3 fue rediseñado para los usuarios de SDR ya que los usuarios de receptores de televisión tienen típicamente requerimientos menos exigentes.

Información Básica:

- Ancho de Banda: estable hasta 2.4 MHz.
- ADC: RTL2832U 8-bits
- Rango Frecuencia: 500 kHz – 1766 MHz (500 kHz – 24 MHz modo direct sampling)
- Impedancia de Entrada típica: 50 Ohms
- Consumo de Corriente típico: 270 – 280 mA

Si bien está limitado sólo a la recepción y no permite la transmisión, es suficiente para recibir la señal que genera y transmite TIAN. En la Figura 5.5 se puede observar una foto del dispositivo.



Figura 5.5 - RTL-SDR Blog V3

5.4.2 SDRAngel

Este software de código abierto provee una serie de herramientas compatibles con distintos SDR. El programa puede utilizar un SDR como dispositivo de entrada, obtener las muestras recibidas y procesarlas con distintos demoduladores y decodificadores. Dependiendo del tipo de entrada, ofrece también herramientas gráficas tales como gráficos del espectro de potencia o gráficos de cascada. Si la señal de entrada provee información de posicionamiento, el software incluye un complemento de ploteo que muestra los datos directamente en un mapa del mundo [21].

Para el receptor AIS, se utilizaron dos funciones de este software. La primera función es el Demodulador AIS. Este complemento utiliza los datos de entrada del RTL-SDR a la frecuencia central establecida por el usuario, tomando los datos de los dos canales AIS simultáneamente, cada uno de ellos separado a 25 kHz de dicha frecuencia. Los datos recibidos son mostrados en dos ventanas dentro del programa como tablas donde se muestra la información contenida en cada mensaje recibido.

La segunda función le permite al programa tomar estos datos decodificados y enviarlos para ser procesados o mostrados mediante otro software a través de una red con una dirección IP y un puerto. Para este banco de pruebas, los datos son enviados a un software cuya función exclusiva es la de ploteo. En la Figura 5.6 se puede observar una captura del programa.

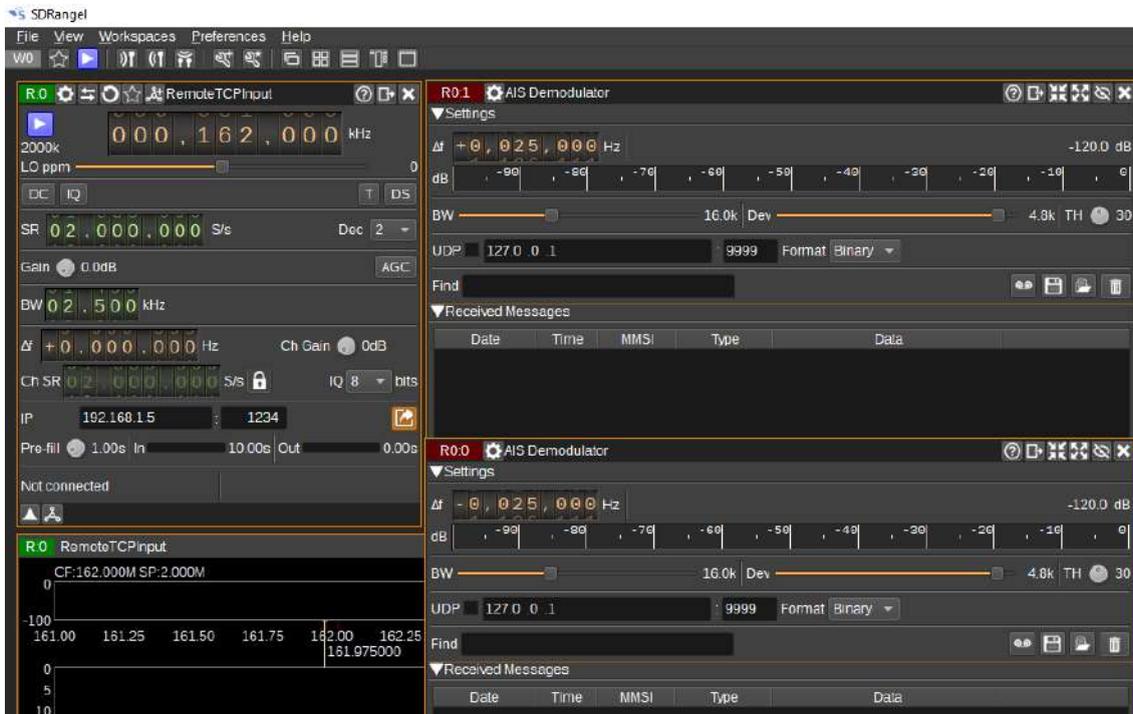


Figura 5.6 - SDRAngel

5.4.3 OpenCPN

El último bloque del receptor es un software cuya única finalidad es mostrar gráficamente la información que se le provee. Configurado para recibir la información procesada por SDRAngel, OpenCPN muestra en su interfaz las embarcaciones que hayan sido detectadas por el RTL-SDR [22]. En la Figura 5.7 se puede apreciar una captura del programa.

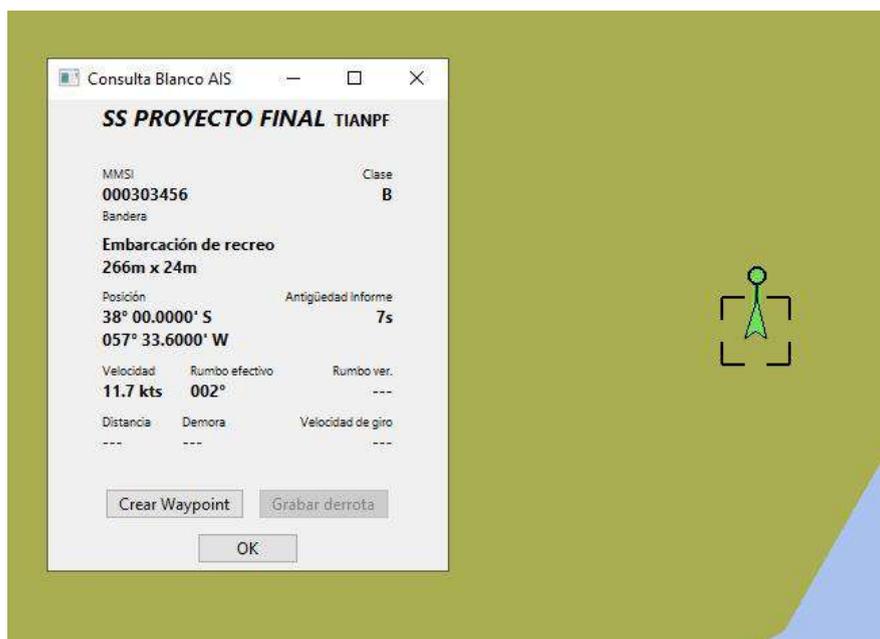


Figura 5.7 - OpenCPN

Cuando se inicia el Simulador de Tráfico de Red, genera 6 tipos distintos de paquetes N2K. De estos paquetes, tres son los considerados útiles por el Adquisidor de Datos. Los paquetes son:

Rapid Position Update [PGN 129025]: Provee los datos de posicionamiento divididos en Latitud y Longitud como coordenadas GPS.

Rapid SOG/COG [PGN 129026]: Provee los datos de velocidad sobre fondo (SOG) y rumbo de la embarcación (COG).

Heading [PGN 127250]: Provee los datos de orientación de la embarcación.

Los tres paquetes restantes generados, son datos de temperatura de la cabina y parámetros ambientales. Estos datos no son de utilidad en este proyecto pero se agregan solo para demostrar la capacidad de filtrado del Adquisidor de Datos. El Simulador envía cada 2 segundos todos los paquetes.

El Adquisidor de Datos recibe esos paquetes y recopila la información como ya se explicó anteriormente. Envía los datos a la Interfaz de Usuario y desde allí es transmitida como mensajes AIS.

El receptor es configurado a la misma frecuencia central que la frecuencia de envío del PlutoSDR, el complemento demodulador de AIS de SDRAngel recibe estos mensajes y los decodifica. En su interfaz se puede observar la llegada de cada mensaje por separado en una tabla que muestra una marca de tiempo de llegada del mensaje, el mensaje codificado y toda la información obtenida de dicho mensaje. Conforme nuevos mensajes llegan, se van agregando al final de la tabla lo que permite analizar los mensajes recibidos de forma sencilla. En esta misma pantalla, se configuró el envío de la información a una dirección IP y un puerto para ser utilizada por el software de ploteo OpenCPN. En este caso, la IP corresponde a la misma computadora por lo que se configura como 127.0.0.1 o localhost.

El último bloque se configura agregando una Conexión que le indique a OpenCPN que dirección IP y puerto escuchar para graficar en el mapa. Se configuró la conexión con los datos usados previamente en SDRAngel y el sistema quedó completo graficando los datos que se reciben.

Este banco receptor fue probado no solo con TIAN, sino que también fue configurado y probado con transmisiones reales en cercanía a la zona portuaria de Mar del Plata. En la Figura 5.8 y 5.9 se puede apreciar el funcionamiento de SDRAngel y OpenCPN respectivamente.

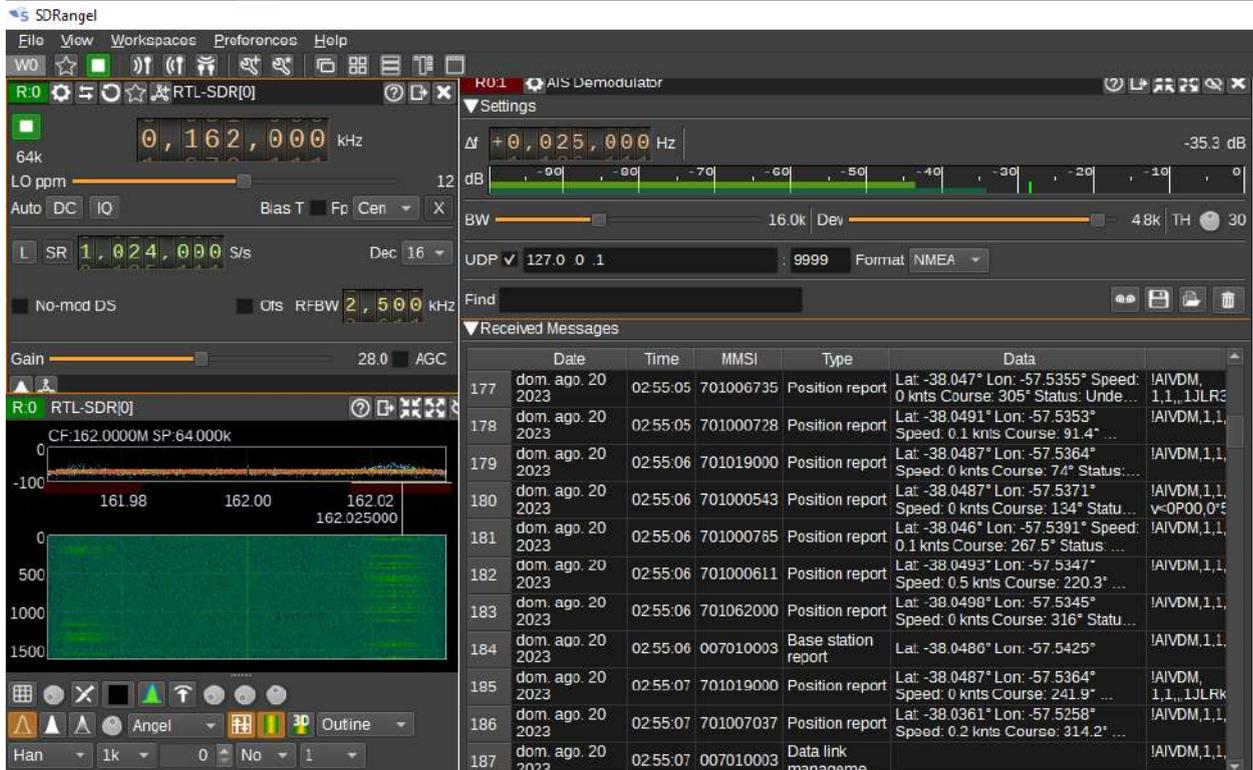


Figura 5.8 - SDRAngel Demodulador AIS

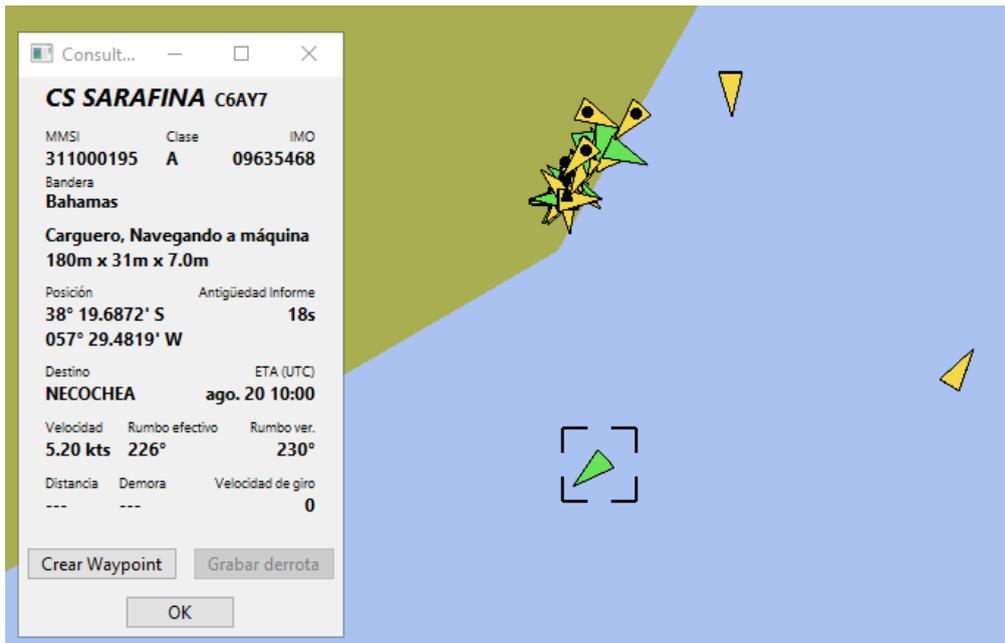


Figura 5.9 - OpenCPN Recepción AIS

5.5 Prueba Integral

Habiendo montado el banco de pruebas se procedió a realizar pruebas de funcionamiento del sistema completo. Resulta importante recordar en este punto que todas las pruebas fueron realizadas sin la implementación del bloque Amplificador de 2 Watts y a una frecuencia distinta de la utilizada por el sistema AIS debido a la normativa establecida por el ENACOM para este tipo de transmisiones [17].

Se conectó el Simulador de Tráfico de Red al Adquisidor de Datos mediante un cable con conectores tipo DB9 preparados para tal fin. Se confirmó la recepción de los datos observando la salida del puerto serie. Para ello se utiliza un monitor serie que muestra la información enviada por el puerto serie desde el Adquisidor hacia la Interfaz de Usuario. En la Figura 5.10 se puede observar dicha salida y como el Adquisidor de Datos responde frente al comando AT.

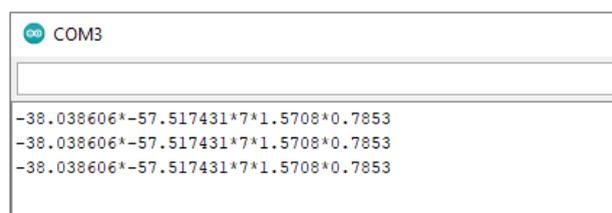


Figura 5.10 - Información enviada por puerto serie.

Para la siguiente prueba se ejecutó el programa que corre la Interfaz de Usuario, debido a la necesidad de correr distintas librerías de Python y GNU Radio en el mismo entorno el programa se corre en una máquina virtual corriendo Linux.

Una vez presentada la interfaz, la sección Vessel Readings debe mostrar los datos recibidos. En este momento se varían los potenciómetros asignados a GPS y SOG para confirmar que los datos son generados, enviados y recibidos por el programa. En la Figura 5.11 se puede observar esta sección recibiendo datos.

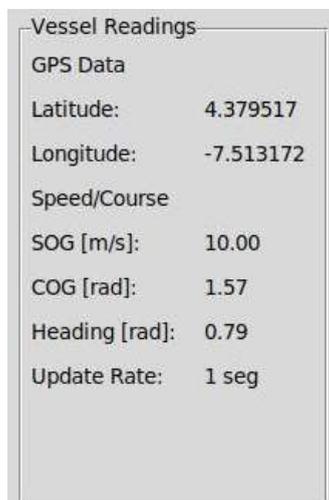


Figura 5.11 - Sección Vessel Readings

El siguiente paso es la prueba de transmisión integral. Para llevarla a cabo se conecta el PlutoSDR mediante el puerto USB de la computadora y se presiona el botón Start. Como se había mencionado, al iniciar el dispositivo, se ejecuta un proceso de calibración. Se realizan 15 mediciones del espectro de potencia a las frecuencias de interés y se obtiene el nivel de piso de ruido. La Figura 5.12 muestra una captura de las mediciones realizadas durante este proceso.

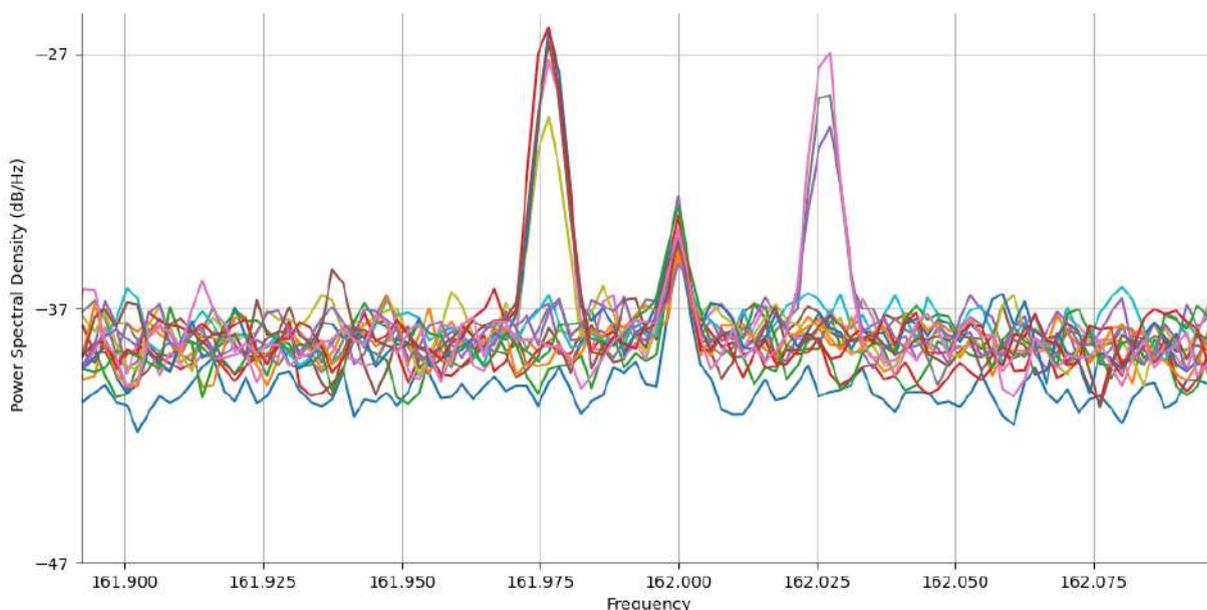


Figura 5.12 - Espectro de potencia medido durante calibración.

A partir de este punto existen varias formas de corroborar el envío de los mensajes.

En la interfaz de Usuario, en la sección AIS Info se podrá apreciar cuál es el último mensaje 18 transmitido. Esto solo será posible en la medida que el programa esté efectivamente transmitiendo.

Otra forma de control es la consola del sistema operativo. Si el modo DEBUG en el programa está activado, la consola mostrará una serie de mensajes que indican en que parte del proceso de envío se encuentra. Se podrá observar todo el proceso relacionado con el envío de mensajes AIS como ser la selección de los 10 slots candidatos, el proceso de medición para el sensado de canal o el envío de los paquetes AIS entre otros datos.

Finalmente, y la forma más ilustrativa de visualizar el funcionamiento del sistema se realiza con el receptor del banco antes mencionado. Se montó en la cercanía de TIAN, una computadora con un RTL-SDR corriendo SDRAngel y OpenCPN. Debido a que ENACOM prohíbe la transmisión a la frecuencia de 162 MHz, la frecuencia programada en TIAN y SDRAngel es 1.1 GHz a los efectos de no transmitir en frecuencias reservadas. Simultáneamente, y para evitar la radiación indiscriminada, la atenuación de salida del PlutoSDR se configuró en -30 dB. La detección del medio, por otro lado, se realiza a la frecuencia establecida por la norma dado que no hay transmisión y sólo recepción.

En este caso, los únicos mensajes recibidos serán los enviados por TIAN. Al igual que en la sección anterior, se podrá observar en SDRAngel las tablas de recepción de los mensajes AIS y en OpenCPN el mapa ploteado con la posición y los datos de la embarcación generados por el Simulador de Tráfico de Red y transmitidos por TIAN. Las Figuras 5.13 y 5.14 muestran las pantallas de los programas corriendo en el receptor.

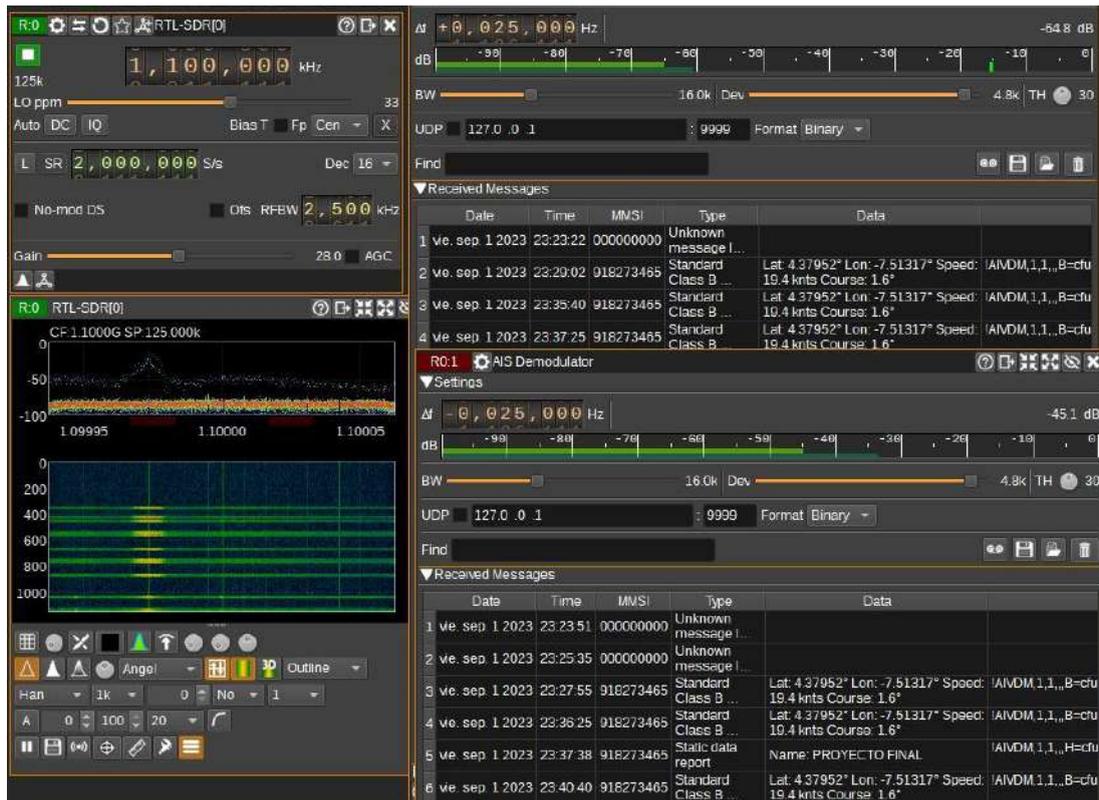


Figura 5.13 - Mensajes TIAN en SDRAngel



Figura 5.14 - Información TIAN en OpenCPN

Capítulo 6: Conclusiones

6.1 Acerca del dispositivo

La implementación final del dispositivo fue un éxito. En la etapa de pruebas se pudo comprobar el correcto funcionamiento de todas las funciones y soluciones implementadas. La activación del modo DEBUG en la Interfaz de Usuario permitió controlar el proceso completo de funcionamiento del sistema como una demostración culminante de todo el desarrollo realizado durante los últimos 22 meses.

Cada etapa del desarrollo llevó aparejado un proceso de aprendizaje específico relacionado con la tecnología a implementar en dicha etapa. Para la etapa correspondiente al Adquisidor de Datos tuve que estudiar el funcionamiento de las redes de las embarcaciones, las cuales hasta ese momento desconocía su existencia. Esto a su vez me llevó a investigar el protocolo NMEA 2000 y la forma en que las distintas empresas comerciales lo implementan en sus dispositivos. También, se definió la tecnología a utilizar dada la disponibilidad de librerías de código abierto y el conocimiento previo de haber cursado la materia Instrumentación Virtual.

La Interfaz de Usuario me dió la oportunidad de iniciarme en la programación en Python. Sabiendo que la implementación del SDR iba a ser en GNU Radio, resultó conveniente aprender las bases de Python con un programa menos elaborado, como es una interfaz visual, antes de afrontar dicha tarea. También en esta etapa, se volvió evidente la necesidad de migrar el proyecto de un entorno en Windows a uno en Linux que permitió una mayor libertad a la hora de utilizar distintas librerías de forma simultánea. La preparación de todo el sistema operativo para correr dichas librerías fue un proceso de aprendizaje en sí mismo.

La etapa de transmisión era la mayor incógnita durante el planeamiento del proyecto. Si bien conocía la existencia de los SDR, no había trabajado nunca con alguno hasta este punto. En el proceso de desarrollo de esta etapa aprendí a utilizar GNU Radio, sus librerías y a exportar scripts que eventualmente modifiqué a mano para que se adapten correctamente al funcionamiento que pretendía para el sistema.

Finalizado el desarrollo y las etapas de pruebas, dediqué mi tiempo a estudiar las distintas posibilidades de implementación con placas de circuito impreso. Si bien tenía alguna experiencia previa fabricando placas durante mis estudios secundarios, la necesidad de realizar un trabajo prolijo y a la altura esperada de un proyecto final de carrera me llevó a analizar nuevas alternativas. Luego de realizar algunas pruebas con métodos caseros y obtener resultados variados, conseguí refinar el proceso lo

suficiente para generar resultados satisfactorios. De esta forma puedo asegurar que adquiriré nuevas habilidades para la fabricación de circuitos impresos.

La implementación del dispositivo no solo representa el conocimiento adquirido a lo largo de la carrera, sino que agrega además los conocimientos específicos que tuve que adquirir conforme avancé sobre los distintos problemas que se fueron presentando. En este sentido, el proyecto no concluye sólo como un proceso de aplicación de conocimientos previos sino como un nuevo estadio de aprendizaje y una demostración de la capacidad de adquirir nuevas herramientas cuando fue necesario.

6.2 Acerca del proceso

Contrario a la idea preestablecida que tenía de lo que sería el proyecto, éste no sólo puso a prueba mis capacidades y conocimientos técnicos, sino que me vi obligado a hacer uso de distintas herramientas administrativas y de gestión que originalmente no formaban parte de las tareas que consideré para el desarrollo del proceso.

Al igual que para la implementación del dispositivo, la gestión del proyecto fue una combinación de herramientas ya adquiridas y otras aprendidas durante el desarrollo mismo. La importancia de una correcta gestión de proyecto tomó forma gracias a los trabajos realizados durante el Seminario de Trabajo Final donde se me proveyó con las herramientas necesarias para llevarla a cabo.

La creación de documentación de seguimiento, documentación técnica e informes preliminares como parte del seminario fueron vitales a la hora de llevar un correcto seguimiento de las tareas realizadas.

Tareas que inicialmente resultaron triviales, tomaron su real dimensión conforme el proyecto avanzó y la cantidad de información a presentar tomó una envergadura considerable. La Bitácora del Proyecto funcionó como hilo conductor de todos los procesos desarrollados y documentados a la hora de redactar este informe. Las constantes actualizaciones al Plan de Proyecto le dieron tangibilidad al tiempo invertido en cada tarea y las distintas especificaciones redactadas ayudaron a visualizar de forma clara los objetivos y su importancia relativa a la hora de tener que priorizar uno sobre otro.

Finalmente, el aporte por parte de los directores asignados al proyecto resultó sumamente importante a lo largo de todo el proceso. Su experiencia y apoyo me permitió mantenerme enfocado en las cuestiones realmente importantes sin desviarme frente a distintas alternativas. Sus consejos y seguimiento del avance me ayudaron a mantener el ritmo de trabajo y el orden en la documentación.

De la misma forma que la implementación del dispositivo representó una tarea de integración técnica, la gestión del proceso cumple la misma función integradora para la experiencia de documentación frente a proyectos de una gran envergadura.

El resultado final no solo se limita a un dispositivo funcional, sino a una experiencia que engloba en su totalidad el alcance de lo aprendido a lo largo de la carrera, no solo desde el ámbito técnico sino también desde un punto de vista de gestión y comunicación a la hora del desarrollo de proyectos de ingeniería.

Apéndice A: Amplificador RF

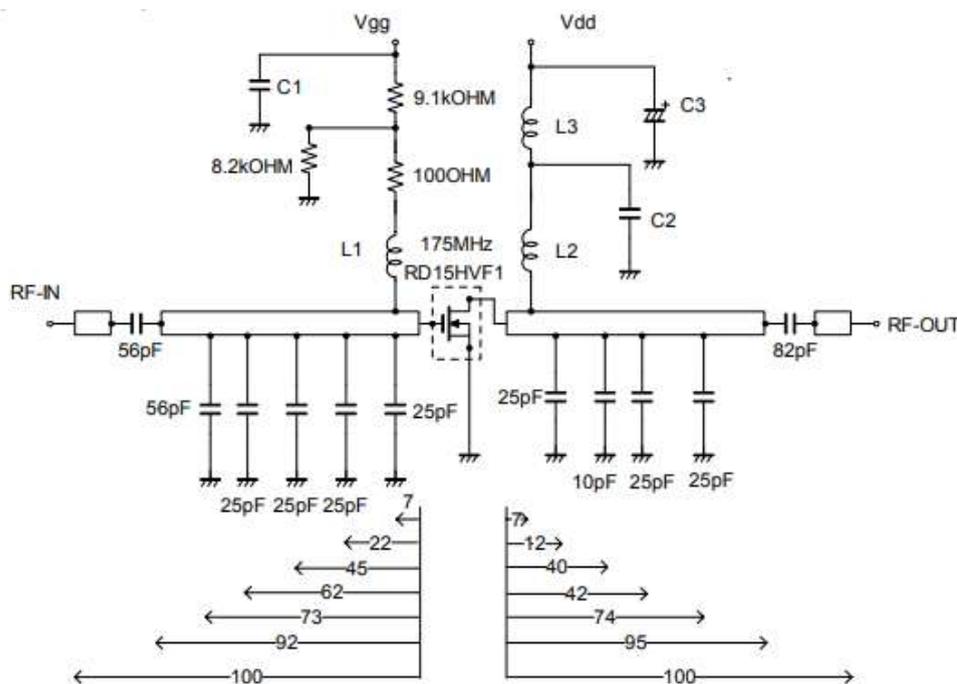
La normativa establece que los dispositivos AIS Clase B transmitan con una potencia máxima de 2 Watt. Para las pruebas de laboratorio no se implementó la etapa de potencia. En este apéndice se propone un circuito integrado de potencia para la implementación de dicha etapa junto con un ejemplo de circuito utilizado normalmente.

A.1 MOSFET de Potencia para RF: RD15HVF1

El RD15HVF1 es un transistor tipo MOSFET específicamente diseñado para implementar en amplificadores de alta potencia para VHF/UHF [23].

Sus principales características son una potencia de salida máxima de 15 Watt tanto para VHF como UHF con ganancia máxima de 14 dB a 175 MHz y 7 dB a 520 MHz.

A.2 Circuito de Prueba a 175 MHz



C1:2200pF 10uF in parallel
 C2:2200pF*2 in parallel
 C3:2200pF,330uF in parallel

L1:4Turns,LD6mm,D1.6mm P=1 silver plated copper wire
 L2:5Turns,LD6mm,D1.6mm P=1 silver plated copper wire
 L3:4Turns,LD6mm,D1.6mm P=1 silver plated copper wire

Note:Board material-Teflon substrate
 micro strip line width=4.2mm/50OHM,er.2.7,t=1.6mm
 Dimensions:mm

Apéndice B: Estudio Preliminar

B.1 Acerca del Equipo

Integrante

Mi nombre es Roberto Sergio Gorak, soy estudiante avanzado de la carrera de Ingeniería Electrónica en la Universidad Nacional de Mar del Plata. Mi experiencia laboral previa es mayormente en la Atención al Público. Suelo dar clases particulares en temas relacionados con mi carrera y más esporádicamente he tenido algunas experiencias de trabajo tipo freelance de diseño web para empresas del país.

Director

Dr. Ing. Alejandro José Uriz, Laboratorio de Comunicaciones, FI UNMdP

Co Directora

MSc. Mónica Liberatori, Laboratorio de Comunicaciones, FI UNMdP

Co Director

Ing. Brian Gluzman, Laboratorio de Comunicaciones, FI UNMdP

B.2 Acerca del Proyecto Final

Descripción del proyecto

El Sistema de Identificación Automática (AIS) provee una forma dinámica y sencilla de identificación para buques en todo el mundo. El sistema permite obtener de forma remota los datos más relevantes de dichos buques.

El sistema requiere por un lado, un receptor compatible que pueda recibir y decodificar la información generada por las embarcaciones, se recomienda su uso a la par de un sistema de radar. Para los buques, se utiliza un transmisor capaz de enviar la información requerida además de proveer a la tripulación una forma de modificar dicha información en caso de ser necesario.

El transmisor cuenta con tres grandes etapas. La primera será un microcontrolador que funcionará como etapa de adquisición de datos de la red de la embarcación. La segunda, un programa de computadora que permitirá el ingreso, modificación y adaptación de los datos obtenidos del adquisidor de datos y los aportados por el usuario, en la trama digital a ser transmitida. La tercera será la etapa de transmisión en RF a las frecuencias requeridas.

En el proyecto se diseñará el transmisor con las funciones mínimas requeridas para probarlo en un entorno de laboratorio. De acuerdo a la escala de TRLs, el proyecto se encuadra en un nivel de TRL = 4.

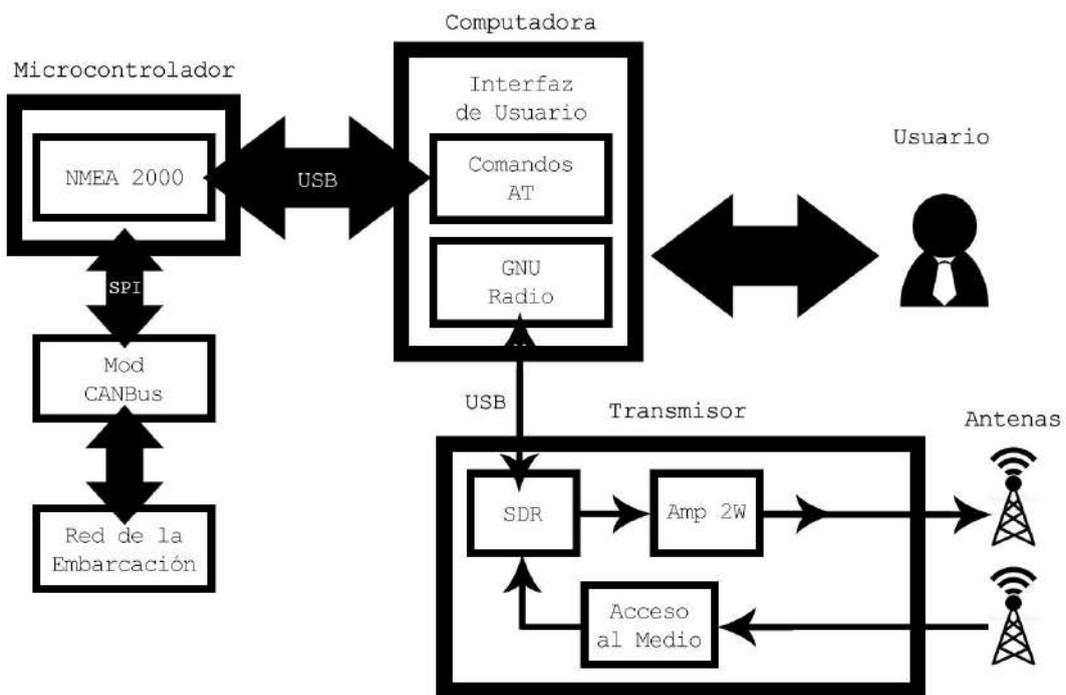
Beneficiario / Cliente

Los posibles beneficiarios del proyecto son aquellas empresas que deseen fabricar este tipo de transmisores sin la necesidad de realizar compras en el exterior. Otros posibles beneficiarios son los astilleros que deseen incluir este transmisor como parte del instrumental base de sus buques. Dispositivos similares ya existen en el mercado, pero deben ser importados de otros países con los costos que esto trae aparejado. El objetivo final de este proyecto es generar un reemplazo de la importación en dichos dispositivos.

Nombre del producto resultante

Transmisor de Identificación Automática para uso Naval (TIAN)

Diagrama en bloques



Grado de Avance

A la fecha de entrega de este estudio, el proyecto se encuentra en etapas de investigación en lo que respecta a la oferta del mercado existente en cuanto a prestaciones y requisitos. También se realizan tareas de investigación relacionadas con las normas establecidas para el uso del espectro de radiofrecuencias para la navegación.

Apéndice C: Plan de Proyecto

C.1 Acerca del Plan

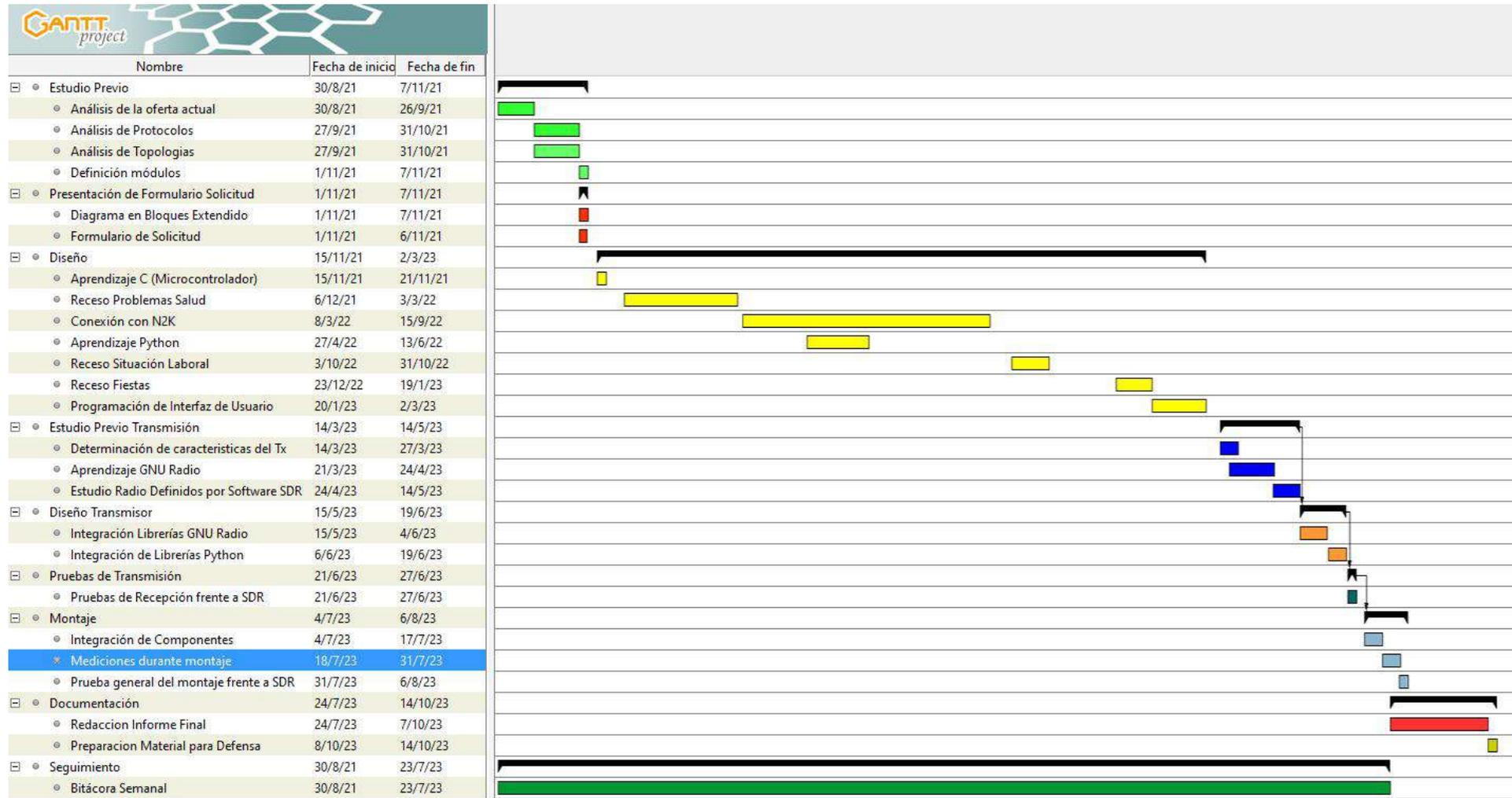
Introducción

En el siguiente cronograma se detallan las tareas a realizar durante la duración del proyecto. Debido al estadio temprano del proyecto, los tiempos son estimados basados en la experiencia propia y podrían ser distintos del tiempo real. Se tomó cada tarea un tiempo que varía en lapsos de 7 días (una semana).

Software

Gantt Project 3.1.3102 (GNU General Public License)

C.2 Diagrama de Gantt



C.3 Riesgos

Riesgo	Probabilidad	Impacto	Plan de Resolución
Imposibilidad de adquirir circuitos integrados debido a las dificultades en la importación.	Media	Alto	Evaluar circuitos integrados alternativos para la etapa comprometida.
Microcontrolador sin memoria suficiente para ejecutar el programa	Media	Medio	Limitar el uso de las librerías y sus opciones. Utilizar un modelo de microcontrolador de mayor capacidad de memoria.
Incompatibilidad de las librerías con entorno Windows	Alta	Alta	Migrar la ejecución de programas a un entorno Linux compatible.
La computadora puede dañarse por picos de tensión en el puerto USB que recibe datos.	Baja	Alto	Utilizar hardware para aislar eléctricamente la adquisición de datos y la computadora.
Transmitir mensajes a potencia y frecuencia reservada sin permiso de ENACOM	Baja	Bajo	Limitar la potencia de transmisión al mínimo necesario para probar el funcionamiento. Transmitir a una frecuencia no reservada.
Retraso en la ejecución de tareas debido a actividad laboral	Alta	Medio	Planificar tiempos dedicados a la ejecución de cada tarea con suficiente antelación.

Apéndice D:

Especificación de Requerimientos

D1. Introducción

Este documento es una Especificación de Requisitos para Transmisor de Identificación Automática para uso Naval (TIAN). Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

D1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo del sistema de identificación y comunicación por RF. Éste será utilizado por estudiantes, profesores y quien quiera entender las capacidades funcionales del sistema en cuestión.

D1.2 Alcance

En este documento se definen los requerimientos funcionales y no funcionales del sistema, como características de desempeño esperadas y limitaciones establecidas para el TIAN.

El documento lista las personas involucradas en el desarrollo del sistema y la documentación que lo acompaña además de sus roles dentro del proyecto. Se describe el funcionamiento del TIAN, el perfil de los usuarios finales y las características y restricciones del montaje final.

Respecto del sistema en sí mismo, se trata de un proyecto que evaluado según la escala de TRL (Technology Readiness Level) se puede considerar en nivel TRL = 4. Los bloques básicos se integran para comprobar que las piezas funcionan como un sistema. Los ensayos incluyen la integración de hardware específico en un laboratorio y pruebas con una variedad de simuladores y pruebas a pequeña escala con condiciones de cargas reales

D1.3 Personal involucrado

Nombre: **Roberto S. Gorak**

Rol: Diseñador, programador, tester.

Categoría: Estudiante

Tareas: Diseño, cálculo, montaje y testeo del sistema.

Contacto: gorakroberto@gmail.com

Nombre: **Dr. Ing. Alejandro José Uriz**

Rol: Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: ajuriz@fi.mdp.edu.ar

Nombre: **Ing. Brian Gluzman**

Rol: Co Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: briangluzman@mdp.edu.ar

Nombre: **MSc. Mónica Liberatori**

Rol: Co Directora de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: mlibera@fi.mdp.edu.ar

D1.4 Definiciones

Sigla	Significado	Descripción
TIAN	Transmisor de Identificación Automática para uso Naval	Dispositivo electrónico transmisor compatible con sistema AIS
AIS	Sistema de Identificación Automática	Sistema de identificación utilizado en embarcaciones que provee variada información.
TRL	Technology Readiness Level	Escala de niveles que permite conocer el grado de madurez de un proyecto tecnológico.
RF	RadioFrecuencia	Radiofrecuencia, medio por el cual se realiza la comunicación entre la baliza y los objetivos
OMI	Organización Marítima Internacional	Organización que dicta las normas internacionales para la navegación marítima.
SDR	Radio Definida por Software (Software Defined Radio)	Sistema de radiocomunicaciones donde varios de los componentes típicamente implementados en hardware son implementados en software
ENACOM	Ente Nacional de Comunicaciones	Ente Nacional que regula las telecomunicaciones en Argentina

D2. Descripción General

D2.1 Perspectiva del Producto

Existen distintos métodos de identificación utilizados para obtener información de embarcaciones. Uno de los más difundidos es el Sistema de Identificación Automática (AIS). Este sistema permite, mediante un receptor compatible, identificar una embarcación y obtener otros datos tales como posición, destino, estado de navegación por nombrar algunos. Para ello, la embarcación a identificar debe utilizar un transmisor compatible con AIS.

Existen distintos modelos de transmisores en el mercado actualmente. Sin embargo, no son de fabricación nacional y deben ser importados al país con los costos que esto acarrea debido al tipo de cambio.

El objetivo de este proyecto es el diseño de un sistema electrónico de transmisión compatible con AIS. Dicho sistema permitirá al usuario ingresar los datos pertinentes al recorrido a realizar, adjuntarlos con datos de posición, estado de navegación y otros suministrados por el instrumental de la embarcación, para generar el mensaje específico a ser transmitido por RF, lo que permitirá a los receptores compatibles identificar correctamente al navío.

Desarrollando este sistema de forma local se busca generar el reemplazo de la importación de dichos transmisores, permitiendo que las empresas que posean embarcaciones y los astilleros que deseen instalar estos transmisores como parte de su instrumental, tengan una alternativa a costo regional. Se considera también la posibilidad de incurrir en mercados internacionales en caso de poder exportar con precios competitivos.

D2.2 Funcionalidad

El sistema diseñado deberá generar una trama específica, cuyas características están definidas por la OMI, para ser transmitida por el medio establecido. El transmisor enviará la trama al medio, de forma que cualquier receptor en el rango de transmisión pueda identificar la embarcación.

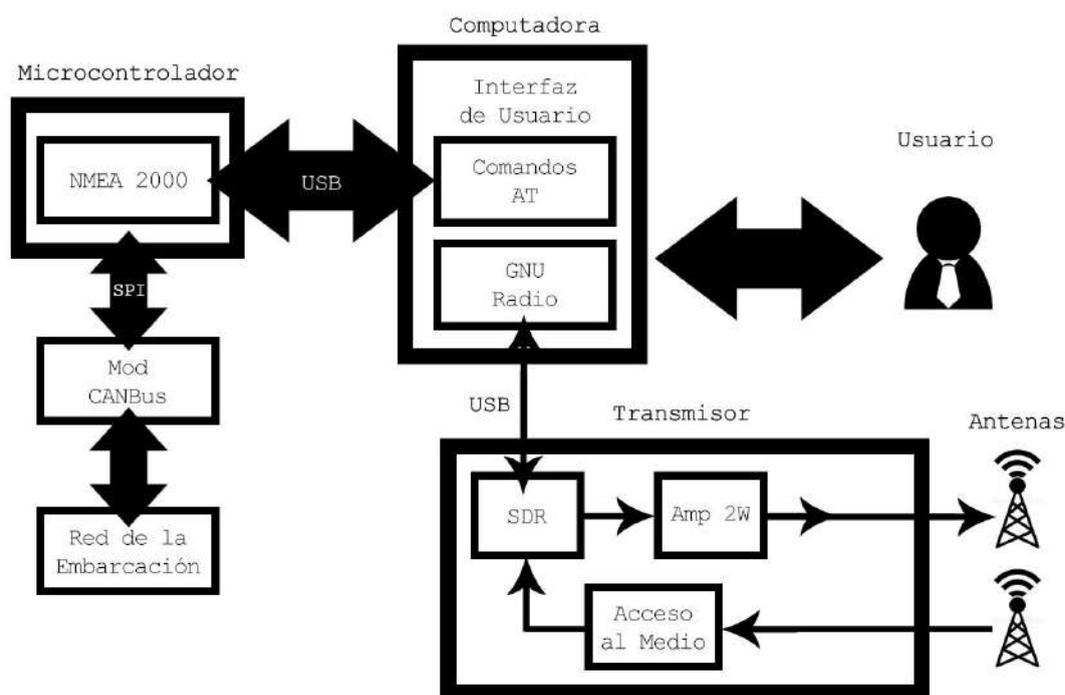
El sistema tendrá tres etapas principales. La primera será un microcontrolador que funcionará como etapa de adquisición de datos de la red de la embarcación. El microcontrolador se programará en su lenguaje correspondiente y será el encargado de obtener los datos de los distintos instrumentos y sensores de la embarcación.

La segunda será un programa de computadora que permitirá el ingreso, modificación y adaptación de los datos obtenidos de los instrumentos de la embarcación y los

aportados por el usuario, en la trama digital a ser transmitida. Ésta etapa será programada en Python.

La tercera será el módulo de transmisión en RF a las frecuencias requeridas. Esta etapa se implementará mediante un SDR.

En la figura a continuación se muestran los bloques y sus interacciones.



D2.3 Características de los usuarios

Montaje: la conexión del dispositivo no requerirá ningún tipo de conocimiento técnico, más que el reconocimiento de los respectivos puertos utilizados para los periféricos.

Testeo del TIAN: se recomienda que personal con conocimientos técnicos y de navegación lleve a cabo las pruebas pertinentes para corroborar el correcto funcionamiento del sistema.

Documentación: se recomienda que los usuarios tengan un mínimo conocimiento de lenguaje técnico para evitar errores de interpretación.

D2.4 Restricciones

Debido a que el ENACOM prohíbe el uso indiscriminado de las bandas de frecuencia a los civiles, el sistema se diseñará e implementará sin etapa de potencia y de forma local.

Considerando la imposibilidad de acceder a una embarcación, el sistema final será montado en un chasis de prueba para comprobar su funcionamiento en las inmediaciones del laboratorio. Para dichas pruebas se utilizará un receptor basado en SDR.

D2.5 Suposiciones y dependencias

Se asume que el usuario tendrá a su disposición periféricos compatibles para utilizar la interfaz de software y que el sistema de transmisión será montado sobre un chasis que permita la transmisión correctamente.

D3. Requisitos Específicos

D3.1 Requisitos comunes de las interfaces

D3.1.1 Interfaces de Usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá permitir el ingreso y modificación de datos respecto del viaje.

D3.1.2 Interfaces de hardware

Será necesario disponer de una computadora para correr el programa y acceder a la interfaz de usuario. El sistema tendrá conectores compatibles con el protocolo NMEA 2000 y USB 2.0.

D3.1.3 Interfaces de Software

La computadora a utilizar debe contar con un sistema operativo compatible con el runtime de Python.

D3.1.4 Interfaces de Comunicación

El sistema debe transmitir la palabra digital generada mediante RF con portadoras de 161,975 MHz y 162,025 MHz.

D3.2 Requisitos Funcionales

D3.2.1 Capacidad de Programación

El sistema debe proveer al usuario una interfaz que le permita establecer los datos estáticos de la embarcación.

D3.2.2 Capacidad de Modificación

El sistema debe proveer al usuario una interfaz que le permita modificar los datos estáticos de la embarcación.

D3.2.3 Código de identificación único

La trama generada por el programa debe incluir la identificación provista por el ente encargado a tal fin, de modo que el sistema permita identificar unívocamente a qué embarcación pertenece.

D3.2.4 Datos de posición en tiempo real

La trama generada por el programa debe ser compatible con los mensajes establecidos por OMI para informar la posición mediante AIS. La frecuencia de actualización de los datos debe seguir la norma establecida.

D3.2.5 Potencia de Transmisión

El rango de potencia permitido para la transmisión de señal de identificación es de 2W para la frecuencia de trabajo y la clase del dispositivo.

D3.2.6 Tensión de Alimentación

El sistema deberá utilizar como fuente de tensión un máximo de 24 volts de corriente continua.

D3.3 Requisitos no Funcionales**D3.3.1 Frecuencia de Transmisión**

El sistema debe transmitir solo en las frecuencias establecidas para tal uso con el objetivo de no generar interferencia en canales destinados a otros usos.

D3.3.2 Información de la Trama

Deberá ser solo la información establecida por la normativa, no deberá haber ningún agregado de información que no sea la especificada.

D3.3.3 Modulación según normativa

La trama debe ser transmitida según las normas para poder ser demodulada correctamente, esta modulación no debe ser alterada.

D3.3.3 Montaje Adecuado

El sistema debe estar preparado para funcionar en las condiciones previstas para una embarcación y las pruebas de laboratorio.

Apéndice E:

Especificación Funcional

E1. Introducción

Existen distintos métodos de identificación utilizados para obtener información de embarcaciones. Uno de los más difundidos es el Sistema de Identificación Automática (AIS). Este sistema permite, mediante un receptor compatible, identificar una embarcación y obtener otros datos tales como posición, destino, estado de navegación por nombrar algunos. Para ello, la embarcación a identificar debe utilizar un transmisor compatible con AIS.

Existen distintos modelos de transmisores en el mercado actualmente. Sin embargo, no son de fabricación nacional y deben ser importados al país con los costos que esto acarrea debido al tipo de cambio.

El objetivo de este proyecto es el diseño de un sistema electrónico de transmisión compatible con AIS. Dicho sistema permitirá al usuario ingresar los datos pertinentes al recorrido a realizar, adjuntarlos con datos de posición, estado de navegación y otros suministrados por el instrumental de la embarcación, para generar el mensaje específico a ser transmitido por RF, lo que permitirá a los receptores compatibles identificar correctamente al navío.

Desarrollando este sistema de forma local se busca generar el reemplazo de la importación de dichos transmisores, permitiendo que las empresas que posean embarcaciones y los astilleros que deseen instalar estos transmisores como parte de su instrumental, tengan una alternativa a costo regional. Se considera también la posibilidad de incurrir en mercados internacionales en caso de poder exportar con precios competitivos.

E1.1. Propósito del documento

El presente documento tiene como propósito proveer información detallada de cómo funcionará el Transmisor de Identificación Automática para uso Naval (TIAN), cuáles serán sus comportamientos deseados y cómo se deberá construir, con base en los requerimientos anteriormente definidos en la Especificación de Requerimientos. Se incluyen en el documento casos de uso, entradas y salidas del sistema, diagramas y maquetas.

Está dirigido al personal técnico encargado de construir el dispositivo, así como a cualquier persona que desee conocer las soluciones propuestas para resolver las cuestiones planteadas en la Especificación de Requerimientos.

E1.2. Alcance del Proyecto

El proyecto fue evaluado según la escala de TRL (Technology Readiness Level), en dicha escala se lo considera en nivel TRL = 4. Los bloques básicos se integran para comprobar que las piezas funcionan como un sistema. Los ensayos incluyen la integración de hardware específico en un laboratorio y pruebas con una variedad de simuladores y pruebas a pequeña escala con condiciones de cargas reales.

E1.3. Alcance del Documento

En este documento se definen las soluciones técnicas propuestas para la implementación de las especificaciones pedidas. Se establece el tipo de sistema embebido utilizado para la adquisición de los datos y generación del mensaje digital, así como los módulos complementarios necesarios tanto para la configuración del sistema y la transmisión. Se define también la forma en que el mensaje será transmitido. Se definen además las capacidades de hardware relacionadas con el funcionamiento general del sistema como tipo de batería, formas de activación del dispositivo, topología del mensaje a enviar.

E1.4. Documentos Relacionados

Componente	Nombre	Descripción
	Transmisor de Identificación Automática para uso Naval (TIAN) – Estudio Preliminar	Estudio preliminar y presentación del Proyecto. Incluye información básica.
	Transmisor de Identificación Automática para uso Naval (TIAN) – Plan de Proyecto	Define las tareas del proyecto y los tiempos requeridos para completarlas.
	Transmisor de Identificación Automática para uso Naval (TIAN) – Especificación de Requerimientos	Define los requerimientos funcionales y no funcionales del sistema además de sus características de desempeño y limitaciones.

E1.5. Siglas y Definiciones

Sigla	Significado	Descripción
TIAN	Transmisor de Identificación Automática para uso Naval	Dispositivo electrónico transmisor compatible con sistema AIS
AIS	Sistema de Identificación Automática	Sistema de identificación utilizado en embarcaciones que provee variada información.
TRL	Technology Readiness Level	Escala de niveles que permite conocer el grado de madurez de un proyecto tecnológico.
RF	RadioFrecuencia	Radiofrecuencia, medio por el cual se realiza la comunicación entre la baliza y los objetivos
OMI	Organización Marítima Internacional	Organización que dicta las normas internacionales para la navegación marítima.
ENACOM	Ente Nacional de Comunicaciones	Ente Nacional que regula las telecomunicaciones en Argentina
SDR	Radio Definida por Software (Software Defined Radio)	Sistema de radiocomunicaciones donde varios de los componentes típicamente implementados en hardware son implementados en software

E1.6. Suposiciones y Riesgos

Se asume que el usuario tendrá a su disposición periféricos compatibles para utilizar la interfaz de software y que el sistema de transmisión será montado sobre un chasis que permita la transmisión correctamente.

E2. Descripción General

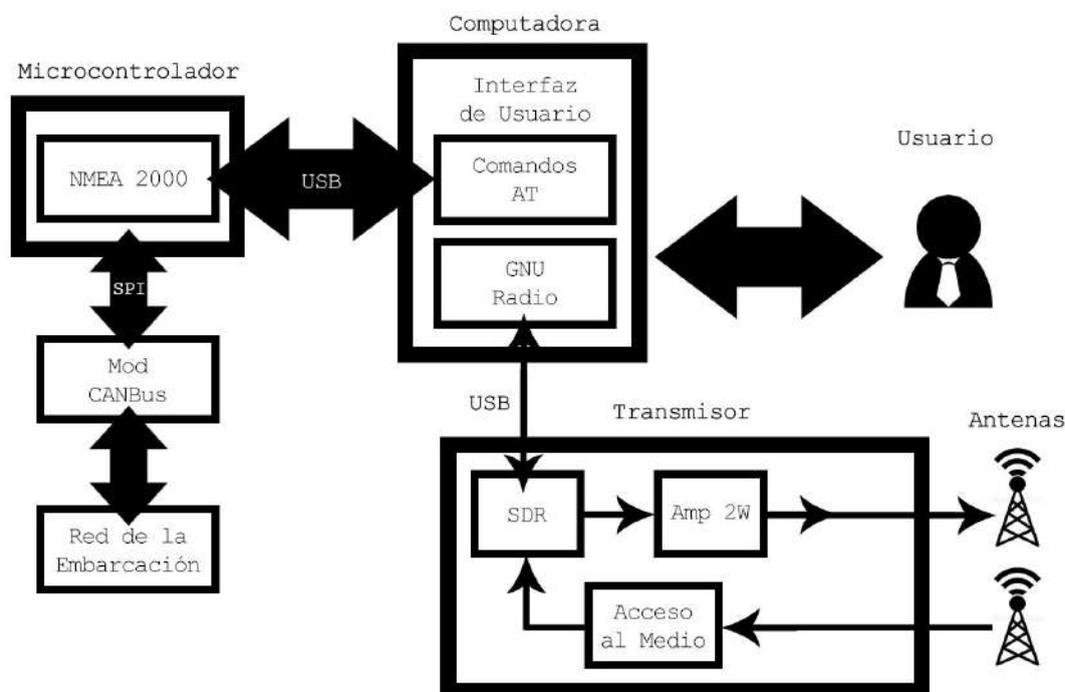
El sistema diseñado deberá generar una trama específica, cuyas características están definidas por la OMI, para ser transmitida por el medio establecido. El transmisor enviará la trama al medio, de forma que cualquier receptor en el rango de transmisión pueda identificar el navío.

El sistema tendrá tres etapas principales. La primera será un microcontrolador que funcionará como etapa de adquisición de datos de la red del buque. El microcontrolador se programará en su lenguaje correspondiente y será el encargado de obtener los datos de los distintos instrumentos y sensores de la embarcación.

La segunda será un programa de computadora que permitirá el ingreso, modificación y adaptación de los datos obtenidos de los instrumentos de la embarcación y los aportados por el usuario, en la trama digital a ser transmitida. Ésta etapa será programada en Python.

La tercera será el módulo de transmisión en RF a las frecuencias requeridas. Esta etapa se implementará mediante un SDR.

E2.1. Diagrama en Bloques



E2.2. Personal Involucrado

2.2.1. Roles y Tareas

Nombre: **Roberto S. Gorak**

Rol: Diseñador, programador, tester.

Categoría: Estudiante

Tareas: Diseño, cálculo, montaje y testeo del sistema.

Contacto: gorakroberto@gmail.com

Nombre: **Dr. Ing. Alejandro José Uriz**

Rol: Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: ajuriz@fi.mdp.edu.ar

Nombre: **Ing. Brian Gluzman**

Rol: Co Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: briangluzman@mdp.edu.ar

Nombre: **MSc. Mónica Liberatori**

Rol: Co Directora de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: mlibera@fi.mdp.edu.ar

E2.3. Dependencias y Cambios

2.3.1. Dependencias del proyecto

2.3.2. Cambios

E3. Especificaciones Funcionales

E3.1. Capacidad de Programación

E3.1.1. Propósito

TIAN deberá permitir al usuario cargar datos estáticos de la embarcación. Estos datos serán provistos por la entidad encargada del registro de los navíos. Estos datos no deberán ser alterados durante los viajes a menos que el dispositivo sea llevado a otra embarcación.

E3.1.2. Caso de Uso

Durante la instalación de TIAN, el usuario debe tener disponible un sector de la interfaz que le permita cargar los datos estáticos.

E3.1.3. Maqueta

E3.1.4. Requerimientos Funcionales

E3.2. Capacidad de Modificación

E3.2.1 Propósito

TIAN deberá permitir al usuario cargar datos dinámicos del buque. Estos datos serán actualizados en cada viaje a realizar. Estos datos pueden ser alterados durante los viajes según el criterio del oficial al mando de la embarcación.

E3.2.2. Caso de Uso

En caso de requerir un cambio de rumbo, una variación en el tiempo de llegada u otra eventualidad, el oficial al mando podrá modificar los datos que utiliza el TIAN para informar a los receptores en rango.

E3.2.3. Maqueta

E3.2.4. Requerimientos Funcionales

E3.3. Código de Identificación único

E3.3.1. Propósito

La trama generada y emitida deberá contener la información de identificación de la embarcación. Esta información debe ser provista por el ente encargado a tal fin. La información no debe ser modificada o adulterada.

E3.3.2. Caso de Uso

Los sistemas receptores de AIS muestran no solo la información del viaje sino también a quién corresponde dicha información. Durante la transmisión de los datos de la embarcación, ésta información debe estar incluida.

E3.3.3. Maqueta**E3.3.4. Requerimientos Funcionales****E3.4. Datos de posición en tiempo real****E3.4.1. Propósito**

En la información transmitida, la posición de la embarcación, debe ser actualizada según la velocidad actual de desplazamiento.

E3.4.2. Caso de Uso

La posición de la embarcación deberá ser actualizada cada 30 segundos si se desplaza con $SOG > 2$ nudos y cada 3 minutos si $SOG \leq 2$ nudos.

E3.4.3. Maqueta**E3.4.4. Requerimientos Funcionales****E3.5. Alimentación y Potencia****E3.5.1. Propósito**

Se utilizará para alimentar al sistema una fuente de tensión de 24 volts máximos. Esta fuente provee la energía necesaria para que el TIAN transmita a una potencia máxima de 2W con portadora en 162 MHz. Con esta potencia de transmisión se asegura el funcionamiento establecido por la OMI para este tipo de dispositivos.

E3.5.2. Caso de Uso

El sistema se alimentará de la propia red de la embarcación. La computadora que ejecute el programa principal podría ser una computadora portátil con su propia fuente a batería.

E3.5.3. Maqueta**E3.5.4. Requerimientos Funcionales****E4. Configuración del Sistema**

Durante la instalación del sistema, se deberá ingresar los datos de identificación de la embarcación por única vez. El programa guardará estos datos siempre que el sistema no sea movido a otra embarcación y deban ser actualizados. Los datos dinámicos de cada viaje, serán cargados al inicio y podrán ser modificados a demanda. Los datos de la red de la embarcación serán directamente controlados por el software sin necesidad de intervención del usuario.

E5. Otros requerimientos

E6. Requerimientos para reportes

E7. Requerimientos de integración

La red de la embarcación debe ser compatible con el protocolo NMEA2000, la entrada de datos para el microcontrolador estará diseñada con ese fin, así como las conversiones realizadas por el software para la computadora.

El chasis sobre el cual se montará la SDR de TIAN debe permitir la transmisión por RF.

E7.1. Manejo de Excepciones y Errores

E8. Requerimientos de Conversión/Migración

E9. Referencias

E10. Incidencias por resolver

Apéndice F: Especificación Técnica

F1. Introducción

Este documento corresponde a la Especificación Técnica del Transmisor de Identificación para uso Naval (TIAN). Esta especificación se ha estructurado basándose en la información mencionada en el documento Especificación Funcional Ver. 0.3.

F1.1. Propósito del documento

El presente documento tiene como propósito proveer información detallada de cómo funcionarán los distintos bloques y el sistema en su totalidad, definiendo cómo se deberán construir y cuales serán sus comportamientos deseados con base en los requerimientos anteriormente definidos en la Especificación Funcional Ver. 0.3.

Se encuentra dirigido a:

- El desarrollador del instrumento, quién lo construirá.
- Los directores y solicitantes del proyecto, así como a quienes lo evaluarán.

F1.2. Alcance del Proyecto

El proyecto fue evaluado según la escala de TRL (Technology Readiness Level), en dicha escala se lo considera en nivel TRL = 4. Los bloques básicos se integran para comprobar que las piezas funcionan como un sistema. Los ensayos incluyen la integración de hardware específico en un laboratorio y pruebas con una variedad de simuladores y pruebas a pequeña escala con condiciones de cargas reales.

F1.3. Personal Involucrado

Nombre: **Roberto S. Gorak**

Rol: Diseñador, programador, tester.

Categoría: Estudiante

Tareas: Diseño, cálculo, montaje y testeo del sistema.

Contacto: gorakroberto@gmail.com

Nombre: **Dr. Ing. Alejandro José Uriz**

Rol: Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: ajuriz@fi.mdp.edu.ar

Nombre: **Ing. Brian Gluzman**

Rol: Co Director de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: briangluzman@mdp.edu.ar

Nombre: **MSc. Mónica Liberatori**

Rol: Co Directora de Proyecto

Categoría: Docente

Tareas: Dirección general del proyecto

Contacto: mlibera@fi.mdp.edu.ar

F1.4. Siglas y Definiciones

Sigla	Significado	Descripción
TIAN	Transmisor de Identificación Automática para uso Naval	Dispositivo electrónico transmisor compatible con sistema AIS
AIS	Sistema de Identificación Automática	Sistema de identificación utilizado en embarcaciones que provee variada información.
TRL	Technology Readiness Level	Escala de niveles que permite conocer el grado de madurez de un proyecto tecnológico.
RF	RadioFrecuencia	Radiofrecuencia, medio por el cual se realiza la comunicación entre la baliza y los objetivos
OMI	Organización Marítima Internacional	Organización que dicta las normas internacionales para la navegación marítima.
ENACOM	Ente Nacional de Comunicaciones	Ente Nacional que regula las telecomunicaciones en Argentina
SDR	Radio Definida por Software (Software Defined Radio)	Sistema de radiocomunicaciones donde varios de los componentes típicamente implementados en hardware son implementados en software

F1.5. Suposiciones y Riesgos

Se asume que el usuario tendrá a su disposición periféricos compatibles para utilizar la interfaz de software y que el sistema de transmisión será montado sobre un chasis que permita la transmisión correctamente.

F2. Descripción General

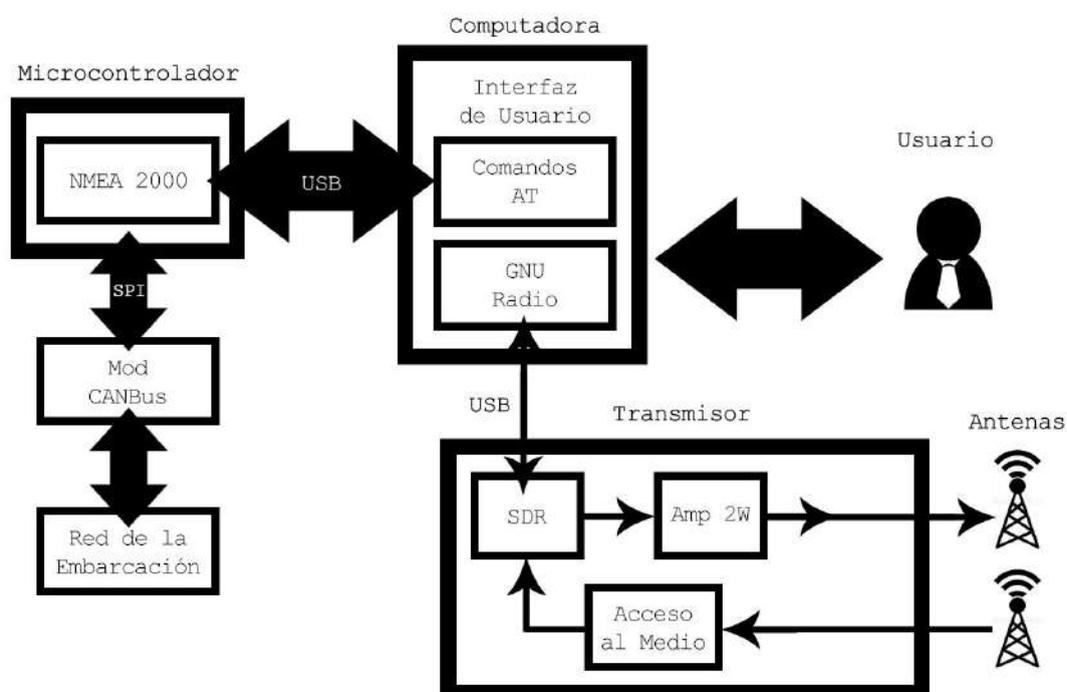
El sistema diseñado deberá generar una trama específica, cuyas características están definidas por la OMI, para ser transmitida por el medio establecido. El transmisor enviará la trama al medio, de forma que cualquier receptor en el rango de transmisión pueda identificar el navío.

El sistema tendrá 3 etapas principales. La primera será un microcontrolador que funcionará como etapa de adquisición de datos de la red del buque. El microcontrolador se programará en su lenguaje correspondiente y será el encargado de obtener los datos de los distintos instrumentos y sensores de la embarcación.

La segunda será un programa de computadora que permitirá el ingreso, modificación y adaptación de los datos obtenidos de los instrumentos de la embarcación y los aportados por el usuario, en la trama digital a ser transmitida. Ésta etapa será programada en Python.

La tercera será el módulo de transmisión en RF a las frecuencias requeridas. Esta etapa se implementará mediante un SDR.

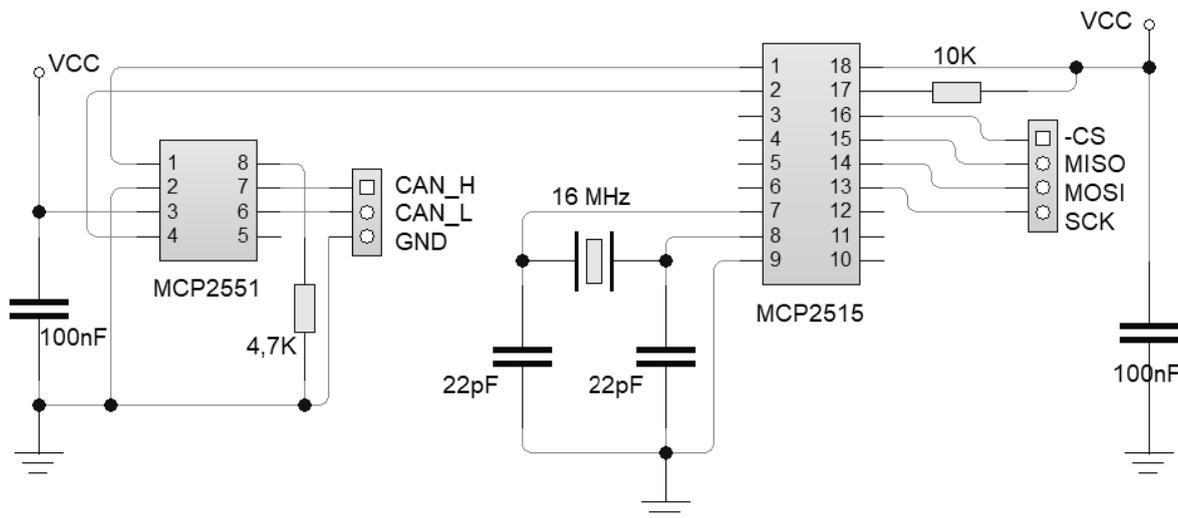
F2.1. Diagrama en Bloques



F3. Descripción de los Módulos

F3.1. Módulo CAN Bus

El módulo CAN Bus proveerá al sistema, una forma de conexión física a la red de la embarcación. La red utilizada para entornos marítimos se conoce como NMEA 2000. Este protocolo está formado por una capa física CAN Bus y un protocolo de alto nivel J1939 modificado. El circuito que se utilizará se muestra en la figura:



Los componentes que configuran a los circuitos integrados serán dimensionados en base al modo de operación de la librería encargada de decodificar los paquetes NMEA 2000.

F3.1.1 MCP2551

Este circuito integrado es un transceptor CAN de alta velocidad de la empresa Microchip. Su función es servir como interfaz entre la red física y el controlador CAN adaptando los niveles y protegiendo al resto de la circuitería de posibles picos o señales espurias que se puedan generar en la red.

Pin	Nombre	Función	Detalle
1	TXD	Entrada de Datos Transmitidos	Recibe los datos a transmitir por el controlador CAN.
2	Vss	Masa	--
3	Vdd	Fuente	--
4	RXD	Salida de Datos Recibidos	Envía los datos recibidos al Controlador CAN

5	Vref	Referencia de Tensión de salida	Sin uso
6	CANH	Línea HIGH CAN	Controla la línea HIGH del bus CAN
7	CANL	Línea LOW CAN	Controla la línea LOW del bus CAN
8	Rs	Control de Slew Rate	Controla el Slew Rate según la corriente

Para este circuito se utilizará una resistencia Rs de 4,7 kΩ para asegurar el máximo Slew Rate posible, evitando las interferencias electromagnéticas. Se agrega además, un capacitor de 100 nF para mejorar el filtrado de la tensión de fuente. Para mayor información acerca de los modos de funcionamiento del MCP2551, referirse a su hoja de datos completa enlazada en las Referencias.

F3.1.2 MCP2515

Este circuito integrado es un Controlador CAN autónomo con interfaz SPI. Su función es el control de los mensajes CAN que se reciben o se desean transmitir. Los pines de transmisión de datos del MCP2515 se conectarán directamente a los del MCP2551. El bus SPI se conectará al bus SPI del microcontrolador que funcionará como Adquisidor de Datos.

Información Básica:

- Compatible con CAN v2.0B a 1 Mb/s
- Bus SPI de alta velocidad (10 MHz)
- Tecnología de bajo consumo CMOS
- Soporte para tramas estándar y extendidas

Para su correcto funcionamiento, este circuito integrado requiere el agregado de un oscilador. Nuevamente, los valores de los componentes serán provistos por la documentación de la librería NMEA 2000. La hoja de datos provee información acerca de la elección de dichos componentes.

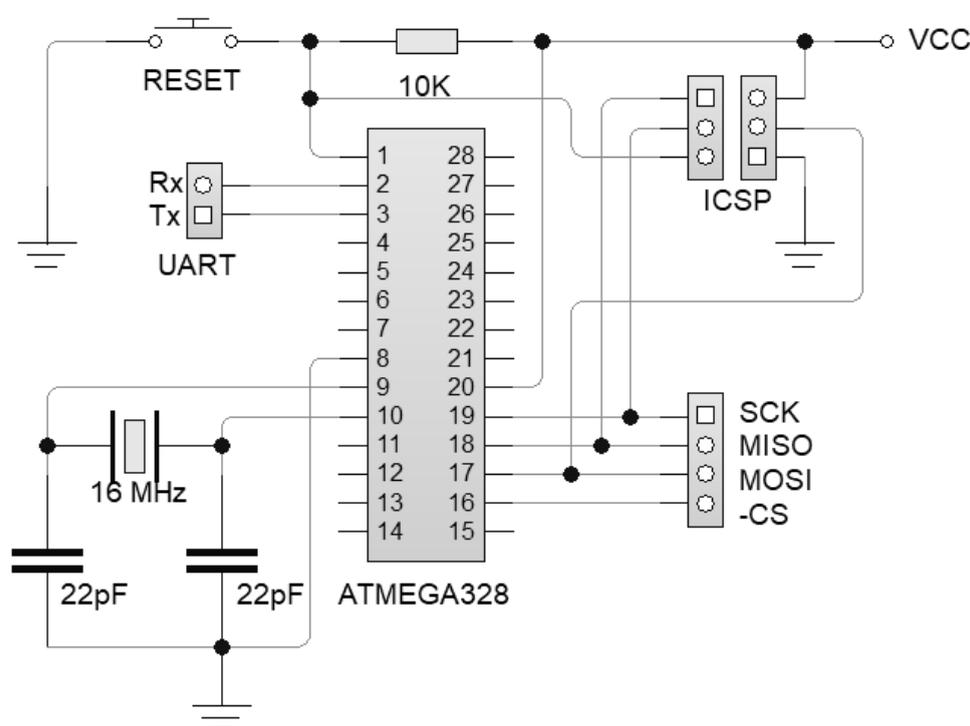
Frecuencia Xs	Valores típicos de Capacitores	
	C1	C2
4 MHz	27 pF	27 pF
8 MHz	22 pF	22 pF
20 MHz	15 pF	15 pF

Para este caso el Controlador funcionará a 16 MHz utilizando un cristal de la misma frecuencia y dos capacitores de 22 pF conectados a los pines 7 y 8. Además, se conectará una resistencia de 10 kΩ desde el pin 17 a la fuente. Esta entrada es el RESET por hardware y es activo bajo. Como se adelantó, el bus SPI será conectado directamente al bus SPI del microcontrolador y los pines 1 y 2, TX y RX, se conectarán al MCP2551.

Este circuito integrado requiere un proceso de configuración por software, sin embargo no se detallará en este documento dicho proceso ya que la librería encargada de controlar el bus CAN configurará el controlador adecuadamente como parte del proceso de inicialización. Para mayor información acerca de los registros y modos de funcionamiento del MCP2515, referirse a su hoja de datos completa enlazada en las Referencias.

F3.2 Módulo Microcontrolador

Las funciones relacionadas a la conectividad quedan a cargo del módulo microcontrolador y la librería NMEA2000 previamente mencionada. El microcontrolador que se utilizará para el desarrollo es el ATMEGA328P de la empresa Atmel. El circuito a utilizar se muestra en la figura:



El circuito de la figura representa la configuración mínima necesaria para el funcionamiento del microcontrolador. Se utilizará un cristal de 16 MHz, con dos capacitores de 22 pF conectados a los pines 9 y 10, como oscilador externo. Se agregará un pulsador y una resistencia de 10 kΩ al pin 1 que provee el RESET por hardware. Se

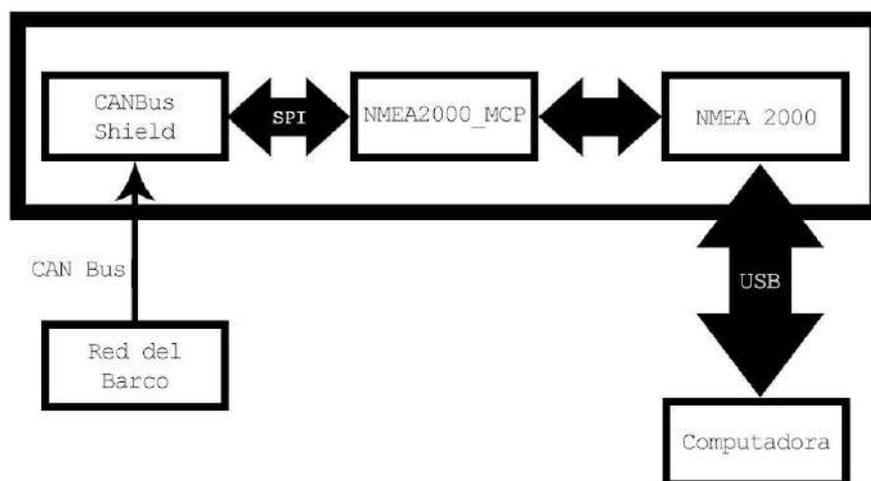
conectará además, los pines 16 a 19 al bus SPI del controlador CAN del módulo previamente desarrollado. Se conectará el puerto UART, pines 2 y 3, al módulo UART-USB que se explicará en una próxima sección. Se recomienda la implementación de un puerto ICSP, en caso que el microcontrolador necesite ser reprogramado en el futuro.

El código que programa a este microcontrolador se puede analizar en las hojas finales de este documento. El código será escrito y cargado mediante Arduino IDE.

F3.2.1 Módulo NMEA2000

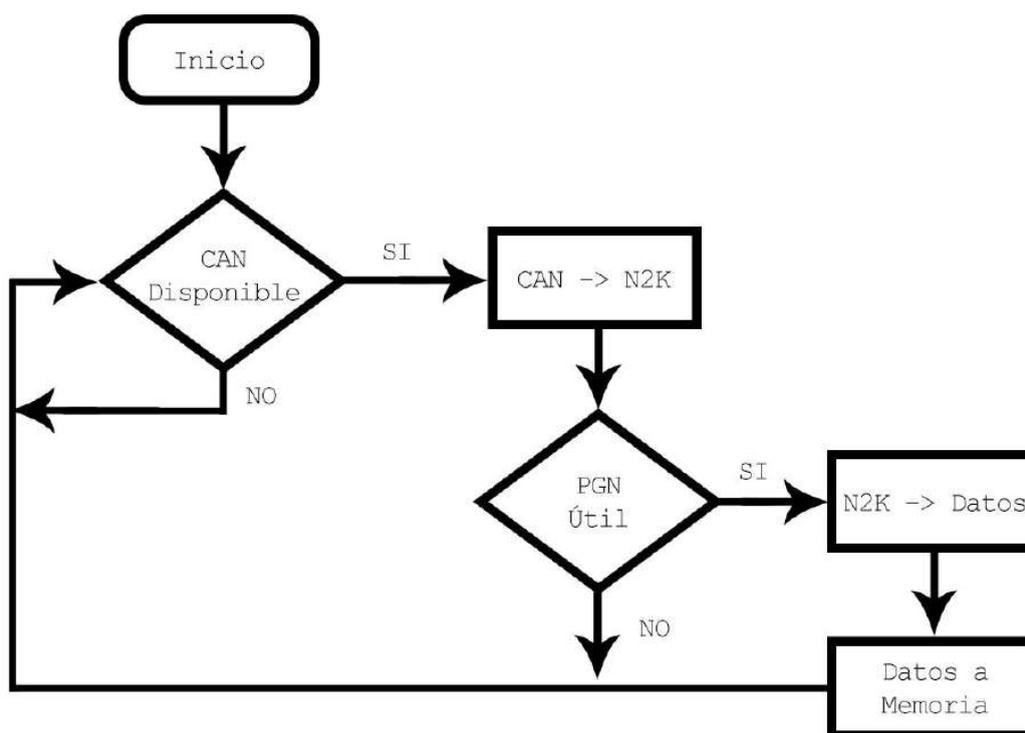
Este módulo será implementado mediante software y representa a la librería encargada de procesar los paquetes NMEA2000 que lleguen desde el bus hasta el microcontrolador.

La librería en cuestión, NMEA2000 Library for C++, es en realidad un grupo de 3 librerías que funcionan en conjunto. La primera de ellas es CAN Bus Shield de SeeedStudio y se encarga de controlar la comunicación con el bus CAN. La segunda librería, de nombre NMEA2000_MCP, desempaqueta los mensajes CAN para obtener su payload. La tercera, de nombre NMEA2000, provee herramientas que permiten manipular los mensajes de N2K como una estructura de datos dentro del programa general. Provee clases y métodos para procesar dichos mensajes. La estructura de la librería se muestra en la figura:

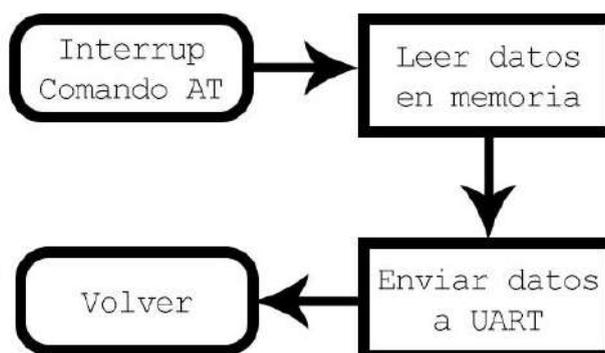


El programa cargado en el microcontrolador realizará dos tareas. La primera será filtrar los paquetes N2K que se consideran necesarios para crear los mensajes AIS de los que no son de utilidad. Una vez recibido un paquete, se identificará según su PGN si es de utilidad o no. En caso afirmativo, se procesará el paquete guardando los datos obtenidos en la memoria del microcontrolador. En caso contrario, el paquete se

descartará sin realizar ninguna acción extra. En la siguiente figura se puede apreciar un diagrama de flujo que representa dicho funcionamiento:



La segunda tarea que realizará este bloque es la de enviar los datos almacenados al programa de la computadora cuando éste lo pida mediante un comando AT. El envío de datos será controlado mediante una rutina de interrupción que se ejecuta como se muestra en la siguiente figura:



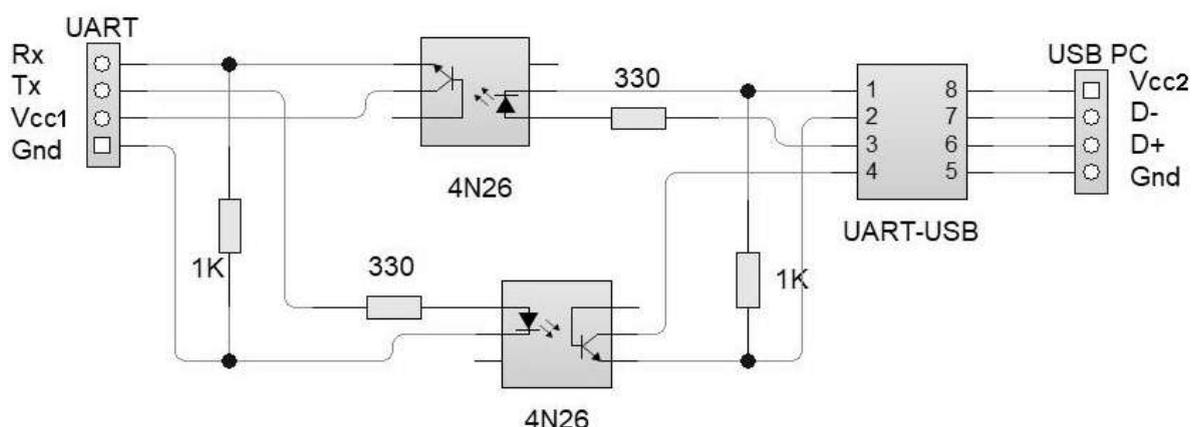
La preparación de la librería sólo requerirá que ésta sea instalada mediante Arduino IDE y que se agreguen al código las sentencias `#include` para poder utilizarla. Para mayor información acerca de la librería referirse a la página en GitHub del autor, enlazada en las Referencias.

F3.3 Módulo UART-USB

Dado que el microcontrolador seleccionado no provee una función nativa de conectividad por puerto USB, se agregará este módulo comercial para cumplir esa función. Este módulo está basado en el circuito integrado PL2303HX que es un controlador puente de USB a serie que provee una solución conveniente para conectar un dispositivo serie asincrónico con capacidad full-duplex a cualquier dispositivo con puertos USB. Durante el montaje, se le retirarán los conectores de entrada y salida para ser montado en una placa diseñada para proveer además, un circuito de aislación eléctrica que proteja a la computadora de posibles señales que puedan dañarla. Se muestra en la foto el módulo comercial a utilizar:



La protección será implementada mediante optoacopladores 4N26 de la empresa Vishay. Se colocará uno en cada línea de datos. Los optoacopladores requieren una serie de resistencias de polarización para su correcto funcionamiento. El circuito completo con la protección se muestra en la figura:



La polarización se realizará teniendo en cuenta los parámetros obtenidos de la hoja de datos: Relación de Transferencia de Corriente (CTR), Corriente Máxima de Entrada (IFmax) y Corriente Máxima de Colector (ICmax).

La relación entre estos 3 parámetros determinará los valores de las resistencias a utilizar. CTR define la relación de corrientes entre I_F e I_C que según la hoja de datos es típicamente del 50% pero puede ser tan baja como 20%. Se recomienda adoptar en este caso un valor intermedio de 33%. I_{Fmax} establece una cota superior para la corriente de entrada al optoacoplador, en este caso de 60mA. I_{Cmax} establece una cota superior para la corriente que puede proveer la salida del optoacoplador, en este caso 50mA.

Sabiendo que la tensión máxima de salida del microcontrolador será de 5V y la corriente máxima será de 40mA, se polarizará el optoacoplador para una corriente de entrada de 15mA según:

$$R = V / I_F = 5V / 15mA = 333\Omega \approx 330\Omega \Rightarrow I_F = 15,15mA$$

Considerando que con un CTR del 33%, a la salida del optoacoplador se obtendrá una corriente I_C de 5,05mA, se calcula la resistencia de salida como:

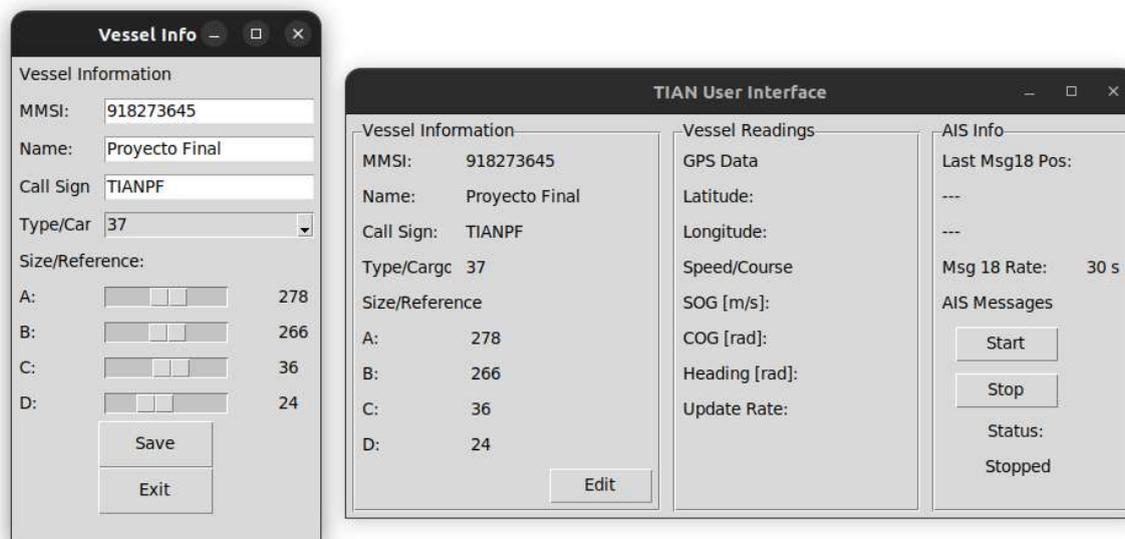
$$R = V / I_C = 5V / 5,05mA = 990\Omega \approx 1k\Omega$$

Finalmente, el último parámetro a considerar es el ancho de banda del optoacoplador. Según la hoja de datos, el tiempo típico de Rise and Fall es de $2\mu s$ bajo una carga de 100Ω . Utilizar una carga 10 veces más grande naturalmente limitará el ancho de banda. Se recomienda que la velocidad del bus UART, a configurar en el microcontrolador, no supere los 9600 baudios.

F3.4 Computadora

F3.4.1 Interfaz de Usuario

Para cumplir con los Requisitos Funcionales RF01, RF02 y RF03 establecidos en la Especificación Funcional, se implementará una Interfaz de Usuario programada en Python. La interfaz permitirá al usuario introducir y modificar la información de la embarcación. Este programa además, realizará otras funciones a detallar en las siguientes secciones. El código para este módulo puede ser analizado al final de este documento. Se muestra en la siguiente figura la interfaz funcionando:



F3.4.2 Comandos AT

Este módulo se encarga de realizar el pedido de datos actualizados al Adquisidor de Datos. Mediante el puerto serie envía un comando cuando requiere actualizar la información. Dado que la frecuencia de envío de mensajes AIS es, en el peor de los casos, cada 30 segundos, se establece una frecuencia de pedido de actualización de 1 Hz.

El programa iniciará una conexión serie al iniciar la ejecución. Como se mencionó en la sección del módulo UART-USB, la velocidad de transferencia no debe superar los 9600 baudios debido al ancho de banda de los optoacopladores.

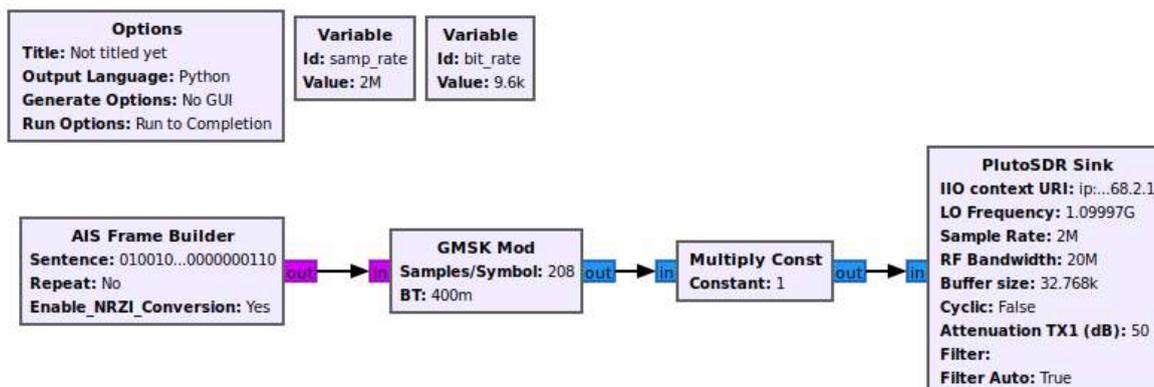
F3.4.3 GNU Radio

GNU Radio es un conjunto de herramientas de desarrollo gratuito y de código abierto que provee bloques de procesamiento de señales para implementar con sistemas de radio mediante software. Este módulo permite controlar el funcionamiento del Pluto SDR que se encarga de la transmisión y recepción. La versión de GNU Radio a utilizar será la v3.8 corriendo sobre un sistema operativo Linux Ubuntu 22.04 LTS.

Además, se utilizarán dos librerías extra. La primera de nombre AISTX del usuario zwh2 en Github. Ésta permitirá codificar los mensajes AIS directamente en GNU Radio. La segunda, de nombre gr-iio, agrega bloques compatibles con algunos SDR comerciales, entre ellos Pluto SDR. Ambas librerías deberán instalarse para el correcto funcionamiento del sistema.

Se implementará la transmisión de mensajes AIS, de acuerdo a los datos de la norma ITU-R M.1371-4 Annex 7, mediante un programa en Python generado en base a un

diagrama de flujo de GNU Radio. El diagrama en cuestión se muestra en la siguiente figura:



Según la norma, la transmisión de los mensajes AIS se implementará con la codificación binaria NRZI a 9600 baudios y modulación GMSK con $B=0.4$. En la figura anterior, se puede apreciar la utilización de los bloques de las librerías mencionadas.

El código Python obtenido de este diagrama, será modificado para aceptar parámetros por línea de comandos. Para ello, se utilizará la librería `OptParser`. La variable de entrada para el bloque AIS Frame Builder es la payload del paquete en código binario. Para generarla, se utilizará la herramienta `Encoder.py` provista por la librería AISTX. Estos programas independientes serán invocados desde el programa principal que muestra la Interfaz de Usuario. La implementación final de este módulo se puede observar en el código al final del documento.

F3.5 Transmisión

F3.5.1 Adalm Pluto SDR

Los bloques anteriores se encargan de generar los mensajes AIS a transmitir. Esta etapa se encarga de la transmisión en sí misma mediante la implementación del dispositivo Pluto SDR.

Adalm Pluto SDR es una herramienta fácil de usar fabricada por Analog Devices. Basado en el transceptor de RF AD9363, el dispositivo ofrece un canal de recepción y un canal de transmisión que pueden ser operados en modo full-duplex con la capacidad de generar o medir señales analógicas de RF desde 325 a 3800 MHz, hasta un máximo de 61.44 Mega Samples por segundo con un ancho de banda de 20 MHz. Mediante la modificación del firmware, el rango de frecuencias puede ser extendido para funcionar desde 70 MHz a 6000 MHz. El dispositivo es alimentado mediante puerto USB y es compatible con los sistemas operativos Windows, OSX y Linux.

La implementación de este sistema considerará que el dispositivo SDR se encuentra conectado a la misma computadora que está corriendo el programa de la Interfaz de Usuario. En caso contrario, deberá modificarse la dirección IP a la que apunta el diagrama de flujo de la sección anterior, de modo que sea capaz de conectarse con el SDR.

F3.5.2 Acceso al Medio

La normativa establece que el sistema AIS debe permitir la coexistencia de distintos dispositivos, para ello se implementa un esquema TDMA. En este esquema, cada minuto es dividido en 2250 slots que son utilizados de a uno por dispositivo.

Para verificar la disponibilidad de slots, los dispositivos clase B utilizan CSTDMA como esquema de acceso al medio. Este método sensa el estado del canal para determinar si es posible realizar la transmisión.

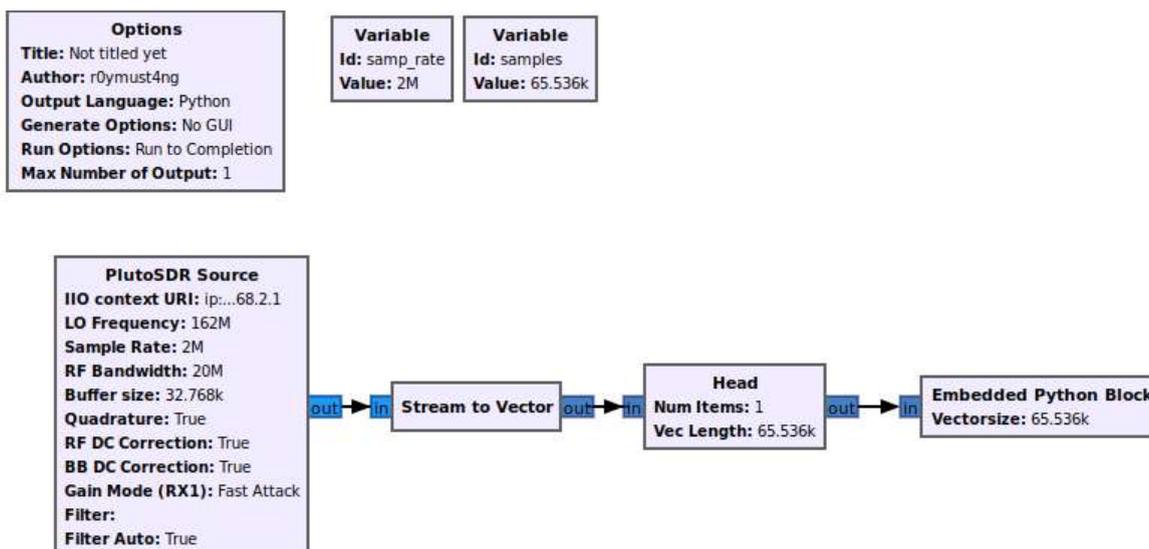
El dispositivo mide la potencia en el canal al inicio de cada slot y la compara con un umbral definido a partir del piso de ruido para determinar si está en uso o disponible. En caso de estar disponible, el mensaje es enviado. En caso de que el canal esté en uso, la transmisión del mensaje se pospone.

Cuando se inicia el sistema, se debe realizar un proceso de calibración. Dicho proceso establecerá el piso de ruido para determinar si los slots están disponibles. El proceso dura un minuto completo. El dispositivo toma una muestra de potencia en el canal cada 4 segundos para un total de 15 muestras. Se identifica la de menor magnitud y se define como valor de piso de ruido. El umbral de sensado se define como:

$$\text{Umbral de Sensado} = \text{Piso Medido} + 10\text{dB}$$

Contra este valor se comparará la potencia medida a la hora de enviar mensajes. En el código, este proceso se implementará utilizando la librería matplotlib de Python que permite un sencillo manejo de las muestras tomadas por el SDR.

El proceso de toma de muestras, se implementa con un diagrama de flujo en GNU Radio. El SDR toma 65536 muestras y las devuelve como un arreglo que será directamente procesado por el programa principal obteniendo el espectro de potencia. La siguiente figura muestra el diagrama de flujo:



Este bloque no existe físicamente en el dispositivo. Su funcionamiento es definido e implementado exclusivamente por software en el código del programa principal. El bloque de código embebido solo cumple la función de mostrar por la consola las muestras capturadas. El código para estas funciones se puede analizar en el código al final del documento.

F3.5.3 Amplificador 2W

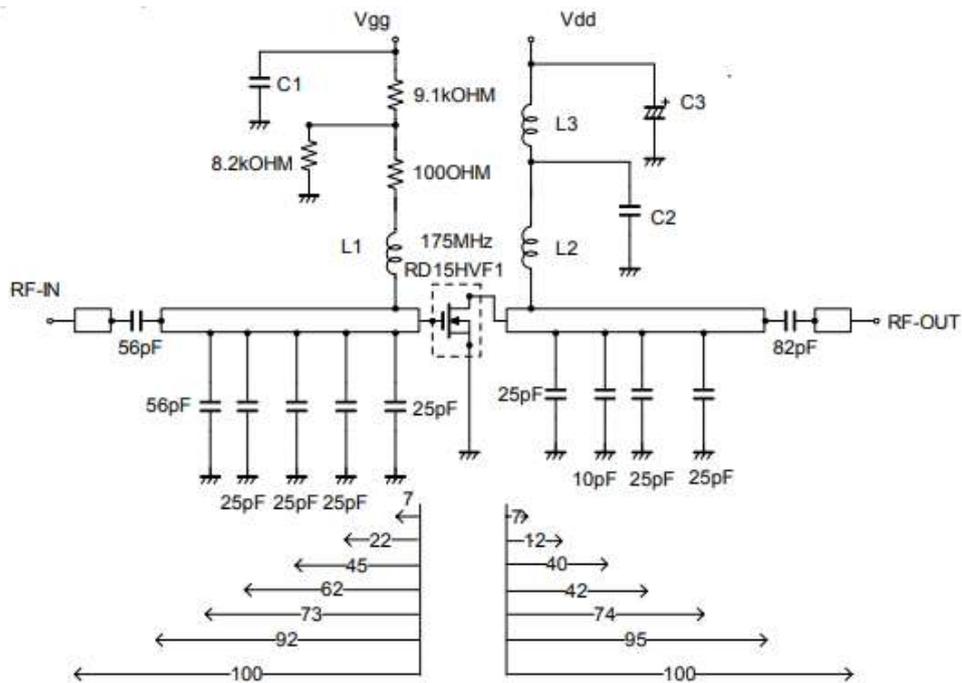
La normativa establece una potencia de transmisión máxima para los transmisores AIS según su clase. Para un transmisor clase B la potencia permitida es de 2 watts. Debido a las limitaciones legales que establece el ENACOM, no está permitido transmitir en las frecuencias reservadas para AIS con esa potencia sin tener los permisos necesarios. Por este motivo el amplificador no será implementado pero se incluye una propuesta de diseño.

F3.5.3.A MOSFET de Potencia para RF: RD15HVF1

El RD15HVF1 es un transistor tipo MOSFET específicamente diseñado para implementar en amplificadores de alta potencia para VHF/UHF.

Sus principales características son una potencia de salida máxima de 15 Watt tanto para VHF como UHF con ganancia máxima de 14 dB a 175 MHz y 7 dB a 520 MHz.

F3.5.3.B Circuito de Prueba a 175 MHz



- C1:2200pF 10uF in parallel
- C2:2200pF*2 in parallel
- C3:2200pF,330uF in parallel

- L1:4Turns,lD6mm,D1.6mm P=1 silver plateted copper wire
- L2:5Turns,lD6mm,D1.6mm P=1 silver plateted copper wire
- L3:4Turns,lD6mm,D1.6mm P=1 silver plateted copper wire

Note:Board material-Teflon substrate
 micro strip line width=4.2mm/50OHM,er=2.7,t=1.6mm
 Dimensions:mm

F4. Referencias

- MCP2551 - High-speed CAN Transceiver.
<https://www.microchip.com/en-us/product/mcp2551>
- MCP2515 - Stand-Alone CAN Controller.
<https://www.microchip.com/en-us/product/mcp2515>
- NMEA2000 library for C++. <https://github.com/ttlappalainen/NMEA2000>
- PL-2303HX USB to Serial.
https://www.prolific.com.tw/US/ShowProduct.aspx?p_id=8&pcid=41
- 4n26 Optocoupler. <https://www.vishay.com/docs/83725/4n25.pdf>
- AISTX por zwh2. <https://github.com/zwh2/aistx>
- RD15HVF1.
<https://www.alldatasheet.es/datasheet-pdf/pdf/143844/MITSUBISHI/RD15HVF1.html>
- ITU-R M.1371-4
https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-4-201004-S!!PDF-E.pdf

Apéndice G: Plan de Pruebas

G1. Acerca del Equipo

Integrante

Roberto S. Gorak. Estudiante Ingeniería Electrónica, FI UNMdP

Director

Dr. Ing. Alejandro José Uriz, Laboratorio de Comunicaciones, FI UNMdP

Co Director

Ing. Brian Gluzman, Laboratorio de Comunicaciones, FI UNMdP

Co Directora

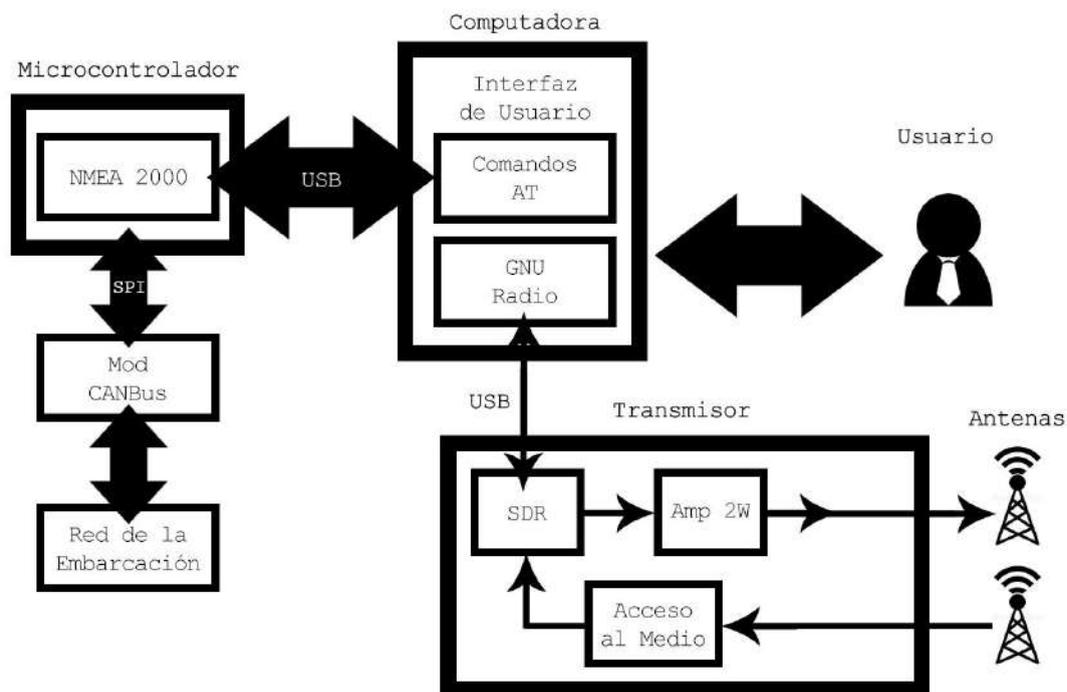
MSc. Mónica Liberatori, Laboratorio de Comunicaciones, FI UNMdP

G2. Acerca del Proyecto Final

El objetivo de este proyecto es diseñar e implementar un prototipo de transmisor compatible con AIS (Sistema de Identificación Automática).

El transmisor cuenta con tres grandes etapas. La primera será un microcontrolador que funcionará como etapa de adquisición de datos de la red de la embarcación. La segunda, un programa de computadora que permitirá el ingreso, modificación y adaptación de los datos obtenidos de los instrumentos del buque y los aportados por el usuario, en la trama digital a ser transmitida. La tercera será el módulo de transmisión en RF a las frecuencias requeridas.

El producto resultante del proyecto será el Transmisor de Identificación Automática para uso Naval (TIAN).



G3. Plan de Pruebas

Alcance

El presente documento abarca los procedimientos y los resultados relacionados con la etapa de pruebas de TIAN. Las pruebas a realizar serán enumeradas junto con su procedimiento. Las pruebas realizadas contarán con detalles y resultados obtenidos durante la ejecución de las mismas.

El objetivo del documento es informar, a quien desee entender el procedimiento de pruebas, acerca de los procedimientos realizados, las dificultades encontradas y los resultados obtenidos.

El proyecto fue evaluado según la escala de TRL (Technology Readiness Level), en dicha escala se lo considera en nivel TRL = 4. Los bloques básicos se integran para comprobar que las piezas funcionan como un sistema. Las pruebas a realizar se clasifican en pruebas unitarias, pruebas de integración y pruebas de homologación.

Ambientes de Prueba

La totalidad de los procedimientos de pruebas se realizarán en el Laboratorio de Comunicaciones de la Facultad de Ingeniería de la UNMdp

Recursos, Herramientas e Instrumentos

- Computadora
- Python IDLE
- Programador ATMEL AVR
- Microcontroladores ATMEGA328P
- Pluto SDR
- RTL-SDR
- GNU Radio
- Actisense NMEA Reader
- Monitor Comunicación Serial
- SDRAngel
- OpenCPN

Políticas de Trabajo

Durante la etapa de pruebas se busca demostrar el funcionamiento correcto del sistema así como de los bloques que lo componen.

Estrategia de Comunicación

Se establecerá como medio principal de comunicación el uso de correos electrónicos. Estos se utilizarán a discreción con la condición que cada mensaje deberá ser enviado a todos los participantes de las pruebas. Se utilizará en menor medida, y previa coordinación de los interesados, el uso de conferencias por videollamada para temas que requieran un mayor dinamismo que el provisto por los correos electrónicos. Los documentos generados como resultado de las pruebas realizadas, deberán ser distribuidos mediante los correos antes mencionados. Además deberán ser archivados en una unidad virtual en la nube que sea de libre acceso para todos los participantes.

G4. Banco de Pruebas

Objetivo

Durante las pruebas se montará un banco que permita apreciar el correcto funcionamiento del sistema y los bloques que lo componen. Para ello, además del desarrollo de TIAN, se implementará un Simulador de Tráfico de Red y un sistema de recepción.

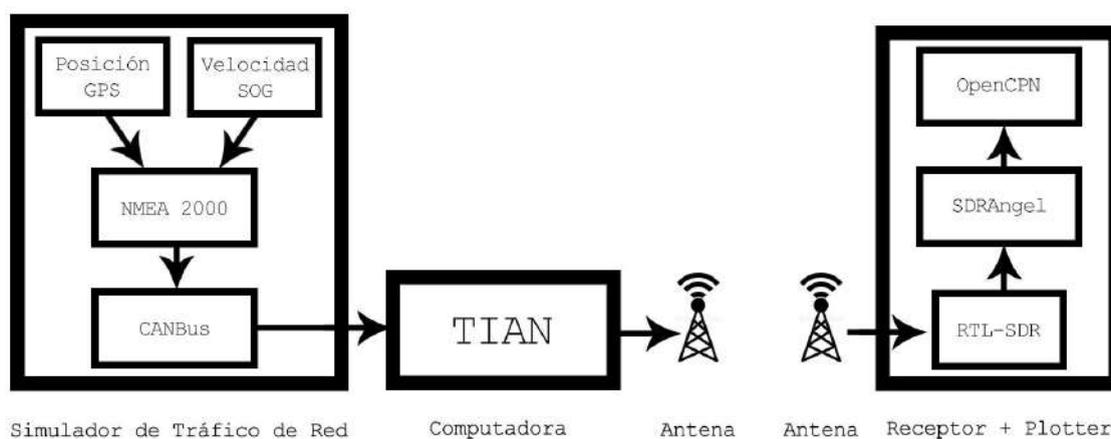
El Simulador cumplirá la función de enviar tramas NMEA 2000 al Adquisidor de Datos de TIAN como si se tratara de una embarcación real. Los mensajes que se enviarán, se programarán directamente en el microcontrolador. Una parte de los mensajes serán los que TIAN necesita para funcionar y otra parte serán mensajes de descarte. Los

mensajes de descarte serán filtrados por el sistema. El simulador contará con dos entradas en forma de potenciómetros que permitirán cambiar en tiempo real los valores de algunos de los mensajes a enviar. Para este banco se asocian éstas entradas con la posición GPS y la velocidad SOG de la nave.

Como última prueba, también se implementará un sistema receptor que pueda procesar la transmisión que realiza el sistema. Para ello se utilizará un RTL-SDR conectado a una PC corriendo SDRAngel y OpenCPN. SDRAngel procesará la información recibida por el SDR y decodificará los paquetes AIS. Los datos ya procesados, serán enviados a OpenCPN. Éste último, graficará los datos recibidos en su interfaz de ploteo. De esta forma, los datos recibidos de TIAN se podrán apreciar en un sistema gráfico preparado para tal fin.

Cabe aclarar que debido a la normativa establecida por ENACOM, todas las pruebas se realizarán sin implementar el bloque del Amplificador de 2 Watt y a una frecuencia de 1.1 GHz para evitar las transmisiones a frecuencias reservadas.

Diagrama en Bloques



G5. Casos de Prueba

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
Mod CAN Bus	#1	Verificación de comunicación por bus CAN	Unitaria	Cargar un programa básico de comunicación en dos microcontroladores. Verificar el envío y recepción de datos entre ambos.	PC, ATMEGA328P, Monitor serie	Los datos enviados desde un extremo deben ser recibidos desde el otro.	2022	1	Gorak, Roberto	No se consigue establecer la comunicación.	Se recomienda realizar mediciones sobre los componentes que configuran a los microcontroladores antes de realizar nuevas pruebas.
	#2	Verificación de comunicación por bus CAN	Unitaria	Cargar un programa básico de comunicación en dos microcontroladores. Verificar el envío y recepción de datos entre ambos.	PC, ATMEGA328P, Monitor serie	Los datos enviados desde un extremo deben ser recibidos desde el otro.	2022	2	Gorak, Roberto	Se establece la conexión esperada.	Los cristales que proveían el oscilador para los microcontroladores estaban defectuosos. Luego de ser reemplazados el sistema responde correctamente.
Simulador de Tráfico de Red	#3	Verificación de generación de tramas NMEA 2000	Unitaria	Cargar un programa que genera tramas NMEA 2000. Enviar las tramas simultáneamente por puerto serie a la PC y por bus CAN.	PC, ATMEGA328P, Monitor serie, Actisense NMEA Reader	El software Actisense NMEA Reader debe reconocer el puerto serie asignado y decodificar correctamente los paquetes generados por el simulador. En el otro extremo, el monitor serie debe mostrar las tramas recibidas.	2022	1	Gorak, Roberto	Se establece la conexión esperada.	Debido a las limitaciones de los puertos serie, no se puede leer un puerto con ANR y el monitor serie simultáneamente.

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
NMEA 2000	#4	Verificación de comunicación por protocolo NMEA 2000	Unitaria	Cargar un programa básico de comunicación en dos microcontroladores. Verificar el envío y recepción de datos entre ambos.	PC, ATMEGA328P, Monitor serie	Los datos enviados desde un extremo deben ser recibidos desde el otro.	2022	1	Gorak, Roberto	Se establece la conexión esperada.	
NMEA 2000	#5	Verificación de filtrado de tramas NMEA 2000 en el extremo final	Unitaria	Cargar un programa que reciba las tramas enviadas. Según su identificador PGN, discriminar las tramas útiles y procesar sus datos. Descartar las tramas que no son de utilidad. Enviar los datos procesados por puerto serie.	PC, ATMEGA328P, Monitor serie	En el monitor serie se debe observar los datos procesados de las tramas útiles recibidas.	2022	1	Gorak, Roberto	El filtrado funciona correctamente.	En este estadio, las tramas cargadas tanto en el Simulador de Tráfico de Red como en el filtro, son solo a modo de prueba. Los valores reales serán cargados para las pruebas integrales.
Comandos AT	#6	Prueba de comunicación serie entre PC y ATMEGA328P	Integral	Cargar un programa en Python que envíe un comando AT al microcontrolador. Verificar la comunicación.	PC, ATMEGA328P	El programa en Python debe enviar un comando AT que el microcontrolador debe interpretar como una señal para enviar los últimos datos guardados que haya recibido del Simulador de Tráfico de Red.	2022	1	Gorak, Roberto	El programa de Python y el integrado responden como era esperado.	
Interfaz de Usuario	#7	Prueba de comunicación entre GUI y Adquisidor de datos	Integral	Conectar el Simulador de Tráfico de Red, Adquisidor de Datos y la PC. Ejecutar el programa en Python para establecer la comunicación.	PC, ATMEGA328P, Monitor serie	Los datos generados por el Simulador de Tráfico de Red deben ser filtrados y visualizados en la sección correspondiente en el programa en Python.	2023	1	Gorak, Roberto	Los datos son mostrados correctamente.	Se formatean los datos recibidos para que coincidan con los datos que utiliza Python.

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
GNU Radio	#8	Pruebas iniciales de compatibilidad de librerías	Unitaria	Instalar el software GNU Radio. Instalar la librerías para envío de mensajes AIS y bloques de conexión con RTL-SDR y PlutoSDR	PC	Las librerías instaladas deben proveer los bloques necesarios para la creación de un diagrama de flujo que permita hacer uso de las capacidades de los SDR mencionados.	2022	1	Gorak, Roberto	Las librerías no pueden ser instaladas en un entorno de Windows	GNU Radio funciona en un entorno llamado Radioconda en Windows. Se recomienda estudiar la documentación para agregar los paquetes faltantes a Radioconda.
GNU Radio	#9	Pruebas iniciales de compatibilidad de librerías	Unitaria	Instalar el software GNU Radio. Instalar la librerías para envío de mensajes AIS y bloques de conexión con RTL-SDR y PlutoSDR	PC	Las librerías instaladas deben proveer los bloques necesarios para la creación de un diagrama de flujo que permita hacer uso de las capacidades de los SDR mencionados.	2022	2	Gorak, Roberto	Las librerías no pueden ser instaladas en el entorno de Radioconda en Windows	La instalación de paquetes en Radioconda es limitada y engorrosa. Habiendo instalado los paquetes se generan nuevos conflictos entre dependencias. Se recomienda migrar la etapa de software de PC a un entorno basado en Linux.
	#10	Pruebas iniciales de compatibilidad de librerías en Ubuntu	Unitaria	Instalar el software GNU Radio. Instalar la librerías para envío de mensajes AIS y bloques de conexión con RTL-SDR y PlutoSDR	PC	Las librerías instaladas deben proveer los bloques necesarios para la creación de un diagrama de flujo que permita hacer uso de las capacidades de los SDR mencionados.	2022	3	Gorak, Roberto	Las librerías se instalaron correctamente y se muestran como se esperaba en GNU Radio.	

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
SDR	#11	Transmisión Estática	Unitaria	Generar un diagrama de flujo en GNU Radio con la librería gr-aistx y PlutoSDR. Cargar el AIS Frame Builder con una payload conocida y generada por el script Encoder proporcionado por la misma librería. Transmitir a muy baja potencia utilizando PlutoSDR. Recibir la transmisión utilizando RTL-SDR conectado a SDRAngel. Plotear los datos recibidos mediante OpenCPN	PC, PlutoSDR, RTL-SDR, SDRAngel, OpenCPN	El decodificador AIS de SDRAngel debe mostrar el mensaje recibido y decodificar correctamente los datos incluidos. Debe enviar estos datos a OpenCPN. OpenCPN debe graficar en su interfaz una nave que se corresponda con los datos ingresados.	2023	1	Gorak, Roberto	SDRAngel muestra los datos recibidos y decodificados. OpenCPN grafica correctamente la nave.	
	#12	Transmisión Estática con parámetros	Unitaria	Generar un script de Python basado en el diagrama de flujo de la prueba #11 y el script de ejemplo provisto con la librería. El script recibe parámetros por consola. Utilizar SDRAngel y OpenCPN para confirmar los datos.	PC, PlutoSDR, RTL-SDR, SDRAngel, OpenCPN	El decodificador AIS de SDRAngel debe mostrar el mensaje recibido y decodificar correctamente los datos incluido. Debe enviar estos datos a OpenCPN. OpenCPN debe graficar en su interfaz una nave que se corresponda con los datos ingresados.	2023	1	Gorak, Roberto	El intérprete de Python presenta errores de ejecución. Los datos no son transmitidos.	El script de demostración incluido en la librería gr-aistx se encuentra muy desactualizado. Se recomienda generar un script totalmente nuevo.

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
SDR	#13	Transmisión Estática con parámetros	Unitaria	Generar un script de Python basado en el diagrama de flujo de la prueba #11. El script recibe parámetros por consola. Utilizar SDRAngel y OpenCPN para confirmar los datos.	PC, PlutoSDR, RTL-SDR, SDRAngel, OpenCPN	El decodificador AIS de SDRAngel debe mostrar el mensaje recibido y decodificar correctamente los datos incluido. Debe enviar estos datos a OpenCPN. OpenCPN debe graficar en su interfaz una nave que se corresponda con los datos ingresados.	2023	2	Gorak, Roberto	SDRAngel muestra los datos recibidos y decodificados. OpenCPN grafica correctamente la nave.	
Acceso al medio	#14	Medición del piso de ruido en el rango de frecuencias de interés	Unitaria	Generar un script de Python usando PlutoSDR que adquiera muestras del rango de frecuencias de interés. Procesarlas y obtener el espectro de potencia. Obtener el menor valor de potencia medido y definirlo como piso de ruido. Graficar las muestras para confirmar visualmente la medición.	PC, PlutoSDR	El script debe tomar un número fijo de muestras, calcular la FFT de las muestras y obtener el mínimo valor de potencia que será el piso de ruido buscado. El piso de ruido será mostrado por la consola. El espectro de potencia calculado debe ser graficado para confirmar la medición.	2023	1	Gorak, Roberto	El intérprete de Python presenta errores de compatibilidad entre la librería pyadi y gr-iiio.	Ambas librerías poseen una dependencia común que requiere distintas versiones de la misma para funcionar.

Módulo	ID	Prueba	Tipo	Procedimiento	Instrumental	Resultado Esperado	Fecha	Iter.	Responsable	Resultado Obtenido	Observaciones
Acceso al medio	#15	Medición del piso de ruido en el rango de frecuencias de interés	Unitaria	Generar un script de Python basado en GNU Radio usando PlutoSDR que adquiera muestras del rango de frecuencias de interés. Procesar las muestras y obtener el espectro de potencia. Obtener el menor valor de potencia medido y definirlo como piso de ruido. Graficar las muestras procesadas para confirmar visualmente la medición.	PC, PlutoSDR	El script basado en GNU Radio debe tomar un número fijo de muestras. El script en Python toma las muestras para calcular la FFT y obtener el mínimo valor de potencia que será el piso de ruido buscado. El piso de ruido será mostrado por la consola. El espectro de potencia calculado debe ser graficado para confirmar la medición.	2023	2	Gorak, Roberto	El gráfico del espectro de potencia coincide con el cálculo del piso de ruido realizado.	Se agrega a ésta etapa la medición de la potencia en los canales AIS para confirmar la ausencia de otras estaciones transmitiendo.
TIAN	#16	Prueba de funcionamiento del sistema completo	Integral	Conectar todos los módulos desarrollados hasta el momento. Ejecutar el programa de Interfaz de Usuario y proceder con la transmisión.	PC, ATMEGA328P, PlutoSDR, RTL-SDR, SDRAngel, OpenCPN	La Interfaz de Usuario debe mostrar los datos estáticos cargados por el usuario, los datos dinámicos obtenidos del Simulador de Tráfico de Red mediante el Adquisidor de Datos y mostrar los datos transmitidos como mensajes AIS a través del PlutoSDR. En una PC por separado, el SDRAngel conectado con RTL-SDR y OpenCPN, debe recibir los datos transmitidos por TIAN y plotearlos en OpenCPN.	2023	1	Gorak, Roberto	El sistema funciona como era esperado.	El sistema no cuenta con módulo GPS propio, como lo establece la norma, con el objetivo de reemplazar estos datos con un potenciómetro que permite apreciar desplazamientos arbitrarios en el software de ploteo. Por restricciones legales del ENACOM, el sistema no cuenta con el amplificador de 2 Watts que establece la norma.

Referencias

- [1] Organización Marítima Internacional. <https://www.imo.org>
- [2] Everything you need to know about AIS. <http://www.allaboutais.com>
- [3] Technology readiness level.
https://en.wikipedia.org/wiki/Technology_readiness_level
- [4] ITU-R M.1371-4: *“Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band”*
https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-4-201004-S!!PDF-E.pdf
- [5] GanttProject. <https://www.ganttproject.biz>
- [6] NMEA 2000. https://es.wikipedia.org/wiki/NMEA_2000
- [7] MCP2551 - High-speed CAN Transceiver.
<https://www.microchip.com/en-us/product/mcp2551>
- [8] MCP2515 - Stand-Alone CAN Controller with SPI Interface.
<https://www.microchip.com/en-us/product/mcp2515>
- [9] *“MCP2515 CAN-BUS Arduino Tutorial – Getting started, Interfacing, Applications”*
<https://www.seeedstudio.com/blog/2020/06/05/mcp2515-can-bus-arduino-tutorial-getting-started-interfacing-applications/>
- [10] ATMEGA328P. <https://www.microchip.com/en-us/product/atmega328p>
- [11] NMEA2000 library for C++. <https://github.com/ttlappalainen/NMEA2000>
- [12] PL-2303HX USB to Serial Bridge Controller.
https://www.prolific.com.tw/US/ShowProduct.aspx?p_id=8&pcid=41
- [13] Optocoupler, Phototransistor Output, with Base Connection.
<https://www.vishay.com/docs/83725/4n25.pdf>
- [14] GNU Radio. <https://www.gnuradio.org>
- [15] ADALM-PLUTO Software-Defined Radio Active Learning Module.
<https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview>
- [16] AISTX por zwh2. <https://github.com/zwh2/aistx>
- [17] Servicio Móvil Marítimo y Aeronáutico.
https://www.enacom.gob.ar/servicio-movil-maritimo-y-aeronautico_p126
- [18] PCB Wizard 3. <http://www.new-wave-concepts.com/ed/wizard.html>
- [19] Actisense NMEA Reader. https://actisense.com/acti_software/nmea-reader/
- [20] RTL-SDR Blog V3. <https://www.rtl-sdr.com/V3>
- [21] SDRangel – Open-source TX & RX Software Defined Radio.
<https://www.sdrangel.org>
- [22] OpenCPN Official Site. <https://opencpn.org>
- [23] Mitsubishi RF Power MosFET RD15HVF1.
<https://www.alldatasheet.es/datasheet-pdf/pdf/143844/MITSUBISHI/RD15HVF1.html>