

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO HARDWARE PARA UN BANCO DE PRUEBAS DEL ESTÁNDAR AIS

Titulación: Grado en Ingeniería en Tecnologías de la

Telecomunicación

Autor: D. Nicolás Molina Padrón

Tutores: Dr. Francisco José Cabrera Almeida

Dr. Víctor Alexis Araña Pulido

Fecha: Julio de 2015



ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO HARDWARE PARA UN BANCO DE PRUEBAS DEL ESTÁNDAR AIS

HOJA DE FIRMAS

Firma de los tutores

Fdo. Dr. Francisco José Cabrera Almeida Fdo. Dr. Víctor Alexis Araña Pulido

Firma del alumno

Fdo. D. Nicolás Molina Padrón

Fecha: Julio de 2015



ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO HARDWARE PARA UN BANCO DE PRUEBAS DEL ESTÁNDAR AIS

HOJA DE EVALUACIÓN

	Calificación: _				
	Presidente		Secretario		Vocal
Fdo.: _		Fdo.: _		Fdo.:	

Fecha: Julio de 2015

Estudia arduamente lo que más te interese, de la forma más indisciplinada, irreverente y original que puedas.

Richard P. Feynman, premio Nobel de Física en 1965.

Agradecimientos

A mis padres, por todo el esfuerzo y los sacrificios que han hecho para que pueda estar hoy aquí, y por darme la fuerza y la confianza que a veces no tengo en mí mismo. Y mi hermano Rafa, que pese a nuestras diferencias, nos apreciamos más de lo que parece.

A mi tía Pepa, por ser desde que tengo memoria mi segunda madre, y a mi tío Vicente, que una vez me dijo que la fórmula del éxito es H_2O (Honestidad, Humildad y Osadía). Y también por darme el mejor regalo posible: mi prima Pepilla, a la que llevo incordiando desde que nació, pero ella sabe que es con cariño. Y a mi tío Octavio, porque siempre ha estado ahí para mi familia y para mí.

A mis amigos Ares y Kito, por todas las carcajadas que nos hemos sacado todos estos años. Bea y Rachel, con quienes que viví ese milagroso ascenso de la U.D. Las Palmas este año (y alguna lagrimilla que preferimos no reconocer en público). A Cris, por todas esas conversaciones sobre libros y viajes que amenizaron estos últimos meses agobiantes. Y a Guzmán, por todo lo que aprendí de él y por creer en mí.

A mis compañeros de clase. Hago especial mención a mi amigo Héctor, que nunca dudó en echarme un cable cuando lo necesité, y a quien eché en falta este último curso por Tafira. Y al resto de compañeros que, de una forma u otra, me ayudaron en muchas etapas de mi formación, especialmente Yoel, Santana y Gara.

Y a mis tutores Francis y Víctor, por la ayuda y el apoyo que me brindaron durante la elaboración de este trabajo, y por la confianza que depositaron en mí. Y a Juan Domingo, por la ayuda técnica que me ofreció en el laboratorio.

Índice general

A	cróni	mos	ΧV					
Ι	Memoria							
1.	Intr	roducción	3					
	1.1.	Antecedentes	3					
	1.2.	Objetivos	7					
	1.3.	Estructura de la memoria	8					
2.	Des	cripción del estándar AIS	9					
	2.1.	Introducción	9					
	2.2.	Regulación y regulación del estándar AIS	9					
	2.3.	Arquitectura de red	10					
		2.3.1. Nivel Físico	12					
		2.3.2. Nivel de Enlace	18					
	2.4.	Tipos de mensajes AIS	28					
	2.5.	Descripción hardware de los transceptores AIS	30					
		2.5.1. AIS clase A	30					
		2.5.2 AIS clase B	31					

3.	Dise	eño mo	odular del prototipo hardware	33			
	3.1.	. Introducción					
	3.2.	Arduino UNO R3					
		3.2.1.	Microcontrolador ATmega328P	34			
		3.2.2.	Entradas y salidas digitales	35			
		3.2.3.	Entradas y salidas analógicas	35			
		3.2.4.	Puerto serie	36			
	3.3.	Módul	lo banda base DV-MEGA GMSK	36			
		3.3.1.	Pines del módem DV-MEGA GMSK	37			
		3.3.2.	CMX589A	38			
		3.3.3.	Circuito de sincronismo externo	39			
		3.3.4.	Etapas de filtrado y ganancia en transmisión	40			
		3.3.5.	Etapas de filtrado y ganancia en recepción	42			
		3.3.6.	Push To Talk	43			
		3.3.7.	Conector Mini-DIN 6	43			
	3.4.	Módul	los paso banda FM Radiometrix	45			
		3.4.1.	Radiometrix TX1	45			
		3.4.2.	Radiometrix RX1	47			
	3.5.	Anten	a de VHF marítimo	50			
4.	Inte	gració	n del banco de pruebas AIS	51			
	4.1.	O	ucción	51			
	4.2.		ación de los módulos	51			
			amación de un firmware de control en Arduino	53			
		4.3.1.	Funciones propias de Arduino	54			

		4.3.2.	Interrupciones en Arduino	56
	4.4.	Desarr	collo del firmware de control del transmisor	57
		4.4.1.	Definiciones y variables globales	58
		4.4.2.	Inicialización del programa	59
		4.4.3.	Programa principal	60
		4.4.4.	Rutina de servicio del transmisor	62
	4.5.	Desarr	collo del firmware de control del receptor	64
		4.5.1.	Definiciones y variables globales	64
		4.5.2.	Inicialización del programa	65
		4.5.3.	Programa principal	65
		4.5.4.	Rutina de servicio del receptor	66
	4.6.	Descri	pción de la interfaz Arduino - Puerto serie	70
		4.6.1.	Interfaz de transmisión	70
		4.6.2.	Interfaz de recepción	71
5 .	Pru	ebas y	resultados	73
	5.1.	Introd	ucción	73
	5.2.	Prueb	as sobre los módulos paso banda	73
		5.2.1.	Pruebas sobre el módulo Radiometrix TX1	73
		5.2.2.	Pruebas sobre el módulo Radiometrix RX1	76
	5.3.	Prueb	as sobre los módulos banda base	78
		5.3.1.	Prueba de transmisión y recepción de una secuencia binaria alternada	79
		5.3.2.	Prueba de transmisión y recepción de una secuencia binaria aleatoria	81
		5.3.3.	Prueba de transmisión y recepción de un carácter conocido	82

	5.4.	Pruebas de integración de los módulos paso banda y banda base	85
		5.4.1. Medidas sobre antenas de VHF marítimo	86
		5.4.2. Pruebas de ancho de banda para tramas AIS	88
		5.4.3. Prueba de envío y recepción de tramas AIS	89
	5.5.	Prueba real de recepción de tramas AIS con el prototipo hardware	91
6.	Con	nclusiones	95
	6.1.	Resultados y Revisión de Objetivos	95
	6.2.	Líneas Futuras	96
II	P	resupuesto	99
Pr	esup	puesto 10	01
	P.1.	Introducción	01
	P.2.	Trabajo tarifado por tiempo empleado	01
	P.3.	Amortización del inmovilizado material	02
		P.3.1. Amortización de los recursos hardware	02
		P.3.2. Amortización de los recursos software	03
	P.4.	Redacción de la documentación	04
	P.5.	Derechos de visado del COITT	05
	P.6.	Gastos de tramitación y envío	06
	P.7.	Aplicación de impuestos y coste total	06
II	\mathbf{I}	Anexos 10)9
Α.	Des	cripción del chip CMX589A	11

ÍNDICE GENERAL	XV
B. PCB del módulo transmisor	129
C. PCB del módulo receptor	131
D. Código del firmware de control del transmisor	133
E. Código del firmware de control del receptor	137
F. Publicación de resultados	141

Índice de figuras

1.1.	Cobertura global AIS por satélite	4
1.2.	Visualización de un sistema radar convencional frente al sistema AIS $$.	5
1.3.	Funcionamiento de SOTDMA entre dos embarcaciones	6
1.4.	Receptor AIS cedido por Marine Traffic al IDeTIC	7
1.5.	Diagrama de bloques del banco de pruebas transmisor AIS	7
1.6.	Diagrama de bloques del banco de pruebas receptor AIS	8
2.1.	Esquema general de la arquitectura de red del estándar AIS	11
2.2.	Descripción de la banda de trabajo del servicio AIS	12
2.3.	Esquema de codificación y de modulación en el estándar AIS	13
2.4.	Ejemplo de codificación NRZI sobre una secuencia binaria	14
2.5.	Comparativa entre los espectros FSK y GMSK	15
2.6.	Respuesta al impulso de un filtro gaussiano ideal	15
2.7.	Respuesta al impulso de un filtro gaussiano con $BT=0.3\ \mathrm{y}\ BT=0.5$.	16
2.8.	Interferencia entre símbolos en GMSK para factores $BT=0.3 \; \mathrm{y} \; BT=0.5$	17
2.9.	Modulador y demodulador FM-GMSK	18
2.10.	(a) Sincronismo UTC directo, (b) Sincronismo UTC indirecto	19
2.11.	Formato de una trama AIS	20
2.12.	Esquema TDMA en AIS para las frecuencia AIS-1 y AIS-2	21

2.13. Esquemas TDMA en AIS en función de su aplicación	21
2.14. Algoritmo de la fase de inicialización en SOTDMA	22
2.15. Selección de NTS en un intervalo SI	23
2.16. Algoritmo de la fase de entrada en red en SOTDMA	24
2.17. Selección de NS definido en un NI	25
2.18. Algoritmo de la fase de operación continua en SOTDMA	25
2.19. Campos incluidos en un mensaje AIS simple	29
2.20. Campos incluidos en un mensaje AIS compuesto	29
2.21. Diagrama de bloques de un transpondedor AIS clase A	31
2.22. Diagrama de bloques de un transpondedor AIS clase B	32
3.1. Diferentes modelos de plataforma Arduino	33
3.2. Microcontrolador Arduino UNO R3	34
3.3. Conectores USB tipo A/B macho y hembra sobre Arduino UNO	36
3.4. Shield DV-MEGA GMSK	37
3.5. Encapsulado del chip CMX589A	38
3.6. Subsistemas integrados al chip CMX589A	39
3.7. Circuito de sincronismo externo al chip CMX589A para 9.8 kbps	40
3.8. Filtro pasivo RC paso bajo	40
3.9. Filtro activo paso bajo	41
3.10. Filtro de transmisión	42
3.11. Filtro de recepción	42
3.12. Circuito PTT del módem DV-MEGA GMSK	43
3.13. Dimensiones del conector Mini-DIN 6 (expresadas en milímetros, mm) .	44
3.14. Pines del conector Mini-DIN 6	44

3.15. Dimensiones físicas del chip Radiometrix TX1	45
3.16. Diagrama de bloques del chip Radiometrix TX1	46
3.17. Dimensiones físicas del chip Radiometrix RX1	47
3.18. Diagrama de bloques del chip Radiometrix RX1	49
3.19. Antena VHF marítimo Kenwood SMA macho	50
4.1. Diagrama de bloques del prototipo hardware del transmisor AIS	52
4.2. Diagrama de bloques del prototipo hardware del receptor AIS	52
4.3. Condiciones de disparo en Arduino	57
4.4. Flujograma del programa principal del firmware de control del transmisor	61
4.5. Flujograma de la rutina de servicio del firmware de control del transmisor	63
4.6. Flujograma del programa principal del firmware de control del receptor	66
4.7. Flujograma de la rutina de servicio del firmware de control del receptor	69
4.8. Banco de pruebas transmisor y receptor con interfaces Serial	70
4.9. Interfaz de la aplicación <i>CoolTerm</i>	71
$4.10.$ Envío de cadenas de caracteres por el puerto serie con $Send\ String$	71
4.11. Interfaz de la aplicación $Serial\ Monitor$ de Matlab 2014b	72
5.1. Montaje de prueba para el chip Radiometrix TX1	74
5.2. Señal portadora a 161.975 MHz generada por el chip Radiometrix TX1	75
5.3. Espectro FM de una señal cuadrada como moduladora	76
5.4. Montaje de prueba para el chip Radiometrix RX1	77
5.5. Señal demodulada por el chip Radiometrix RX1	77
5.6. Montaje de prueba para el módem DV-MEGA GMSK	78
5.7. Algoritmo de prueba de transmisión de una secuencia binaria alternada	79

5.8.	Algoritmo de prueba de recepción de una secuencia binaria alternada .	80
5.9.	Visualización en osciloscopio multicanal de una secuencia binaria alternada en el módem DV-MEGA GMSK	80
5.10.	Algoritmo de prueba de transmisión de una secuencia binaria aleatoria	81
5.11.	Visualización en osciloscopio multicanal de una secuencia binaria aleatoria en el módem DV-MEGA GMSK	82
5.12.	Algoritmo de prueba de transmisión de una secuencia binaria conocida	83
5.13.	Visualización en osciloscopio multicanal de un dato conocido en el módem DV-MEGA GMSK	84
5.14.	Recepción del carácter 52 desde el módem GMSK hasta Serial Monitor	84
5.15.	Montaje de prueba para la visualización por pantalla de tramas AIS	85
5.16.	Banco de pruebas para la visualización por pantalla de tramas AIS $$. $$.	86
5.17.	Medida del parámetro S_{11} para la antena VHF transmisora	87
5.18.	Medida del parámetro S_{11} para la antena VHF receptora	87
5.19.	Montaje de medida del ancho de banda de transmisión de tramas AIS .	88
5.20.	Trama AIS de test sobre la interfaz $Send\ String\ de\ CoolTerm$	88
5.21.	Trama AIS modulada en FM-GMSK por el prototipo hardware del transmisor	89
5.22.	Trama AIS modulada en FM-GMSK por el prototipo hardware transmisor	90
5.23.	Envío de la trama AIS durante 30 iteraciones ($CoolTerm0$ como transmisor, $CoolTerm1$ como receptor)	90
5.24.	Montaje del banco de pruebas de recepción de tramas AIS reales	91
5.25.	Montaje en laboratorio del banco de pruebas de recepción de tramas AIS reales	91
5.26.	Banda de trabajo AIS a la frecuencia 161.975 MHz	92
5.27.	Forma de onda de la trama AIS recibida	93
5.28.	Comprobación de los slots en las tramas AIS recibidas	94

A.1. Encapsulado del chip CMX589A	111
A.2. Subsistemas integrados al chip CMX589A	112
A.3. Diagrama de pines del chip CMX589A	113
A.4. Topología de sincronismo local con oscilador XTAL	115
A.5. Topología de sincronismo externo para $EXTCLK > 10.24~\mathrm{MHz}$	116
A.6. Topología de sincronismo externo para $EXTCLK \leq 10.24~\mathrm{MHz}$	116
A.7. Selección de R_B en función de f_{xtal} y los estados sobre ClkDivA y ClkDivE	3117
A.8. Mínimas conexiones del CMX589A para $R_B=4800bps$	117
A.9. Filtro RC para la selección del factor BT	118
A.10. Tabla de selección de R_1 y C_1 en función de R_B y factor BT	119
A.11. Tabla de modos de funcionamiento del PLL en recepción	121
A.12.Modos de funcionamiento de control en recepción	122
A.13. Relación entre BER y relación S/N para distintos factores BT $\ .\ .\ .\ .$.	123
A.14. Relación entre tiempos de adquisición y relación S/N para distintos factores BT	124
A.15.Respuesta al impulso del filtro de transmisión	125
A.16.Encapsulado del modelo CMX589AE2	127
A.17.Diagrama de pines del chip CMX589AD5	127
A.18.Diagrama de pines del chip CMX589AD2	128
A.19.Diagrama de pines del chip CMX589AP4	128
B.1. Esquemático del módulo transmisor para el banco de pruebas AIS $$. $$.	129
B.2. Diseño PCB sobre Eagle PCB Design del transmisor FM	130
B.3. Implementación física de la PCB del transmisor	130
C.1. Esquemático del módulo receptor para el banco de pruebas AIS	131

				[G]			

37	37	T.	т
X	X		ı

C.2.	Diseño PCB sobre Eagle PCB Design del receptor FM	132
C.3.	Implementación física de la PCB del receptor	132

Índice de tablas

2.1.	Parámetros radio generales del estándar AIS	12
2.2.	Características especiales para transmisores AIS	13
2.3.	Características especiales para receptores AIS	13
2.4.	Campos de datos de un mensaje AIS codificado con NMEA 0183	30
3.1.	Descripción de los pines del módem DV-MEGA GMSK	38
3.2.	Características del chip Radiometrix TX1	47
3.3.	Características del chip Radiometrix RX1	49
3.4.	Características de la antena de VHF marítimo Kenwood	50
P.1.	Tabla de precios del material hardware fungible	103
P.2.	Tabla de precios del material hardware no fungible	103
P.3.	Tabla de precios del material software	104
P.4.	Presupuesto según el trabajo tarifado y la amortización del inmovilizado material	105
P.5.	Presupuesto incluyendo la redacción de la memoria	106
P.6.	Presupuesto final del TFG	107

Acrónimos

Acrónimo	Descripción
ADC	Analog-to-Digital Converter
AIS	Automatic Identification System
AtoN	Aid to Navigation
BER	Bit Error Rate
COITT	Colegio Oficial de Ingenieros Técnicos de Telecomunicación
CRC	Código de Redundancia Cíclica
CSTDMA	Carrier Sense Time Division Multiple Access
DGPS	Differential Global Positioning System
DIN	Deutsches Institut für Normung
DLS	Data Link Service
EEPROM	Electrically Erasable Programmable Read Only Memory
FATDMA	Fixed Access Time Division Multiple Access
\mathbf{FM}	Frequency Modulation
GMDSS	Global Maritime Distress and Safety System
GMSK	Gaussian Minimum Shift Keying
GPS	Global Positioning System
GSM	Global System for Mobile communications
HDLC	High-Level Data Link Control
IDE	Integrated Development Environment
IDeTIC	Instituto para el Desarrollo Tecnológico y la Innovación en
	Comunicaciones
IMO	International Maritime Organization
ISR	Interrupt Service Routine
ITDMA	Incremental Time Division Multiple Access
ITU	International Telecommunication Union
KB	Kilobyte
LME	Link Management Entity
MAC	Medium Access Control
MMSI	Maritime Mobile Service Identity
MN	Milla náutica
MSC	Maritime Safety Committee
NI	Nominal Interval
NMEA	National Marine Electronics Association
NRZI	Non Return Zero Inverted
NS	Nominal Slot
NSS	Nominal Start Slot
NTS	Nominal Transmission Slot

OPAMP | Operational Amplifier

OSI Open System Interconnection

PATDMA | Pre-Announced Time Division Multiple Access

PCB Printed Circuit Board

PTT Push To Talk

PWM Pulse-width Modulation

RATDMA | Random Access Time Division Multiple Access

RF Radio Frequency
RI Reporting Interval
RR Report Rate

SART | Search and Rescue Transponder

SMA SubMiniature version A SMD Surface Mount Device SOLAS Safety Of Life At Sea

SOTDMA | Self Organized Time Division Multiple Access

SRAM Static Random Access Memory
TDMA Time Division Multiple Access

TFG Trabajo Fin de Grado

UART Universal Asynchronous Receiver - Transmitter

USB Universal Serial Bus

VCO Universal Time Coordinated Voltage Controlled Oscillator

VHF Very High Frequency
VTS Vessel Tracking Service

Parte I

Memoria

Capítulo 1

Introducción

1.1. Antecedentes

La preocupación por mejorar la seguridad en las travesías marítimas se remonta a 1912, cuando ocurrió la catástrofe naval del Titanic. Fue entonces cuando la seguridad en los océanos dejó de ser un tema trivial y los gobiernos de numerosas naciones decidieron crear un organismo que regulara ciertas normas que avalaran un incremento en la seguridad marítima. De esta manera, en 1914 se crea el Convenio SOLAS (Safety Of Life At Sea) con el objetivo de garantizar la seguridad en los océanos. Este convenio, desde sus orígenes, ha sufrido un alto número de enmiendas, en parte para incluir normas que se adapten a los nuevos cambios en lo que respecta al tráfico marítimo [1].

En 1948, la ONU crea la IMO (*International Maritime Organization*) con el objetivo de regular las normas adoptadas por el Convenio SOLAS a nivel global. Su objetivo es promover la seguridad en los océanos y garantizar la supervivencia humana en caso de catástrofe. Para ello, la IMO incorporó el MSC (*Maritime Safety Committee*) para llevar a cabo estas regulaciones, así como para encargarse de la incorporación de las enmiendas a las que se vea sometido el Convenio SOLAS.

Pero no fue hasta 1991 cuando se presenta el primer documento acerca de las especificaciones técnicas que caracterizarían al estándar AIS (Automatic Identification System), un estándar internacional y de libre acceso que actualmente es adoptado por un gran número de embarcaciones de todo tipo y que tiene como objetivo principal garantizar una mayor seguridad en las travesías marítimas. Este sistema es capaz de monitorizar la actividad naval en una misma zona de cobertura e identificar la posición de los buques, su velocidad o su rumbo, para así como evitar posibles colisiones a partir del conocimientos de estos datos por el resto de embarcaciones [2].

Aunque desde la presentación del estándar AIS se han modificado algunos aspectos técnicos del estándar, no fue hasta 2001 cuando comenzaron a realizarse las primeras pruebas de conformidad para el estándar AIS. Finalmente, en el año 2002, la IMO exige el uso obligatorio de AIS para aquellas embarcaciones con un tonelaje bruto superior a 500 toneladas, los buques de más de 300 toneladas que realicen travesías internacionales y todo buque de pasajeros (independientemente de su tonelaje) [3].

4 1.1. Antecedentes

Además, el estándar AIS se incorpora al sistema GMDSS (Global Maritime Distress and Safety System) [4], vigente desde 1999, que se trata de un conjunto de procedimientos, equipos y protocolos de comunicación diseñados con el objetivo de incrementar la seguridad en las travesías marítimas y facilitar la navegación y el rescate de las embarcaciones que se encuentren en situación de peligro. Un ejemplo de otro sistema incorporado a GMDSS con fines similares es COSPAS/SARSAT [5].

Entre las ventajas más relevantes que pueden citarse acerca del estándar AIS, son de especial interés las que se mencionarán a continuación:

- Fácil implementación, mantenimiento y actualización.
- Bajo costo operativo (incluso a veces inexistente).
- Estandarización internacional.
- Cobertura entre 20 MN y 100 MN (dependiendo de las alturas de las antenas y de los obstáculos de la transmisión).
- Robustez frente al estado del mar y las condiciones meteorológicas adversas.
- Auge de incorporación de este sistema en embarcaciones pesqueras y de recreo.

Además, el estándar AIS presenta cobertura global por satélite [5]. Esto permite que las zonas marítimas más alejadas de la costa, donde no hay una gran confluencia de tráfico marítimo y la cobertura es escasa o inexistente, estén provistas de suficiente cobertura para que los buques que integran AIS puedan realizar transmisiones de sus datos de travesía. Estos datos son recibidos por satélites AIS, como es el caso *Orbcomm* y *exactEarth*, que direccionan los mensajes hasta las estaciones base AIS que a su vez, transmiten los datos al cliente o a la autoridad con derecho para utilizarlos en diversos fines, como puede ser salvamento marítimo (Figura 1.1).



Figura 1.1: Cobertura global AIS por satélite

1. Introducción 5

Otra de las ventajas que presenta el sistema AIS es su robustez frente a los elementos topográficos. Es de especial mención esta ventaja, dado que los sistemas radar destinados para los mismos fines sí se ven afectados por la orografía circundante, especialmente en las zonas costeras donde los acantilados pueden ocultar ciertas embarcaciones que no se encuentren en visibilidad directa con el radar (Figura 1.2).

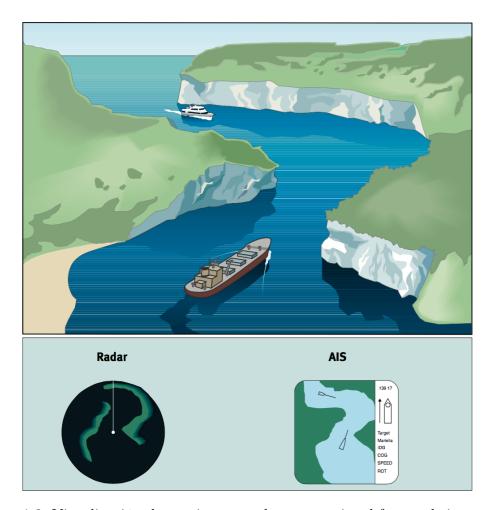


Figura 1.2: Visualización de un sistema radar convencional frente al sistema AIS

No obstante, AIS puede ser integrado en los sistemas radar en dispositivos conocidos como AIS-VTS (*Automatic Identification System - Vessel Tracking Service*), considerado como uno de los principales avances en la tecnología de comunicaciones marítima desde la invención del radar. En los dispositivos AIS-VTS, se combina la visualización AIS y radar, consiguiendo manejar las ventajas de cada uno de estos en un mismo sistema [6].

Sin embargo, hay buques que no están sometidos al Convenio SOLAS. Un ejemplo de ello son las embarcaciones militares y los buques que prestan un servicio no comercial a un Estado (además de las embarcaciones de recreo, donde el uso de AIS es opcional). Esto genera el problema de que al no estar todas las embarcaciones obligadas a identificarse mediante este sistema, podría ser empleado para operaciones de piratería, terrorismo o pesca ilegal.

6 1.1. Antecedentes

Otra de las peculiaridades que hace de AIS un sistema cada vez más atractivo, incluso para aquellas embarcaciones que no se ven sometidas al Convenio SOLAS, es que a diferencia de otros sistemas móviles, como es el caso de GSM, los fallos en las estaciones base de control AIS no implican la caída total del sistema, pues cada estación móvil autogestiona su envío y recepción de datos hacia otras estaciones. Esto se consigue a través del principal aspecto técnico de este estándar: el esquema de multiplexación temporal SOTDMA (Self-Organized Time Division Multiple Access).

SOTDMA es una técnica de multiplexación que incorpora una sincronización UTC (*Universal Time Coordinated*) que permite a las estaciones AIS poder advertir al resto de estaciones su intención de transmitir en un determinado slot de tiempo. En caso de que este slot esté ocupado por otra estación, nuestra estación transmisora hará uso del algoritmo SOTDMA hasta encontrar un slot libre al que acceder. Este proceso será repetido por el resto de estaciones, asegurando que en ningún momento dos estaciones puedan interferirse ocupando el mismo slot para transmitir sus datos [7].

A través de los slots, cada embarcación transmitirá su información tanto estática (identificación) como dinámica (posición, rumbo, velocidad) para que el resto de embarcaciones puedan conocerla y así evitar conflictos de trayectoria en una misma zona de tránsito marítimo. De esta forma, todas las estaciones conocerán los datos del resto de estaciones (Figura 1.3).

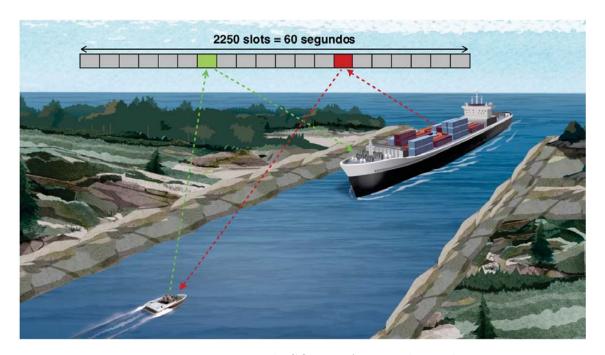


Figura 1.3: Funcionamiento de SOTDMA entre dos embarcaciones

Desde abril de 2013, el IDeTIC (*Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones*) cuenta con una estación receptora de AIS que fue cedida por la empresa *Marine Traffic* para realizar diversos estudios sobre operaciones portuarias y marítimas (Figura 1.4). Presenta un rango de cobertura que varía en función de las condiciones meteorológicas, pero ronda entre las 82.73 y 685.52 MN, y es capaz de recibir datos de cerca de 100 embarcaciones [8].

1. Introducción 7

Rango de alimentación eléctr	ica: 9 - 30 Voltios DC
Consumo de energía:	180mA @ 12VDC
Velocidad de transmisión:	38400 Baud (38.4Kb)
Formato:	ITU/ NMEA 0183
Mensaje de salida:	VDM
Frecuencia de receptor:	Canal A 161.975 MHz Canal B 162.025 MHz
Separación entre canales:	25KHz
Sensibilidad:	-112dBm
Demodulación:	GMSK
Velocidad de datos:	9600
Impedancia de antena:	50 ohms
Dimensiones físicas:	L 165 x W 105 x H 46mm Weight: 600g
Montaje:	4 agujeros de montaje en placa
Conectores:	Antena BNC
Puerto Ethernet:	RJ-45
Puerto USB:	Tipo B
Alimentación:	Clavija fonográfica de 2.1mm
Diseñado para cumplir:	IEC 61993-2 Aprobación CE a EN 60945: EN 6100-6-1/2 FCC parte 15



Figura 1.4: Receptor AIS cedido por Marine Traffic al IDeTIC

1.2. Objetivos

El principal objetivo de este Trabajo de Fin de Grado es el diseño y la implementación física de un prototipo hardware modular destinado a la realización de pruebas sobre el estándar AIS. Para este banco de pruebas, se atenderá al diseño de un transmisor y un receptor.

En el caso del transmisor, desde una interfaz conectada al puerto serie de un microcontrolador, es posible enviar tramas de datos. El microcontrolador dará formato a estas tramas y serán moduladas en banda base a través de un módem GMSK. Seguidamente, los datos modulados en banda base volverán a ser modulados en paso banda con un modulador FM, que a su salida entregará la señal generada a través de una antena de VHF marítimo (Figura 1.5).

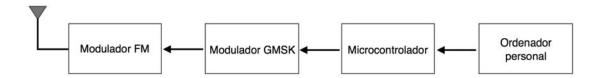


Figura 1.5: Diagrama de bloques del banco de pruebas transmisor AIS

Por otra parte, el receptor recibirá los datos modulados en FM por una antena de VHF marítimo. Estos datos serán demodulados en FM y posteriormente, en GMSK, entregando a la entrada del microcontrolador los datos enviados desde el transmisor.

Por último, una interfaz conectada al puerto serie del microcontrolador permite que los datos recibidos queden representados en una pantalla de ordenador (Figura 1.6).

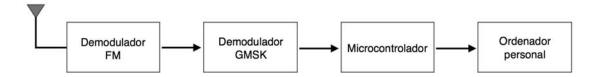


Figura 1.6: Diagrama de bloques del banco de pruebas receptor AIS

Para alcanzar el objetivo principal de este TFG, los pasos que se siguen son los siguientes:

- Estudio en profundidad del estándar AIS.
- Diseño modular del prototipo para el banco de pruebas.
- Implementación e integración física de los módulos.
- Programación del firmware de control para el banco de pruebas.

1.3. Estructura de la memoria

En el Capítulo 1 de esta memoria se describen algunas consideraciones generales sobre el estándar AIS y el marco en el que se sitúa este trabajo, aportando una breve introducción que permita dar paso Capítulo 2, donde se realiza un estudio minucioso de este estándar. En este segundo capítulo, se estudia con mayor detalle el estándar con el objetivo de reconocer los requisitos que deben cumplir los dispositivos hardware que lo integran.

En el Capítulo 3 se se detallan las características técnicas de los módulos escogidos para el desarrollo del prototipo hardware. Este tercer capítulo permite abordar las consideraciones de diseño relativas al conexionado y funcionamiento del prototipo, que se describe en el Capítulo 4.

El Capítulo 5 recoge todas las pruebas realizadas para verificar el correcto funcionamiento del prototipo hardware y su viabilidad. Las conclusiones alcanzadas tras la realización de estas pruebas y las propuestas de mejoras para este trabajo se recogen en el Capítulo 6.

Seguidamente, se incluye la bibliografía que ha sido referenciada en la redacción de este trabajo y el presupuesto de este TFG. Se incluye como anexos alguna documentación relativa al hardware que se emplea en el prototipo hardware, destacando la inclusión de los resultados publicados en base a este trabajo.

Capítulo 2

Descripción del estándar AIS

2.1. Introducción

En este capítulo se realiza un estudio previo del estándar AIS, que está especificado en la recomendación ITU-R M.1371-5 [9]. En primer lugar, se habla de la regulación del estándar AIS y la obligatoriedad de incorporación del mismo en ciertos tipos de embarcaciones. Por último, se describe la arquitectura de red de este estándar, entrando en mayor detalle con los niveles Física y de Enlace, pues tienen una repercusión directa con este trabajo.

2.2. Regulación y regulación del estándar AIS

Desde 2003, la IMO exige la incorporación obligatoria del estándar AIS en ciertos tipos de embarcaciones en función de su tonelaje bruto y el servicio al que se destinan.

Los buques con un tonelaje bruto superior a 500 toneladas, superior a 300 toneladas y que realicen travesías internacionales y todo buque de pasajeros, independientemente de su tonelaje bruto, están obligados a incorporar transpondedores AIS en sus equipos de abordo.

Aquellos buques que prestan un servicio no comercial para un Estado, los buques militares y las pequeñas embarcaciones de recreo o de pesca, no están obligados a integrar este sistema. En cualquier caso, la mayoría de los buques militares integran AIS como sistema de respaldo para otros equipos de a bordo, pero no están obligados a identificarse en el sistema.

En este conjunto de embarcaciones exentas del cumplimiento del estándar AIS se encuentran también las embarcaciones de recreo (dimensiones de eslora máxima de 24 metros), los barcos pesqueros de pequeñas dimensiones y cualquier otra embarcación con dimensiones y tonelaje inferiores a las que exige la IMO en relación a la obligatoriedad de integración del estándar AIS.

La obligatoriedad de integración de los equipos AIS para ciertos tipos de embarcaciones hace que este sistema diferencie entre dos tipos de transpondedores, llamados AIS clase A y AIS clase B, que difieren entre sí tanto en prestaciones técnicas como en sus esquemas de acceso al medio.

- AIS clase A: se trata del sistema con mayores prestaciones dentro de la gama de transpondedores AIS. Su uso es obligatorio en buques con un tonelaje bruto superior a 500 toneladas, superior a 300 toneladas que realicen travesías internacionales y cualquier buque de transporte de pasajeros, independientemente de su tonelaje bruto.
- AIS clase B: es un sistema con menores prestaciones que los transpondedores AIS clase A, y su calidad depende principalmente de las técnicas de acceso al medio implementadas. Se destina a aquellos buques para los que el uso de AIS no sea obligatorio, como pueden ser las embarcaciones de recreo o los buques pesquero de pequeñas dimensiones.

Por otro lado, el estándar AIS puede ser integrado en diversos dispositivos dependiendo de los fines a los que se destina. En función de su uso, se establece la siguiente clasificación:

- AIS BS (*Base Station*): estaciones de control instaladas en la costa, que integran AIS para para proveer de cobertura a los barcos que se encuentran próximos a los puertos y realizan operaciones de atraque.
- AtoN (Aid to Navigation): sistema integrado en boyas de salvamento marítimo, que prestan cobertura en las zonas más hostiles del entorno marino.
- AIS SART (Search and Rescue Transponder): dispositivos AIS integrados en numerosos vehículos terrestres, marítimos y aéreos, con el objetivo de determinar la posición de los buques de salvamento y así poder servir como red de apoyo en operaciones de rescate.

2.3. Arquitectura de red

El estándar AIS presenta una arquitectura de red basada en el modelo de referencia OSI (*Open System Interconnection*), lo que significa que está formada por 7 capas o niveles relacionados entre sí (Figura 2.1). Sin embargo, el estándar AIS solamente describe el funcionamiento de los tres primeros niveles de esta arquitectura:

• Nivel Físico: en él se realizan todas las conexiones entre los sistemas que pretenden comunicarse en formato digital a través de un mismo estándar, de manera que se puede interpretar como la interfaz entre el sistema de comunicación y el canal.

- Nivel de Enlace: se relaciona con el manejo de las tramas de bits, así como con la detección y corrección de errores que se realiza previamente al envío de paquetes por la red. Por ello, es la capa que asegura la fiabilidad del flujo binario. A su vez, se compone de tres subniveles:
 - MAC (*Medium Access Control*): asegura la correcta transferencia de datos desde el medio físico hasta el Nivel de Enlace.
 - **DLS** (*Data Link Service*): realiza la detección y el control de errores tras realizarse la transferencia de datos sobre el Nivel de Enlace.
 - LME (*Link Management Entity*): controla las operaciones que realiza las subcapas MAC y DLS.
- Capa de Red: en esta capa se realiza el envío y recepción de paquetes de datos. Realiza las tareas de descongestión de paquetes en red, en caso de existir flujos masivos de datos que pusiesen en riesgo el intercambio de información entre sistemas bajo un mismo estándar.

En lo que respecta a las capas de Transporte, Sesión, Presentación y Aplicación, la única exigencia del estándar es que se implementen de forma coherente a los paquetes de datos que son intercambiados desde el Nivel de Red.

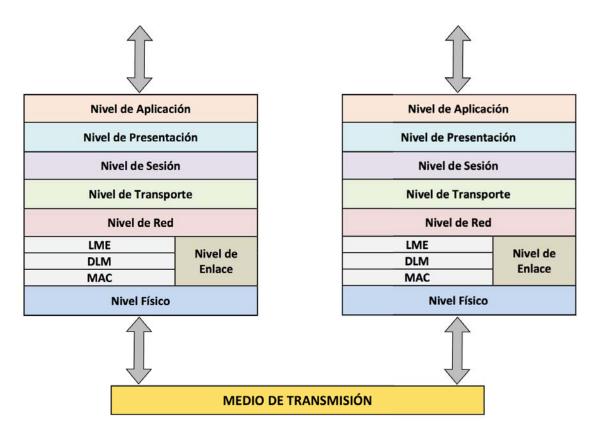


Figura 2.1: Esquema general de la arquitectura de red del estándar AIS

2.3.1. Nivel Físico

El Nivel Físico hace referencia a la interfaz que existe entre el propio sistema de comunicación y el canal. En esta capa se transmiten y reciben las secuencias de bits de los datos del protocolo implementado, y se caracteriza por todos aquellos parámetros ligados al hardware del sistema.

2.3.1.1. Banda de trabajo y parámetros radio

El servicio AIS se establece en la banda de VHF marítimo (156.00 MHz - 162.025 MHz), tomando dentro de esta banda dos posibles canales: **AIS-1** (161.975 MHz) y **AIS-2** (162.025 MHz), también llamados canales 87 y 88, respectivamente (Figura 2.2).

La canalización de la banda de trabajo en AIS es 25 KHz, establecido por la recomendación ITU-R M.1084-5 [11].

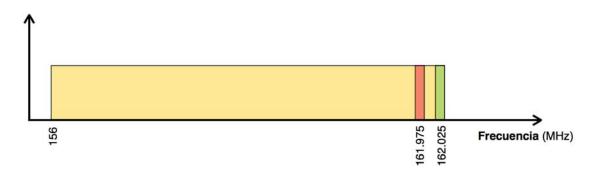


Figura 2.2: Descripción de la banda de trabajo del servicio AIS

Los parámetros radio definen aquellas características que los equipos deben satisfacer en la banda de trabajo donde operan. Para el estándar AIS, estas características quedan recogidas en la Tabla 2.1.

Parámetro radio	Valor
Banda de trabajo del servicio	156.025 - 162.025 MHz
Frecuencia de trabajo AIS-1	161.975 MHz
Frecuencia de trabajo AIS-2	$162.025~\mathrm{MHz}$
Canalización del servicio	$25~\mathrm{KHz}$
Velocidad de transmisión	9600 baudios
Potencia transmitida en AIS clase A	12.5 W
Potencia transmitida en AIS clase B	1 W
Factor BT en transmisión (aproximado)	0.4
Factor BT en recepción (aproximado)	0.5
Índice de modulación	0.5

Tabla 2.1: Parámetros radio generales del estándar AIS

Además, el estándar AIS también especifica las características propias de los equipos de transmisión (Tabla 2.2) y de recepción (Tabla 2.3).

Parámetro radio	Valor
Desviación de frecuencia de la portadora	$\pm 500~\mathrm{Hz}$
Desviación de potencia de la portadora	$\pm 1.5 \text{ dB}$
Respuesta de rechazo para emisiones espúreas	-36 dBm (9 KHz - 1 GHz)
	-30 dBm (1 GHz - 4 GHz)
Atenuación de intermodulación	≥ 40 dB

Tabla 2.2: Características especiales para transmisores AIS

Parámetro radio	Valor
Sensibilidad	-107 dB
Tasa de error por saturación	1% para -77 dBm
	1% para -7 dBm
Selectividad para canales adyacentes	70 dB
Selectividad para el propio canal	10 dB
Respuesta de rechazo para emisiones espúreas	-57 dBm (9 KHz - 1 GHz)
	-47 dBm (1 GHz - 4 GHz)
Atenuación sobre emisiones espúreas	≥ 70 dB
Atenuación de intermodulación	≥ 74 dB

Tabla 2.3: Características especiales para receptores AIS

Por último, el estándar AIS establece sus propios esquemas de codificación y modulación. Emplea un esquema de codificación NRZI sobre los datos digitales transmitidos. Estos datos son modulados en banda base con GMSK, que se combina con una modulación paso banda FM (Figura 2.3).

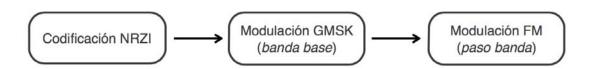


Figura 2.3: Esquema de codificación y de modulación en el estándar AIS

En los siguientes puntos se trata el estudio del esquema de codificación y de modulación del estándar AIS en mayor profundidad.

2.3.1.2. Esquemas de codificación

El estándar AIS incopora un tipo concreto de *codificación diferencial*, que son un grupo de codificaciones ampliamente utilizadas en comunicaciones por radio y satélites,

que se emplean para la recepción precisa de los bits cuando la señal se ve sometida a algún tipo de modulación digital.

Estas técnicas permiten que los datos transmitidos no dependan del bit actual únicamente, sino también del bit anterior que ha sido transmitido. De esta manera, hay un mayor control en el flujo de los datos transmitidos, aunque cabe destacar que son muy sensibles a los llamados errores de multiplicación por los que una detección incorrecta de un símbolo conlleva a la generación de dos símbolos incorrectos, duplicando de esta forma la tasa de error de bit (BER).

La codificación diferencial empleada por el estándar AIS se denomina NRZI (Non Return to Zero inverted). Esta codificación mantiene constante el nivel de tensión durante la duración de un bit, de forma que los datos están codificados mediante la presencia o ausencia de una transición de la señal al principio del intervalo del tiempo de bit [12].

Cuando se tiene un 1 lógico, se codifica mediante la transición (sea desde nivel bajo hasta nivel alto, o viceversa) al principio del intervalo de bit. Por el contrario, un θ lógico representa la ausencia de una transición, de manera que cuando el valor del bit es 1, la señal cambia de estado después de que el ciclo de reloj lo indique, y si es θ , la señal no cambia de estado (Figura 2.4).

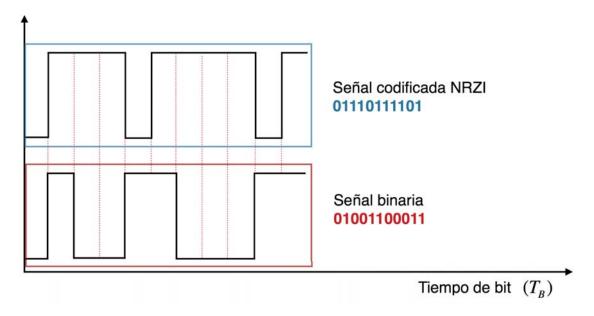


Figura 2.4: Ejemplo de codificación NRZI sobre una secuencia binaria

Esta codificación presenta importantes ventajas que justifican la elección de la misma en el estándar AIS, como son las siguientes:

- Precisión en la detección o ausencia de señal.
- Inmunidad frente a ruido e interferencias en el canal.
- Facilidad para detectar la transición entre símbolos.

En cuanto a sus desventajas, la principal es que cuando existe una corriente continua durante una secuencia de ceros, se perturba el sincronismo entre transmisor y receptor. Por este motivo, las tramas AIS deben presentar un formato sobre el que, al realizar la codificación NRZI, se eviten las largas cadenas de θ lógicos que afecten al sincronismo entre los sistemas transmisor y receptor.

2.3.1.3. Esquema de modulación

El estándar AIS emplea la combinación de una modulación paso banda FM (Frequency Modulation) con una modulación banda base GMSK (Gaussian Minimum Shift Keying), conocida como modulación FM-GMSK.

La modulación GMSK es un tipo particular de modulación MSK en banda base, que emplea un filtrado previo sobre los datos para reducir el ancho de banda de la señal. Más concretamente, es capaz de eliminar aquellas componentes relativas a los lóbulos secundarios del espectro [13].

En la Figura 2.5 se muestra una comparativa en el dominio de la frecuencia entre una modulación FSK y una modulación GMSK. Se observa, pues, la mejora que sufre el espectro en lo que respecta a la reducción del ancho de banda en banda base al filtrar los lóbulos secundarios.

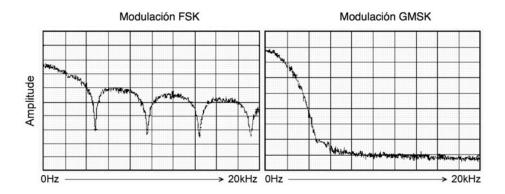


Figura 2.5: Comparativa entre los espectros FSK y GMSK

El tipo de filtro que se emplea para ello es un filtro gaussiano [14]. Se trata de un filtro paso bajo con una respuesta al impulso capaz de transformar una delta de Dirac a su entrada en una forma de onda que, idealmente, se corresponde con una curva de Gauss (Figura 2.6).



Figura 2.6: Respuesta al impulso de un filtro gaussiano ideal

Matemáticamente, la respuesta al impulso g(t) del filtro gaussiano viene dada por de la ecuación (2.1):

$$g(t) = \frac{1}{2} \cdot \left[Q\left(-\frac{2\pi B}{\sqrt{2 \cdot \ln(2)}} \cdot t \right) - Q\left(-\frac{2\pi B}{\sqrt{2 \cdot \ln(2)}} \cdot (t - T_B) \right) \right]$$
(2.1)

donde B representa el ancho de banda a -3 dB de la señal filtrada a través del filtro gaussiano, T_B el periodo de bit y Q es una función de distribución de Gauss dada por la expresión (2.2):

$$Q(t) = \int_{t} \frac{1}{\sqrt{2}} \cdot e^{-\frac{x^2}{2}} dx \tag{2.2}$$

Del espectro de la señal modulada en GMSK se pueden extraer algunos parámetros fundamentales para caracterizar esta modulación [15]. Se encuentra así el factor BT, que marca una relación entre la frecuencia de corte del filtro gaussiano a $-3 \ dB$ y la tasa binaria de la señal (2.3).

$$BT = \frac{f_{-3 \ dB}}{R_B} \tag{2.3}$$

Hay dos valores del factor BT con los que normalmente trabajan los sistemas de comunicaciones digitales. Estos son BT = 0.3 y BT = 0.5. En la Figura 2.9 puede observarse la diferencia entre estos dos valores al definir un eje de abscisas para el periodo de bit (T_B) y la respuesta al impulso normalizada en el eje de ordenadas.

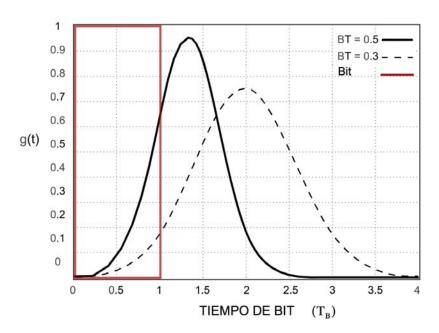


Figura 2.7: Respuesta al impulso de un filtro gaussiano con BT=0.3 y BT=0.5

En la Figura 2.7 se observa cómo al emplear un factor BT=0.5, la respuesta al impulso del filtro gaussiano se extiende 2 periodos de bits. En cambio, con un factor BT=0.3, la respuesta se extiende hasta cerca de 3 periodos de bits. En el dominio del tiempo, la diferencia entre los dos factores BT radica en la interferencia entre símbolos (IES).

Se evidencia que en el caso de un factor BT = 0.3, la IES es mayor que en el caso de BT = 0.5. No obstante, para un factor BT = 0.5, la tasa binaria máxima aceptada por el filtro para una misma frecuencia de corte es menor, dado que el factor BT y la tasa binaria son variables inversamente proporcionales.

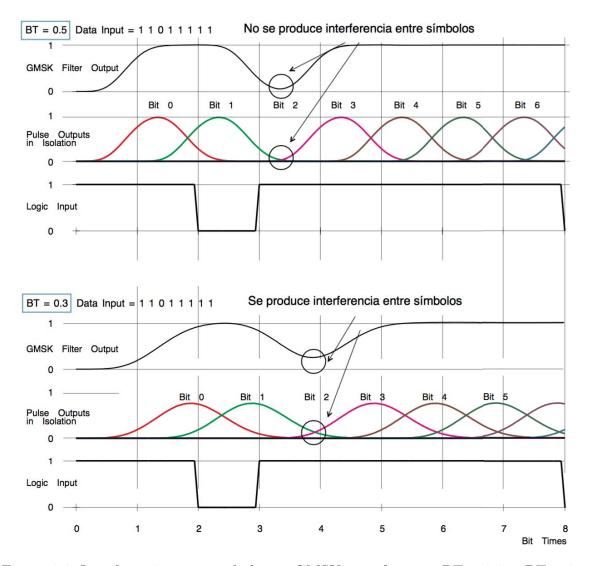


Figura 2.8: Interferencia entre símbolos en GMSK para factores BT = 0.3 y BT = 0.5

En la Figura 2.8 se observa el incremento de la IES para un factor BT = 0.3. Sin embargo, para BT = 0.5 también existe IES, pero en comparación al factor BT = 0.3 y con respecto al resto de modulaciones digitales, como es MSK, se considera que la IES es aceptable. Por tanto, hay un compromiso entre eficiencia espectral e interferencia entre símbolos durante la elección del factor BT.

El esquema de modulación del estándar AIS, a nivel de sistemas, se puede describir con la inclusión de un modulador o demodulador GMSK en banda base conectado a un modulador o demodulador FM, respectivamente. De esta forma, se engloban las virtudes de la modulación GMSK con la robustez frente al canal radio de la modulación FM sobre un mismo sistema.

El diagrama de bloques genérico de un modulador y un demodulador FM-GMSK se representa en la Figura 2.9.

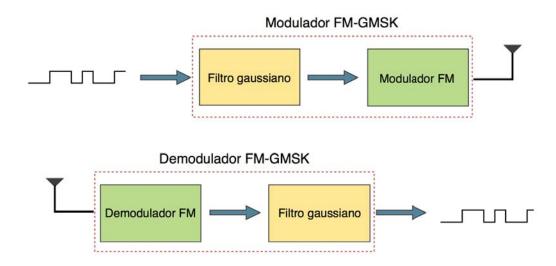


Figura 2.9: Modulador y demodulador FM-GMSK

2.3.2. Nivel de Enlace

El Nivel de Enlace de cualquier arquitectura de red se encarga de que realizar la transferencia fiable de los datos hasta el Nivel de Red. En esta capa se genera la detección y corrección de errores en las tramas.

Asimismo, el Nivel de Enlace para el estándar AIS se divide en tres subniveles que se denominan, en sentido ascendente: MAC (*Medium Access Control*), DLS (*Data Link Service*) y LME (*Link Management Entity*). Seguidamente, se describe la función de cada uno de estos subniveles.

2.3.2.1. Subcapa MAC

La subcapa MAC (*Medium Access Control*) se encarga de garantizar que la comunicación del Nivel Físico por el medio de transmisión se realice de una forma fiable. Para ello, el estándar AIS incluye las referencias de tiempo común entre las estaciones AIS que integran la red.

Todas las estaciones AIS deben estar sincronizadas bajo una referencia de tiempo común denominada UTC (*Universal Time Coordinated*), que viene dada por la red de satélites GPS (*Global Position System*).

Hay diferentes formas de acceder al sincronismo UTC por partes de las estaciones, diferenciando entre dos formas elementales: sincronismo UTC directo y sincronismo UTC indirecto (Figura 2.10).

- Sincronismo UTC directo: el sincronismo UTC deriva de la propia red GPS. El transpondedor AIS debe incluir un receptor GPS que tome esta referencia temporal para sincronizarse a un tiempo común entre estaciones.
- Sincronismo UTC indirecta: el sincronismo UTC es heredado de aquellas estaciones que se sincronizan de forma directa con la red GPS.

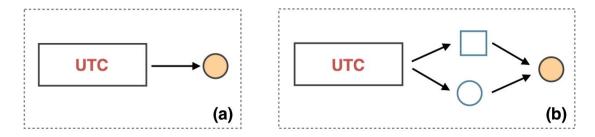


Figura 2.10: (a) Sincronismo UTC directo, (b) Sincronismo UTC indirecto

El sincronismo entre las estaciones AIS permite la actualización de las tasas de informes de posición. Esto quiere decir que en función del estado de la embarcación, más concretamente su velocidad de desplazamiento, los informes de posición se envían más rápidamente dependiendo de que el barco también aumente su velocidad.

2.3.2.2. Subcapa DLS

La subcapa DLS (*Data Link Service*) se encarga de dar formato a las tramas AIS que son enviadas o recibidas por el sistema. Cada trama AIS está formada por 32 bytes (Figura 2.11), organizados de la siguiente manera:

- **Preámbulo**: es una cadena de 24 bits configurada como una secuencia alternada de 1 y 0 lógicos. Tiene la forma binaria 01010101010101010101010101, que en hexadecimal se representa como tres bytes con la forma 0x55.
- Cabecera de inicio: son 8 bits dispuestos como un flag HDLC (*High-Level Data Link Control*) que alertan del inicio de los datos transmitidos. Tiene la forma binaria 01111110, que en hexadecimal se representa como un byte con la forma 0x7E.
- Datos: son 168 bits de datos en los que se incluye la información, tanto estática como dinámica, que conforma cada paquete AIS que transmite una estación. Sin embargo, el contenido de estos datos no queda definido en la subcapa DLS, sino desde capas superiores como la de red. En caso de que un paquete tenga un tamaño superior a 168 bits, se emplean técnicas de segmentación de paquetes.

- CRC: se trata de un polinomio generador de 16 bits que emplea un código de redundancia cíclica (CRC) definido en la norma *ISO/CEI 3309:1993* para realizar el control de errores que puede aparecer durante una transmisión.
- Cabecera de fin: tiene la misma forma que la cabecera de inicio (0x7E), pero a diferencia de esta, indica el final de los datos almacenados en un paquete.
- Buffer temporal: es una secuencia adicional de 24 bits que complementa a la trama para las siguientes tareas:
 - Relleno de bits (4 bits).
 - Retardo por distancia (12 bits).
 - Retardo del repetidor (4 bits).
 - Efectos por *jitter* (4 bits).



Figura 2.11: Formato de una trama AIS

En lo que respecta al campo de datos de las tramas AIS, la codificación de los mensajes se describe en la recomendación *ITU-R M.1371-5 (Anexo VIII)*, donde se recogen los diferentes tipos de mensajes que pueden enviar las estaciones AIS y el significado de cada uno de sus campos.

2.3.2.3. Subcapa LME

La subcapa LME (*Link Management Entity*) implementa el esquema de acceso al medio TDMA (*Time Division Multiple Access*). De esta manera, cada estación AIS tiene un intervalo de tiempo asignado (slot) para acceder al medio compartido con el resto de estaciones y realizar la transferencia de datos. Una vez agotado este slot, la estación desocupa el slot y permite a otra estación acceder al mismo.

Generalmente, los esquemas TDMA exigen que las estaciones estén sincronizadas. En el caso de las estaciones AIS, el sincronismo viene dado por el tiempo de referencia UTC, común a todas las estaciones AIS integradas en una misma región de cobertura.

En el estándar AIS, cada slot tiene una duración de $26.67 \ ms$. Si cada trama AIS tiene una duración temporal de $60 \ s$, el número de slots totales en una trama AIS es:

$$N = 60 \cdot \Delta t_{slot} = 60 \cdot 26.67 \simeq 2250 \ slots$$
 (2.4)

En la Figura 2.12 se observa el esquema TDMA para los dos canales AIS disponibles, verificando una duración por cada slot de 26.67 ms.

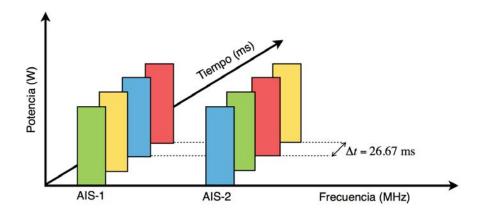


Figura 2.12: Esquema TDMA en AIS para las frecuencia AIS-1 y AIS-2

El estándar AIS incorpora diferentes esquemas de acceso al medio TDMA, pero los principales son el esquema SOTDMA, integrado en las estaciones AIS clase A, y el esquema CSTDMA, integrado en las estaciones AIS clase B [16].

En la Figura 2.13 se contemplan los usos de los diferentes esquemas TDMA en AIS en relación a las posibles aplicaciones de AIS que se describen en el punto 2.2.

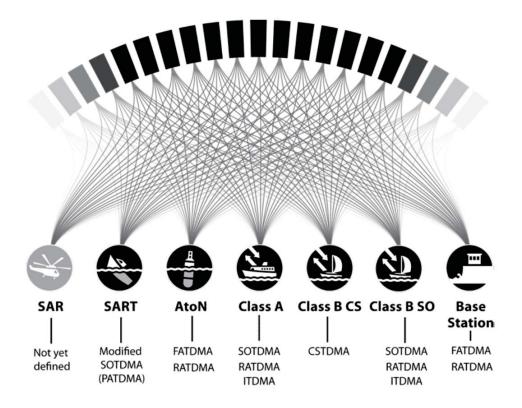


Figura 2.13: Esquemas TDMA en AIS en función de su aplicación

A continuación, se explica el funcionamiento de cada uno de estos esquemas de acceso al medio, dando un mayor enfoque al esquema SOTDMA.

SOTDMA

El esquema SOTDMA (Self-Organized Time Division Multiple Access) es el principal esquema de multiplexación que caracteriza al estándar AIS, y se integra en los transpondedores AIS clase A.

Este esquema contempla los siguientes modos de operación:

- Modo autónomo y continuo: las estaciones AIS ejecutan continuamente este modo para poder realizar la transmisión y recepción de mensajes AIS en todo momento.
- Modo asignado: es programado mediante una autoridad competente que configura los datos en función del área donde se realiza la travesía.
- Modo interrogación: se ejecuta únicamente cuando una estación AIS solicita una respuesta por parte de otra estación AIS sobre un determinado evento durante la travesía.

El modo autónomo y continuo comienza con la fase de inicialización, que se establece en el momento en que la estación es activada. El primer paso consiste en monitorizar los canales VHF marítimo del sistema AIS durante 60 segundos. De esta manera, cada estación puede identificar al resto de estaciones activas en su misma región de cobertura, además del estado de los slots a los que pretende acceder (Figura 2.14).

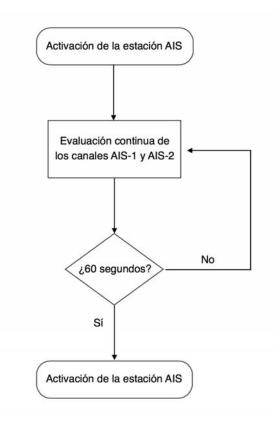


Figura 2.14: Algoritmo de la fase de inicialización en SOTDMA

Una vez la estación monitoriza la actividad de las estaciones en los canales AIS-1 y AIS-2, debe seleccionar un primer slot para presentarse en el sistema. Esta es la *fase de entrada en red*, que consigue que el resto de estaciones conozcan la incorporación de esta nueva estación AIS, y así gestionar el acceso a los slots, teniendo en cuenta que un elemento nuevo competirá con ellas para dicho acceso.

Para ello, lo primero que hace la estación es seleccionar para los canales AIS-1 y AIS-2 los NSS (Nominal Start Slot), llamados NSS_A y NSS_B . Considerando un cierto NI (Nominal Increment), se define la siguiente relación:

$$NSS_B = NSS_A + NI \tag{2.5}$$

donde el incremento nominal NI se define como:

$$NI = \frac{2250}{R_r} \tag{2.6}$$

siendo R_r (Report Rate) el número de informes de posición que el sistema genera cada minuto, de forma que si RI (Reporting Interval) es el intervalo de duración de cada informa, entonces:

$$R_r = \frac{60}{RI} \tag{2.7}$$

En la Figura 2.15 se observa la selección un intervalo SI con los posibles slots destinados a la transmisión y del NTS, que es el slot que realiza la transmisión.

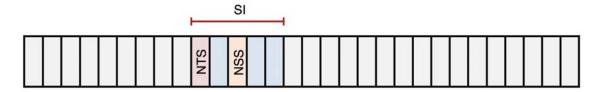


Figura 2.15: Selección de NTS en un intervalo SI

Cabe destacar que el comienzo de la selección del slot se establece a partir de un NTS (*Nominal Transmission Slot*), que tiene especial relevancia en la tercera etapa de funcionamiento del esquema de acceso SOTDMA (Figura 2.16).

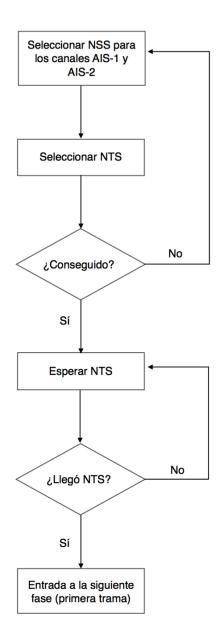


Figura 2.16: Algoritmo de la fase de entrada en red en SOTDMA

La tercera etapa de funcionamiento de SOTDMA es la fase de la primera trama. En esta etapa, cada estación AIS localiza de forma continua los distintos NTS y transmite sobre estos sus informes de posición. Al alcanzar el primer NTS, se seleccionan unos nuevos NS (Nominal Slot) y NTS para las próximas transmisiones que llevará a cabo.

Dicho de otra manera, cuando se selecciona un slot sobre el que se va a realizar una transmisión, se selecciona otro para la próxima transmisión, y así sucesivamente hasta cumplir 60 segundos desde que se alcanza el primer NSS de la transmisión.

Para llevar a cabo este proceso, se almacena en un registro el número de transmisiones efectuadas hasta el momento en el canal, con un tamaño que va desde 0 a $R_r - 1$. De esta manera, la selección de NS debe efectuarse conforme a la siguiente expresión:

$$NS = NSS + (n \cdot NI) \qquad \forall \ n \in [0, R_r - 1] \tag{2.8}$$

En la Figura 2.17 se observa la selección del siguiente NS en función del NSS con el que se realiza la transmisión del informe de posición en el slot actual.

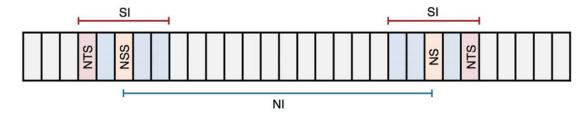


Figura 2.17: Selección de NS definido en un NI

Por último, se establece la fase de operación continua. En esta fase, la estación realiza de forma continua las transmisiones sobre los slots asignados. Un contador (slot time-out) marca los tiempos en que es ocupado el slot de transmisión, y cuando este contador se pone a 0, se selecciona un nuevo NTS sobre el intervalo SI de igual forma a como se establece en la fase de la primera trama (Figura 2.18).

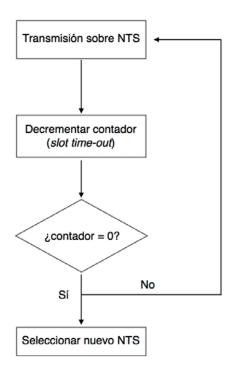


Figura 2.18: Algoritmo de la fase de operación continua en SOTDMA

Esta fase se mantiene hasta que el equipo se apaga, pasa al modo asignado o al modo interrogación o bien si cambia la tasa de actualización de los informes.

RATDMA

El esquema RADTMA (Random Access Time Division Multiple Access) se emplea para anunciar previamente la transmisión aperiódica de slots.

Las estaciones transmisoras que emplean RATDMA usan un *mapa de slots* interno que selecciona aleatoriamente los slots que no están siendo ocupados por otras estaciones. No se anuncia el uso de ningún slot usado para transmisiones aperiódicas (actuales o futuras).

No es aconsejable usar RATDMA en transmisiones periódicas para la asignación de slots. Esto se debe a que RATDMA no permite el reconocimiento de estos slots por otros dispositivos que integren AIS, ya que provienen del mapa de slots interno del dispositivo. De hecho, si se usara para transmisiones periódicas, implicaría serios problemas de colisión de datos que comprometerían el correcto funcionamiento del sistema.

Por ello, RATDMA se suele emplear como técnica de entrada en red por las estaciones AIS clase A. Esto ocurre cuando un dispositivo que integra AIS clase A se conecta a la red y pretende anunciarse por primera vez en la misma. La selección aleatoria del slot inicial que realiza RATDMA mejora considerablemente esta etapa para otros esquemas de mayor calidad, como SOTDMA.

ITDMA

El esquema ITDMA (*Incremental Time Division Multiple Access*) se emplea cuando una estación necesita asignar un slot, pero esta asignación no ha sido anunciada previamente.

A diferencia de RATDMA, con el esquema ITDMA sí se anuncia al resto de estaciones qué slots están siendo ocupados en todo momento. De esta manera, permite que una estación transmisora pueda anunciar un cambio temporal en el intervalo de información que ocupa un mensaje periódico. Por tanto, es posible completar envíos de mensajes aperiódicos (por ejemplo, un mensaje relacionado con la seguridad de la embarcación).

Este esquema se suele emplear, al igual que RATDMA, para la fase de entrada en la red. Sin embargo, necesita apoyarse del esquema SOTDMA, por lo que no se usa como un esquema de acceso independiente como ocurre con RATDMA.

FATDMA

El esquema FATDMA (*Fixed Access Time Division Multiple Access*) se gestiona manualmente para todos los dispositivos AIS que están configurados para utilizar los slots a los que se accede para el resto de transmisiones.

Las estaciones configuradas con FATDMA transmiten un mensaje DLM (Data Link Management), que informan al resto de estaciones que integran este esquema sobre la

asignación de slots, y permiten bloquear estos slots para que no sean ocupados por otras estaciones cercanas.

Este esquema TDMA se usa exclusivamente en las estaciones base AIS y en el sistema AtoN. De hecho, se intenta que su integración no abunde entre dispositivos AIS, ya que puede influir en la dinamización de la asignación de slots en la red AIS.

CSTDMA

El esquema CSTDMA (Carrier Sense Time Division Multiple Access) se emplea como técnica de acceso al medio para transpondedores AIS clase B. Estos transpondedores deben ser compatibles con SOTDMA y garantizar que, en todo momento, SOTDMA tenga prioridad sobre cualquier otro esquema.

Este esquema TDMA no necesita sincronismo UTC directo para operar correctamente, por lo que emplea sincronismo UTC indirecto. De esta manera, el sincronismo entre las estaciones que emplean este esquema deriva de otras estaciones con sincronismo UTC directo (por GPS).

En cuanto a su funcionamiento, las estaciones con CSTDMA monitorizan continuamente el nivel de ruido de fondo en los canales AIS y toman como referencia este nivel de ruido. De esta forma, se logra medir el nivel de la señal al inicio de cada slot. Usando una asignación aleatoria de slots, se mide el nivel de señal al inicio del slot, seleccionado aleatoriamente, y toma la siguiente decisión:

- Si el nivel de señal es superior al nivel de ruido de referencia, asume que ese slot está en uso y, por tanto, no accede al mismo.
- Si el nivel de señal es inferior al nivel de ruido de referencia, asume que ese slot está libre y, por tanto, accede al mismo.

La principal desventaja de este sistema, frente a la ventaja económica indiscutible que supone la integración de este esquema de acceso sobre transpondedores AIS, es que requiere la existencia de estaciones AIS que empleen otros esquemas de mayor calidad, como SOTDMA, para poder sincronizarse con el resto de estaciones.

Además, una mala medida del nivel de referencia de ruido en el canal puede ocasionar que las estaciones que integran CSTDMA interfieran sobre los slots en uso y provocar la colisión de paquetes. Es por ello por lo que normalmente se emplea en equipos receptores.

PATDMA

El esquema PATDMA (*Pre-Announced Time Division Multiple Access*), también conocido como *SOTDMA modificado*, es un esquema de acceso basado en SOTDMA que se utiliza para anunciar futuras transmisiones. Está destinado únicamente a los dispositivos de transmisión.

Las estaciones PATDMA seleccionan de forma aleatoria el slot que se utiliza para transmitir. La primera transmisión anuncia la intención de utilizar este slot durante 8 minutos para la siguiente transmisión, y una vez transcurre este primer periodo, se selecciona aleatoriamente el siguiente slot. Por ello, las transmisiones se realizan en ráfagas de 8 mensajes (un mensaje por minuto), garantizando de esta manera que la transmisión tenga éxito cuando se establece la comunicación cerca de la superficie del mar, pues podría bloquearse por el oleaje u otros fenómenos adversos.

El hecho de que PATDMA use un sistema de acceso aleatorio a los slots implica que el esquema sea más sensible a las colisiones de paquetes. Por ello, suele aplicarse casi exclusivamente en boyas de salvamento marítimo, puesto que no reciben una cantidad masiva de datos como ocurre en el resto de estaciones AIS.

2.4. Tipos de mensajes AIS

La información compartimentada entre las estaciones AIS depende de la tecnología del transpondedor AIS que ofrece el servicio. En cualquier caso, la clasificación de los mensajes AIS para cualquier tipo de transpondedor AIS es la siguiente:

- Información estática: está referida a todos aquellos parámetros propios de una embarcación que permiten identificarla, de manera unívoca, frente al resto de estaciones:
 - MMSI (Maritime Mobile Service Identity).
 - Nombre de la embarcación.
 - Tipo de embarcación.
 - Dimensiones físicas.
 - Bandera del país de origen de la embarcación.
- Información dinámica: está referida a la información variable durante una travesía y que es de especial importancia para un correcto control de las operaciones marítimas provistas del sistema AIS:
 - Posición.
 - Velocidad.
 - Trayectoria.
 - Estado de la navegación.
- Información sobre el viaje: el usuario puede introducir manualmente algunos datos relevantes acerca de la travesía marítima y que no son monitorizados de forma automática por el sistema AIS.

Además de la clasificación de los mensajes AIS según su aplicación, se abre una nueva clasificación que diferencia en mensajes simples y mensajes compuestos [17].

Los mensajes simples contienen la información más básica de la embarcación (Figura 2.19), y se envían en diferentes intervalos de tiempo en función de la velocidad del buque. Si el barco está anclado o atracado, cada 3 minutos envía un informe de posición. Si está en movimiento, con una velocidad menor a 14 nudos, los informes se envían cada 10 segundos, y si supera los 14 nudos, las tasas de actualización varían entre 6 y 2 segundos, dependiendo si la embarcación cambia su rumbo.

MENSAJE SIMPLE	
1. MMSI	
2. Nombre	
3. Latitud	
4. Longitud	
5. Velocidad	
6. Rumbo	
7. Status	
8. Fecha	
9. Hora	

Figura 2.19: Campos incluidos en un mensaje AIS simple

Por otro lado, los mensajes compuestos incorporan información adicional sobre la travesía de la embarcación que envía este mensaje (Figura 2.20), y son enviados cada 6 minutos.

MENSAJE COMPUESTO			
1. IMO	10. Tipo embarcación 19. Eslora		
2. MMSI	11. Indicativo radio	20. Manga	
3. Latitud	12. Bandera 21. Calado		
4. Longitud	13. Puerto actual 22. Tonelaje bruto		
5. Velocidad	14. Puerto anterior 23. Tonelaje peso mue		
6. Rumbo	15. Puerto anterior (fecha) 24. Año de construcción		
7. Status	16. Puerto anterior (hora)		
8. Fecha	17. Destino		
9. Hora	18. Fecha de llegada estimada		

Figura 2.20: Campos incluidos en un mensaje AIS compuesto

Además, es posible agregar información complementaria a la que se provee en estos mensajes, como puede ser la asociación de la velocidad a un modo de navegación de la embarcación, o considerar el total de embarcaciones y el tiempo de navegación en aquellos cuadrantes de referencia geográfica conformados entre meridianos y paralelos para permitir la realización estudios estadísticos sobre la confluencia de tráfico marítimo en una región determinada.

El formato que define los mensajes AIS se recopila en la especificación NMEA 0183 [18]. Este formato común para la transmisión de datos entre dispositivos electrónicos destinados a equipos marítimos combina ciertas especificaciones eléctricas con los datos recibidos por GPS, permitiendo así la comunicación en tiempo real entre embarcaciones y estaciones marítimas.

La comunicación con formato NMEA 0183 se realiza través del puerto serie de los equipos marítimos. Los mensajes se presentan como una ráfaga de datos AIVDM que son transmitidos hasta un host como datos binarios ASCII y se codifican a través de los protocolos de red TCP/IP o UDP/IP.

Un ejemplo de mensaje NMEA 0183 es el siguiente:

!AIVDM,1,1, ,B,19NWrrP02sbuuuuhM86hA0=n2<0:,0*12.

En la Tabla 2.4 se describe el significado de las diferentes partes de un mensaje con formato NMEA 0183 como el que se presenta en el ejemplo anterior. Los campos del mensaje y su descripción en esta tabla se ordenan conforme a una lectura del mismo de izquierda a derecha.

Descripción	Campos NMEA
Tipo de mensaje NMEA	!AIVDM
Número de líneas del mensaje	1
Línea número 1 del mensaje	1
Espacio en blanco para separar mensajes multilínea	
Tipo de canal AIS $(A = AIS-1, B = AIS-2)$	B
Datos AIS codificados	$19NWr \cdots = n2 < 0:$
Fin de los datos AIS	0*
Checksum NMEA	12

Tabla 2.4: Campos de datos de un mensaje AIS codificado con NMEA 0183

2.5. Descripción hardware de los transceptores AIS

El estándar AIS diferencia entre dos clases de transpondedores AIS: AIS clase A y AIS clase B [19]. La diferencia entre estas dos clases de transpondedores depende de la calidad del hardware y las prestaciones que ofrece el sistema, permitiendo así que por un coste económico considerable cualquier embarcación pueda sumarse a la red AIS, pese a no estar obligadas a ello.

2.5.1. AIS clase A

Los transceptores AIS clase A son los transpondedores AIS de mayor calidad y mejores prestaciones, motivo por el que su integración es obligatoria para los buques

que deben cumplir con el Convenio SOLAS. Se caracterizan por operar con potencias de hasta 12.5 W y por implementar el protocolo SOTDMA (Figura 2.21).

Los subsistemas que integran estos transpondedores son:

- Un transmisor VHF sintonizable para los canales AIS-1 y AIS-2.
- Dos receptores de VHF para los canales AIS-1 y AIS-2.
- Un procesador de comunicaciones marítimas con una interfaz para comunicaciones IEC 61162/NMEA 0183 y donde esté implementado el protocolo SOTDMA.
- Un receptor GPS/DGPS para establecer el sincronismo entre estaciones.
- Una unidad de visualización del servicio, con la posibilidad de combinarse con la visualización radar.

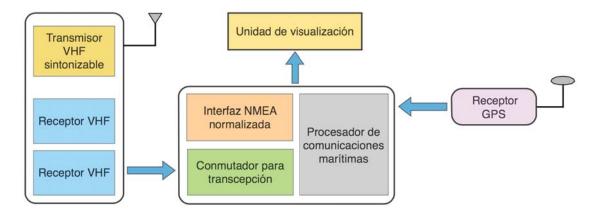


Figura 2.21: Diagrama de bloques de un transpondedor AIS clase A

2.5.2. AIS clase B

Los transpondedores AIS clase B son la opción económicamente más asequible para aquellas embarcaciones que no se ven inmersas en la obligatoriedad de cumplimiento del Convenio Solas, y por tanto, la integración de transpondedores AIS pasaría a ser opcional. Se caracterizan por operar con potencias de hasta $2\ W$ y por implementar el protocolo CSTDMA, de manera que el alcance es menor. Las embarcaciones que integren este modelo no se ven obligadas a transmitir (Figura 2.22).

Los subsistemas que integran estos transpondedores son:

- Un transmisor VHF sintonizable para uno de los posibles canales, AIS-1 o AIS-2.
- Un receptor de VHF para uno de los canales, AIS-1 o AIS-2.

- Un procesador de comunicaciones marítimas, que permite ser configurada conjuntamente a la instrumentación de abordo y donde esté implementado el protocolo CSTDMA.
- Un receptor GPS externo para establecer el sincronismo entre estaciones.
- Una unidad de visualización del servicio.

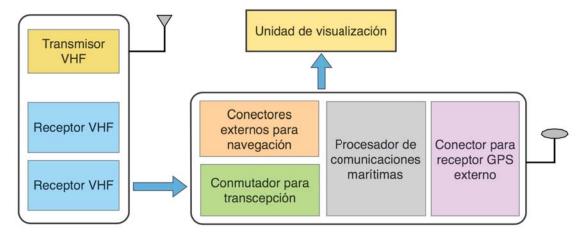


Figura 2.22: Diagrama de bloques de un transpondedor AIS clase B

Capítulo 3

Diseño modular del prototipo hardware

3.1. Introducción

En este capítulo se estudian las características técnicas de cada uno de los módulos que conforman el prototipo hardware del banco de pruebas AIS, con el objetivo de justificar el diseño realizado, que se expone en el Capítulo 4.

3.2. Arduino UNO R3

Arduino es una plataforma de hardware libre programable, cada vez más extendida desde sus orígenes en 2005, que se destinada a la realización de prototipos electrónicos para una gran variedad de propósitos. Está basada en un microcontrolador y un entorno de desarrollo, incluyendo un lenguaje de programación propio basado en Processing, con una sintaxis muy similar a C++ (Figura 3.1).



Figura 3.1: Diferentes modelos de plataforma Arduino

Cada plataforma Arduino está formada por una placa con un microcontrolador Atmel AVR y un cierto número de puertos de entrada y salida con los que el procesador se comunica con el exterior a través de sensores y otros dispositivos destinados a cumplir los propósitos de la aplicación que se quiera realizar. A estos sistemas añadidos se les suele llamar *shields*.

El modelo Arduino UNO R3 (Figura 3.2) es una placa de circuito impreso de dimensiones $68.6 \times 53.3 \ mm$ formada por un microcontrolador ATmega328P, 14 pines de entrada y salida digitales, 6 pines de entrada analógica y un puerto serie por el que se cargan los programas y realiza comunicaciones serie. Presenta una toma de alimentación externa de hasta 12 V que permite conseguir mayores niveles de tensión y corriente que los que aporta la conexión con el puerto serie, lo cual llega a ser preciso para ciertas aplicaciones que exijan un consumo de potencia elevado.

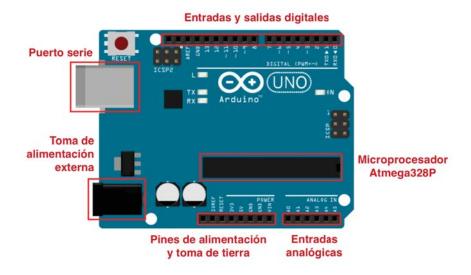


Figura 3.2: Microcontrolador Arduino UNO R3

3.2.1. Microcontrolador ATmega328P

El microcontrolador ATmega328P es un microcontrolador de la familia Atmel AVR de 8 bits. Integra una arquitectura RISC avanzada, con la capacidad para ejecutar hasta 131 instrucciones en un único ciclo de reloj. Además, presenta 32 registros de 8 bits cada uno.

Algunas de las características más relevantes son:

- Oscilador cerámico a 16 MHz.
- Memoria FLASH de 32 KB (0.5 KB son usados por el bootloader).
- Memoria SRAM de 2 KB.
- Memoria EEPROM de 1 KB.

3.2.2. Entradas y salidas digitales

Se presentan 14 pines digitales que se puede usar tanto como entradas como salidas. Funcionan a +5 V, y cada pin puede suministrar hasta 40 mA. Estos pines permiten escribir o leer niveles de tensión altos (HIGH) y bajos (LOW), que se asocian a los dos posibles estados binarios, 1 y 0, respectivamente.

Internamente, cada uno de estos pines presenta una resistencia de pull-up cuyo valor se comprende entre $20~k\Omega$ y $50~k\Omega$. Inicialmente, esta resistencia está desconectada salvo que el usuario dé instrucciones de que ocurra lo contrario.

Algunos pines digitales, además de tener la función genérica que se especifica, presentan otra función particular. Para el modelo Arduino UNO R3, estos pines especiales son los siguientes:

- Pines Serial: los pines 0 (TX) y 1 (RX) sirven para introducir pulsos TTL en el puerto serie.
- Pines de interrupciones: los pines 2 y 3 atienden las interrupciones externas *INT0* e *INT1*.
- Pines PWM: los pines 3, 5, 6, 9, 10 y 11 pueden presentar señales de salida con modulación por ancho de pulsos, que operan con el ciclo de trabajo de una señal pulso para poder controlar ciertos circuitos analógicos.
- Pines SPI: los pines 10, 11, 12 y 13 permiten la comunicación SPI cuando se implementa la correspondiente librería. De esta manera, sigue esta configuración:
 - SS (Slave Select): pin 10
 - MOSI (Master Output Slave Input): pin 11
 - MISO (Master Input Slave Output): pin 12
 - SCLK (Serial Clock): pin 13

3.2.3. Entradas y salidas analógicas

Se presentan 6 pines analógicos únicamente de entrada, que permiten leer datos analógicos en un rango de tensión entre 0 V - 5 V. Se designan como A_i , con i = 0...6.

Las entradas analógicas en Arduino se basan en un ADC con una resolución de 10 bits, lo que implica $2^{10}=1024$ posibles niveles digitales. Por tanto, dado que la tensión de alimentación de Arduino es de +5~V, implica una resolución en voltaje de 2.44~mV y una precisión relativa respecto a la señal de entrada de 0.1~%.

Para solventar el problema de detección de niveles de tensión en señales de entrada cuyos rangos de variación son inferiores a la tensión V_{CC} , Arduino incluye un pin adicional llamado V_{ref} (Analog Reference), que permite sustituir V_{CC} por la referencia del nivel de tensión máximo de variación de la señal de entrada (V_{ref}) , y así conservar la precisión original.

Las entradas analógicas en Arduino son esenciales para capturar valores de sensores conectados a este microcontrolador, como pueden ser para magnitudes analógicas como temperatura, humedad y presión, entre otras.

3.2.4. Puerto serie

La mayor parte de los modelos Arduino integran al menos un puerto serie, también conocido como UART (*Universal Asynchronous Receiver-Transmitter*), que opera con niveles de tensión entre 0 V-5 V. Este puerto serie se usa para transferir el código de los programas que son desarrollados sobre el IDE Arduino hacia la placa de prototipado, pero esta no es su única función.

Arduino presenta ciertas funciones a nivel software que son exclusivas del puerto serie. De esta manera, permite presentar resultados de operaciones internas del microcontrolador, ya sea la captura de valores analógicos o la transmisión y recepción de caracteres por los pines digitales, a través del monitor serie que integra el propio IDE Arduino.

En lo que respecta al hardware, Arduino UNO presenta en su placa un conector hembra USB tipo B por el que se cargan los programas que se desarrollan sobre el IDE a través de un cable formado por un conector macho USB tipo B (conectado directamente a la placa Arduino) y un conector macho USB tipo A (conectado al ordenador desde donde se ejecuta el IDE Arduino con el programa que se quiere implementar sobre la placa), tal y como se ve en la Figura 3.3.

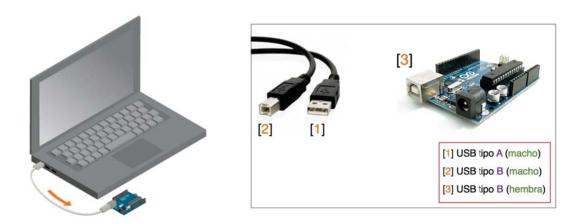


Figura 3.3: Conectores USB tipo A/B macho y hembra sobre Arduino UNO

3.3. Módulo banda base DV-MEGA GMSK

El módem DV-MEGA GMSK es un shield para Arduino diseñado por la compañía DV-MEGA, que puede tanto modular como demodular datos digitales siguiendo un esquema de modulación digital GMSK (Figura 3.4).

Este módem puede comportarse como punto de acceso o como repetidor en una comunicación. Su funcionamiento, basado en el chip CMX589A, está orientado a la integración del mismo sobre microcontroladores ATmega. Como nodo, se destina fundamentalmente a realizar comunicaciones con host, tanto sean un PC como una Raspberry Pi.

Puede operar con una tasa de transmisión de 9600 baudios para el intercambio de paquetes vía radio, y permite que la comunicación sea de tipo half-dúplex o full-dúplex. El funcionamiento de un modo u otro depende de las características del equipo radio al que se encuentre conectado.

Entre sus usos más habituales, destaca el de repetidor en equipos de radio D-STAR, para los que la compañía DV-MEGA ha desarrollado un firmware de control particular que es compatible con el software que manejan estos equipos. Sin embargo, esto no exime al resto de equipos de radio a integrar el módem DV-MEGA GMSK y desarrollar un firmware de control propio para cubrir sus necesidades.



Figura 3.4: Shield DV-MEGA GMSK

Para justificar su diseño, se describen los esquemas de funcionamiento que sigue este shield, entrando en detalle en el funcionamiento del chip CMX589A y los subsistemas de filtrado que integra el dispositivo en función de este chip.

3.3.1. Pines del módem DV-MEGA GMSK

En la Tabla 3.1 se presentan los pines relativos a la transmisión y recepción de datos del shield DV-MEGA GMSK, detallando su función y el modo (entrada o salida) en cada uno de los casos.

Pin	Modo	Descripción
TX SERIAL	Salida	No tiene conexiones con el módem.
RX SERIAL	Entrada	No tiene conexiones con el módem.
RX CLK	Entrada	Señal de reloj de la línea de datos recibidos.
TX CLK	Entrada	Señal de reloj de la línea de datos transmitidos.
S/N	Salida	Marca la relación S/N según el nivel de los datos recibidos.
RXD	Entrada	Línea de datos recibidos.
TXD	Salida	Línea de datos transmitidos.
RXDCACQ	Salida	Opera sobre el PLL del chip CMX589A.
PTT	Entrada	Permite la operación half-dúplex del módem.
PLLACQ	Salida	Opera sobre el PLL del chip CMX589A.
LED TX	Entrada	Enciende el LED de transmisión (verde).
LED COS	Entrada	Enciende el LED COS (rojo).
LED PWR	Entrada	Enciende el LED PWR (amarillo).
ARDU LED	Entrada	Enciende el LED integrado sobre Arduino.

Tabla 3.1: Descripción de los pines del módem DV-MEGA GMSK

3.3.2. CMX589A

El chip CMX589A es un encapsulado de 24 pines diseñado por la empresa CML Microcircuits que se describe como un módem GMSK banda base que puede operar siguiendo tanto un esquema de comunicación full-dúplex como half-dúplex (Figura 3.5).

Este chip permite tasas de transmisión comprendidas entre 4~kbps y 200~kbps. En lo que respecta al factor BT, presenta un pin configurable para seleccionar este factor en dos posibles valores: 0.3~y~0.5.

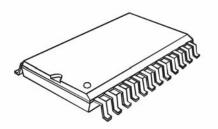


Figura 3.5: Encapsulado del chip CMX589A

La función principal de este chip es realizar una serie de filtrados sobre las señales de entrada y de salida siguiendo los esquemas de una modulación GMSK. Presenta pines de transmisión y recepción de datos que trabajan bit a bit, además de sus respectivas señales de reloj, que son las que permiten el sincronismo de los datos transmitidos y recibidos, respectivamente.

Este chip requiere circuitos externos para completar su funcionamiento, ya sean unidades de amplificación como filtros y moduladores paso banda, además de los cir-

cuitos de sincronismo que, entre otros factores, determinan la tasa binaria a la que puede operar el dispositivo.

Tal y como se observa en el diagrama de bloques de la Figura 3.6, los pines principales con los que opera el CMX589A son los pines de transmisión y recepción de datos $(TX\ Data\ y\ RX\ Data)$ y sus correspondientes pines de sincronismo $(TX\ Clock\ y\ RX\ Clock)$, que se conectan directamente a la UART o a un microcontrolador con el que se pretende comunicar para realizar la transferencia de datos.

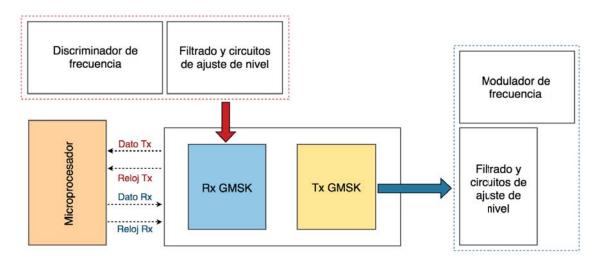


Figura 3.6: Subsistemas integrados al chip CMX589A

A este sistema se le suman algunos subsistemas relativos a la ganancia y al filtrado de la señal. Debido a que el CMX589A no puede asegurar que a su salida se entregue el nivel de señal que se requiere para una determinada aplicación, emplea diversos mecanismos de amplificación y filtrado para ajustar los niveles de señal y rechazar frecuencias indeseables a la entrada y a la salida del chip.

Los detalles técnicos del chip CMX589A, así como su configuración y selección de componentes y circuitos externos al propio chip, se recogen en el Anexo [A].

A continuación, se describen algunas redes externas que se integran en el dispositivo DV-MEGA GMSK, y son las siguientes:

- Circuito de sincronismo externo.
- Etapa de filtrado y ganancia en transmisión y recepción.
- Controlador PTT.

3.3.3. Circuito de sincronismo externo

El módem DV-MEGA GMSK emplea una red de sincronismo externa basada en un cristal de 9.8304 MHz para establecer las líneas de sincronismo TXCLK y RXCLK que se encuentran en el chip CMX589A (Anexo [A]).

Este cristal fija una señal de reloj de 9.8304 MHz para establecer una tasa binaria de 9.8 kbps. Para ello, la configuración binaria que deben seguir las líneas ClkDivA y ClkDivB del chip CMX589A es ClkDivA = 1 y ClkDivB = 1, pues se fija el divisor de frecuencia a N = 1024 y se establece que:

$$R_B = \frac{f_{XTAL}}{N} = \frac{9.8304 \cdot 10^6}{1024} = 9.8 \text{ kbps}$$
 (3.1)

La ecuación (3.1) queda recogida en el Anexo [A]. Se indica, además, que para el correcto funcionamiento de la red representada en la Figura 3.7, los condensadores C_2 y C_3 se fijan a 30 pF y la resistencia $R_2 = 1 M\Omega$.

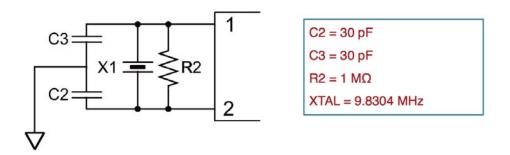


Figura 3.7: Circuito de sincronismo externo al chip CMX589A para 9.8 kbps

3.3.4. Etapas de filtrado y ganancia en transmisión

A la salida del pin TXOUT del chip CMX589A se colocan dos filtros. El primero de estos filtros se encarga de fijar la tasa binaria de salida del dispositivo, mientras que el segundo realiza un filtrado paso bajo sobre los datos e incluye, a través de un potenciómetro, el control de ganancia a la salida del módem.

En una primera etapa, el filtro pasivo RC paso bajo formado por la resistencia $R_1 = 47k\Omega$ y el condensador $C_1 = 470~pF$ determina el factor BT, según la tasa binaria seleccionada, y elimina el ruido de alta frecuencia (Figura 3.8).

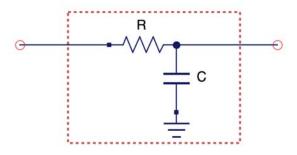


Figura 3.8: Filtro pasivo RC paso bajo

A partir de las especificaciones del chip CMX589A, si el factor BT de este dispositivo es BT = 0.5, el valor de la tasa binaria a la salida, según los valores de R_1 y C_1 anteriormente fijados, se calcula de la siguiente manera:

$$R_B = \frac{0.22}{R_1 \cdot C_1} = \frac{0.22}{47 \cdot 10^3 \cdot 470 \cdot 10^{-12}} \approx 9.8 \text{ kbps}$$
 (3.2)

A esta etapa de filtrado le sigue otra etapa que incluye un filtro activo paso bajo basado en un amplificador operacional (Figura 3.9):

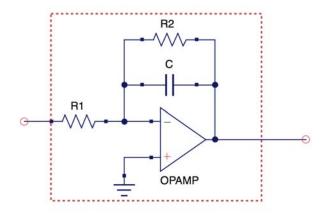


Figura 3.9: Filtro activo paso bajo

Su función de transferencia viene determinada por la ecuación (3.3):

$$H(j\omega) = -\frac{R_2}{R_1} \cdot \frac{\frac{1}{R_2 \cdot C}}{j\omega + \frac{1}{R_2 \cdot C}}$$

$$(3.3)$$

Por tanto, a partir de la ecuación (3.3) es posible determinar la ganancia del filtro activo, que viene dada por la expresión (3.4):

$$H_0 = -\frac{R_2}{R_1} \tag{3.4}$$

La expresión (3.4) indica que al menos uno de los dos valores resistivos de este filtro debe ser variable para que a la salida que se conecta al puerto del módem GMSK pueda ajustarse el nivel de señal. El valor variable que integra un potenciómetro es R_1 , que oscila entre 500 $k\Omega$ y 1 $M\Omega$. Esto implica que las variaciones de ganancia sean del orden de entre 7 dB hasta 10 dB.

Este filtro consigue que la señal que llega al modulador FM pueda variar linealmente la frecuencia en función de la tensión que regula el potenciómetro, de manera que se trata de un VCO (Voltage-controlled oscillator) [20].

La conjunción de estos filtros y elementos de control de ganancia en el transmisor, colocados a la salida del pin TXOUT, se representa en la Figura 3.10.

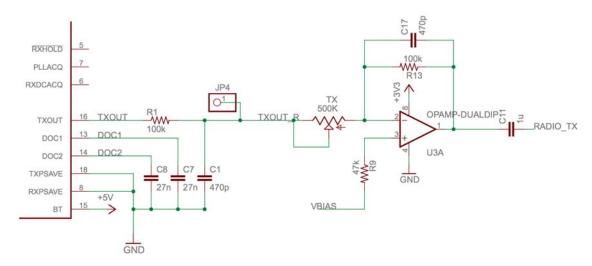


Figura 3.10: Filtro de transmisión

3.3.5. Etapas de filtrado y ganancia en recepción

A la entra del pin RX IN SIGNAL del chip CMX589A se coloca un filtro de topología similar a la estudiada para la etapa de transmisión (punto 3.3.4).

Este filtro, fundamentalmente, controla la ganancia del receptor. Consta de un potenciómetro que hace variar las señales de entrada para que no superen 1 V_{pp} a la entrada del chip CMX589A, como se indica en el Anexo [A], pues es la amplitud con la que opera internamente este chip. Se consigue, además, filtrar algunas componentes espúreas que pueden pasar desde el demodulador de FM.

Seguido de esta topología, se coloca un filtro RC paso bajo descrito en el Anexo [A], para el que se establecen los valores $R_3=100~k\Omega$, $R_4=100~k\Omega$ y $C_6=22~\rm pF$ para configurar que la tasa binaria con la que opera el chip CMX589A sea inferior a 64 kbps.

Estos filtros y los elementos de control de ganancia en el receptor, colocados a la entrada del pin RX IN SIGNAL, se representan en la figura 3.11.

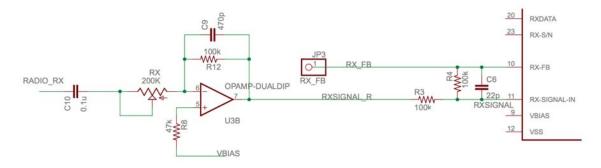


Figura 3.11: Filtro de recepción

3.3.6. Push To Talk

Los circuitos PTT (*Push To Talk*) suelen integrarse en los sistemas de radiofrecuencia para activar y desactivar los subsistemas de transmisión y de recepción. Normalmente, se asocian a los principios de operación de un walkie-talkie: cuando se pulsa el PTT, el equipo entra en modo transmisión y el usuario puede enviar voz, mientras que si no está pulsado, el usuario puede recibir la señal que llega al dispositivo desde otros equipos de radio.

Por tanto, se puede definir el PTT como el circuito que permite controlar la comunicación half-dúplex en una comunicación por radiofrecuencia. En el caso del módem DV-MEGA, este circuito realiza un proceso de modulación o de demodulación GMSK a partir de una señal de activación generada por el sistema de comunicación FM, aunque su incorporación no excluye el uso del modo símplex en este módem.

El circuito PTT está formado por una topología inversora. De esta manera, a través de una resistencia de pull-up, que mantiene el nivel que llega a la entrada del PTT, se consigue que en caso de que el equipo transmisor no esté transmitiendo voz y/o datos, se desconecte el modo transmisión y el modo recepción pasa a estar activo, o viceversa (Figura 3.12).

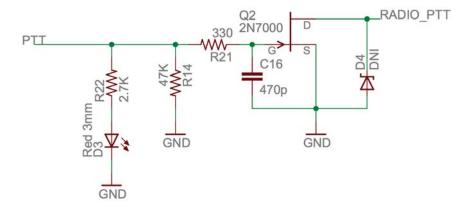


Figura 3.12: Circuito PTT del módem DV-MEGA GMSK

3.3.7. Conector Mini-DIN 6

Los conector Mini-DIN son una familia de conectores estandarizados por el organismo de estandarización alemán DIN ($Deutsches\ Institut\ f\"ur\ Normung$). Presentan sección circular, con un diámetro genérico de 9.5 mm, y el número de pines en su interior es variable dependiendo del modelo evaluado (Figura 3.13).

El módem DV-MEGA GMSK integra un conector Mini-DIN 6. Este conector se caracteriza por presentar 6 pines, de los cuales dos quedan inhabilitados para esta aplicación (Figura 3.14).

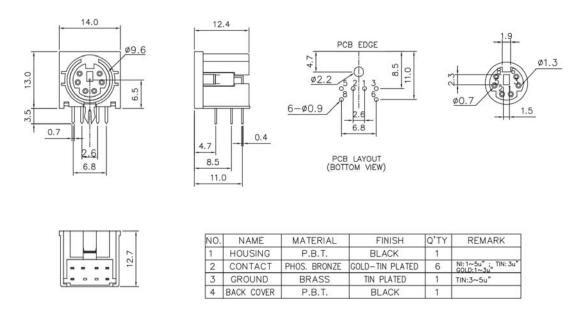


Figura 3.13: Dimensiones del conector Mini-DIN 6 (expresadas en milímetros, mm)

A continuación, se presenta la descripción de los pines habilitados para el módem:

- Pin 1 (OUT): salida de datos del módem, de manera que en relación a los circuitos implementados en el módem, presenta una conexión directa con la salida de las etapas de ajuste de ganancia y filtrado a la salida del pin TX OUT del chip CMX589A.
- Pin 2 (GND): toma de tierra del módem.
- Pin 3 (PTT): referido a la habilitación y deshabilitación de los modos de transmisión y de recepción en el módem.
- Pin 4 (IN): entrada de datos del módem, de manera que en relación a los circuitos implementados en el módem, presenta una conexión directa con la entrada de las etapas de ajuste de ganancia y filtrado a la entrada del pin RX SIGNAL del chip CMX589A.



Figura 3.14: Pines del conector Mini-DIN 6

3.4. Módulos paso banda FM Radiometrix

Dado que el estándar AIS especifica que su esquema de modulación debe presentar una modulación FM, se escogen dos dispositivos que operan como modulador y demodulador FM. La opción elegida son los chips Radiometrix TX1 (modulador) y Radiometrix RX1 (demodulador), ambos operando a la frecuencia 161.975 MHz.

3.4.1. Radiometrix TX1

El chip Radiometrix TX1 a la frecuencia 161.975 MHz es un dispositivo modulador de FM que opera sobre los datos que la llegan a su entrada, entregándolos a su salida modulados en FM y siendo emitidos por una antena que opera a la frecuencia de trabajo del propio chip.

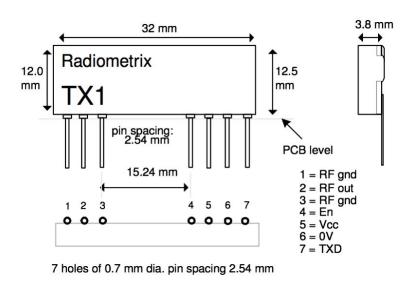


Figura 3.15: Dimensiones físicas del chip Radiometrix TX1

Tal y como se observa en la Figura 3.15, este dispositivo presenta siete pines de entrada y salida sobre su encapsulado. A continuación, se describe la función de cada uno de estos pines:

- RF GND: toma de tierra del dispositivo; los pines 1, 3 y 6 están conectados internamente, de manera que solo se necesita una conexión a tierra en alguno de estos pines para que el dispositivo funcione correctamente.
- RF Out: salida del modulador de FM que entrega a la salida una potencia de aproximadamente $10 \ dBm$ para una alimentación superior a $2.8 \ V$.
- Enable: señal que habilita la transmisión en el dispositivo. En caso de alimentar este pin con una tensión entre 0.15~V-1.7~V (asegurando que la corriente suministrada no sea inferior a $1~\mu A$), empieza a transmitir la portadora.

- Vcc: alimentación del dispositivo, comprendida en unos valores de tensión entre 2.2 V 12 V. No debe superar un rizado de $0.1 V_{pp}$. Además, la corriente máxima suministrada no puede ser superior $14 \ mA$.
- **TXD**: entrada para la señal moduladora con la que se realiza la modulación FM. Acepta señales de datos digitales comprendidas entre $0 \ V 3 \ V$.

A nivel de diagrama de bloques, el chip Radiometrix TX1 consiste en un filtro paso bajo a la entrada de la señal de datos TXD con una frecuencia de corte de 7 KHz. La señal filtrada pasa por un VCO que oscila a la frecuencia de 161.975 MHz, de manera que se encarga de generar la frecuencia portadora para hacer posible la modulación en frecuencia de los datos de entrada. Un filtro paso banda se encarga de ajustar la banda modulada previamente a la etapa de amplificación, generando a la salida una potencia máxima de RF que va en función del voltaje que entrega el regulador de tensión del dispositivo (Figura 3.16).

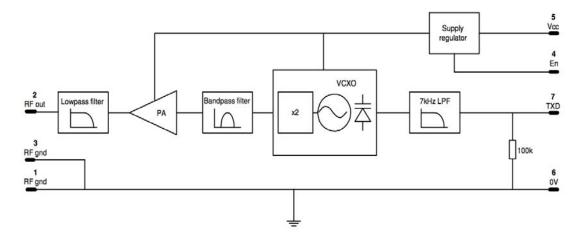


Figura 3.16: Diagrama de bloques del chip Radiometrix TX1

La salida del diagrama de bloques se conecta a una antena de VHF, ya sea de forma directa o incluyendo atenuadores o amplificadores, según el diseño que se quiera llevar a cabo. En cualquier caso, debe asegurarse una correcta adaptación a 50 Ω que permita que la señal modulada se transmita con el menor coeficiente de reflexión posible.

En la Tabla 3.2 se muestran algunas de las características técnicas más relevantes del chip Radiometrix TX1, clasificando cada parámetros según sus valores mínimo, nominal y máximo:

Parámetro	Mínimo	Norminal	Máximo
Tensión de alimentación (V)	2.2	3	12
Corriente de alimentación (mA)		9.5	11
Potencia de salida (dBm) ($V_{CC} = 2.2 V$)	+4.5	+6	7.5
Potencia de salida (dBm) ($V_{CC} \ge 2.8 \ V$)	+8.5	+10	+11.5
Atenuación de la banda espuria (dBc)		-70	-55
Desviación de frecuencia (KHz)	-2	0	+2
Ancho de banda a -3 dB (KHz)	0		7
Nivel de señal de entrada (LOW) (V)	-0.2	0	0.2
Nivel de señal de entrada (HIGH) (V)	2.8	3	3.2

Tabla 3.2: Características del chip Radiometrix TX1

3.4.2. Radiometrix RX1

El chip Radiometrix RX1 a la frecuencia 161.975 MHz es un dispositivo demodulador de FM que opera sobre los datos modulados en FM que llegan a su entrada desde una antena de VHF y los entrega a su salida demodulados (banda base) y convertidos a formato digital.

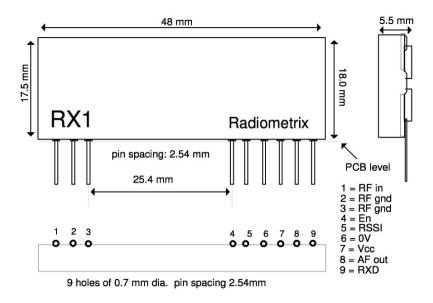


Figura 3.17: Dimensiones físicas del chip Radiometrix RX1

Tal y como se observa en la Figura 3.17, este dispositivo presenta nueve pines de entrada y salida sobre su encapsulado. A continuación, se describe la función de cada uno de estos pines:

■ RF GND: toma de tierra del dispositivo; los pines 2, 3 y 6 están conectados internamente, de manera que la conexión a tierra solamente es necesaria en alguno de estos pines para el correcto funcionamiento del dispositivo.

- RF In: entrada del demodulador de FM, con una sensibilidad de aproximadamente $-119 \ dBm$, dependiendo de la alimentación que se suministra al dispositivo.
- Enable: señal que habilita la recepción en el dispositivo. En caso de alimentar este pin con una tensión entre $0.15 \ V 1.7 \ V$ (asegurando de que la corriente sea inferior a $1 \ \mu A$), puede recibir la señal modulada.
- **RSSI**: indicador que permite medir la calidad del enlace en función del nivel de RF que aporta la señal de entrada. Entrega un cierto nivel de tensión, lo que permite realizar estudios de rango de señal superiores a 80 dB.
- Vcc: alimentación del dispositivo comprendida en unos valores de tensión entre 2.7 V-12~V. No debe superar un rizado de 0.1 V_{pp} . Además, la corriente suministrada debe estar comprendida entre 7mA-14mA.
- **AF Out**: presenta la señal demodulada, en formato analógico, que se recibe a la entrada del dispositivo. Por tanto, este pin muestra la señal moduladora que se ha sido enviada desde el transmisor.
- RXD: salida digital del demodulador, por lo que a través de esta salida se obtiene la señal digital que ha sido enviada desde el transmisor.

A nivel de diagrama de bloques, el chip Radiometrix RX1 consiste en una topología superheterodina de doble conversión a FI (Figura 3.18). Las señales que llegan a la entrada del dispositivo pasan por la etapa de RF, donde se realiza una preamplificación seguida de un filtrado paso banda. Una vez se realiza el filtrado en RF, la señal pasa por un mezclador con otra entrada conectada a un oscilador de cristal que resuena a la frecuencia de 161.975 MHz, permitiendo de esta manera que la señal de RF baje en frecuencia hacia la primera etapa de FI.

Al tratarse de una topología superheterodina de doble conversión, la señal pasa por un filtro paso banda a la frecuencia intermedia FI1 con su respectiva amplificación, y seguidamente vuelve a pasar por un mezclador que permite bajar la frecuencia de entrada hasta la frecuencia intermedia FI2 (con su respectiva amplificación), permitiendo de esta manera mejorar la selectividad.

Una vez realizada la doble conversión a FI, la señal pasa por un discriminador de frecuencia que convierte las variaciones de frecuencia en variaciones de amplitud de forma lineal, de manera que la señal modulada en FM realiza una conversión a señal modulada en AM. Debido a que la información se encuentra en la envolvente de la señal AM, lo más común es utilizar un detector de envolvente que permita extraer la envolvente de la señal modulada y eliminar la señal portadora, quedando únicamente la señal moduladora tras un proceso de demodulación. Para conseguir una buena selectividad en la última etapa de filtrado, se añade un filtro paso bajo de frecuencia de corte de 7 kHz.

Por último, con el objetivo de adaptar las tasas de transmisión a diferentes tipos de protocolos para los que puede ser empleado este dispositivo, se añade un buffer de recepción que permite almacenar los datos recibidos durante un cierto periodo de tiempo. A medida que este buffer se va llenando, el dispositivo permite que los datos vayan saliendo a una cierta velocidad por el pin RXD.

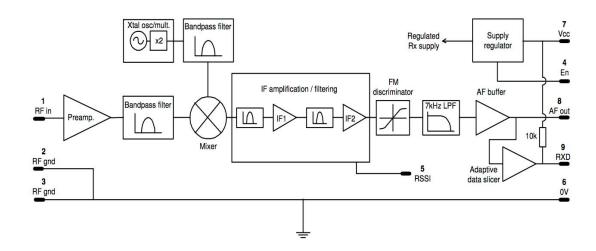


Figura 3.18: Diagrama de bloques del chip Radiometrix RX1

Cabe mencionar que a la entrada del Radiometrix RX1 se conecta una antena de VHF correctamente adaptada a 50 Ω para permitir que la señal modulada llegue al receptor RX1 con un COE adecuado.

En la Tabla 3.3 se muestran algunas de las características técnicas más relevantes del chip Radiometrix RX1, clasificando cada parámetros según sus valores mínimo, nominal y máximo:

Parámetro	Mínimo	Nominal	Máximo
Tensión de alimentación (V)	2.7	3	12
Corriente de alimentación (mA)		12	14
Sensibilidad en RF (dBm) $(S+N)/N$		-119	-115
Potencia de entrada IP3 (dBm)		-28	
Umbral RSSI (dBm)		-127	
Rango RSSI (dB)		80	90
Ancho de banda de FI (kHz)		15	
Atenuación en banda imagen (dB)	50	55	
Atenuación en canales adyacentes (dB)	50	55/60	
Atenuación de componentes espúreas (dB)	70	100	
Fugas en OL (conductores) (dBm)		-70	-65
Fugas en OL (radiadas) (dBm)		-70	-60
Ancho de banda a -3 dB en banda base (kHz)	0.05		6
Nivel AF (mV_{PP})		400	
Nivel de offset a la salida AF (V)	0.7	1	1.3
Distorsión en AF (%)		1	10

Tabla 3.3: Características del chip Radiometrix RX1

3.5. Antena de VHF marítimo

Las antenas de VHF marítimo que se emplean para la transmisión y recepción de señales moduladas en FM-GMSK en cada uno de los módulos del banco de pruebas son idénticas en cuanto a sus parámetros característicos. Son antenas Kenwood de VHF marítimo, con conectores SMA macho, que se emplean normalmente para walkie-talkie y radios de distintos servicios marítimos (Figura 3.19).



Figura 3.19: Antena VHF marítimo Kenwood SMA macho

En la Tabla 3.4 se presentan los parámetros que caracterizan a la antena Kenwood de VHF marítimo seleccionada para el prototipo hardware:

Parámetro	Valor	
Banda de frecuencia (MHz)	136 - 175	
Ancho de banda (MHz)	5	
Ganancia (dBi)	3	
Impedancia (Ω)	50	
COE	1.8	
Polarización	Lineal	
Tipo de conector (macho)	SMA	
Longitud (cm)	16.3	
Diámetro máximo (cm)	1.7	
Peso (g)	33	

Tabla 3.4: Características de la antena de VHF marítimo Kenwood

Capítulo 4

Integración del banco de pruebas AIS

4.1. Introducción

En este capítulo se detalla la implementación física de los módulos que componen el prototipo del banco de pruebas AIS. Se incluye el desarrollo del firmware de control para transmisión y recepción sobre el microcontrolador Arduino UNO R3, que permite realizar pruebas que se describen en el Capítulo 5.

4.2. Integración de los módulos

A partir de las características extraídas de los módulos descritos en el Capítulo 3, se detalla el diseño modular del banco de pruebas AIS para el transmisor y el receptor.

En el caso del transmisor, los elementos que caracterizan el diseño modular que se lleva a cabo son los siguientes:

- Ordenador personal con Matlab
- Microcontrolador Arduino UNO R3
- Modulador FM Radiometrix TX1
- Módem GMSK DV-MEGA
- Antena de VHF marítimo

Las conexiones entre cada uno de los módulos que se lleva a cabo para implementar el sistema de transmisión del prototipo hardware se presenta en la Figura 4.1.

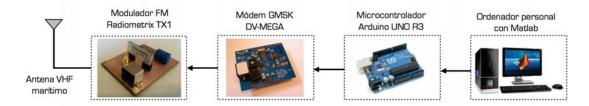


Figura 4.1: Diagrama de bloques del prototipo hardware del transmisor AIS

En el caso del receptor, los elementos que caracterizan el diseño modular que se lleva a cabo son los siguientes:

- Ordenador personal con Matlab
- Microcontrolador Arduino UNO R3
- Demodulador FM Radiometrix RX1
- Módem GMSK DV-MEGA
- Antena de VHF marítimo

Las conexiones entre cada uno de los módulos que se llevan a cabo para implementar el sistema de recepción del prototipo hardware se presentan en la Figura 4.2.

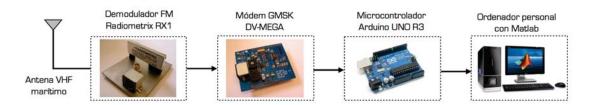


Figura 4.2: Diagrama de bloques del prototipo hardware del receptor AIS

Para realizar las conexiones entre los módulos de cada sistema se emplean dos tipos de cables:

- Para la conexión entre el ordenador personal y el microcontrolador Arduino, se emplea un cable USB con conectores tipo A hembra tipo B macho.
- Para la conexión entre el módem DV-MEGA GMSK y los dispositivos Radiometrix TX1 y Radiometrix RX1, se emplea un cable MINI DIN 6 con conectores macho macho.

Cabe señalar que en el caso de los módulos Radiometrix TX1 y Radiometrix RX1, se diseña una placa de circuito impreso de dimensiones físicas idénticas a las del Arduino UNO R3. El diseño PCB de las placas del transmisor y del receptor VHF queda especificado en los anexos [B] y [C], respectivamente.

En el caso de la conexión del módem DV-MEGA GMSK con el microcontrolador Arduino, tanto para el caso del transmisor como del receptor, al ser este módem un shield de Arduino, basta con acoplarlo sobre la plataforma Arduino para completar la conexión entre ambos.

4.3. Programación de un firmware de control en Arduino

La estructura básica del lenguaje de programación Arduino se compone al menos de dos partes bien diferenciadas: *setup* y *loop*, que forman en el IDE la estructura inicial de un *sketch* en Arduino [21].

```
void setup() {
   // estamentos del setup

void loop() {
   // estamentos del loop
}
```

A continuación, se describe la función de cada una de estas secciones de código para justificar su programación en relación al firmware de control:

- setup(): esta función es invocada una sola vez (al inicio del programa), y permite configurar inicialmente los modos de trabajo de los pines que estemos utilizando, del puerto serie o de las variables de trabajo que se declaran en el programa.
- loop(): tras la inicialización, esta función ejecuta de forma cíclica los estamentos que están definidos entre sus llaves. Esto permite que el programa pueda responder a los posibles eventos que puedan producirse en la placa.

Necesariamente, cada sketch en Arduino debe contener estas dos funciones. Esto significa que aunque alguna de las dos funciones no lleve definidos estamentos, debe ser declarada en el sketch, ya que forma parte de la estructura básica del programa.

Previamente a la declaración del setup(), se incluyen las definiciones y las variables globales que se emplean en el programa.

Las definiciones del programa son declaradas a través de la directiva define. Esta crea una macro asociada al nombre o identificador de la definición y le asigna un

valor referenciado en cualquier sección del código, permitiendo de esta manera que la programación sea más compacta.

```
#define ejemplo_definicion 2015
```

Con respecto a las variables, se declaran asignándole un tipo de variable, un nombre y si procede, un valor inicial. Los tipos que incluye el lenguaje de programación Arduino son los mismos que se incluyen en otros lenguajes de programación como C.

```
int variable_entero = 3;
float variable_decimal = 9.5;
char variable_caracter;
boolean variable_booleana = false;
```

Además, el lenguaje Arduino permite crear funciones que cumplan ciertas operaciones cuando son llamadas por el programa. Las funciones se caracterizan por tener un tipo, un nombre y si procede, parámetros. Estas mismas funciones pueden ser rutinas de servicio que atiendan las interrupciones que ocurren en el diseño.

```
char funcion1() {
   // estamentos de la funcion1
}

int funcion2(int valor1, int valor 2) {
   // estamentos de la funcion2
}
```

4.3.1. Funciones propias de Arduino

El lenguaje Arduino se caracteriza por presentar algunas funciones propias que facilitan la programación de los dispositivos. Es conveniente indicar algunas de las funciones más relevantes en Arduino para concretar la descripción del funcionamiento del firmware de control del transmisor (punto 4.4) y del receptor (punto 4.5).

En primer lugar, las funciones básicas de configuración, lectura y escritura de datos a través de los puertos digitales de Arduino son las siguientes:

• pinMode(pin, modo): permite configurar un pin según el modo que se especifique en la inicialización del programa. Puede ser modo entrada (*INPUT*) o modo salida (*OUTPUT*).

- digitalWrite(pin, estado): permite la escritura de niveles lógicos sobre un pin.
 El estado del nivel lógico lo marcan las constantes HIGH (nivel alto) y LOW (nivel bajo).
- digitalRead(pin): permite la lectura de niveles lógicos sobre un pin. El estado del nivel lógico lo marcan las constantes HIGH y LOW.
- delay(milisegundos): introduce un retardo durante la ejecución del programa, y se especifica como parámetro los milisegundos que se desea retardar una cierta rutina. Suele ser una función de gran utilidad para la visualización por pantalla de los datos transmitidos y recibidos a través del puerto serie.

Otras funciones de especial importancia son aquellas relativas al puerto serie, y que pertenecen a la clase *Serial* definida en Arduino. Estas funciones permiten la lectura y escritura de datos digitales a través del puerto serie, de manera que pueden ser representados por pantalla o ser enviados a otros dispositivos mediante un cable USB para su almacenamiento, procesado o visualización.

Para indicar que son funciones del puerto serie, se debe especificar la clase *Serial* en la instrucción antes de declarar la función:

```
Serial . funcion (parametros);
```

Las funciones principales del puerto serie son las siguientes:

- begin(velocidad, configuración): inicializa el puerto serie según el protocolo de la UART que se desee implementar. Se especifica para ello la tasa de símbolos (baudios) y la configuración del protocolo de la UART. Este último parámetro no es estrictamente necesario, ya que por defecto Arduino implementa el protocolo 8N1 (8 bits de datos, sin paridad y 1 bit de stop).
- write(dato): permite la escritura de un carácter por el puerto serie. Se diferencia de las funciones print() y println() en que estas escriben los caracteres según su equivalencia en la tabla ASCII.
- read(dato): permite la lectura de un carácter por el puerto serie.
- available(): devuelve el número de bytes disponibles para ser leídos desde el puerto serie. Suele emplearse como condición para la habilitación del puerto serie.
- flush(): realiza el borrado del buffer de transmisión y/o de recepción del puerto serie durante la inicialización del programa. Suele ser útil para asegurar que los datos previos a la ejecución del programa actual no interfieran con los nuevos datos.

4.3.2. Interrupciones en Arduino

Una interrupción es una señal de un dispositivo conectado al microprocesador que provoca la detención del programa principal con el fin de ejecutar una rutina externa al mismo. A esta rutina externa se le suele denominar rutina de servicio (ISR).

Las rutinas de servicio son secciones de código en un programa que permiten atender una interrupción que llega al procesador. De esta manera, la ISR se habilita y toma el control del procesador una vez llegue la interrupción, y se deshabilita cuando la interrupción haya sido atendida.

Generalmente, los microcontroladores pueden atender interrupciones *por hardware* y *por software*. Sin embargo, en el caso de los microcontroladores Arduino, actualmente solamente pueden ser procesadas las interrupciones hardware, aunque no se descarta que diseños futuros permitan la inclusión de las interrupciones software.

En los microcontroladores Arduino, dependiendo del modelo elegido, hay dos o más pines sobre los que pueden llegar interrupciones. El modelo Arduino UNO R3 presenta dos pines digitales sobre los que se atienden las interrupciones previamente programadas en el dispositivo. Estos pines son:

- Pin 2: atiende la interrupción *INT0*.
- Pin 3: atiende la interrupción *INT1*.

Cuando se activa una rutina de servicio para atender una interrupción, es necesario incluir una condición sobre los flancos de las señales que llegan al pin, a las que se les denomina *condiciones de disparo*. En el caso de Arduino, su programación permite incluir esta clasificación en las condiciones de disparo:

- LOW: la interrupción se dispara cuando el pin se encuentra a nivel bajo.
- CHANGE: la interrupción se dispara cuando el pin pasa de nivel alto a nivel bajo, o viceversa.
- RISING: la interrupción se dispara en flanco de subida (cuando el pin pasa de nivel bajo a nivel alto).
- **FALLING**: la interrupción se dispara en flanco de bajada (cuando el pin pasa de nivel alto a nivel bajo).

En la Figura 4.3 se incluye una descripción gráfica de estos cuatro posibles estados por los que se habilita una ISR.

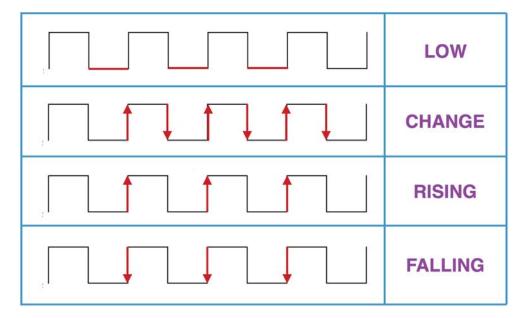


Figura 4.3: Condiciones de disparo en Arduino

Por tanto, cuando se define una interrupción en Arduino, hay dos funciones para habilitar y deshabilitar una interrupción: attachInterrupt (habilita la ISR) y detachInterrupt (deshabilita la ISR).

Para la función *attachInterrupt*, se definen tres parámetros: el número de la interrupción, el nombre de la ISR y la condición de disparo. Es decir:

```
attachInterrupt(interrupt, ISR_name, mode);
```

Para la función detachInterrupt, se define únicamente como parámetro el número de la interrupción:

```
detachInterrupt(interrupt);
```

4.4. Desarrollo del firmware de control del transmisor

El firmware de control del transmisor se diseña con el objetivo de que el microcontrolador Arduino que actúa como transmisor AIS en el banco de pruebas pueda recibir datos desde un ordenador personal a través del puerto serie empleando atención por polling. El propio Arduino genera el formato de la tramas AIS, la codifica en NRZI y la envía a través del módem DV-MEGA GMSK hasta el modulador FM por medio de una rutina de servicio, donde las tramas son moduladas y, posteriormente, enviadas a través de la antena de VHF marítimo del banco de pruebas del transmisor.

4.4.1. Definiciones y variables globales

En esta sección de código se definen los pines del módem GMSK, los valores límite que son usados en el programa y las variables globales con las que se trabaja en el mismo.

En lo que respecta a la definición de los pines, cabe señalar que se usa la directiva define y se declara el nombre del pin y su número asociado en el módem GMSK.

```
1 #define pinRXCLK
                       2
2 #define pinTXCLK
                       3
3 #define pinSN
                       4
4 #define pinRXD
                       5
5 #define pinTXD
                       6
6 #define pinPTT
                       8
7 #define pinPLLACQ
                       9
8 #define pinLEDTX
                       10
9 #define pinLEDCOS
                       11
10 #define pinLEDPWR
                       12
11 #define pinArduinoLED 13
```

Los valores límites también usan la directiva define y la misma declaración que los pines. Se asocian a los valores que sirven para establecer condiciones y hacer la programación más compacta. De esta forma, se tiene MAX DATA (indica el número máximo de datos que puede almacenar una trama AIS), LONG HEADER (indica el número de bytes fijos en la trama AIS) y MAX FRAME (indica el tamaño máximo de la trama AIS).

```
#define MAX_DATA 23
2 #define LONG_HEADER 4
3 #define MAX_FRAME 2*LONG_HEADER + MAX_DATA
```

Las variables globales usadas son las siguientes:

```
uint8_t txdato;
uint8_t txByte;
uint8_t datoGMSK;
uint8_t datoGMSK_NRZI = 0;
int contadorBitTx = 0;
int contadorByteTx = 0;
int contadorTx = 0;
unsigned char bufferTx[MAX_FRAME];
```

4.4.2. Inicialización del programa

En esta sección del código se inicializan el puerto serie, los pines del módem GMSK y los elementos del vector de la trama que son fijos.

El puerto serie se inicializa con una velocidad de transmisión de 9600 baudios a través de la función begin() de la clase Serial. Además, es importante asegurar que el puerto serie no tenga datos anteriores que puedan ser transmitidos por error, de manera que se emplea la función flush() de la clase Serial para borrar cualquier dato presente en el puerto serie antes de iniciar la ejecución del programa.

```
Serial.begin(9600);
Serial.flush();
```

En la inicialización de los pines, cabe señalar que aunque se trabaja casi en exclusividad con los pines TXCLK y TXD, es conveniente inicializar todos los pines con el fin de que el código de control del transmisor pueda ser reutilizado para otros trabajos posteriores. Para ello se emplea la función pinMode() que recibe como parámetro el pin y el modo en el que se quiera trabajar con dicho pin, pudiendo ser de salida (OUTPUT) o de entrada (INPUT).

```
pinMode(pinTXCLK, INPUT);
pinMode(pinRXCLK, INPUT);
pinMode(pinSN, INPUT);
pinMode(pinRXD, INPUT);
pinMode(pinTXD, OUTPUT);
pinMode(pinLEDTX, OUTPUT);
pinMode(pinLEDCOS, OUTPUT);
pinMode(pinLEDPWR, OUTPUT);
pinMode(pinLEDPWR, OUTPUT);
pinMode(pinPTT, OUTPUT);
```

Puede comprobarse como el pin TXD se inicializa como salida debido a que los datos que van a ser generados por el microcontrolador Arduino son enviados a través de este pin hacia la interfaz radio. Sin embargo, los pines TXCLK y RXCLK deben actuar como entradas debido a que recogen las señales de reloj asociadas a la transmisión y la recepción, respectivamente, para poder establecer el sincronismo con los datos transmitidos y recibidos.

Por último, una trama AIS contiene 4 bytes iniciales relativos al preámbulo y a la cabecera de inicio. Dado que la trama AIS se declara como un vector, los 4 primeros elementos de este vector deben ser fijos. Lo mismo ocurre con los 4 últimos bytes de la trama AIS, que se corresponden con la cabecera de fin y el buffer temporal.

```
bufferTx[0] = 0x55;
```

```
bufferTx[1] = 0x55;
bufferTx[2] = 0x55;
bufferTx[3] = 0x7E;
bufferTx[LONG_HEADER + MAX_DATA] = 0x7E;
bufferTx[LONG_HEADER + MAX_DATA+1] = 0x00;
bufferTx[LONG_HEADER + MAX_DATA+2] = 0x00;
bufferTx[LONG_HEADER + MAX_DATA+3] = 0x00;
```

Cabe señalar que los 3 bytes relativos al buffer temporal, debido a que se relacionan fundamentalmente con niveles superiores de la arquitectura de red AIS y que no son objeto de este trabajo, se inicializan a 0x00 para conservar el formato de la trama pero sin implementar sus valores adecuados, que dependen de la transmisión y el sistema sobre el que vienen integrados.

4.4.3. Programa principal

En esta sección del código, primero se comprueba si en el puerto serie se reciben bytes. Para ello se establece una condición que indique que en caso de no haber bytes disponibles, no se ejecutarán las siguientes líneas del código. Se usa para ello la función available() de la clase Serial.

```
while(!Serial.available());
```

En caso de que haya bytes en el puerto serie, se reciben los bytes a través de la función read() de la clase Serial y se almacenan en el vector de la trama AIS hasta que se hayan almacenado 23 bytes correspondientes a los campos de datos y CRC. Además, a través de la función write() de la clase Serial, se presentan por pantalla los datos introducidos con un retardo de 10 ms entre carácter y carácter, determinado por la función delay().

```
for(int k=0; k<MAX_DATA; k++) {
   delay(10);
   txdato = Serial.read();
   bufferTx[LONG_HEADER + k] = txdato;
   Serial.write(txdato);
}</pre>
```

Cuando los datos recibidos estén almacenados en el vector de la trama AIS y en disposición de ser enviados por el módem GMSK, se reinician los contadores de bits y bytes (que están declarados en la rutina de servicio del transmisor) y se habilita la interrupción de transmisión a través de la función attachInterrupt().

```
contadorBitTx = 0;
contadorByteTx = 0;
attachInterrupt(1, txISR, RISING);
```

En la Figura 4.4 se presenta el flujograma del proceso que sigue el programa principal del firmware de control del transmisor.

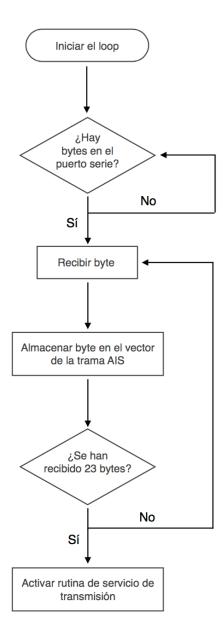


Figura 4.4: Flujograma del programa principal del firmware de control del transmisor

4.4.4. Rutina de servicio del transmisor

En primer lugar, la rutina de servicio extrae los bytes almacenados en el vector de la trama AIS. Sobre cada byte, toma cada uno de los bits que lo componen a través de un desplazamiento lógico a la derecha. Estos bits se escriben sobre el pin TXD a través de la función digitalWrite().

```
txByte = bufferTx[contadorByteTx];
datoGMSK = (txByte >> (7-contadorBitTx)) & 0x01;
// CODIFICACION NRZI
digitalWrite(pinTXD, datoGMSK ? HIGH:LOW);
```

Previamente a que los bits sean escritos sobre el pin TXD, se establece la codificación NRZI sobre los bits a partir del siguiente bloque condicional:

- Si el dato que se va a transmitir está a nivel alto (HIGH), se realiza una operacion XNOR entre el estado del pin de transmisión y el propio bit a transmitir.
- Si el dato que se va a transmitir está a nivel bajo (LOW), se lee el estado del pin de transmisión y se realiza la escritura del bit resultante sobre el pin TXD.

```
if(datoGMSK == HIGH) {
  datoGMSK_NRZI = digitalRead(pinTXD) ^ datoGMSK;
  digitalWrite(pinTXD, datoGMSK_NRZI);
} else {
  datoGMSK_NRZI = digitalRead(pinTXD);
  digitalWrite(pinTXD, datoGMSK_NRZI);
}
```

Seguidamente, la escritura de los bits que forman cada byte se realiza hasta que el contador de bits llegue a los 8 bits, indicando que se ha escrito un byte y por tanto, el contador de bytes aumenta una unidad. Este proceso se repite mientras el contador de bytes no alcance el valor de 23 bytes.

En caso de que el contador de bytes llegue a 23, se deshabilita la rutina de servicio del transmisor a través de la función detachInterrupt(). De esta manera, vuelven a almacenarse los datos recibidos desde el puerto serie desde el programa principal y se vuelve a realizar todo el proceso descrito anteriormente en la rutina de servicio del transmisor.

```
if(contadorBitTx >= 7) {
  contadorBitTx = 0;
  if(contadorByteTx >= MAX_FRAME-1) {
     contadorByteTx = 0;
}
```

En la Figura 4.5 se presenta el flujograma del proceso que sigue la rutina de servicio del firmware de control del transmisor.

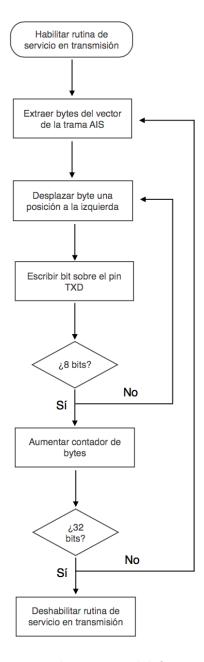


Figura 4.5: Flujograma de la rutina de servicio del firmware de control del transmisor

4.5. Desarrollo del firmware de control del receptor

El firmware de control del receptor se diseña con el objetivo de que el microcontrolador Arduino que actúa como receptor AIS en el banco de pruebas pueda enviar los datos, mediante técnicas de polling, por el módem GMSK hasta un ordenador personal a través del puerto serie. El microcontrolador Arduino se encarga de filtrar el preámbulo y las cabeceras de la trama AIS para así poder extraer los datos que son enviados desde el transmisor, decodificarlos en NRZI y representarlos por pantalla, usando para ello una rutina de servicio.

4.5.1. Definiciones y variables globales

En esta sección de código se definen los pines del módem GMSK, los valores límite que son usados en el programa y las variables globales con las que se trabaja en el mismo.

La definición de los pines en el programa del receptor es la misma que en el caso del transmisor que se estudia en el punto 4.4.1.

Los valores límites que se usan son MAX DATA (indica el número máximo de datos que puede almacenar una trama AIS) y HEADER (marca el valor 0x7E que caracteriza a las cabeceras de inicio y de fin de la trama), declarados a partir de la directiva define.

```
#define MAX_DATA 23
2 #define HEADER 0x7E
```

Las variables globales usadas son las siguientes:

```
uint8_t rxdato;
uint8_t rxdatoNRZI;
uint8_t rxdatoNRZI_anterior = 0;
uint8_t rxByte;
unsigned char rxHeader = 0;
unsigned char rxTail = 0;
uint8_t contadorBitRx = 0;
uint8_t contadorByteRx = 0;
uint8_t contadorByteRx = 0;
unsigned char bufferRx[MAX_DATA];
int flagComienzoTrama = 0;
int flagRx = 0;
```

4.5.2. Inicialización del programa

La inicialización que se hace para el puerto serie y los pines del módem GMSK será idéntica a la que se estudia en el firmware del transmisor en el apartado 4.4.2. Por tanto, la única diferencia entre la inicialización del firmware del receptor con respecto a la del transmisor es que se inicializa la rutina de servicio del receptor y no se incluye la inicialización de los valores fijos del vector de la trama AIS ya que no la está generando.

```
attachInterrupt(0, rxISR, FALLING);
```

4.5.3. Programa principal

En esta sección del código, se comprueba previamente si la rutina de servicio en recepción ha recibido los bytes de datos que integra la trama AIS enviada desde el transmisor. En caso afirmativo, el dispositivo activa un flag de recepción que le permite enviar cada byte almacenado en el vector de datos recibidos por el puerto serie, empleando para ello la función write() de la clase Serial.

Una vez se envían los 23 bytes del vector de datos recibidos a través del puerto serie, se reinician las variables de la rutina de servicio en recepción y vuelve a activarse la misma para poder seguir recibiendo los datos que llegan al pin RXD.

```
if(flagRx) {
    for(int k=0; k<MAX_DATA; k++) {</pre>
2
3
       delay(10);
       Serial.write(bufferRx[k]);
4
5
6
      flagRx = 0;
      rxHeader = 0;
7
8
      rxTail = 0;
9
      attachInterrupt(0, rxISR, FALLING);
10 }
```

En la Figura 4.6 se presenta el flujograma del proceso que sigue el programa principal del firmware de control del receptor.

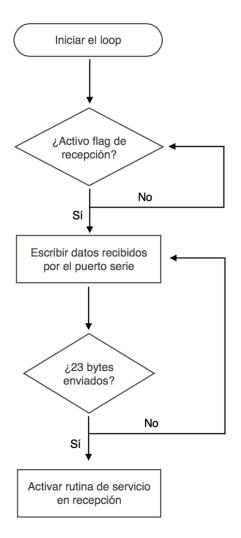


Figura 4.6: Flujograma del programa principal del firmware de control del receptor

4.5.4. Rutina de servicio del receptor

En primer lugar, la rutina de servicio en recepción lee por el pin RXD los bits que llegan desde el transmisor y los almacena en una variable, usando para ello la función digitalRead().

```
1 rxdato = digitalRead(pinRXD);
```

Los datos que se leen desde este pin se encuentran codificados en NRZI, de manera que se establece un bloque condicional para deshacer el proceso de codificación NRZI del transmisor. Este bloque condicional opera de la siguiente manera:

- Si se recibe un bit a nivel alto (HIGH), se realiza una operación XNOR entre el dato anterior y el que se lee por el pin TXD, se almacena en la variable del dato actual y se actualiza el dato anterior.

- Si se recibe un bit a nivel bajo (LOW), se almacena el dato anterior en la variable del dato actual y se actualiza la variable del dato anterior

```
if(rxdatoNRZI == HIGH) {
    rxdato = rxdatoNRZI_anterior ^ rxdatoNRZI;
} else {
    rxdato = rxdatoNRZI_anterior;
}
rxdatoNRZI_anterior = rxdatoNRZI;
```

A partir de este punto, se evalúa con una estructura switch/case los diferentes campos que constituyen la trama AIS que se recibe.

```
switch(flagComienzoTrama) {
1
2
    case 0:
3
      // código de filtro del preámbulo
4
    break;
5
6
    case 1:
      // código de filtro de la cabecera de inicio
7
    break;
8
9
10
    case 2:
11
      // código de filtro de los datos y CRC
12
    break;
13 }
```

Se distingue así un primer campo de 3 bytes referido a la secuencia alternada del preámbulo. El código que corresponde a este caso combina un desplazamiento lógico a la izquierda con una operación OR para que los datos recibidos por el pin RXD se almacenen en una variable, que representa al byte.

```
rxHeader = (rxHeader << 1) | rxdato;
```

A continuación, se compara el byte recibido con la cabecera de inicio 0x7E. En caso de que sea igual, la variable evaluada por el *switch* se pone a $\mathbf 1$ y pasa a evaluar el segundo caso.

```
if(rxHeader == 0x7E) {
  flagComienzoTrama = 1;
}
```

En el segundo caso, se evalúa la recepción de la cabecera de inicio de la trama AIS. De esta manera, se almacenan los bits recibidos en una variable que representa a un byte y se van almacenando los bytes en un vector de recepción. Este proceso se repite hasta llegar al máximo de datos que contiene la trama AIS, y una vez se llega al final de la misma, la variable de evaluación del *switch* pasa a tomar el valor 2.

```
rxByte = (rxByte << 1) | rxdato;</pre>
  if(contadorBitRx >= 7) {
    bufferRx[contadorByteRx] = rxByte;
    contadorByteRx++;
4
    contadorBitRx = 0;
5
    rxByte = 0;
6
    if(contadorByteRx>=MAX_DATA) {
7
      flagComienzoTrama = 2;
8
    }
9
    else {
10 }
    contadorBitRx++;
11
12 }
```

Por último, el tercer caso evalúa si el siguiente byte recibido es la cabecera de fin (0x7E). En caso de serlo, se reinician a 0 los contadores de bits y bytes, además de la variable de evaluación del *switch*, y se deshabilita la rutina de servicio en recepción a través de la función *detachInterrupt()*. Además, debe activarse el flag de recepción para que los datos almacenados en el vector de recepción puedan enviarse a través del puerto serie desde el programa principal.

```
rxTail = (rxTail << 1) |</pre>
                              rxdato;
  if(contadorBitRx >= 7) {
2
    flagComienzoTrama = 0;
3
    contadorBitRx = 0;
4
    contadorByteRx = 0;
5
    detachInterrupt(0);
    flagRx = 1;
   else {
8
    contadorBitRx++;
9
10 }
```

En la Figura 4.7 se presenta el flujograma del proceso que sigue el programa principal del firmware de control del receptor.

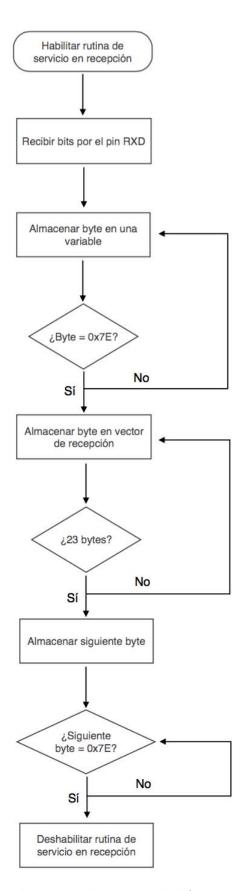


Figura 4.7: Flujograma de la rutina de servicio del firmware de control del receptor

4.6. Descripción de la interfaz Arduino - Puerto serie

Se emplean dos aplicaciones para el manejo de los sistemas de pruebas en transmisión y en recepción. En ambos casos, el objetivo de las aplicaciones utilizadas es realizar la comunicación vía puerto serie entre un ordenador personal y Arduino.

Para el caso se transmisión, se emplea el software *CoolTerm* (Figura 4.9), que permite enviar una cadena de caracteres a través del puerto serie hasta el dispositivo Arduino transmisor.

Para recepción, se utiliza la aplicación $Serial\ Monitor$ incluida en Matlab 2014b (Figura 4.11), que permite al usuario visualizar los datos recibidos desde el puerto serie al que se conecta el dispositivo Arduino receptor, además de almacenar estos datos en un fichero de extensión (.dat).

En la Figura 4.8 se presenta un esquema del montaje final del banco de pruebas, incluyendo tanto los elementos hardware que caracterizan tanto al transmisor como al receptor (microcontrolador, módem GMSK y módulos paso banda) como las interfaces de transmisión y recepción de datos en cada uno de los casos.

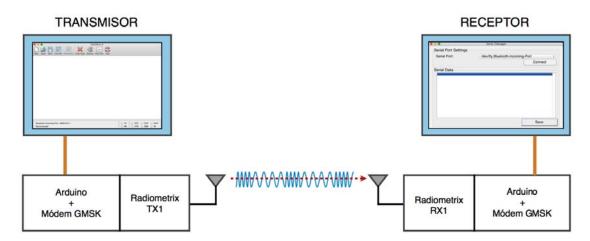


Figura 4.8: Banco de pruebas transmisor y receptor con interfaces Serial

4.6.1. Interfaz de transmisión

En la Figura 4.9 se presenta la interfaz del software *CoolTerm*, utilizado para el envío de cadenas de caracteres que se incluyen en el campo de datos de las tramas AIS generadas por Arduino para el sistema transmisor.

CoolTerm ofrece diferentes utilidades dependiendo de la aplicación que se esté evaluando. Las principales funciones que se manejan para el control del prototipo hardware se encuentran en la ventana Connection, donde se puede establecer la conexión con el puerto serie donde se conecta el prototipo transmisor.

Además, se presenta el comando *Send String*, que envía una cadena de caracteres que se reciben en Arduino para transmitir tramas AIS (Figura 4.10).

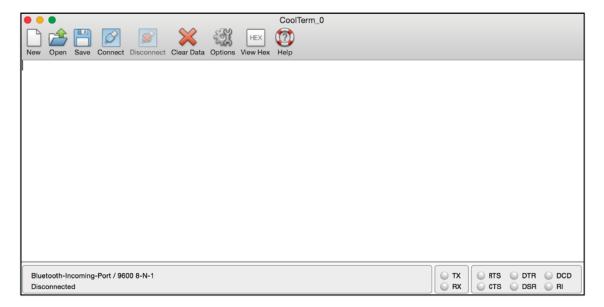


Figura 4.9: Interfaz de la aplicación *CoolTerm*

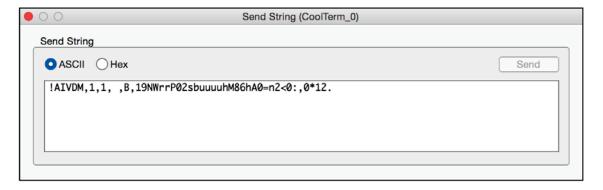


Figura 4.10: Envío de cadenas de caracteres por el puerto serie con Send String

4.6.2. Interfaz de recepción

En la Figura 4.11 se presenta la interfaz *Serial Monitor* que se utiliza para la visualización de los datos recibidos desde Arduino.

Esta interfaz incluye una ventana llamada Serial Data. Sobre esta ventana, los datos recibidos quedan representados por pantalla, y con la opción Save, se permite almacenar los datos recibidos en un fichero Matlab. Esto permite operar con los datos recibidos desde Matlab a través de diferentes pruebas que se recogen en el Capítulo 5.

Por último, se debe especificar el puerto de conexión a través de *Serial Port*. Una vez se selecciona el puerto serie al que está conectado el microcontrolador Arduino que opera como receptor, basta con seleccionar *Connect* para comenzar a recibir datos.

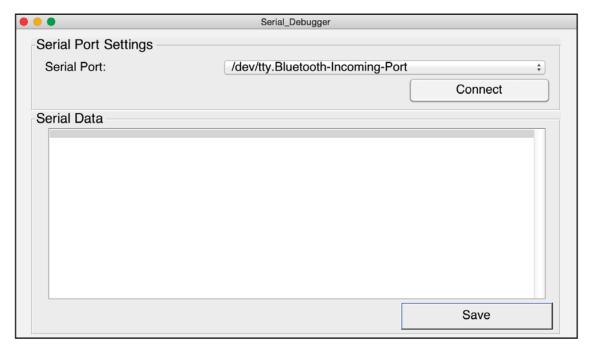


Figura 4.11: Interfaz de la aplicación $Serial\ Monitor$ de Matlab 2014b

Capítulo 5

Pruebas y resultados

5.1. Introducción

En este capítulo se describen las pruebas de conformidad para cada uno de los módulos que forman parte del prototipo hardware, así como las pruebas finales de viabilidad al integrar estos módulos sobre un mismo sistema de transmisión y recepción.

5.2. Pruebas sobre los módulos paso banda

Se describen las pruebas individuales realizadas sobre cada de uno de los módulos paso banda que integran el prototipo hardware. Los dispositivos que se evalúan para este propósito son los chips Radiometrix TX1 (3.4.1) y Radiometrix RX1 (3.4.2).

El objetivo de estas pruebas es comprobar el funcionamiento de estos módulos, empleando para ello señales de test que puedan simular el comportamiento de los dispositivos previamente a su integración con el módulo banda base.

5.2.1. Pruebas sobre el módulo Radiometrix TX1

Para realizar las pruebas de verificación del chip Radiometrix TX1, se requieren los siguientes materiales:

- Radiometrix TX1.
- Fuente de alimentación.
- Generador de señal.
- Analizador de espectros.
- Atenuador fijo.

Con estos elementos, se realiza el montaje de la Figura 5.1.

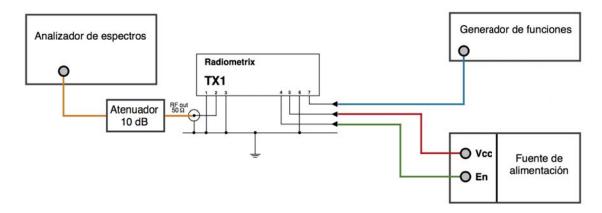


Figura 5.1: Montaje de prueba para el chip Radiometrix TX1

En este montaje se aprecian dos líneas relativas a la fuente de alimentación. La línea roja representa la tensión de alimentación $V_{CC} = 3 V$ y una corriente de 9.5 mA aplicada sobre el dispositivo, mientras que la línea verde entrega la tensión $V_{EN} = 1.5 V$ y 0.1 mA para que el dispositivo se encuentre operativo.

La conexión entre la salida del Radiometrix TX1 y la entrada del analizador de espectros a través de un cable coaxial rígido está adaptada a 50 Ω , y depende de la entrada del chip $(RF\ IN)$ que se encuentra conectada por medio de otro cable coaxial a la salida del generador de funciones.

El generador de funciones se ajusta para entregar a su salida una señal cuadrada de 4.8 KHz, con una amplitud de 1.5 V_{pp} y un nivel de offset de 500 mV_{pp} . Cada pulso tiene un ciclo de trabajo del 50 %. De esta manera, se consigue representar una tasa binaria de 9600 bps.

Una vez se realiza el montaje de la Figura 5.1, la primera medida tomada para estas pruebas es la potencia de portadora. Se realiza desconectando el generador de funciones de la entrada del modulador de FM. De esta forma, el modulador Radiometrix TX1 únicamente emite su portadora a 161.975 MHz.

En la Figura 5.2 se observa que el analizador de espectro detecta una potencia de pico de portadora de $-2.8 \ dBm$. Se trata de un resultado lógico si se tiene en cuenta que el modulador transmite una potencia cercana a $7 \ dBm$ y se le aplica a la señal de salida una atenuación de $10 \ dB$, por lo que se esperaría una potencia teórica de $-3 \ dBm$.

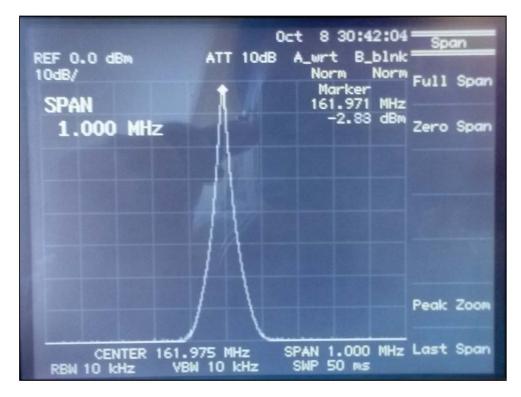


Figura 5.2: Señal portadora a 161.975 MHz generada por el chip Radiometrix TX1

La siguiente medida es el ancho de banda de la señal modulada. Para ello, basta con conectar el generador de funciones ajustado tal y como se indica anteriormente, de manera que se tiene una señal moduladora en forma de señal cuadrada combinada con una portadora a 161.975 MHz sobre una modulación FM.

Para esta medida, es necesario buscar una justificación a partir de la regla de Carson (5.1). Considerando una desviación de frecuencia $\Delta f = \pm 2.5$ KHz y que la tasa de transmisión es el doble de la frecuencia moduladora al tratarse de una señal cuadrada, $V_T = 9.6$ KHz, se obtiene un ancho de banda de 24.2 KHz.

$$B_T \simeq 2 \cdot (\Delta f + V_T) \tag{5.1}$$

Teniendo este dato presente, el ancho de banda registrado a partir de la utilidad *POWER MEASURE* del analizador de espectros es de 26.1 KHz, por lo que hay un error de 1.9 KHz entre el cálculo teórico a partir de la regla de Carson y la medida obtenida en el laboratorio (Figura 5.3).

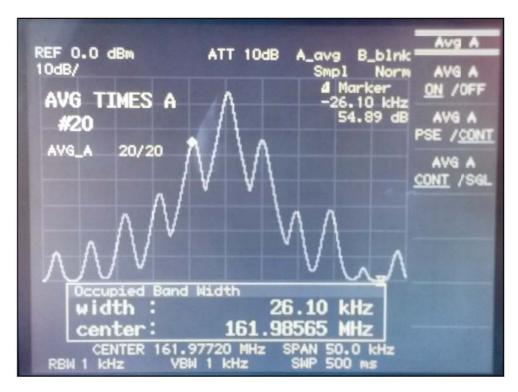


Figura 5.3: Espectro FM de una señal cuadrada como moduladora

5.2.2. Pruebas sobre el módulo Radiometrix RX1

Para realizar las pruebas de verificación del chip Radiometrix RX1, se requieren los siguientes materiales:

- Radiometrix RX1.
- Radiometrix TX1.
- Fuente de alimentación.
- Generador de señal.
- Analizador de espectros.
- Atenuador fijo.

Se requiere mantener el montaje de la Figura 5.1 para que se envíe una señal modulada en FM que el chip Radiometrix RX1 se encarga de demodular. Con estos elementos, junto con los del punto 5.2.1, se realiza el montaje de la Figura 5.4. En este se conecta a través de un cable coaxial rígido (línea azul) la salida del módulo Radiometrix TX1 con la entrada del módulo Radiometrix RX1 por medio de los conectores SMA montados sobre la PCB de cada uno de los módulos (Anexos B y C).

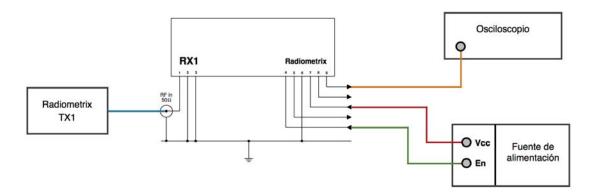


Figura 5.4: Montaje de prueba para el chip Radiometrix RX1

La prueba de verificación de este chip consiste en observar mediante el osciloscopio si sobre el pin de salida del módulo se obtiene la señal moduladora, correctamente demodulada, que entrega el módulo Radiometrix TX1 (Figura 5.5).

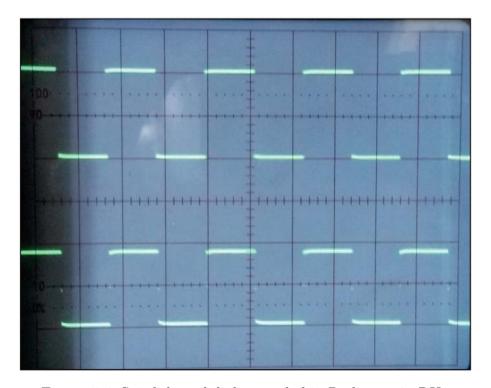


Figura 5.5: Señal demodulada por el chip Radiometrix RX1

Se observa a la salida del chip Radiometrix RX1 la señal moduladora que se utiliza para testear el conjunto transmisor-receptor en paso banda, con una amplitud algo menor a la transmitida debido a las pérdidas del cable que conecta al módulo TX1 con RX1 y algunas pérdidas relacionadas con la desadaptación de impedancias.

Por tanto, se concluye que la combinación entre los módulos paso banda es adecuada para realizar las pruebas de integración de los módulos banda base.

5.3. Pruebas sobre los módulos banda base

Se describen las pruebas individuales realizadas sobre el módem DV-MEGA GMSK (3.3) que se integra en las etapa de transmisión y recepción del prototipo hardware.

El objetivo de estas pruebas es comprobar el funcionamiento del módem GMSK y sus características técnicas, y verificar que los procesos de modulación y demodulación de datos transmitidos y recibidos por el microcontrolador Arduino en cada etapa se realiza correctamente.

Para realizar las pruebas de verificación del módem DV-MEGA GMSK, se requieren los siguientes materiales:

- Microcontrolador Arduino UNO R3 (\times 2).
- Módem DV-MEGA GMSK (×2).
- Osciloscopio multicanal.

Para ello, se realiza en montaje de la Figura 5.6. En este montaje se conectan a través de un cable la salida y la entrada Mini DIN 6 del módem transmisor y receptor, respectivamente, y se visualiza la salida de datos a través del osciloscopio (línea D3). Seguidamente, se conecta la línea de la señal de reloj del transmisor (línea D0) y las líneas de transmisión (D1) y de recepción (D2) de los datos a través de Arduino.

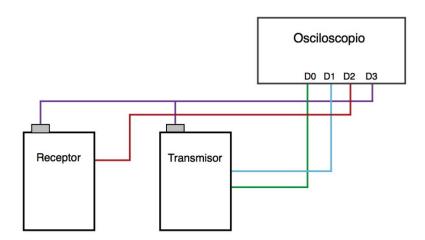


Figura 5.6: Montaje de prueba para el módem DV-MEGA GMSK

Se realizan las siguientes pruebas a partir de los diferentes códigos de test que se implementan sobre los microcontroladores Arduino:

- Transmisión y recepción de una secuencia binaria alternada.
- Transmisión y recepción de una secuencia binaria aleatoria.
- Transmisión y recepción de un dato (byte) conocido.

5.3.1. Prueba de transmisión y recepción de una secuencia binaria alternada

Una vez se realiza el montaje de la Figura 5.6, se carga sobre cada uno de los módulos de transmisión y recepción un firmware diferente para cada caso.

En el caso del transmisor, se genera una secuencia alternada de 1 y 0 lógicos a través de una rutina de servicio que se activa y desactiva cada vez que se realiza una escritura del bit sobre el pin de transmisión del módem GMSK (Figura 5.7).

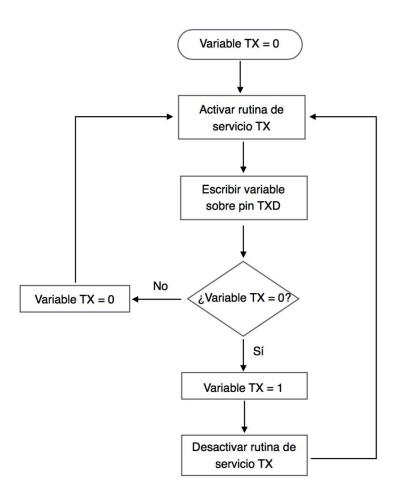


Figura 5.7: Algoritmo de prueba de transmisión de una secuencia binaria alternada

Para el receptor, se recibe la secuencia binaria alternada a través de una rutina de servicio sencilla que está activa durante toda la ejecución del programa (Figura 5.8).

En la Figura 5.9 se observa la forma de onda senoidal resultante tras el proceso de modulación GMSK, además de las secuencias binarias pertinentes del proceso de transmisión y recepción con una velocidad de transmisión de 9600 baudios. Además, se aprecia un cierto retardo entre las líneas de transmisión y recepción.

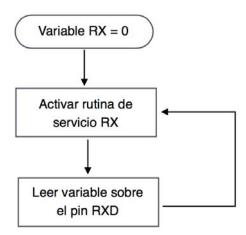


Figura 5.8: Algoritmo de prueba de recepción de una secuencia binaria alternada

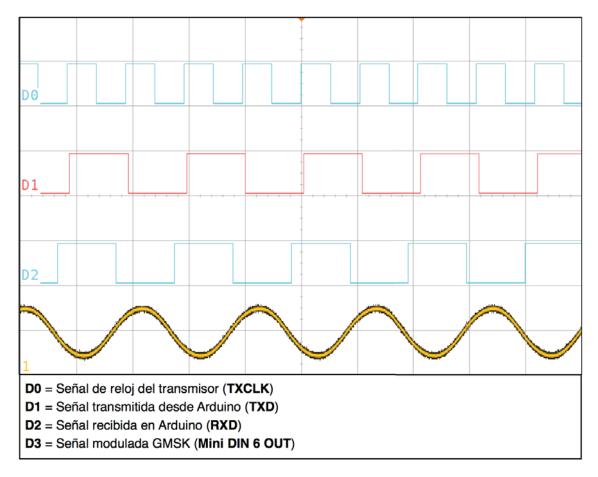


Figura 5.9: Visualización en osciloscopio multicanal de una secuencia binaria alternada en el módem DV-MEGA GMSK

5.3.2. Prueba de transmisión y recepción de una secuencia binaria aleatoria

En esta prueba, el firmware que se implementa en el transmisor genera una secuencia aleatoria de de $1 \ y \ 0$ lógicos (Figura 5.10), mientras que el del receptor no se modifica respecto a la prueba de la secuencia alternada de bits (Figura 5.8), ya que únicamente se encarga de recibir los bits que llegan al pin de recepción del módem.

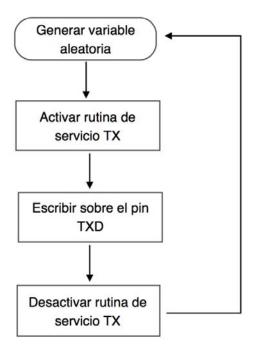


Figura 5.10: Algoritmo de prueba de transmisión de una secuencia binaria aleatoria

De esta manera, una vez cargado el firmware de transmisión, se genera una secuencia de bits aleatoria que se corresponde con la Figura 5.11, donde al igual que ocurría en la Figura 5.9, se aprecia la forma de onda de la secuencia binaria modulada en GMSK y a su vez, los datos aleatorios que se generan desde Arduino.

Con esta prueba se determina el retardo que existe entre la señal transmitida y la señal recibida, algo que es imposible de calcular si se genera una secuencia binaria alternada, como ocurre en la prueba anterior. Este retardo está calculado y señalado sobre la Figura 5.11.

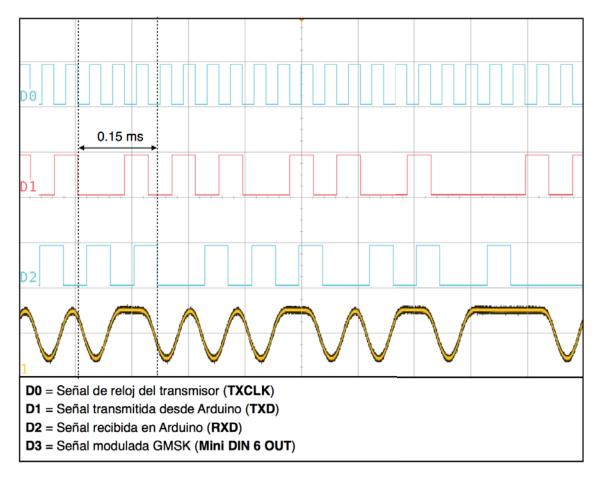


Figura 5.11: Visualización en osciloscopio multicanal de una secuencia binaria aleatoria en el módem DV-MEGA GMSK

5.3.3. Prueba de transmisión y recepción de un carácter conocido

La siguiente prueba que se describe es el envío y recepción de un carácter conocido. Se mantiene para esta prueba el montaje de la Figura 5.6, pero se modifican los algoritmos del firmware cargado sobre los microcontroladores Arduino.

El firmware para transmisión consiste en enviar un carácter conocido sucesivamente a través de Arduino, y el receptor toma cada 8 bits recibidos y los almacena en una variable para representar un byte (Figura 5.12). Para esta prueba, se elige el número decimal 52, que en binario se representa como 00110100.

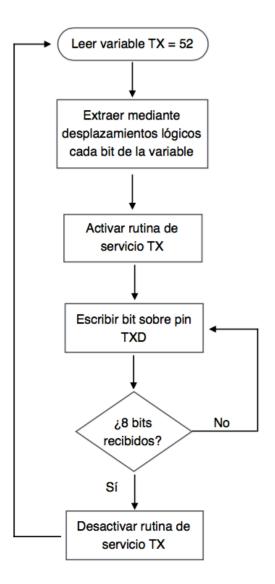


Figura 5.12: Algoritmo de prueba de transmisión de una secuencia binaria conocida

En la Figura 5.13 se observa la forma de onda de la secuencia binaria modulada en GMSK y a su vez, el dato 00110100 que se genera continuamente desde Arduino.

En esta prueba se añade la conexión por medio de USB entre Arduino y la interfaz *Serial Monitor* de Matlab 2014b (4.6.2). De esta manera, se consigue que el carácter que se recibe a través del puerto serie desde el módem GMSK aparezca sucesivamente en la pantalla de la interfaz empleada (Figura 5.14).

Por tanto, este conjunto de pruebas, y más concretamente la última prueba, verifica que es posible el envío de caracteres modulados en GMSK a través del módem DV-MEGA y su recepción y representación a través de Matlab. Se abre así la posibilidad de generar ráfagas de datos en forma de tramas, que son las pruebas que se realizan a continuación.

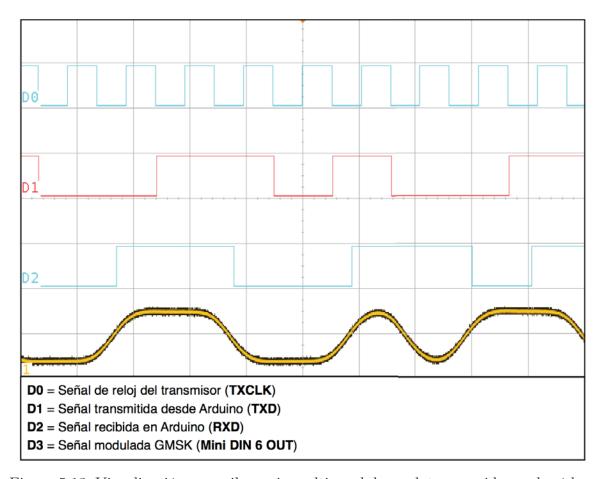


Figura 5.13: Visualización en osciloscopio multicanal de un dato conocido en el módem DV-MEGA GMSK

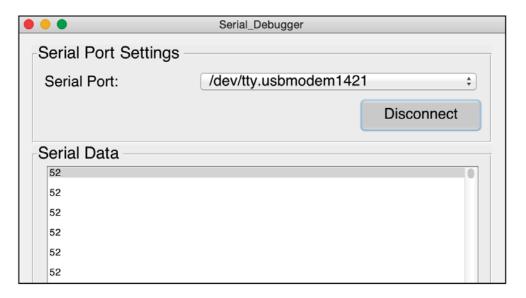


Figura 5.14: Recepción del carácter 52 desde el módem GMSK hasta Serial Monitor

5.4. Pruebas de integración de los módulos paso banda y banda base

Se describen las pruebas de integración de los módulos paso banda con los módulos banda base realizadas sobre los módulos Radiometrix TX1 y RX1 junto al módem DV-MEGA GMSK.

El objetivo de estas pruebas es comprobar el funcionamiento del módem GMSK al ser combinado con los dispositivos paso banda que se escogen para el prototipo hardware. De esta manera, se consigue integrar la modulación FM-GMSK que caracteriza al estándar AIS, dando paso a las pruebas de viabilidad del prototipo hardware en virtud del envío y recepción de tramas AIS.

Para realizar las pruebas de verificación del módem DV-MEGA GMSK, se requieren los siguientes materiales:

- Radiometrix TX1.
- Radiometrix RX1.
- Microcontrolador Arduino UNO R3 (\times 2).
- Módem DV-MEGA GMSK (×2).
- Antena VHF marítimo (×2).
- Ordenador personal.

En la Figura 5.15 se describe el montaje para estas pruebas de integración.

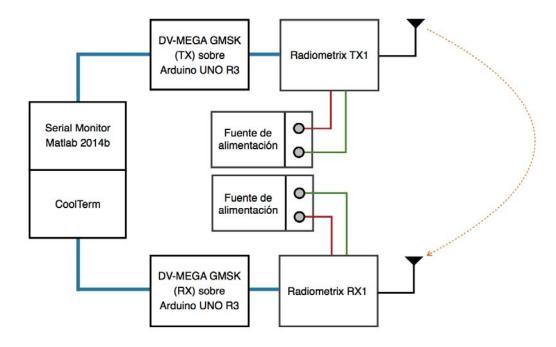


Figura 5.15: Montaje de prueba para la visualización por pantalla de tramas AIS

En la Figura 5.16 se incluye el banco de pruebas real donde se realizan las pruebas de transmisión y recepción de tramas AIS.

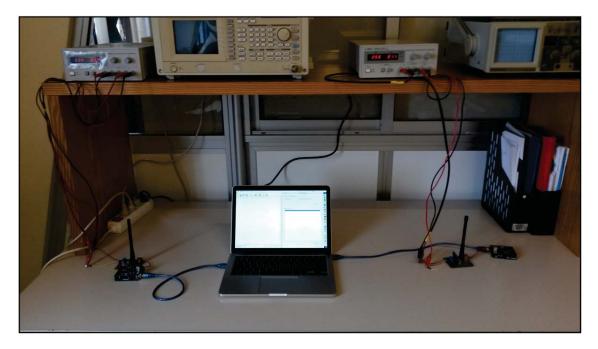


Figura 5.16: Banco de pruebas para la visualización por pantalla de tramas AIS

Es importante realizar previamente a estas pruebas la caracterización de las antenas de VHF marítimo escogidas y la medida del ancho de banda de la señal modulada en FM-GMSK cuando se realiza el envío continuo de tramas AIS.

5.4.1. Medidas sobre antenas de VHF marítimo

Las pruebas realizadas con las antenas de VHF marítimo que se describen en 3.5 es la medida del módulo del parámetro S_{11} , que permite medir el grado de reflexión que sufren las señales al ser radiada con estas antenas.

$$|S_{11}| = |\rho_{in}| = \frac{COE - 1}{COE + 1}$$
 (5.2)

Tal como indica la Ecuación 5.2, a partir de la medida del coeficiente de reflexión de entrada de cada antena, es posible calcular el COE que mide la calidad de la antena en función en lo que respecta a la potencia que puede radiar.

Para esta prueba, se realiza la medida del módulo parámetro S_{11} de las dos antenas empleadas para el prototipo hardware a través de un analizador de redes escalar previamente calibrado, y a partir de esta medida, se calcula de forma teórica el COE de ambas.

En la Figura 5.17 se observa la medida obtenida para la antena que se usa para el

prototipo del transmisor, mientras que en la Figura 5.18 se observa la misma medida, pero para la antena que se usa en recepción.



Figura 5.17: Medida del parámetro S_{11} para la antena VHF transmisora

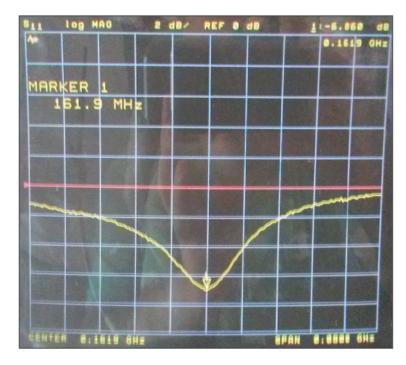


Figura 5.18: Medida del parámetro S_{11} para la antena VHF receptora

Por tanto, se obtiene que la antena VHF transmisora presenta un COE = 1.51, mientras que la receptora un COE = 1.35 según los datos recogidos en las medidas en el analizador de redes. Se concluye, pues, que ambas antenas presentan un COE alto, lo que puede suponer serios problemas durante la transmisión y recepción de tramas en el prototipo hardware.

5.4.2. Pruebas de ancho de banda para tramas AIS

En esta prueba se analiza el ancho de banda ocupado al enviar sucesivamente tramas AIS desde el prototipo hardware del transmisor. Para ello, se realiza el montaje de la Figura 5.19.

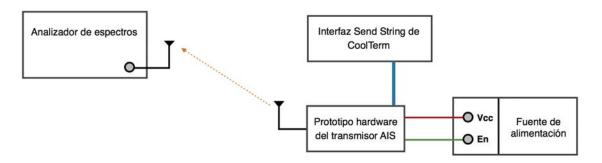


Figura 5.19: Montaje de medida del ancho de banda de transmisión de tramas AIS

Se toma el analizador de espectros como receptor y se conecta a su entrada una de las antenas de VHF marítimo empleadas para este prototipo hardware. Con ello, se implementa el firmware de control del transmisor descrito en el punto 4.4 y a través de la interfaz *CoolTerm*, se envía sucesivamente una trama AIS (Figura 5.20).

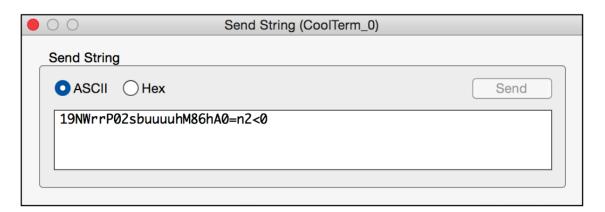


Figura 5.20: Trama AIS de test sobre la interfaz Send String de CoolTerm

En la Figura 5.21 se visualiza el espectro de la señal transmitida tras el envío sucesivo de tramas a través del prototipo hardware del transmisor, obteniendo para ello un ancho de banda de transmisión de 9.3 KHz.

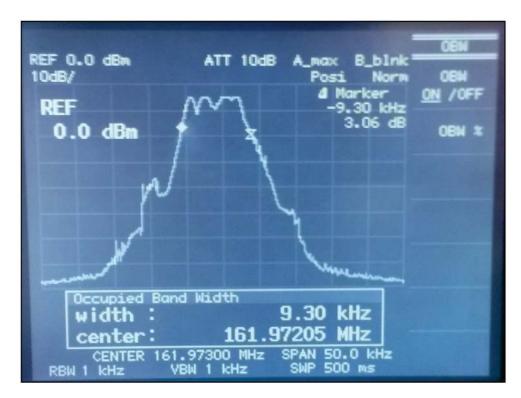


Figura 5.21: Trama AIS modulada en FM-GMSK por el prototipo hardware del transmisor

5.4.3. Prueba de envío y recepción de tramas AIS

En esta prueba se describe el proceso de envío y recepción de tramas AIS, utilizando para ello el prototipo hardware implementado, tal y como se ha hecho en las pruebas anteriores.

Para ello, se realiza el montaje de la Figura 5.22. En este montaje, se emplea *CoolTerm* para la transmisión y recepción de tramas AIS, puesto que esta interfaz ofrece más facilidades para detectar posibles errores durante la comunicación entre los dos sistemas del banco de pruebas.

Para ello, se implementa sobre Arduino cada uno de los códigos del firmware de control del transmisor y del receptor recogidos en los puntos 4.4 y 4.5, respectivamente.

El transmisor genera desde Arduino el formato de la trama AIS, tomando los datos que recibe por el puerto serie desde la interfaz transmisora *CoolTerm*, presentándose además por pantalla. Seguidamente, estos datos se modulan en GMSK y posteriormente en FM, y se envían por la antena transmisora de VHF.

El receptor, por otra parte, recibe la trama AIS que se envía desde el transmisor por la antena receptora, y extrae de ella únicamente el campo de datos. De esta manera, los datos extraídos se envían por el puerto serie hasta la interfaz receptora *CoolTerm*, que los representa por pantalla.

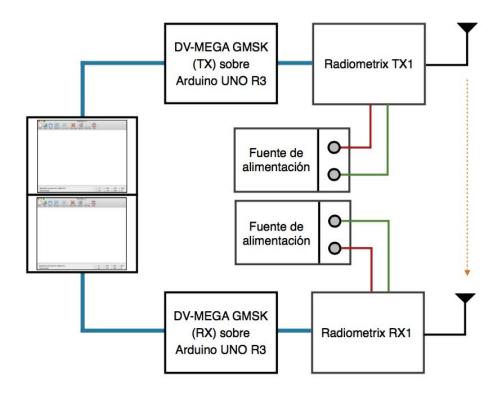


Figura 5.22: Trama AIS modulada en FM-GMSK por el prototipo hardware transmisor

En la Figura 5.23 aparece el proceso de transmisión y recepción de una misma secuencia de datos, que se corresponde con las 23 primeras letras del abecedario.

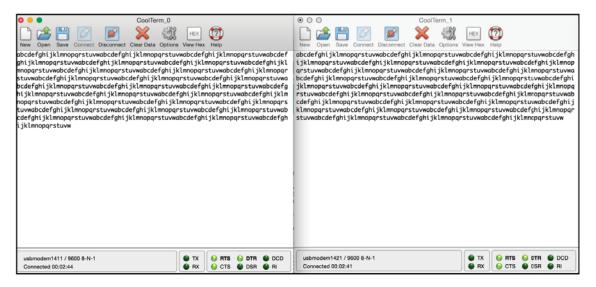


Figura 5.23: Envío de la trama AIS durante 30 iteraciones (*CoolTerm0* como transmisor, *CoolTerm1* como receptor)

Se comprueba que la trama de datos enviada se recibe correctamente, por lo que el diseño del prototipo hardware y el firmware de control implementado es correcto. Esta prueba se ha repetido 10 veces, y el porcentaje de acierto se mantiene en un $100\,\%$.

5.5. Prueba real de recepción de tramas AIS con el prototipo hardware

El objetivo de esta última prueba es evaluar si, conforme a las pruebas realizadas anteriormente, es posible recibir tramas AIS de las embarcaciones que se encuentran en la misma región de cobertura del dispositivo y que integran este servicio.

Para ello, se realiza el montaje de la Figura 5.24, mientras que en la Figura 5.25 se observa el montaje realizado en el laboratorio.

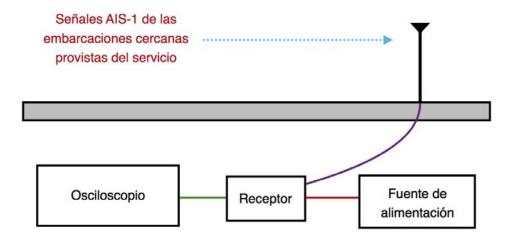


Figura 5.24: Montaje del banco de pruebas de recepción de tramas AIS reales



Figura 5.25: Montaje en laboratorio del banco de pruebas de recepción de tramas AIS reales

El montaje de la Figura 5.25 se realiza a través de una antena de VHF marítimo instalada en la azotea del Pabellón B de la Escuela de Ingeniería Electrónica y Telecomunicación (EITE) de la Universidad de Las Palmas de Gran Canaria (ULPGC). A través de un cable coaxial que va desde la salida de la antena hasta el prototipo hardware del receptor, se capturan los datos que llegan a la misma y sufren el proceso de demodulación por parte de este prototipo.

La visualización de la banda de trabajo de AIS que recibe la antena de VHF marítimo de este montaje se encuentra en la Figura 5.26. Esta antena se conecta directamente al analizador de espectros y se selecciona un *SPAN* de 100 KHz, centrando la banda en la frecuencia 161.975 MHz. Esta medida permite determinar que el dispositivo Radiometrix RX1 realiza correctamente la selección del canal AIS-1, ya que el espectro de esta señal se encuentra dentro de la canalización de 25 KHz.

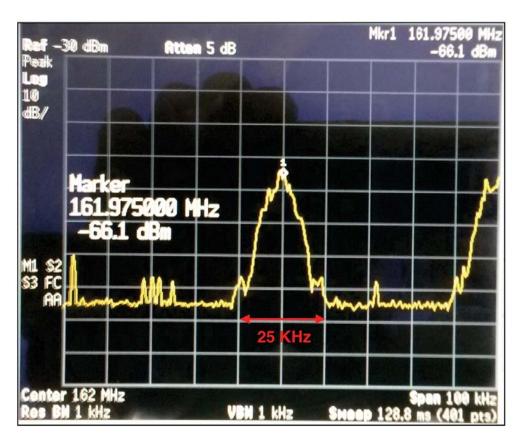


Figura 5.26: Banda de trabajo AIS a la frecuencia 161.975 MHz

En la Figura 5.28 se visualiza a través del osciloscopio una parte de las tramas AIS donde se remarca con la utilidad CURSORS un intervalo temporal de 26.67 ms, correspondiente al slot de una trama. Se comprueba la presencia de datos en toda la trama.

Se comprueba que el proceso de demodulación de las tramas se realiza correctamente. De esta manera, se determina una tasa binaria de 9.6 kbps, medida en el osciloscopio, que corresponde a un bit (104 μs) (Figura 5.27).

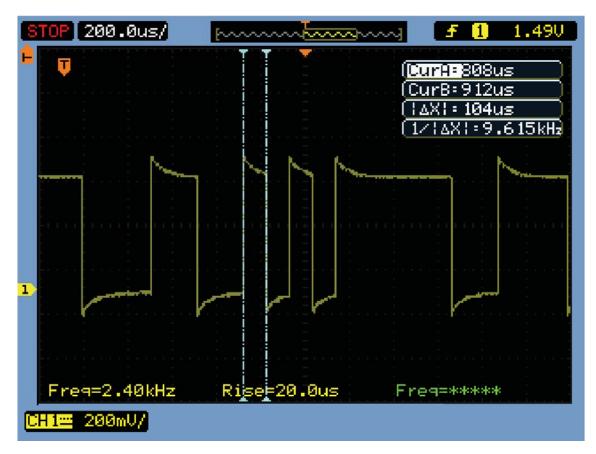


Figura 5.27: Forma de onda de la trama AIS recibida

El prototipo hardware del receptor decodifica las tramas AIS, pero no se ha implementado el proceso de decodificación del campo de datos. Por tanto, los datos que se reciben y que se representan a través de la interfaz de Matlab no aportan información sobre el servicio.

Por último, una vez comprobada que la demodulación es correcta, se verifica la decodificación de las tramas AIS. Esta comprobación se realiza a través de la interfaz serie de Matlab, siendo los resultados de esta prueba satisfactorios, puesto que el prototipo hardware envía información a la interfaz serie. Sin embargo, no se ha implementado la decodificación del campo de datos de las tramas, y por tanto, la visualización de las mismas a través de la interfaz serie no es interpretable.

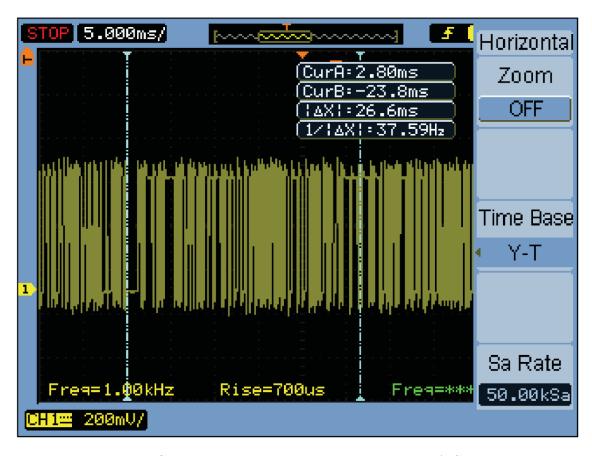


Figura 5.28: Comprobación de los slots en las tramas AIS recibidas

Capítulo 6

Conclusiones

6.1. Resultados y Revisión de Objetivos

La línea de trabajo de este TFG está orientada a la evaluación del estándar AIS a partir de un prototipo hardware modular basado en la gama de microcontroladores Arduino. A través de este prototipo, se evalúa el Nivel Físico y de Enlace que sigue la arquitectura de este estándar, extrayendo diferentes conclusiones acerca del mismo.

Este trabajo se compone de una etapa previa de documentación sobre el estándar AIS estrictamente necesaria, donde se entra en detalle acerca de las características que debe cumplir este estándar. Esto permite la evaluación y posterior elección de los módulos que integran el prototipo hardware para el banco de pruebas.

A través del diseño de un firmware de control para un transmisor y un receptor que operan en la frecuencia AIS-1 (161.976 MHz), se consigue evaluar características propias de la modulación FM-GMSK que sigue este estándar, además de la codificación NRZI, y se cubre completamente el estudio del Nivel Físico en AIS.

Es importante destacar la capacidad del chip Radiometrix RX1 para demodular las tramas AIS del servicio establecido, aunque la decodificación del campo de datos y su posterior procesado no es objeto de este TFG. Sin embargo, con las pruebas realizadas al respecto, se contempla la viabilidad del prototipo hardware implementado en cuanto a las características de recepción.

En lo que respecta a la codificación NRZI, se comprueba la necesidad de la misma en el proceso de transmisión y de recepción. Esta codificación se implementa por software en el propio firmware de control, aunque en los modelos comerciales suele venir dada, de forma más eficiente, como una solución hardware [22].

También se consigue evaluar la capacidad del microcontrolador Arduino UNO R3 como núcleo para este banco de pruebas. Realizando el firmware de control del prototipo hardware a través de interrupciones, se ha comprobado la viabilidad de un diseño modular para este estándar.

Los resultados de este TFG han sido publicados en 1st Radio Science Conferen-

ce (URSI-AT-RASC), celebrada entre los días 18 y 25 de mayo de 2015 en Maspalomas (Gran Canaria), bajo el título *Design of a Low Cost Prototype of Automa6c Iden6fica6on System (AIS) Receiver* (Anexo F).

6.2. Líneas Futuras

En este apartado se evalúan las posibles líneas de trabajo que toman como base los resultados obtenidos en este Trabajo Fin de Grado.

Una posible línea de trabajo que se abre a partir de los resultados obtenidos es el desarrollo de los elementos necesarios para permitir que el prototipo hardware actual, que realiza la comunicación entre transmisor y receptor en modo símplex, pueda comunicarse en los modos half-dúplex y/o full-dúplex. Para esta línea de trabajo, es necesario entrar en detalle en el funcionamiento del circuito PTT incluido en el módem DV-MEGA GMSK y modificar las rutinas de servicio de cada sistema para así desarrollar un único firmware de control destinado a operar tanto en recepción como en transmisión. Se debe modificar, además, el diseño de los módulos paso banda para que operen sobre una misma PCB.

Otra posible línea de trabajo es integrar la decodificación de mensajes AIS sobre PC o Raspberry Pi. En este trabajo, se ha conseguido establecer correctamente el enlace de comunicación entre los dispositivos transmisor y receptor a partir de la plataforma Arduino, manteniendo el formato de las tramas según se indica en el estándar AIS. Sin embargo, los datos no incluyen el formato NMEA que siguen las tramas, ni tampoco se contempla la interpretación de los datos por parte del sistema. Se propone integrar un dispositivo GPS sobre el prototipo hardware diseñado para que pueda sincronizarse con el resto de estaciones AIS activas y poder transmitir mensajes AIS, tal y como hacen los dispositivos comerciales [8].

A partir de los resultados obtenidos acerca de los parámetros radio de cada módulo que integra el prototipo hardware, cabe la posibilidad de estudiar diferentes topologías de amplificación que al integrarse sobre el prototipo hardware, permitan realizar estudios sobre cobertura que mejoren el diseño hardware existente.

El desarrollo del firmware de control del banco de pruebas se encuentra actualmente en una fase preliminar. Se propone como línea de trabajo futura la mejora de este firmware de control con el fin de optimizar el código e incluir las técnicas de control de errores (CRC-16) sobre Arduino, evitando así que el control de errores se lleve a cabo en otros dispositivos, como puede ser un PC o una Raspberry Pi, mejorando así la eficiencia del sistema implementado.

Por último, el sistema AIS puede permitir un inventario de emisiones que permitiría identificar perfiles operativos y contaminantes en los puertos debido a la emisión aérea de contaminantes por parte de los barcos, y en función de su tipo de operación (hotelling, maniobras de atraque...) [23].

Bibliografía

- [1] Guerci, A. (2009). Red de datos AIS y Mareógrafos. Buenos Aires (Argentina), Fl.: Universidad de Belgrano.
- [2] Mathapo, K. F. (2007). A software-defined radio implementation of maritime AIS (Doctoral dissertation, Stellenbosch: University of Stellenbosch).
- [3] Patel, P. R. (2013). Implementation of a Single Channel Automatic Identification System (AIS) on a Field Programmable Gate Array (FPGA) (Doctoral dissertation, University of Dayton).
- [4] CMSA (2008). GMDSS equipment operation assessment standards.
- [5] Maggio, F., Rossi, T., Cianca, E., and Ruggieri, M. (2014). Digital beamforming techniques applied to satellite-based AIS receiver. *Aerospace and Electronic Systems Magazine*, *IEEE*, 29(6), 4-12.
- [6] Captain/Senior Pilot Benny Pettersson. (2004). Automatic Identification System (AIS) for ships in the future. Swedish Maritime Administration. Retrieved from http://www.sjofartsverket.se.
- [7] Harchowdhury, A., Bandyopadhyay, K., Sarkar, B. K., and Bhattacharya, A. (2012). Generalized Mechanism of SOTDMA and Probability of Reception for Satellite-based AIS. In *Computers and Devices for Communication (CODEC)*, 2012 5th International Conference on, (pp. 1-4). IEEE.
- [8] Lanier, F. (2012). Feature Loaded High-end VHFs. ICOM 2012.
- [9] Recommendation ITU-R M.1371-5 (2014). Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band. M Series: *Mobile*, radiodetermination, amateur and related satellite services. IEEE Publications.
- [10] Hasegawa, K., Hata, K., Niwa, K., and Fukuto, J. (2008). Transmission evaluation of Ship-borne Automatic Identification System (AIS) in congested waterways. In ITS Telecommunications, 2008. ITST 2008. 8th International Conference on (pp. 18-23). IEEE.
- [11] Recommendation ITU-R M.M.1084-5 (2012). Interim solutions for improved efficiency in the use of the band 156-174 MHz by stations in the maritime mobile service. M Series: *Mobile*, radiodetermination, amateur and related satellite services. *IEEE Publications*.

98 BIBLIOGRAFÍA

[12] Bernard, S. (2001). Digital communications fundamentals and applications. *Prentice Hall, USA*.

- [13] de Dieuleveult, F. (2000). Electrónica aplicada a las altas frecuencias. Thomson-Paraninfo.
- [14] Souissi, M. G., Grati, K., Ghazel, A., and Kouki, A. (2008). Software efficient implementation of GMSK modem for an automatic identification system transceiver. In *Electrical and Computer Engineering*, 2008. CCECE 2008. Canadian Conference on (pp. 000601-000606). IEEE.
- [15] Murota, K., and Hirade, K. (1981). GMSK modulation for digital mobile radio telephony. *Communications*, *IEEE Transactions on*, 29(7), 1044-1050.
- [16] All about AIS: TDMA schemes. (n.d.). Retrieved March, 4, 2015, from http://www.allaboutais.com.
- [17] Song, J. H., Oh, K. R., Kim, I. K., and Lee, J. Y. (2010). Application of maritime AIS (Automatic Identification System) to ADS-B (Automatic Dependent Surveillance? Broadcast) transceiver. In *Control Automation and Systems (ICCAS)*, 2010 International Conference on (pp. 2233-2237). IEEE.
- [18] NMEA 0183 version 3.01 (2002). Standard For Interfacing Marine Electronic Devices, from *International Electrotechnical Commission*.
- [19] Xing, J., Lou, S., Zhang, D., Luo, X., and Zhang, J. (2008). Research and integration of marine embedded AIS information terminal. In *Intelligent Control and Automation*, 2008. WCICA 2008. 7th World Congress on (pp. 3586-3589). IEEE.
- [20] Pérez, M. S., Iragüen, B. G., Jambrina, J. L. F., and Castañer, M. S. (2003). *Electrónica de comunicaciones*. Pearson Educación.
- [21] Getting Started with Arduino. (n.d.). Retrieved July, 7, 2015, from https://www.arduino.cc.
- [22] CML Microcircuits: Marine Comms. (n.d.). Retrieved May, 10, 2015, from http://www.cmlmicro.com/products/
- [23] Tichavska, M., and Tovar, B. (2014). Port-city Exhaust Emission Model: an approach to Cruise and Ferry operations in Las Palmas Port. In *IAME Conference in Virginia*, *USA*.

Parte II

Presupuesto

Presupuesto

P.1. Introducción

En este capítulo se incluye el presupuesto total de este Trabajo de Fin de Grado. Este presupuesto se divide en cinco partes:

- Trabajo tarifado por tiempo empleado.
- Amortización del inmovilizado material.
 - Amortización de los recursos hardware.
 - Amortización de los recursos software.
- Redacción de la documentación.
- Derechos de visado del COITT.
- Gastos de tramitación y envío.

Una vez analizados los gastos de cada uno de los puntos anteriores, se aplican los impuestos vigentes y se establece el coste total de este TFG.

P.2. Trabajo tarifado por tiempo empleado

En esta sección, se contabilizan los gastos de mano de obra a partir del salario que corresponde a la hora de trabajo de un Ingeniero Técnico empleando la fórmula (P.1):

$$H = 14,48 \times H_n + 20,27 \times He \in$$
 (P.1)

donde:

- H son los honorarios totales por tiempo empleado.
- H_n son las horas normales trabajadas durante la jornada laboral.

• H_e son las horas especiales trabajadas durante la jornada laboral.

Se estima que para la elaboración de este TFG se ha trabajado durante 5 meses, desde **febrero de 2015** hasta **junio de 2015**, con una media de 3 horas diarias. Considerando que en un mes hay 20 días laborales, el número de horas normales trabajadas asciende a la siguiente cifra:

$$H_n = 5 \times 20 \times 3 = 300 \text{ horas}$$
 (P.2)

Se asume que $H_e = 0$ debido a que no se han trabajado horas fuera de la jornada laboral establecida, de manera que el coste total de los honorarios es:

$$H = 14,48 \times 300 + 20,27 \times 0 = 4.344,00 \in (P.3)$$

Por tanto, el trabajo tarifado por tiempo empleado asciende a la cantidad de *cuatro* mil trescientos cuarenta y cuatro euros.

P.3. Amortización del inmovilizado material

En el inmovilizado material se consideran tanto los recursos hardware como software empleados para la realización de este TFG. Se analizan de forma independiente.

Se estipula el coste de amortización para un periodo de 3 años, utilizando un sistema de amortización lineal donde se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula haciendo uso de la ecuación (P.4):

$$Cuota anual = \frac{Valor de adquisición - Valor residual}{Número de años de vida útil}$$
(P.4)

donde el valor de adquisición se refiere al coste unitario de cada recurso y el valor residual es el valor teórico que se supone que presenta el elemento evaluado tras cumplir su vida útil.

P.3.1. Amortización de los recursos hardware

Al ser la duración de este TFG de 5 meses, y esta es una cantidad inferior a los 3 años estipulados para la amortización de los elementos evaluados, los costes que se evalúan son los que corresponden a los 5 primeros meses.

En las siguientes tablas se muestran los recursos hardware empleados para la elaboración de este TFG, incluyendo para cada recurso su valor de adquisición, valor residual y coste de amortización, estableciendo un tiempo de uso de 5 meses. En lo que respecta

al valor residual, se estima que su valor es un 60/% del coste unitario del material, independientemente del recurso evaluado.

Es importante hacer una escisión entre recursos hardware fungibles (Tabla P.1) y recursos hardware no fungibles (Tabla P.2), entendiendo por fungible aquellos materiales que son consumidos al uso y no admiten reutilización, y por tanto, no se les aplica una amortización.

Concepto	Número de unidades	Coste unitario (€)	
Arduino UNO R3	2	24,29	
Radiometrix TX1	1	20,6	
Radiometrix RX1	1	29,71	
Shield DV-MEGA GMSK	2	60	
Antena VHF marítimo	2	12,5	
Placa PCB	2	1,5	
Resistencia SMD	7	0,01	
Condensador SMD	3	0,05	
Conector Mini DIN 6	2	0,45	
Conector SMA	2	0,05	
Clema PCB	2	0,05	
Cable USB tipo A/B	2	2,35	
Coste total		252.91 €	

Tabla P.1: Tabla de precios del material hardware fungible

Concepto	Coste unitario (€)	Coste residual (€)
MacBook Pro 13.3" Retina Intel i5	1.649,00	989,4
Osciloscopio digital Agilent DSO1002A	996,00	597,6
Analizador espectral Advantest R1331A	3.970,00	2.382,00
Generador de funciones Agilent 33250A	1.567,00	940,20
Fuente de alimentación Agilent E3620A	610,30	366,18
Multímetro digital FLUKE 77III	213,89	128,33
Sonda con punta de prueba Tektronix	7,65	4,59
Cables de prototipado	2,45	1,47
Coste total	9.016,29 €	5.409,77 €

Tabla P.2: Tabla de precios del material hardware no fungible

El coste total de los recursos hardware empleados, sumando las cantidades del material hardware fungible y no fungible, asciende a cinco mil seiscientos sesenta y dos euros con sesenta y ocho céntimos.

P.3.2. Amortización de los recursos software

De forma análoga a los recursos hardware no fungibles, se estiman los costes de los recursos software relativos a los 5 primeros meses, que se corresponden con el periodo

de realización de este TFG.

En la Tabla P.3 se muestran los recursos software empleados para la elaboración de este TFG, incluyendo para cada recurso su valor de adquisición, valor residual y coste de amortización, y estableciendo un tiempo de uso de 5 meses.

Concepto	Coste unitario (€)	Coste residual (€)	
MATLAB 2014b (UNIX)	2.000,00	1.200,00	
IDE Arduino 1.6	0,00	0,00	
CoolTerm	0,00	0,00	
CadSoft Eagle PCB Design 7.2	0,00	0,00	
Microsoft Office 2011	119,00	71,4	
Editor LaTeX TexMaker	0,00	0,00	
Fritzing	0,00	0,00	
Coste total	2.119,00 €	1.271,4 €	

Tabla P.3: Tabla de precios del material software

El coste total de los recursos software empleados asciende a mil doscientos setenta y un euros con cuarenta céntimos.

P.4. Redacción de la documentación

Se emplea la fórmula (P.5) para establecer el coste asociado a la redacción de la documentación relativa a la memoria de este TFG:

$$R = 0.05 \times P \tag{P.5}$$

donde:

- R son los honorarios por la redacción de la memoria.
- \blacksquare P es el presupuesto.

El valor del presupuesto P se calcula como la suma de los costes de las secciones anteriores correspondientes al trabajo tarifado por tiempo empleado y a la amortización del inmovilizado material (tanto para recursos hardware como software), es decir:

$$P = H + C_h + C_s \tag{P.6}$$

donde:

• H son los honorarios totales por tiempo empleado.

- C_h es el coste total de los recursos hardware.
- C_s es el coste total de los recursos software.

Esta suma total se refleja en la Tabla P.4.

Concepto	Coste (€)
Trabajo tarifado por tiempo empleado	4.344,00
Amortización del material hardware	5.662,68
Amortización del material software	1.271,4
Coste total	11.278,08 €

Tabla P.4: Presupuesto según el trabajo tarifado y la amortización del inmovilizado material

Por tanto, los honorarios por la redacción de la memoria de este TFG son:

$$R = 0.05 \times 11.278, 08 = 563.9 \in (P.7)$$

El coste total de la redacción de la memoria asciende a quinientos sesenta y tres euros con noventa céntimos.

P.5. Derechos de visado del COITT

El COITT establece que para la redacción de trabajos en Integración de Sistemas, Equipos y Aplicaciones Electrónicas y Redes y Aplicaciones Telemáticas, los derechos de visado para 2015 se calculan en base a la fórmula (P.8):

$$V = 0,0035 \times P \times C \tag{P.8}$$

donde:

- V es el coste de visado del trabajo.
- \blacksquare P es el presupuesto.
- C es el es el coeficiente reductor en función del presupuesto.

El valor del presupuesto P se calcula sumando los costes de las secciones anteriores, que se corresponden con el trabajo tarifado por tiempo empleado, la amortización del inmovilizado material (tanto hardware como software) y la redacción de la memoria, es decir:

$$P = H + C_h + C_s + R \tag{P.9}$$

Esta suma total se refleja en la Tabla P.5.

Concepto	Coste (€)
Trabajo tarifado por tiempo empleado	4.344,00
Amortización del material hardware	$5.662,\!68$
Amortización del material software	1.271,4
Redacción de la memoria	563,9
Coste total	11.841,98 €

Tabla P.5: Presupuesto incluyendo la redacción de la memoria

En este caso, el coeficiente reductor C es el asignado para un presupuesto de hasta $30.050 \in$, cuyo valor es C = 1. Por tanto, el coste de visado del trabajo es:

$$V = 0,0035 \times P \times C = 0,0035 \times 11.841,98 \times 1 = 41,45 \in (P.10)$$

El coste total del visado del trabajo asciende a cuarenta y un euros con cuarenta y cinco céntimos.

P.6. Gastos de tramitación y envío

Los gastos de tramitación y envío se estipulan en seis euros $(6 \in)$ por cada documento visado de forma telemática.

P.7. Aplicación de impuestos y coste total

La realización del presente TFG está gravada por el Impuesto General Indirecto Canario (I.G.I.C.), en un *siete por ciento* (7%). De esta manera, en la Tabla P.6 se presenta el presupuesto final sobre el que se aplican estos impuestos.

Concepto	Coste (€)
Trabajo tarifado por tiempo empleado	4.344,00
Amortización del material hardware	5.662,68
Amortización del material software	1.271,4
Redacción de la memoria	563,9
Derechos de visado del COITT	41,45
Gastos de tramitación y envío	6,00
Coste total	11.889,43 €

Tabla P.6: Presupuesto final del TFG

Por lo tanto, el coste total de este Trabajo de Fin de Grado con título "Diseño e implementación de un prototipo hardware para un banco de pruebas del estándar AIS" asciende a once mil ochocientos ochenta y nueve euros con cuarenta y tres céntimos.

El graduado

Fdo.: D. Nicolás Molina Padrón

En Las Palmas de Gran Canaria a 15 de Julio de 2015

Parte III

Anexos

Apéndice A

Descripción del chip CMX589A

El chip CMX589A es un encapsulado de 24 pines diseñado por la empresa CML Microcircuits que se describe como un módem GMSK banda base que puede operar siguiendo tanto un esquema de comunicación full-dúplex como half-dúplex . Este chip permite velocidades de transmisión comprendidas entre $4\ kbps$ y $200\ kbps$, y presenta un pin configurable para seleccionar el factor BT deseado (que puede ser $0.3\ o\ 0.5$).

Entre los usos más habituales del chip CMX589A, se encuentran las aplicaciones de transmisión y recepción de datos vía inalámbrica, principalmente en telefonía móvil, donde la modulación GMSK suele estar presente en los principales servicios, como es el caso de GSM. También suele ser utilizado en la recepción de datos GPS y DGPS y en los módulos de transmisión y recepción de ordenadores portátiles e impresoras.

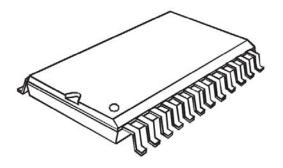


Figura A.1: Encapsulado del chip CMX589A

La función principal de este chip es realizar una serie de filtrados sobre las señales de entrada y de salida siguiendo los esquemas de una modulación GMSK. En esencia, presenta pines de transmisión y recepción de datos binarios, y para asegurar un correcto funcionamiento de la comunicación, estos pines deben estar sincronizados con sus respectivas señales de reloj.

Además, este chip requiere circuitos externos para completar su funcionamiento y poder realizar aplicaciones como las que se han mencionado anteriormente, ya sean unidades de amplificación como filtros y moduladores paso banda.

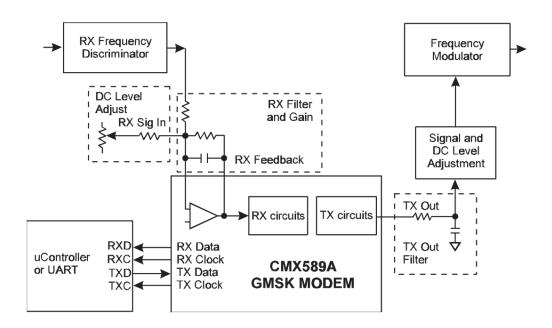


Figura A.2: Subsistemas integrados al chip CMX589A

Tal y como se observa en este diagrama de bloques, los pines principales con los que opera el CMX589A son los pines de transmisión y recepción de datos (TX Data y RX Data) y sus correspondientes pines de sincronismo (TX Clock y RX Clock), que van conectados directamente a la UART o a un microcontrolador con el que se pretende comunicarse para realizar la transferencia de datos.

A este sistema se le suman algunos subsistemas relativos a la ganancia y al filtrado de la señal. Debido a que el CMX589A no puede asegurar que a su salida se entregue el nivel de señal que se requiere para una determinada aplicación, o bien no es capaz de atenuar y filtrar la señal que le llega a su entrada, emplea diversos mecanismos de amplificación y filtrado para ajustar los niveles de señal y rechazar frecuencias indeseables a la entrada y a la salida del chip.

Por último, recordando que el chip CMX589A se define como módem GMSK en banda base, cabe señalar la incorporación de dos elementos indispensables para realizar comunicaciones paso banda: discriminador de frecuencia (para la etapa de recepción) y modulador de frecuencia (para la etapa de transmisión).

Pines del chip CMX589A

En este apartado se describe la función de cada uno de los pines que conforman el encapsulado del chip CMX589A, lo que permite detallar la funcionalidad de los subsistemas integrados internamente en este encapsulado.

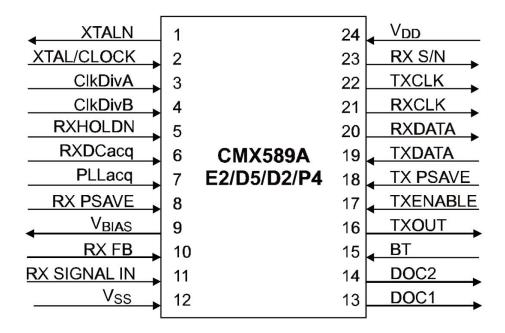


Figura A.3: Diagrama de pines del chip CMX589A

- XTALN: salida del oscilador a cristal interno.
- XTAL/CLOCK: entrada del oscilador a cristal interno cuando se necesite utilizar una señal de reloj externa.
- ClkDivA, ClkDivB: controles de niveles de señal lógicos del divisor de reloj interno (ver TABLA 4).
- RxHOLDN: si se pone un 0 lógico en esta entrada, se congelan los circuitos de extracción de la señal de reloj y medición de nivel a menos que se encuentren en modo Acquire.
- RxDacq: si se pone un 1 lógico en esta entrada, se introducen los circuitos de medición de nivel del Rx para el modo Acquire (ver TABLA 6).
- PLLacq: si se coloca un 1 lógico en esta entrada, se introducen los circuitos de extracción de la señal de reloj y medición para el modo Acquire (ver TABLA 5).
- Rx PSAVE: si se coloca un 1 lógico en esta entrada, todos los circuitos de recepción se ponen en modo ahorro de energía excepto para la salida Rx CLK (que sigue funcionando a la velocidad de transmisión) y hace que aparezca un 0 lógico en las salidas Rx Data y RX S/N.

- V_{bias} : alimentación interna del dispositivo; obliga a que se coloque un condensador en su pin de entrada.
- Rx FB: salida del amplificador Rx de la entrada.
- Rx Signal In: entrada del amplificador Rx de la entrada.
- V_{SS} : alimentación negativa (GND).
- DOC1: conexiones para los circuitos de medida de nivel de Rx; obliga a conectar un condensador desde este pin hasta V_{SS} .
- DOC2: conexiones para los circuitos de medida de nivel de Rx; obliga a conectar un condensador desde este pin hasta V_{SS} .
- BT: selecciona el factor $B \cdot T$ del módem; si se pone un 1 lógico se selecciona $B \cdot T = 0.5$ y si se pone un 0 lógico se selecciona $B \cdot T = 0.3$.
- Tx Out: señal Tx filtrada a través de un filtro gaussiano. Cuando se encuentra en modo ahorro de energía, este pin se encuentra en alta impedancia.
- Tx Enable: si se coloca un 1 lógico en esta entrada, habilita la ruta de transmisión de datos, a través del filtro de transmisión hasta el pin Tx Out. Un 0 lógico aplicado sobre este pin coloca el pin Tx Out en alta impedancia.
- Tx PSAVE: si se coloca un 1 lógico en esta entrada, todos los circuitos de Tx se pondrán en modo ahorro de energía excepto para el Tx CLK.
- Tx Data: nivel lógico de entrada del dato que va a ser transmitido; este dato debe estar sincronizado con Tx CLK.
- Rx Data: nivel lógico de salida del dato que es recibido; este dato debe estar sincronizado con Rx CLK.
- Rx CLK: nivel lógico de salida de reloj de la salida para la tasa binaria de recepción.
- Tx CLK: nivel lógico de salida de reloj de la salida para la tasa binaria de transmisión.
- Rx S/N: nivel lógico de salida que puede ser utilizado como indicador de la calidad de la señal recibida.
- V_{DD} : alimentación positiva; obliga a a conectar un condensador externo.

Circuitos externos del chip CMX589A

En este apartado se describe el diseño de los circuitos externos al chip CMX589A, imprescindibles para la realización de aplicaciones prácticas. De esta forma, se detallan los siguientes puntos:

- Circuitos de reloj.
- Selección del factor BT y tasa binaria.
- Filtrado y ganancia en transmisión.
- Filtrado y ganancia en recepción.

Circuitos de reloj

Hay dos formas posibles para implementar los circuitos de reloj en el chip CMX589A: sincronismo local y sincronismo externo.

En lo que respecta al **sincronismo local**, se debe incluir un oscilador a cristal sobre los pines XTALN y XTAL/CLOCK que siga una topología similar a la que vemos a continuación:

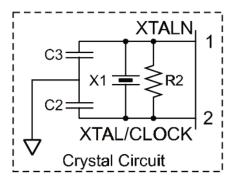


Figura A.4: Topología de sincronismo local con oscilador XTAL

Para esta topología, se escoge un valor fijo de $R_2 = 1 M\Omega$, independientemente de la frecuencia de oscilación. Sin embargo, la elección de los condensadores C_2 y C_3 va en función de la frecuencia de oscilación escogida para X1, además de la alimentación V_{DD} .

Como una primera estimación orientativa, si $C_2 = C_3 = 33 \ pF$ sobre una frecuencia de oscilación de 1 MHz, se presenta una caída de 18 pF sobre la máxima frecuencia posible. Si $V_{DD} = 3 \ V$ y se asumen los mismos valores de capacitancia anteriores, se observa una caída de 18 pF sobre 5 MHz. Cabe señalar que la resistencia equivalente de X1 debe cumplir que $R_{eq} < 2 \ k\Omega$, con una caída de 150 Ω sobre la máxima frecuencia posible.

En caso de emplear **sincronismo externo**, se proponen dos formas de conseguir el sincronismo, que van en función de la frecuencia de oscilación que se inyecta externamente.

■ EXT CLK > 10.24 MHz: se escoge un valor fijo de $R_2 = 1 M\Omega$. El valor de C_9 no es muy crítico, ya que suele ser utilizado como bloqueador de DC.

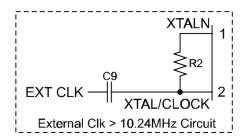


Figura A.5: Topología de sincronismo externo para EXTCLK > 10.24 MHz

■ EXT CLK ≤ 10.24 MHz: se escoge un valor fijo de $R_2 = 1~M\Omega$. El valor de C_9 no es muy crítico, ya que suele ser utilizado como bloqueador de DC.

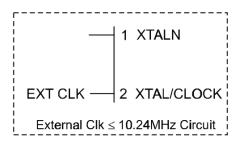


Figura A.6: Topología de sincronismo externo para $EXTCLK \leq 10.24 \text{ MHz}$

Selección de la tasa binaria

La tasa binaria del dispositivo CMX589A se determina a partir del pin XTALN, que presenta una conexión con divisores de frecuencia capaces de reducir la frecuencia de oscilación que ha sido generada por un oscilador a cristal interno del CMX589A (on-chip) o externamente a través de los circuitos de sincronismo de reloj estudiada en el punto anterior.

El factor de división de estos divisores de frecuencia vienen controlados por los niveles lógicos que se introducen sobre los pines de entrada sobre ClkDivA y ClkDivB. La ecuación que maneja la relación entre la tasa binaria y la frecuencia de oscilación es la siguiente:

$$R_B = \frac{f_{xtal}}{N} \tag{A.1}$$

siendo f_{xtal} la frecuencia de reloj que ha sido generada interna o externamente el chip CMX589A y N el factor de división de los divisores de frecuencia.

En la siguiente tabla, se recogen los ejemplos más usuales de tasas binarias conseguidas en relación a los cuatro posibles estados binarios que presentan los pines ClkDivA y ClkDivB y a la frecuencia de reloj que llega a los divisores de frecuencia:

			Xtal/Clock Frequency (MHz)					
			24.576*	8.192	4.9152	4.096	2.4576	2.048
Inp	uts		12.288/3 12.288/5 6.144/3					6.144/3
ClkDivA		Data Rate (kbps)						
CIKDIVA	ClkDivB	Data Rate				180 18 16		
0	0	128	192*	64*	38.4*	32	19.2	16
0	1	256	96*	32	19.2	16	9.6	8
1	0	512	48*	16	9.6	8	4.8	4
1	1	1024	24*	8	4.8	4		

^{*} V_{DD} ≥ 4.5V, external clock

Figura A.7: Selección de R_B en función de f_{xtal} y los estados sobre ClkDivA y ClkDivB

Cabe señalar que el conexionado mínimo que exige el dispositivo CMX589A para establecer un correcto sincronismo para una tasa binaria dada, poniendo como ejemplo $R_B = 4.8 \ kbps$ con sincronismo externo basado en un oscilador a cristal de 4.9152 MHz, será como el que se presenta en la siguiente figura:

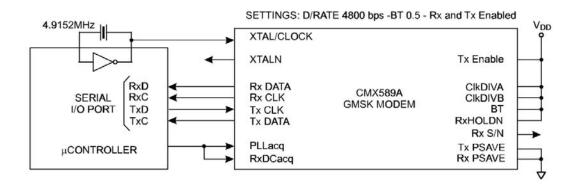


Figura A.8: Mínimas conexiones del CMX589A para $R_B = 4800bps$

Selección del factor BT

Para la selección del factor BT del chip CMX589A, surge la necesidad de colocar un filtro RC entre la salida del pin 16 (TXOUT) y la entrada del modulador FM. Esta red puede incluir algún tipo de ajuste de ganancia o de nivel de DC, lo que permite controlar los niveles máximos y mínimos de potencia a la salida del chip.

Los valores que determinan el factor BT son los que componen el filtro RC: R_1 y C_1 , que como ya fue mencionado, ocupan la siguiente localización en relación con los pines del CMX589A:

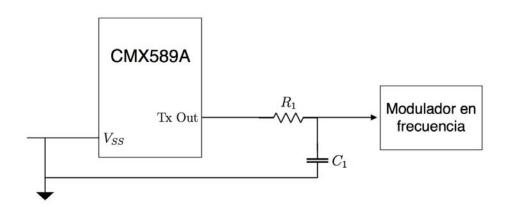


Figura A.9: Filtro RC para la selección del factor BT

Considerando que en todo momento debe cumplirse que $R_1 \ge 20 \ k\Omega$, los valores de R_1 y C_1 pueden ser calculados a partir de las siguientes expresiones en función de la tasa binaria deseada y el factor BT escogido:

- Para un factor
$$BT = 0.3$$

$$R_1 \cdot C_1 = \frac{0.34}{R_B} \tag{A.2}$$

- Para un factor
$$BT=0.5$$

$$R_1 \cdot C_1 = \frac{0.22}{R_B} \tag{A.3} \label{eq:A.3}$$

A continuación, se adjunta una tabla con los valores de R_1 y C_1 en función de los dos posibles valores de BT seleccionables y las tasas binarias más habituales:

Data Rates (kbps)	BT = 0.3		BT = 0.5		
	R1	C1	R1	C1	
4	120kΩ	680pF	120kΩ	470pF	
4.8	100kΩ	680pF	100kΩ	470pF	
8	91kΩ	470pF	120kΩ	220pF	
9.6	91kΩ	390pF	47kΩ	470pF	
16	47kΩ	470pF	91kΩ	150pF	
19.2	100kΩ	180pF	91kΩ	120pF	
32	47kΩ	220pF	47kΩ	150pF	
38.4 *	47kΩ	180pF	47kΩ	120pF	
64 *	56kΩ	100pF	51kΩ	68pF	
80 *			39kΩ	68pF	
128 *			82kΩ	22pF	
144 *			68kΩ	22pF	
160 *			62kΩ	22pF	
176 *			56kΩ	22pF	
192 *			51kΩ	22pF	
* V _{DD} ≥ 4.5V, external clock					

Figura A.10: Tabla de selección de R_1 y C_1 en función de R_B y factor BT

Filtrado y ganancia en recepción

Las principales funciones que cumple el chip CMX589A para la recepción de datos son las siguientes:

- Ajustar el nivel de la señal de entrada.
- Filtrar la señal de entrada.
- Proporcionar los niveles de DC adecuados para las señales de reloj y para los circuitos externos.
- Proporcionar el sincronismo de la señal de reloj para la extracción de datos y para los circuitos externos.
- Conformar los datos recibidos en formato binario.
- Evaluar la calidad de la señal recibida e indicar la relación S/N.

La salida del discriminador de frecuencia del receptor radio debe conectarse al filtro RX con una ganancia adecuada y un nivel de DC correctamente ajustado. Este filtro, además, debe ser añadido previamente a la etapa de amplificación del receptor. La ganancia debe ser ajustada de forma que el nivel de la señal de entrada no supere 1 V_{pp} (para una alimentación $V_{DD} = 0.5 \ V$ centrado sobre V_{bias} cuando se recibe un patrón de datos binarios continuos de la forma 1111000011110000.

Los ciclos positivos de la señal de entrada generan un θ lógico, mientras que los ciclos negativos de la señal de entrada generan un 1 lógico sobre la salida de datos recibidos.

La señal recibida es filtrada a través de un filtro pasobajo con una atenuación de -3 dB para 0.56 veces la tasa binaria de datos. Este filtro se sitúa antes de la región de extracción de datos y generación de las señales de reloj del chip.

El bloque de ajuste de nivel consiste en dos detectores de voltaje: uno de ellos mide los ciclos positivos (niveles más altos que $\frac{V_{DD}}{2}$) y otro los negativos (niveles más bajos que $\frac{V_{DD}}{2}$). Los condensadores externos son utilizados para estos detectores a través de los pines DOC1 y DOC2, formando así el voltaje para los circuitos de hold e integración. Estos dos niveles se emplean para establecer los decisores de umbrales óptimos para el nivel DC del circuito de reloj y para la extracción de datos, y dependen de la amplitud de la señal recibida y sus niveles de offset.

Modos de control en recepción

Las características operativas de control de los circuitos de medida de nivel y de extracción de reloj en el modo receptor del dispositivo están controlados a través de los niveles lógicos que se introducen a la entrada de los pines PLLacq, Rx HOLDN y RxDCacq, que también pueden ser usados para cambiar las condiciones de recepción, tal y como se muestra en la siguiente tabla:

PLLacq	Rx HOLDN	PLL Action		
1	1	Acquire	Sets the PLL bandwidth wide enough to allow a lock to the received signal in less than 8 zero crossings. This mode will operate as long as PLLacq is a logic "1".	
1 to 0	1	Medium Bandwidth	The correction applied to the extracted clock is limited to a maximum of ±1/16th bit-period for every two received zero-crossings. The PLL operates in this mode for a period of about 30 bits immediately following a 1 to 0 transition of the PLLacq input, provided that the Rx HOLDN input is a logic '1'.	
0	1	Narrow Bandwidth	The correction applied to the extracted clock is limited to a maximum of ±1/64th bit-period for every two received zero-crossings. The PLL operates in this mode whenever the Rx HOLDN Input is a logic '1' and PLLacq has been a logic '0' for at least 30 bit periods (after Medium Bandwidth operation for instance).	
0	0	Hold	The PLL feedback loop is broken, allowing the Rx Clock to freewheel during signal fade periods.	

Figura A.11: Tabla de modos de funcionamiento del PLL en recepción

En general, una transmisión de datos comienza con un cierto preámbulo, como puede ser 1100110011001100, lo que permite a los circuitos receptores establecer los tiempos de adquisición de reloj tan rápidamente como fuera posible. Una vez se haya detectado la portadora en recepción, y durante el tiempo en el que se espera el preámbulo de la trama de datos recibida, las entradas PLLacq y RxDCacq deben ser conmutadas de θ lógico a θ lógico para que el control de ganancia y la extracción de reloj cumplan sus funciones correctamente.

La entrada Rx HOLDN normalmente debe mantenerse con un 1 lógico mientras se reciban datos, aunque puede establecerse un θ lógico para congelar el nivel de medición de los circuitos de extracción de reloj durante un proceso de desvanecimiento. Si el desvanecimiento es menor a 200 periodos de bit, puede retroceder a su modo de funcionamiento normal introduciendo un 1 lógico sobre la entrada Rx HOLDN. En caso de desvanecimientos superiores a 200 periodos de bit, es recomendable restablecer los circuitos de control de ganancia poniendo un 1 lógico en el pin RxDCacq durante un intervalo de tiempo entre 10 a 20 periodos de bits.

La entrada Rx HOLDN no tiene ningún efecto sobre los circuitos de control de ganancia siempre que RxDCacq siga con un 1 lógico en su entrada, y tampoco tiene efecto sobre el PLL siempre que el pin PLLacq tenga un 1 lógico. Un θ lógico en Rx HOLDN no desactiva la salida del reloj en recepción (Rx Clock), y los circuitos de extracción de reloj y de medición de la relación S/N siguen estando operativos.

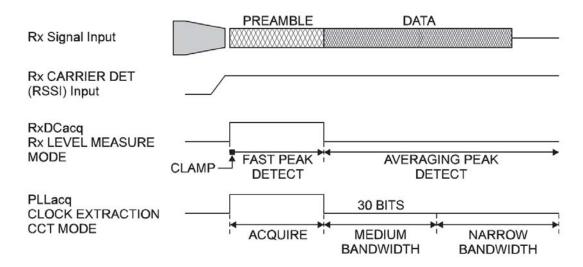


Figura A.12: Modos de funcionamiento de control en recepción

Extracción de reloj en recepción

Los circuitos de extracción de la señal de reloj del receptor controlan la salida de reloj del receptor, que se encuentra sincronizada mediante un circuito PLL de cruces por cero de la señal de entrada. Incluso la salida de reloj del receptor internamente es usada por los circuitos de extracción de datos.

Las características del PLL pueden variar en el circuito de control del receptor, sobre los pines PLLacq y RX HOLDN, dependiendo del modo de operación seguido (tal y como se indicaba en las tablas anteriores).

Extracción de datos en recepción

Los circuitos de extracción de datos del receptor muestrean la señal recibida y decide si el valor adquirido es un 1 o un 0 lógicos, después del filtrado de esta misma señal. La comparación de estos valores se realiza a través de los circuitos de medición de nivel a través de un umbral.

Este umbral se adapta bit a bit para reducir la interferencia entre símbolos (IES) y la limitación de la ruta global de transmisión y de premodulación gaussiana. Los datos extraídos se envían desde el pin de recepción de datos y se deben tomar muestras externamente en el flanco ascendente de la señal de reloj del receptor.

Detección de relación S/N en recepción

El sistema de detección de la relación S/N clasifica los cruces por cero como buenos y malos dependiendo de los tiempos que transcurren entre los cruces que se producen realmente con respecto al tiempo esperado, que viene dados por los circuitos de extrac-

ción de reloj del PLL. Esta información es procesada para dar seguidamente un nivel lógico a la salida del pin RX S/N a partir de esta clasificación:

- Si tenemos un 1 lógico, los cruces por cero serán buenos.
- Si tenemos un 0 lógico, los cruces por cero serán malos.

Según esta medida, es posible derivar una medida de la relación S/N, y por tanto, puede estimarse la tasa de error de bit (BER) de la señal recibida para un factor BT establecido. Esto se puede representar a modo de ejemplo, para una tensión de alimentación $V_{DD} = 5$ V, como se observa en la siguiente gráfica:

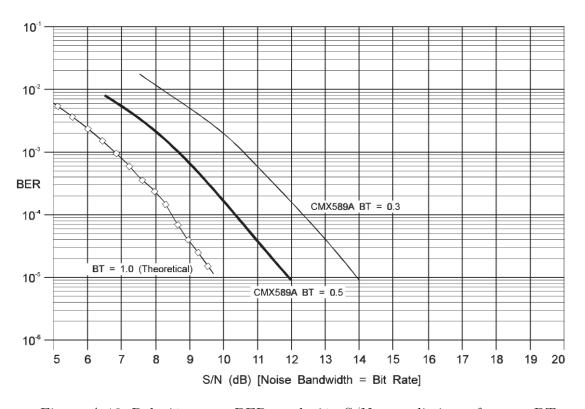


Figura A.13: Relación entre BER y relación S/N para distintos factores BT

Cabe señalar que esta gráfica es independiente de la tasa binaria (aunque se considera el ancho de banda de ruido para calcular la tasa binaria empleada), el rendimiento radio (por ejemplo, la distorsión en el filtro de FI), el voltaje aplicado (para detectar un nivel de bit alto se necesita que $V_{DD} \ge 4.5 \text{ V}$) y otros factores reales.

Calidad de la señal recibida

Para medir la calidad de la señal recibida, se emplea una gráfica que marca la tendencia lineal de los tiempos de adquisición de señales con un cierto factor BT, expresados como medida porcentual, en función de la relación S/N, como es la que se presenta a continuación:

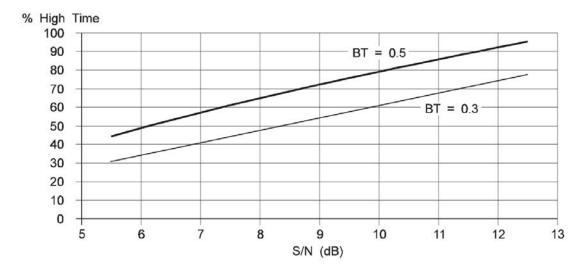


Figura A.14: Relación entre tiempos de adquisición y relación S/N para distintos factores BT

Filtrado y ganancia en transmisión

Los datos binarios que se aplican a la entrada TX son remitidos por el chip CMX589A en cada flanco de subida de la señal de reloj de transmisión, para luego convertirse en una señal binaria con una amplitud de 1 V_{PP} centrada en V_{BIAS} para una tensión $V_{DD} = +5~V$.

En caso de que la entrada de habilitación del transmisor, TX ENABLE, se encuentra a nivel alto, esta señal binaria se conecta a la entrada del filtro paso bajo del transmisor, y la salida de este filtro se conecta al pin de salida TX OUT. Por el contrario, si la entrada de habilitación del transmisor se encuentra a nivel bajo, se conecta la entrada del filtro de transmisión a V_{BIAS} y se desconecta la salida del filtro, TX OUT, a través de una resistencia normalizada de 500 $k\Omega$.

El filtro de transmisión tiene una respuesta al impulso gaussiana aproximada (ver figura A.15), que minimiza la distorsión de amplitud y de fase de la señal binaria, a la vez que introduce la suficiente atenuación para evitar las interferencias con los canales adyacentes a la banda de trabajo del dispositivo.

En lo que respecta al ancho de banda del filtro de transmisión, este viene determinado por la tasa binaria con la que opera el dispositivo (dependiente de la red de sincronismo externa) y la configuración del factor BT, pudiendo ser esta BT = 0.3 o BT = 0.5.

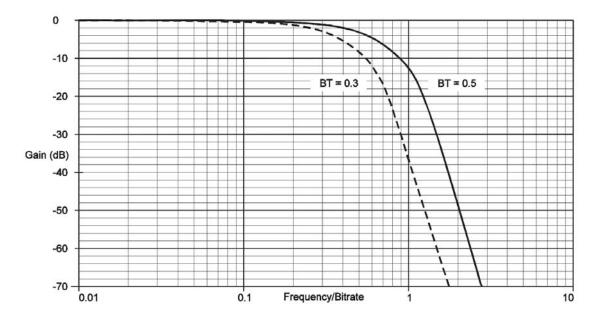


Figura A.15: Respuesta al impulso del filtro de transmisión

Entre la salida del pin TX OUT y la entrada del modulador de frecuencia, es necesario colocar una red RC. Esta red se puede adaptar a cualquier cambio de los niveles de DC y los ajustes del nivel de salida (ganancia) de la señal, y forma una parte importante para el filtrado de la señal de transmisión.

La conexión a tierra del condensador C_1 de esta red RC debe ser tal que permita in-

troducir la máxima atenuación posible para el ruido de alta frecuencia en el modulador de frecuencia.

La señal TX OUT se centra alrededor de V_{BIAS} , que es positiva cuando se reciben 1 lógicos y negativa cuando se reciben θ lógicos.

Encapsulado de los modelos de chip CMX589A

En este apartado se muestran las dimensiones físicas de los cuatro tipos de encapsulados del chip CMX589A que se encuentran en el mercado, aunque el funcionamiento de estos modelos es exactamente igual en todos los casos.

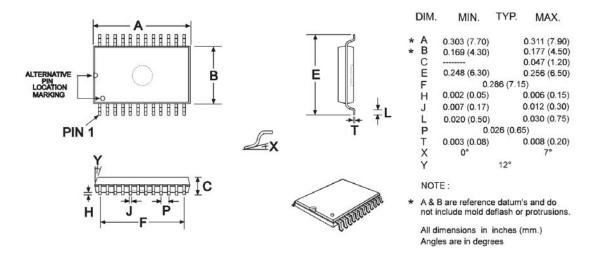


Figura A.16: Encapsulado del modelo CMX589AE2

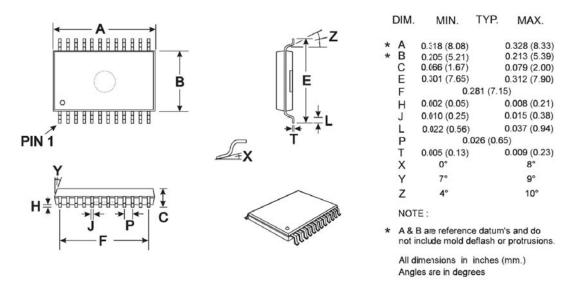


Figura A.17: Diagrama de pines del chip CMX589AD5

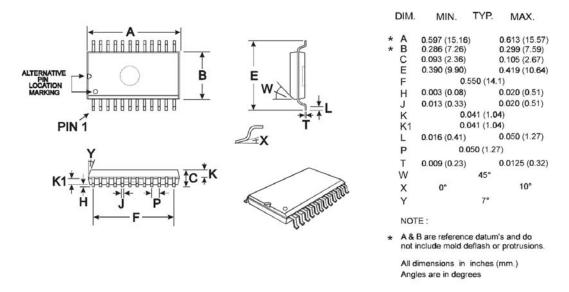


Figura A.18: Diagrama de pines del chip CMX589AD2

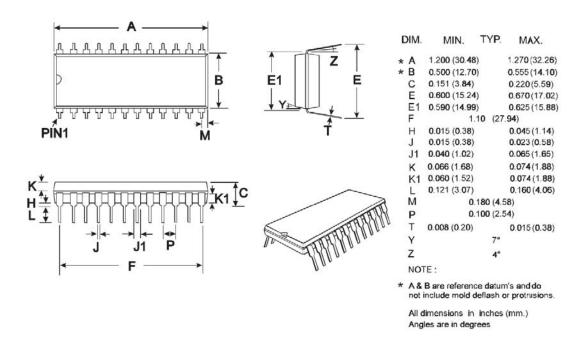


Figura A.19: Diagrama de pines del chip CMX589AP4

Apéndice B

PCB del módulo transmisor

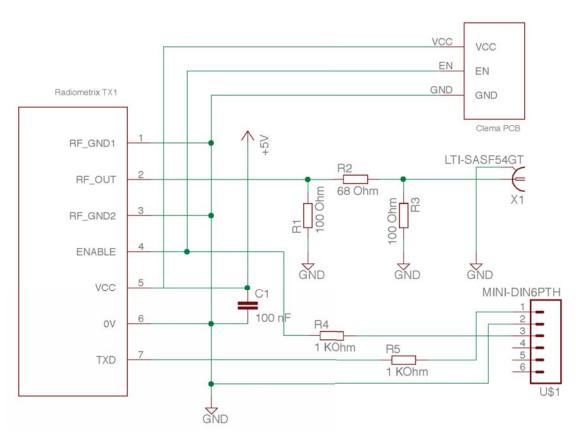


Figura B.1: Esquemático del módulo transmisor para el banco de pruebas AIS

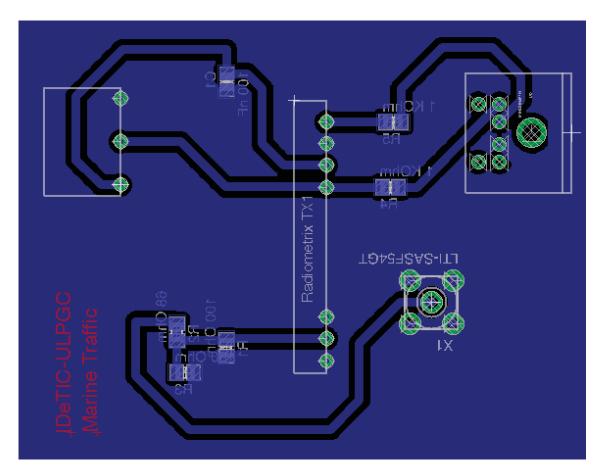


Figura B.2: Diseño PCB sobre Eagle PCB Design del transmisor FM

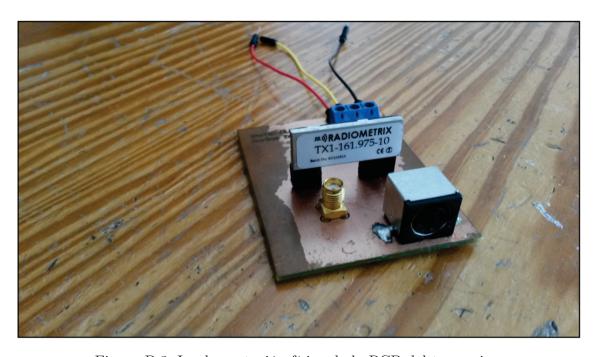


Figura B.3: Implementación física de la PCB del transmisor

Apéndice C

PCB del módulo receptor

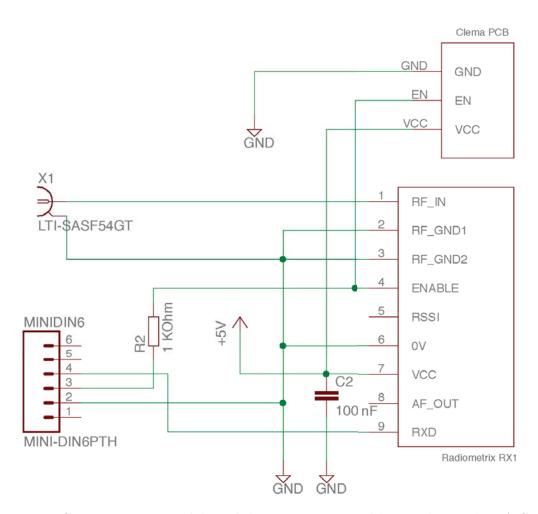


Figura C.1: Esquemático del módulo receptor para el banco de pruebas AIS

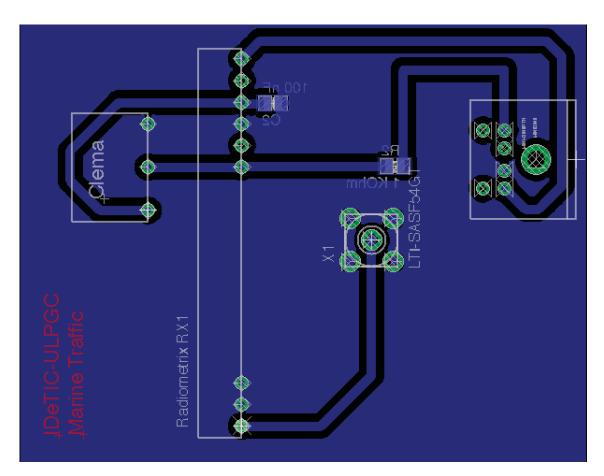


Figura C.2: Diseño PCB sobre Eagle PCB Design del receptor FM



Figura C.3: Implementación física de la PCB del receptor

Apéndice D

Código del firmware de control del transmisor

```
1 // DEFINICIONES DE PINES
2 #define pinRXCLK
3 #define pinTXCLK
4 #define pinSN
5 #define pinRXD
6 #define pinTXD
7 #define pinPTT
8 #define pinPLLACQ
                      10
9 #define pinLEDTX
10 #define pinLEDCOS
                      11
11 #define pinLEDPWR
12 #define pinArduinoLED 13
14 // DEFINICIONES DE LIMITES DE VARIABLES
15 #define MAX_DATA 23
16 #define LONG_HEADER 4
17 #define MAX_FRAME LONG_HEADER + MAX_DATA
18
19 // VARIABLES GLOBALES
20 uint8_t txdato;
21 uint8_t txByte;
22 uint8_t datoGMSK;
23 uint8_t datoGMSK_NRZI = 0;
24 int contadorBitTx = 0;
25 int contadorByteTx = 0;
26 int contadorTx = 0;
27 unsigned char bufferTx[MAX_FRAME];
29 // INICIALIZACION DEL PROGRAMA
30 void setup() {
31 // inicializacion del puerto serie
    Serial.begin (9600);
```

```
Serial.flush();
33
    // inicializacion de los pines
34
    pinMode(pinTXCLK, INPUT);
35
    pinMode(pinRXCLK, INPUT);
36
    pinMode(pinSN, INPUT);
37
    pinMode(pinRXD, INPUT);
38
39
    pinMode(pinTXD, OUTPUT);
    pinMode(pinLEDTX, OUTPUT);
40
    pinMode(pinLEDCOS, OUTPUT);
41
    pinMode(pinLEDPWR, OUTPUT);
42
    pinMode(pinPTT, OUTPUT);
43
44
    pinMode(pinPLLACQ, OUTPUT);
45
    // inicializacion de la trama
    bufferTx[0] = 0x55;
46
    bufferTx[1] = 0x55;
47
    bufferTx[2] = 0x55;
48
    bufferTx[3] = 0x7E;
49
    bufferTx[LONG_HEADER + MAX_DATA] = 0x7E;
50
    bufferTx[LONG_HEADER + MAX_DATA+1] = 0x00;
51
    bufferTx[LONG_HEADER + MAX_DATA+2] = 0x00;
52
    bufferTx[LONG_HEADER + MAX_DATA+3] = 0x00;
53
54 }
55
56 // PROGRAMA PRINCIPAL
  void loop() {
57
    // mientras no se reciban datos por el puerto serie, no
58
    // se hace ninguna operacion
59
    while(!Serial.available());
60
    // se recorre el vector de datos y se envia uno a uno
61
    // los bytes con el formato de la trama AIS
62
    for(int k=0; k<MAX_DATA; k++) {</pre>
63
      delay(10);
64
      txdato = Serial.read();
65
66
      bufferTx[LONG_HEADER + k] = txdato;
      Serial.write(txdato);
67
    }
68
    // una vez enviada la trama, se reinician los contadores
69
    // de bits y bytes y se vuelve a activar la ISR
70
    contadorBitTx = 0;
71
    contadorByteTx = 0;
72
73
    attachInterrupt(1, txISR, RISING);
74 }
75
76 // RUTINA DE SERVICIO EN TRANSMISION
77 void txISR() {
    // se toman los bytes almacenados en el vector
78
79
    // de la trama de datos
  txByte = bufferTx[contadorByteTx];
```

```
// se escribe bit a bit en el pin TXD los bytes
     // almacenados en la trama
82
     datoGMSK = (txByte >> (7-contadorBitTx)) & 0x01;
83
     // se realiza la codificación NRZI
84
     if (datoGMSK == HIGH) {
85
       // se modifica el estado respecto al bit anterior
86
       datoGMSK_NRZI = digitalRead(pinTXD) ^ datoGMSK;
87
       digitalWrite(pinTXD, datoGMSK_NRZI);
88
      } else {
89
       // se mantiene el estado del bit anterior
90
       datoGMSK_NRZI = digitalRead(pinTXD);
91
92
       digitalWrite(pinTXD, datoGMSK_NRZI);
93
94
     // si el contador de bits llega a 8, se reinicia este
     // contador de bits y se incrementa el de bytes
95
     if(contadorBitTx >= 7) {
96
       contadorBitTx = 0;
97
       // si el contador de bytes llega al tamaño maximo de la
98
       // trama de datos, se reinicia el contador de bytes y
99
          se
       // desactiva la interrupcion; en caso contrario, se ira
100
       // aumentando el contador de bytes cada vez que el
101
       // contador de bits llegue a 8
102
       if(contadorByteTx >= MAX_FRAME-1) {
103
         contadorByteTx = 0;
104
         detachInterrupt(1);
105
106
       } else {
         contadorByteTx++;
107
108
     } else {
109
       contadorBitTx++;
110
111
112 }
```

Apéndice E

Código del firmware de control del receptor

```
1 // DEFINICIONES
2 #define pinRXCLK
3 #define pinTXCLK
4 #define pinSN
5 #define pinRXD
6 #define pinTXD
7 #define pinPTT
8 #define pinPLLACQ
                      10
9 #define pinLEDTX
10 #define pinLEDCOS
                      11
11 #define pinLEDPWR
12 #define pinArduinoLED 13
14 // DEFINICIONES DE LIMITES DE VARIABLES
15 #define MAX_DATA 23
16 #define HEADER 0x7E
17
18 // VARIABLES GLOBALES
19 uint8_t rxdato;
20 uint8_t rxdatoNRZI;
21 uint8_t rxdatoNRZI_anterior = 0;
22 uint8_t rxByte;
23 uint8_t datoGMSK;
24 unsigned char rxHeader = 0;
25 unsigned char rxTail = 0;
26 uint8_t contadorBitRx = 0;
27 uint8_t contadorByteRx = 0;
28 unsigned char bufferRx[MAX_DATA];
29 int flagComienzoTrama = 0;
30 int flagRx = 0;
31
32 // INICIALIZACION DEL PROGRAMA
```

```
void setup() {
    // inicializacion del puerto serie
34
    Serial.begin (9600);
35
    Serial.flush();
36
    // inicializacion de los pines
37
    pinMode(pinSN, INPUT);
38
39
    pinMode(pinRXD, INPUT);
    pinMode(pinTXD, OUTPUT);
40
    pinMode(pinLEDTX, OUTPUT);
41
    pinMode(pinLEDCOS, OUTPUT);
42
    pinMode(pinLEDPWR, OUTPUT);
43
44
    pinMode(pinPTT, OUTPUT);
    pinMode(pinPLLACQ, OUTPUT);
45
    // inicializacion de la interrupcion
46
    attachInterrupt(0, rxISR, FALLING);
47
48 }
49
  // PROGRAMA PRINCIPAL
50
  void loop() {
51
    // si se llega al final de la trama de datos
52
    // filtrada, los bytes almacenados se envian
53
    // por el puerto serie
54
    if(flagRx) {
55
      for(int k=0; k<MAX_DATA; k++) {</pre>
56
        delay(10);
57
        Serial.write(bufferRx[k]);
58
      }
59
      // se reinician las variables para el filtro
60
      // de las tramas
61
      flagRx = 0;
62
      rxHeader = 0;
63
      rxTail = 0;
64
      // se vuelve a activar la interrupcion
65
66
      attachInterrupt(0, rxISR, FALLING);
    }
67
68 }
69
70 // RUTINA DE SERVICIO EN RECEPCION
  void rxISR() {
71
    // se leen los bits recibidos en el pin RXD
72
73
    rxdato = digitalRead(pinRXD);
    // se realiza la decodificacion NRZI
74
    if(rxdatoNRZI == HIGH) {
75
      rxdato = rxdatoNRZI_anterior ^ rxdatoNRZI;
76
77
    } else {
      rxdato = rxdatoNRZI_anterior;
78
79
    rxdatoNRZI_anterior = rxdatoNRZI;
80
```

```
// se evaluan los campos de la trama AIS
     switch(flagComienzoTrama) {
82
        // PREAMBULO: [0x55][0x55][0x55]
83
        case 0:
84
          rxHeader = (rxHeader << 1) | rxdato;</pre>
85
          if(rxHeader == 0x7E) {
86
87
            flagComienzoTrama = 1;
          }
88
          break;
89
        // CABECERA: [0x7E]
90
        case 1:
91
92
          rxByte = (rxByte << 1) | rxdato;</pre>
          if(contadorBitRx >= 7) {
93
            bufferRx[contadorByteRx] = rxByte;
94
            contadorByteRx++;
95
            contadorBitRx = 0;
96
            rxByte = 0;
97
            if(contadorByteRx>=MAX_DATA) {
98
               flagComienzoTrama = 2;
99
            }
100
          } else {
101
            contadorBitRx++;
102
          }
103
104
          break;
        // DATOS & CRC
105
        case 2:
106
          rxTail = (rxTail << 1) | rxdato;</pre>
107
          if(contadorBitRx >= 7) {
108
            flagComienzoTrama = 0;
109
            contadorBitRx = 0;
110
            contadorByteRx = 0;
111
            detachInterrupt(0);
112
            flagRx = 1;
113
114
          } else {
            contadorBitRx++;
115
116
117
          break;
     }
118
119 }
```

Apéndice F

Publicación de resultados

Design of a Low Cost Prototype of Automatic Identification System (AIS) Receiver

Francisco Cabrera ⁽¹⁾, Nicolás Molina ⁽¹⁾, Miluše Tichavska* ⁽²⁾ and Víctor Araña ⁽¹⁾ (1) IDeTIC, Universidad de Las Palmas de Gran Canaria, http://www.ulpgc.es (2) MarineTraffic, http://www.marinetraffic.com

The GMDSS system is the Global System of Maritime Distress and Safety. It is conformed by a variety of systems that allow communication and a fairly complete coverage on vessel tracking (GMDSS equipment operation assessment standards", CMSA, Mar, 2008). The AIS is part of this system and allows the user to acknowledge: the position, speed, heading and additional vessel information on real-time. Numerous commercial devices integrate the AIS. (Frank Lanier, "Feature Loaded High-end VHFs", ICOM 2012) These, allow users to access specific information once it has been correctly transmitted, received within a range of coverage and, transmitted into a general database (i.e. MarineTraffic). Nevertheless, the acquisition of a receiving device, which integrates features of interest, implies a considerable economic cost.

With this purpose, the design of a low cost AIS receiver that enables the analysis and viability of the system is proposed. To develop this low cost receiver, a complete system (as presented in Figure 1) must be designed. To do this, its different parts must be taken into consideration: an antenna that receives the signal, a radiofrequency stage that demodulates the signal to base band and the digital stage to perform an error control algorithm using a Cyclic Redundancy Code (CRC) to then, decode the AIS frames.



Figure 1: Signal processing before being displayed to the user.

The marine band VHF antenna chosen is a low-gain omnidirectional whip and adapted in the frequency band of the AIS. The FM demodulator chosen is the RX1-161.975-10 of Radiometrix. Once the signal is demodulated, it is demodulated once again in base band GMSK through a DV-MEGA modem plate (shield of Arduino) at a speed of 9600 bps. At last, the Microcontroller MEGA 2560 of Arduino is chosen to perform the actions related to error control (CRC) and decoding frames of AIS for the later transmission of decoded signals to a PC or a Raspberry PI.

The performance of the proposed low-cost AIS receiving hardware (< 125 Euros) complies with the expected reception performance under an academic or research environment. Indeed, athe extension of its use will not only facilitate the access to information of great value but it will enable the building of computing systems and products. (Tichavska, M. and Tovar, B. Maritime Economists Conference, Norfolk, Virginia, 2014).

Furthermore, results may support applied and theoretical research in the areas of innovative system architectures for the future internet of things, transport systems, Geographical Information Systems, hardware sensors and low power computer, vessel and route efficiency, port optimization and others.