

UNIVERSIDAD NACIONAL DEL CALLAO

ESCUELA DE POSGRADO

UNIDAD DE POSGRADO DE LA FACULTAD DE

INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**“DISEÑO E IMPLEMENTACIÓN DEL TELEGRAMA DE
DATOS ACTUALES CON INTERFAZ GRÁFICA PARA EL
TORPEDO ANTISUPERFICIE SST-4”**

**SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO
ACADÉMICO DE MAESTRO EN CIENCIAS DE LA
ELECTRÓNICA CON MENCIÓN EN CONTROL Y
AUTOMATIZACIÓN”**

**CÉSAR RICARDO ARANGÜENA DELAUDE
OSCAR ANTONIO DOMÍNGUEZ COLÁN**

Callao, 2019

PERÚ

HOJA DE REFERENCIA DEL JURADO

MIEMBROS DEL JURADO

Dr.	: JACOB ASTOCONDOR VILLAR	PRESIDENTE
Msc.	: JULIO CÉSAR BORJAS CASTAÑEDA	SECRETARIO
Mg.	: WILBERT CHÁVEZ IRAZABAL	MIEMBRO
Msc.	: ABILIO BERNARDINO CUZCANO RIVAS	MIEMBRO
Dr.	: NICANOR RAÚL BENITES SARAVIA	ASESOR

Nº DE LIBRO : 01

FOLIO : 091

FECHA DE APROBACIÓN : 12 de diciembre del 2019

RESOLUCION DIRECTORAL : 068-2019-DUPFIEE

DEDICATORIA

Dedicamos este trabajo de investigación a nuestros padres por sus principios y valores éticos que nos permitieron inculcarnos la necesidad de ser mejor cada día como persona y como profesionales.

A nuestros maestros, quienes nos orientaron a través de sus enseñanzas y experiencias, otorgándonos los conocimientos y herramientas necesarias para ser realidad este proyecto.

AGRADECIMIENTO

Quisiéramos agradecer al Ing. Jorge ESCAJADILLO Lara por todo el apoyo desde el inicio de este proyecto.

También al personal de la Estación de Armas Submarinas de la Marina de Guerra del Perú que nos apoyaron durante el desarrollo de la tesis, ya que sin sus conocimientos y apoyo no se hubiera podido cumplir las metas trazadas.

ÍNDICE

ÍNDICE	1
Índice de figuras	5
Índice de tablas	10
RESUMEN.....	11
ABSTRACT	13
INTRODUCCIÓN.....	15
CAPÍTULO I.....	17
PLANTEAMIENTO DEL PROBLEMA	17
1.1 Determinación del Problema.....	17
1.2 Formulación del Problema	19
1.2.1 Problema general:.....	19
1.2.2 Problemas específicos:	19
1.3 Definición de objetivos	20
1.3.1 Objetivo general.....	20
1.3.2 Objetivos específicos	20
1.4 Justificación	21
CAPÍTULO II.....	22
MARCO TEÓRICO	22
2.1 Estado del arte.....	22
2.2 Marco teórico	27
2.2.1 Torpedo SST-4	27
2.2.1.1 Descripción General.....	28
2.2.1.2 Sistema de detección acústica.....	31
2.2.1.3 Sistema de comunicación del torpedo SST-4.....	36
2.2.1.4 Sistema de Estabilización y control del torpedo SST-4.	42
2.2.2 Plataforma Arduino	47
2.2.2.1 Características del hardware.....	48
2.2.2.2 Modelos de tarjetas Arduino.....	49
2.2.2.3 Arduino Mega pro-mini.....	50
2.2.2.4 Arduino nano.....	51
2.2.3 Amplificadores operacionales.....	52
2.2.3.1 Amplificador diferencial.	53

2.2.3.2	Amplificador Inversor	53
2.2.4	Optoacoplador.	54
2.2.5	Regulador LM2596.....	55
2.2.6	Convertor ADC.....	56
2.2.6.1	Resolución	56
2.2.6.2	Circuito ADS1115.....	57
2.2.7	Visual Studio 2012	58
2.2.7.1	Visual C#	60
2.2.8	Eagle Cadsoft	61
CAPÍTULO III.....		62
VARIABLES E HIPOTESIS		62
3.1	Definición de las variables.....	62
3.1.1	Variable dependiente.	62
3.1.2	Variables independientes.....	62
3.2	Operacionalización de las variables.	63
3.3	Hipótesis general e hipótesis específicas.....	65
3.3.1	Hipótesis general.	65
3.3.2	Hipótesis específicas.	65
CAPÍTULO IV		66
METODOLOGÍA.....		66
4.1	Tipo de investigación.	66
4.2	Diseño de la investigación.....	66
4.3	Población y muestra.....	68
4.4	Técnicas e instrumentos de recolección de datos.	69
4.5	Procedimiento de recolección de datos.....	70
4.5.1	Señales digitales del torpedo SST-4.	71
4.5.1.1	Modo de guiado.	72
4.5.1.2	Distancia de seguridad.....	73
4.5.1.3	Detección de blanco.....	73
4.5.1.4	Modo Acústico.	75
4.5.1.5	Motor ON.	75
4.5.1.6	Rumbo.	76
4.5.2	Señales analógicas del torpedo SST-4.	77
4.5.2.1	Cabeceo.	78

4.5.2.2	Balanceo.....	81
4.5.2.3	Distancia en activo.....	82
4.5.2.4	Correlador de 0°.....	84
4.5.2.5	Profundidad.....	84
4.5.3	Diseño del DAQ de señales digitales.....	87
4.5.3.1	Hardware del DAQ de señales digitales.....	87
4.5.3.2	Software del DAQ de señales digitales.....	89
4.5.4	Diseño del DAQ de señales analógicas.....	96
4.5.4.1	Hardware del DAQ de señales digitales.....	96
4.5.4.2	Software del DAQ de señales Analógicas.....	106
4.5.5	Diseño del Codificador de Telegrama Actual.....	114
4.5.5.1	Hardware del Codificador de Telegrama Actual.....	114
4.5.5.2	Software del Codificador de Telegrama Actual.....	116
4.5.6	Diseño del Decodificador de Telegrama Actual.....	118
4.5.6.1	Hardware del Decodificador de Telegrama Actual.....	118
4.5.6.2	Software del Decodificador de Telegrama Actual.....	120
4.5.8	Diseño y desarrollo de la interfaz gráfica.....	122
4.6	Procesamiento estadístico y análisis de los datos.....	128
CAPÍTULO V		129
RESULTADOS		129
5.1	Resultados.....	129
5.1.1	Tarjeta de adquisición de señales digitales.....	129
5.1.2	Tarjeta de adquisición de datos de señales analógicas.....	131
5.1.3	Tarjeta codificador del telegrama de datos actuales.....	132
5.1.4	Tarjeta decodificador del telegrama de datos actuales.....	133
5.1.5	Interfaz gráfica.....	134
CAPÍTULO VI		160
DISCUSIÓN DE RESULTADOS.....		160
6.1	Contrastación de hipótesis con los resultados.....	160
CAPÍTULO VII		163
CONCLUSIONES.....		163
CAPÍTULO VIII		165
RECOMENDACIONES.....		165
CAPÍTULO IX		166

REFERENCIAS BIBLIOGRÁFICAS	166
ANEXOS	170
Anexo 1: Programa de la tarjeta DAQ digital.....	170
Anexo 2: Programa de la tarjeta DAQ Analógico	174
Anexo 3: Programa Codificador de Telegrama Actual	178
Anexo 4: Programa Decodificador de Telegrama Actual.....	192

Índice de figuras

Figura 1: Secciones principales del torpedo SST-4.....	27
Figura 2: Partes del torpedo SST-4, versión de combate y ejercicio.	28
Figura 3: Sistema de detección acústica del torpedo SS-4.	31
Figura 4: Arreglo esquemático del sonar del torpedo SS-4.	32
Figura 5: Condiciones para TC y TD en modo pasivo.	34
Figura 6: Condiciones para TC y TD en modo activo.	34
Figura 7: Condiciones para Perdida de blanco en modo pasivo.....	35
Figura 8: Sistema de comunicación del torpedo SS-4.....	36
Figura 9: Telegrama de comando del torpedo SST-4.	38
Figura 10: Telegrama Actual del torpedo SST-4.	39
Figura 11: Modulo NVS-A del torpedo SST-4.	40
Figura 12: Modulo SPU-A del torpedo SST-4.	41
Figura 13: Control de Profundidad y Cabeceo del torpedo SS-4.....	44
Figura 14: Control de Rumbo y Balanceo del torpedo SS-4.....	46
Figura 15: Hardware y software de la plataforma Arduino.....	47
Figura 16: Diagrama de bloques del controlador Atmega.	48
Figura 17: Modelos de placa Arduino.....	49
Figura 18: Arduino Mega pro mini.....	50
Figura 19: Arduino nano	51
Figura 20: Simbología del Amplificador operacional.	52
Figura 21: Amplificador operacional modo diferencial	53
Figura 22: Amplificador inversor	53
Figura 23: Circuito del optoacoplador.	54

Figura 24: Módulo DC-DC Converter LM2596.	55
Figura 25: Pines del ADS1115.	57
Figura 26: Configuración del ADS1115.	58
Figura 27: Logo de la plataforma Visual Studio.	59
Figura 28: Entorno de desarrollo C#.	60
Figura 29: IDE Eagle Cadsoft, diagramador y esquema	61
Figura 30: Etapas de procesamiento del telegrama de datos actuales.	70
Figura 31: Señales digitales en el (SPU-A).	72
Figura 32: Modo de detección acústica.	74
Figura 33: Señales analógicas en el (SPU-A).	77
Figura 34: Conversión de voltaje a ángulo de cabeceo.	79
Figura 35: Recorrido del pulso en modo activo.	82
Figura 36: Conversión de voltaje a pies.	86
Figura 37: Circuito optoacoplador.	87
Figura 38: Circuito de lectura de las señales digitales.	88
Figura 39: Diagrama de flujo del programa del algoritmo Digital.	89
Figura 40: Diagrama de flujo de la función Lectura-Digital.	91
Figura 41: Diagrama de flujo de la función “Leer: Motor_On y Dist_Cub”	92
Figura 42: Diagrama de flujo de la función “Leer: Homing, Modo_Acust”	93
Figura 43: Diagrama de flujo de la función “Leer_Rumbo”	94
Figura 44: Diagrama de flujo de la función “Leer_TC_TD_HH”	95
Figura 45: Amplificador diferencial.	96
Figura 46: Amplificador diferencial con voltaje de referencia.	97
Figura 47: Diagrama de corrientes en el amplificador diferencial.	97
Figura 48: Amplificador inversor.	101

Figura 49: Amplificador inversor con referencia o voltios.	101
Figura 50: Diagrama de corriente en el amplificador inversor.	102
Figura 51: Circuito acondicionador de nivel de cabeceo y rolido.....	103
Figura 52: Circuito acondicionador de la señal profundidad.....	105
Figura 53: Circuito DAQ Analógico.	105
Figura 54: Diagrama de flujo del programa de señales analógicos.	106
Figura 55: Diagrama de flujo de la función “Lectura_Analogo”.....	107
Figura 56: Diagrama de flujo de la función “Leer_Cabeceo”.	109
Figura 57: Diagrama de flujo de la función “Leer_Rolido”.	110
Figura 58: Diagrama de flujo de la función “Leer_Profundidad”.	111
Figura 59: Diagrama de flujo de la función “Leer_Correlador”.....	112
Figura 60: Diagrama de flujo de la función “Leer_Distancia”.....	113
Figura 61: Esquema del Codificador de Telegrama Actual.	115
Figura 62: Diagrama de flujo del Codificador de Telegrama Actual.....	116
Figura 63: Diseño del telegrama de datos actuales.	117
Figura 64: Esquema del Decodificador de Telegrama Actual.....	118
Figura 65: Esquema de conversión serial TTL a USB.....	119
Figura 66: Programa del Decodificador del Telegrama Actual.	120
Figura 67: Función de lectura del telegrama de datos actuales.	121
Figura 68: Cuota del mercado mundial de S.O. en el 2019.....	124
Figura 69: Plataforma de control original del Test Stand SST-4.....	127
Figura 70: Consola de pruebas Test-Stand.....	127
Figura 71: Vista en osciloscopio de la adaptación de la señal digital.	129
Figura 72: Tarjeta de adquisición de datos digitales.	130
Figura 73: Tarjeta de adquisición de datos analógicos.....	131

Figura 74: Tarjeta Codificadora del telegrama de datos actuales.....	132
Figura 75: Tarjeta Decodificadora del telegrama de datos actuales.....	133
Figura 76: Logo del software del Test-stand SST-4	134
Figura 77: Diagrama general del funcionamiento del software.....	135
Figura 78: Ventana inicial de control.....	136
Figura 79: Botones de ventanas de control.....	136
Figura 80: Botones de reporte, ayuda e información.....	137
Figura 81: Información de usuario y del entorno	137
Figura 82: Panel de sub-Ventanas de control	137
Figura 83: Plataforma original vs interfaz gráfica.	139
Figura 84: Ventana de Telegramas. Interfaz Test Stand SST-4.....	140
Figura 85: Modernización de los instrumentos de medición de los timones.	141
Figura 86: Modernización de los botones del sistema. Primer grupo.	142
Figura 87: Modernización de los botones del sistema. Segundo grupo.....	143
Figura 88: Modernización de los botones del sistema. Tercer grupo.....	144
Figura 89: Modernización de los botones del sistema. Cuarto grupo.	145
Figura 90: Modernización de los botones de profundidad.	146
Figura 91: Modernización de los botones de rumbo.....	147
Figura 92: Modernización de los botones de program.....	148
Figura 93: Modernización de los botones del modo acústico.	149
Figura 94: Modernización de los botones de velocidad.....	150
Figura 95: Modernización de los botones de Homing.	151
Figura 96: Modernización de la llave de encendido.	152
Figura 97: Summary	153
Figura 98: Ventana de indicadores	154

Figura 99: Instrumentos e indicadores de rumbo, rolo y cabeceo	155
Figura 100: Instrumentos e indicadores de velocidad y profundidad.....	155
Figura 101: Instrumentos e indicadores de distancia y correlador.....	156
Figura 102: Indicadores digitales.	156
Figura 103: Indicador Status.....	157
Figura 104: Control de acceso a la interfaz gráfica.	158
Figura 105: Ingreso correcto a la interfaz gráfica.	159
Figura 106: Administrador de usuario de la interfaz gráfica.	159
Figura 107: Telegrama de datos Actuales instalado.	160
Figura 108: Señal de sincronismo del telegrama de datos actuales.....	161
Figura 109: Tarjeta Decodificadora instalado en el Test-stand.....	162
Figura 110: Pruebas del torpedo SST-4 en el Test-Stand.....	162

Índice de tablas

Tabla 1. Especificaciones técnicas de la placa Mega pro-mini.....	50
Tabla 2. Especificaciones técnicas del microcontrolador Atmega328.	51
Tabla 3. Características técnicas el optoacoplador 4N35.....	54
Tabla 4. Especificaciones técnicas del regulador LM2596.	55
Tabla 5. Señales digitales del telegrama de datos actuales.	71
Tabla 6. Señal modo de guiado.	72
Tabla 7. Señal de la distancia de seguridad.....	73
Tabla 8. Señales de detección de blanco TC, TD y HH.	74
Tabla 9. Señal modo acústico.....	75
Tabla 10. Señal de Motor On.	75
Tabla 11. Señales digitales de las señales de rumbo.	76
Tabla 12. Señales analógicas del telegrama de datos actuales.	77
Tabla 13. Voltajes del ángulo de cabeceo.....	78
Tabla 14. Voltajes del ángulo de rolido.	81
<i>Tabla 15. Valores calculados de la distancia al blanco.</i>	<i>83</i>
Tabla 16. Valores de la relación señal/ruido.	84
Tabla 17. Valores de la profundidad actual.	85
Tabla 18. Valores de conversión de voltajes +/-12 voltios.....	100
Tabla 19. Conversión de voltajes negativos a positivos.	103
Tabla 20. Conversión de voltajes negativos a positivos.	104

RESUMEN

El torpedo SST-4, es un arma submarina guiado remotamente desde el sistema de control de tiro de las unidades submarinas por medio de un cable de comunicación. El proyecto de investigación trata de la Modernización del telegrama de datos actuales con interfaz gráfica para el torpedo SST-4, la finalidad es mantener la información actualizada del torpedo durante su desplazamiento bajo el agua, el contar con la información permite al sistema de control de tiro la mejora del sistema táctico de operación y control.

Actualmente el sistema de comunicación del telegrama de retorno de datos actuales del torpedo SST-4 es limitado, debido a su tecnología proveniente de la década del 70, el cual solo transmite las señales de “pulso de control y detección de blanco”, limitando la operación táctica en el sistema de disparo y control de las unidades submarinas.

La presente tesis de Maestría trata del diseño e implementación del telegrama de datos actuales con interfaz gráfica para el torpedo anti superficie SST-4, aplicado al sector militar de Defensa Nacional, el cual permitirá contribuir al sistema de disparo y control de las unidades submarinas pertenecientes a la Marina de Guerra del Perú, orientado a mejorar sus operaciones tácticas de combate, al tener la ubicación real del torpedo durante su corrida efectiva mediante el telegrama de datos actuales

el torpedo podrá transmitir su performance acústica y dinámica de esta manera los datos serán visualizados en una interfaz gráfica instalado en las unidades submarinas.

La investigación es de tipo experimental y aplicativo (enfoque cuantitativo), el propósito de esta investigación es aplicar el conocimiento científico para transmitir el telegrama de datos actuales, mediante la investigación experimental podemos manipular las variables relacionadas con las señales acústicas y dinámicas, por último la investigación aplicada nos permite asignar el resultado obtenido de una investigación experimental con la necesidad de buscar nuevas herramientas tecnológicas que permitan solucionar el problema del uso de tecnologías discontinuadas.

PALABRAS CLAVE: Telegrama de datos actuales, Rendimiento acústico y dinámico, torpedo SST-4, Sistema de control de tiro.

ABSTRACT

The SST-4 torpedo is an underwater weapon remotely guided from the submarine units' control system by means of a communication cable. The research project deals with the Modernization of the current data telegram with graphic interface for the SST-4 torpedo, the purpose is to keep the torpedo updated information of acoustic and dynamic performance during its movement under water, having the information allows to the system of control of shot the improvement of the tactical system of operation and control.

Currently the communication system of the current data return telegram of the SST-4 torpedo is limited, due to its technology from the 70s, which only transmits the "control pulse and target detection" signals, limiting the tactical operation in the firing and control system of submarine units.

This Master's thesis deals with the design and implementation of the current data telegram with graphic interface for the anti-surface torpedo SST-4, applied to the military sector of National Defense, which will contribute to the system of firing and control of the submarine units belonging to the Navy of Peru, oriented to improve its tactical operations of combat, to have the real location of the torpedo during its effective run through the current data telegram the torpedo will be able to transmit its acoustic and dynamic performance, in this way the data will be visualized in a graphic interface installed in the submarine units.

The research is experimental and applicative (quantitative approach), the purpose of this research is to apply scientific knowledge to transmit the current data telegram, through experimental research we can manipulate variables related to acoustic and dynamic signals, finally the applied research allows us to assign the result obtained from an experimental research with the need to look for new technological tools that allow solving the problem of the use of discontinued technologies.

INTRODUCCIÓN

El torpedo SST-4 (“Special Surface Target 4), es un arma submarina que puede ser disparado desde una unidad submarina o de superficie cuya finalidad es destruir blancos de superficie, está diseñado para ser guiado remotamente desde el sistema de control de tiro (SCT) de la unidad lanzadora a través de un cable de comunicación de doble núcleo.

Actualmente el sistema de comunicación del telegrama de retorno de datos actuales del torpedo SST-4 es limitado, debido a su tecnología proveniente de la década del 70, el cual solo transmite las señales de “pulso de control y detección de blanco”, limitando la operación táctica en el sistema de disparo y control de las unidades submarinas.

La presente tesis de Maestría trata del diseño e implementación del telegrama de datos actuales con interfaz gráfica para el torpedo anti superficie SST-4, aplicado al sector militar de Defensa Nacional, el cual permitirá contribuir al sistema de disparo y control de las unidades submarinas pertenecientes a la Marina de Guerra del Perú, orientado a mejorar sus operaciones tácticas de combate, al tener la ubicación real del torpedo durante su corrida efectiva mediante el telegrama de datos actuales el torpedo podrá transmitir su performance acústica y dinámica, de esta manera los datos serán visualizados en una interfaz gráfica instalado en las unidades submarinas.

De esta manera este plan de tesis tratará en el Capítulo I, sobre la presentación del problema objeto de estudio, es decir el planteamiento y formulación del mismo, los objetivos, la justificación y por último la aplicación hacia la cual está dirigido el trabajo.

En el Capítulo II, se expone los antecedentes relevantes a nivel nacional e internacional, además el marco teórico relacionado al concepto del sistema de comunicación, la estructura del diseño del protocolo de comunicación, las señales analógicas y digitales de los sensores o componentes electrónicos usados en la adquisición de datos, así como el lenguaje de programación usado en los controladores y el lenguaje de programación usado en la interfaz gráfica.

En el Capítulo III, se definen las variables utilizadas y la operacionalización de éstas, además de la hipótesis general e hipótesis específicas.

En el Capítulo IV, se analiza los lineamientos metodológicos empleados para cumplir el objetivo de innovación tecnológica, enfocando el tipo de investigación, diseño, así como también el universo, muestra y variables.

Finalmente se presenta el cronograma de actividades y presupuesto.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Determinación del Problema.

El torpedo SST-4 fue recibido entre los años 1974 y 1975 por la Armada Peruana conjuntamente con los submarinos tipo 209, se trataba entonces, de una de las armas submarinas más modernas de la época.

El torpedo SST-4 tiene un sistema de comunicación mediante un cable filo guiado de 26 Kilómetros conectado al sistema de control de tiro (SCT) para ser guiado remotamente a través del telegrama de comando, así mismo el torpedo envía información mediante pulsos generados, conocido como telegrama de retorno. El telegrama de retorno de los torpedos SST-4 es muy limitado, ocasionando un gran margen de error al guiar el torpedo hacia el blanco, debido a que el operador del sistema de control de tiro (SCT) no tiene la información en tiempo real de la ubicación del torpedo durante su corrida efectiva, las señales del telegrama de retorno son:

- **Pulso de Control** (Señal de comunicación entre el torpedo y la unidad submarina).
- **Distancia de seguridad cubierta** (Señal que indica que el torpedo cubrió las 300 yardas desde el punto de disparo).
- **Señales del blanco “TC, TD, HH”** (Señal proveniente del sonar de la cabeza acústica con respecto al blanco).

Los torpedos SST-4 fabricados exclusivamente para la Marina Alemana cuentan con un telegrama de retorno que utilizan un protocolo serial compuesto por 16 palabras de 16 bits cada uno, en este telegrama se envía información de los sensores del torpedo y permiten tener en tiempo real la ubicación del torpedo durante su recorrido, la información que está compuesto el mencionado telegrama consta de las señales de rumbo, profundidad, balanceo, cabeceo, velocidad e información del blanco, este último producido por el sonar de la cabeza acústica.

En la década del 80 se hicieron cotizaciones para dotar a los torpedos SST-4 con el Telegrama de retorno, pero por razones presupuestales no se hicieron por el alto costo aproximadamente de \$ 35,000.00 (TREINTA Y CINCO MIL DOLARES AMERICANOS) por unidad (torpedo), sin incluir el costo de modificación en el Sistema de Control de Tiro.

En el año 2007 se presentó la propuesta de Investigación y Desarrollo por un grupo de investigadores de la Marina de guerra del Perú, la propuesta se enfocaba en reemplazar el sistema de comunicación actual del torpedo SST-4, la propuesta consistía en cambiar el cable de filo guía a fibra óptica donde se incluía como parte del proyecto incluir el telegrama de retorno, el mencionado proyecto no se llegó a desarrollar por razones presupuestales y el alto costo del cambio del cable de filo guía por el de fibra óptica.

1.2 Formulación del Problema

1.2.1 Problema general:

¿Cómo mejorar la operación táctica en el sistema de disparo y control de las unidades submarinas pertenecientes a la Marina de Guerra del Perú al no tener la ubicación real del torpedo SST-4 durante su corrida efectiva?

1.2.2 Problemas específicos:

P.E.1 ¿Qué dispositivo electrónico permitirá leer correctamente las señales digitales y analógicas de los sensores y componentes electrónicos del torpedo SST-4 para diseñar el telegrama de datos actuales?

P.E.2 ¿Cómo codificar y fabricar el telegrama de datos actuales con los parámetros obtenidos de las señales digitales y analógicas para ser adaptados al sistema de comunicación del torpedo?

P.E.3 ¿Cómo decodificar el telegrama de datos actuales obtenidos en el módulo receptor del sistema de comunicación de las unidades submarinas y de la plataforma de Pruebas (Test-Stand) de la Estación de Armas Submarinas?

P.E.4 ¿Qué software permite obtener una interfaz gráfica de alta calidad y performance para la visualización del telegrama de datos actuales en la plataforma de pruebas (Test-Stand)?

1.3 Definición de objetivos

1.3.1 Objetivo general

Implementar el Telegrama de datos actuales con interfaz gráfica para el torpedo anti-superficie SST-4, el cual permitirá que el torpedo tenga la capacidad de transmitir la información de los sensores de los sistemas de control hacia las unidades submarinas para validar el correcto funcionamiento y a la vez tener la ubicación real del mismo.

1.3.2 Objetivos específicos

O.E.1 Diseñar y confeccionar tarjetas electrónicas para adquirir los datos de las señales digitales y analógicas de los sensores y componentes electrónicos del torpedo SST-4.

O.E.2 Diseñar y confeccionar una tarjeta electrónica que permita codificar los datos obtenidos de los sensores en un protocolo serial denominado telegrama de datos actuales, para ser transmitidos al sistema de comunicación del torpedo SST-4.

O.E.3 Diseñar y confeccionar una tarjeta electrónica en el sistema de control de tiro, que permita decodificar telegrama de datos actuales provenientes del torpedo.

O.E.4 Desarrollar una interfaz gráfica (GUI), en la plataforma Visual Studio en lenguaje de programación C#, para visualizar en forma gráfica la información obtenida del telegrama de datos actuales del torpedo SST-4.

1.4 Justificación

La antigüedad y limitación del sistema de comunicación de los torpedos SST-4 comparada con las opciones actuales dentro del campo de la tecnología de las armas submarinas, hace que los usuarios del sistema de control de tiro enfrenten dificultades para el empleo táctico del arma. El torpedo SST-4 está discontinuado por la empresa Alemana Atlas Elektronik no contando con ningún soporte técnico para la modernización del telegrama de retorno en el sistema de comunicación.

Debido a los avances tecnológicos la Marina de Guerra del Perú viene apostando en la investigación y desarrollo tecnológico con el fin de reducir la dependencia de los países fabricantes al desarrollar tecnología propia a un bajo costo y así prolongar la vida útil de los diferentes sistemas de armas. Dentro de los proyectos de investigación desarrollados por la Marina de Guerra del Perú se cuentan con los sistemas de Control de Tiro del tipo ARGOS y KALLPA, desarrollados para el control y disparo de los torpedos tipo SST-4 y SUT-264, estos proyectos están limitados con respecto al torpedo SST-4, debido a la limitación del telegrama de retorno.

El diseño y desarrollo del telegrama de datos actuales a los torpedos modelo SST-4 resulta una propuesta adecuada a la modernización de las unidades submarinas que fácilmente se integraría a los nuevos sistemas de control tiro tipo ARGOS y KALLPA, permitiendo tener la posición real del torpedo con los datos actuales del rumbo, velocidad y profundidad.

CAPÍTULO II

MARCO TEÓRICO

2.1 Estado del arte

En la revisión de investigaciones relacionadas al diseño e implementación del telegrama de datos actuales para el torpedo SST-4, se evidenció que no existen investigaciones relacionadas al tema debido a que la información de este tipo de armamento resulta ser reservado y confidencial por el fabricante, motivo por el cual se fundamenta la importancia de realizar este estudio en beneficio de la Marina de Guerra del Perú. A continuación, se describirán los manuales del torpedo y una serie de investigaciones que están relacionados con los fundamentos teóricos del trabajo de investigación, las cuales servirán como base científica para resolver la problemática de la presente investigación a desarrollar.

(Alemania Patente nº 1392263-10, 1970) Manual: “Descripción de la cabeza acústica del torpedo SST-4”, en este manual se especifica el funcionamiento de la cabeza acústica, desde la recepción de la señal de un blanco para ser evaluado hasta obtener la información del ángulo de recepción acústica, Este manual consta de la parte teórica y una parte de diagramas eléctricos, los cuales serán utilizados para medir los niveles de voltajes de las señales analógicas (Guiado horizontal y Voltaje de control) así como las señales digitales (Pulso de control, Contacto de blanco, Detección de blanco y Pérdida de blanco), que provienen del sistema de

recepción, estas señales son enviadas a la unidad SPU-A (Unidad de procesamiento de señal) de la sección electrónica del torpedo SST-4 donde se implementará la tarjeta de adquisición de datos.

(Alemania Patente nº 1392263-11, 1970) Manual: “Descripción de la sección electrónica del torpedo SST-4”, en este manual se especifica el funcionamiento de las unidades que se encuentran en esta sección como: El Giróscopo que informa el Rumbo, cabeceo y balanceo; las celdas de presión que informan la profundidad actual; las señales que llegan de la cabeza acústica TC, TD, HH y guiado horizontal; los telegramas del sistema de comunicación, entre otros, toda esta información es procesada en la unidad de procesamiento de señales (SPU-A) donde se colocará la tarjeta de adquisición de datos, así mismo mediante los diagramas eléctricos se describen los niveles de tensión de todas las señales antes mencionados.

(Alemania Patente nº 1392263-13, 1970) Manual: “Descripción de la plataforma de pruebas Test-Stand del torpedo SST-4”, en este manual se especifica el funcionamiento de la consola de pruebas, el cual es utilizado para efectuar las pruebas generales que se realizan a los torpedos SST-4, dentro de esta plataforma se encuentra el sistema de comunicación para enviar el telegrama de comando (Ordenes al torpedo) así como para visualizar el telegrama de datos actuales, el cual es visualizado en una lámpara indicadora, en esta plataforma se encuentra la unidad de procesamiento de señales (SPU-B) el cual recibe el telegrama de datos

actuales proveniente del torpedo, en esta unidad se confeccionará la tarjeta decodificadora del telegrama de datos actuales, el cual enviará dicha información a la computadora para ser visualizados en una interfaz gráfica.

(De la Horra, 2013) investigó: “Sistemas de adquisición de datos basados en la plataforma Arduino: aplicaciones a Matlab, Simulink y Android”. Su estudio realizado permitió desarrollar el hardware de un sistema básico de adquisición de datos para su empleo a nivel educativo, se usó las tarjetas electrónicas de la plataforma arduino. Durante el desarrollo de su proyecto se logró realizar la comunicación entre el controlador y la computadora por medio de un módulo que utiliza el protocolo Ethernet, de esta manera el hardware se puede enlazar con la computadora para obtener la información de los sensores, como herramienta utilizo la plataforma del el programa Matlab donde se utiliza un algoritmo de control, permitiendo de esta manera transmitir la información de los sensores utilizando el protocolo Ethernet a una estación remota para monitorear la información obtenida.

(Nuñez, 2014) investigo: “Diseño e integración de un sistema de adquisición de datos mediante el uso de Arduino y Raspberry-pi”. Este trabajo se basó en la integración de una tarjeta electrónica para adquirir datos de señales digitales y analógicas de los sensores del laboratorio de control de procesos con la finalidad que los alumnos tengan la capacidad de monitorear los datos obtenidos. El monitoreo de los sensores es visualizado en una computadora mediante el desarrollo de un entorno gráfico para el cual

utilizaron como herramienta el software Labview, el hardware desarrollado se comunica con la computadora por medio de un adaptador de comunicación serial de TTL a USB. La herramienta utilizada para el desarrollo del entorno grafico es el lenguaje de programación C#, partiendo desde una función básica para integrar un diseño avanzado donde se declaran las propiedades y métodos de las clases para realizar un programa estructurado y por último se detalla la forma de la compilación del desarrollo del código.

(Sharp, 2015) Publico: "Microsoft Visual C# Step by Step". El desarrollo de este libro se orienta en la fundamentación de los conocimientos básicos para el desarrollo de los algoritmos de uno de los lenguajes de programación orientado a objetos más utilizado, el libro explica paso a paso como aprender a utilizar la herramienta de programación orientada a objetos utilizando la plataforma de Microsoft Visual C#, desde un enfoque básico y sencillo, de esta manera el investigador puede contar con una herramienta que le permita interactuar con el desarrollo de la programación orientado a objetos. Este libro está bien estructurado para iniciar el aprendizaje del lenguaje de programación C#, donde el autor detalla los conceptos y funciones básicas de los algoritmos a ser utilizados y desarrolla diversas aplicaciones, partiendo desde niveles básicos hasta llegar a realizar aplicaciones de niveles más complejos.

(Espinoza, Sistema de combate Kallpa I, 2015) Desarrollo “SISTEMA DE COMBATE “KALLPA I”, en este proyecto se reemplazó el sistema de combate original SINBADS VM8/24-74PE por un nuevo sistema, emulando las bondades de su predecesor e incorporando una nueva interface hombre-máquina, producto de la experiencia y participación de los usuarios. Con la realización de este proyecto se ha recuperado y elevado la confiabilidad del sistema de armas de las unidades submarinas tipo “ANGAMOS” así como la incorporación de nuevos componentes disponibles en el mercado internacional, sin caer en la dependencia del proveedor exclusivo y disponer de una arquitectura abierta a la incorporación de más tareas y, funciones que se requieran para un óptimo funcionamiento de la plataforma de combate.

Asimismo, este proyecto comprende también el desarrollo de algoritmos para estimar los parámetros de los blancos, una mejor interfaz hombre - máquina del sistema de armas para resolver el problema de disparo de los torpedos SUT-264 y SST-4.

(Espinoza, Sistema de combate Kallpa II, 2015), desarrollo el “SISTEMA DE COMBATE KALLPA II”, en este proyecto se mejoró el proyecto “KALLPA I” para el cual se desarrolló un módulo de disparo para las unidades submarinas el cual es capaz de realizar la secuencia de disparo de un torpedo y una vez disparado guiarlo a través de un cable de comunicaciones.

2.2 Marco teórico

2.2.1 Torpedo SST-4

El torpedo SST-4 es una versión modificada del Torpedo Alemán “Seal” y ha sido desarrollado especialmente como un arma para atacar blancos de superficie, puede ser disparado tanto desde Submarinos como Buques de Superficie. (AEG TELEFUNKEN, Funtional description, 1970).

El SST-4 es un torpedo de dos velocidades, controlado y propulsado eléctricamente mediante un sistema buscador acústico activo, pasivo. Utiliza un cable de cobre de 26 kilómetros, como medio de comunicación entre el torpedo y el buque lanzador. Fue fabricado por la empresa AEG-TELEFUNKEN en Alemania, en la figura 1 se observa el torpedo de combate SST-4, completamente ensamblado con todas sus secciones.

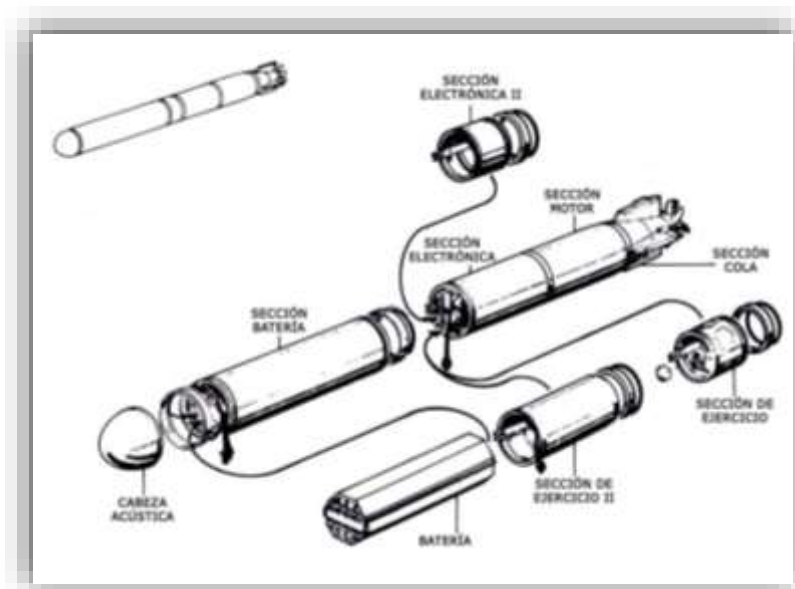


Figura 1: Secciones principales del torpedo SST-4.
Fuente: (AEG TELEFUNKEN, Funtional description, 1970)

2.2.1.1 Descripción General.

El torpedo consiste de un cuerpo cilíndrico compuesto por varias secciones llamadas componentes principales, las que se encuentran conectadas entre sí por medio de aros de unión (AEG TELEFUNKEN, Funtional description, 1970), en la figura 2 se observa las secciones que conforman el torpedo SST-4.

1. Cabeza acústica.
2. Sección explosiva.
3. Sección batería
4. Sección electrónica II
5. Sección electrónica
6. Sección motor y cola
7. Sección cola

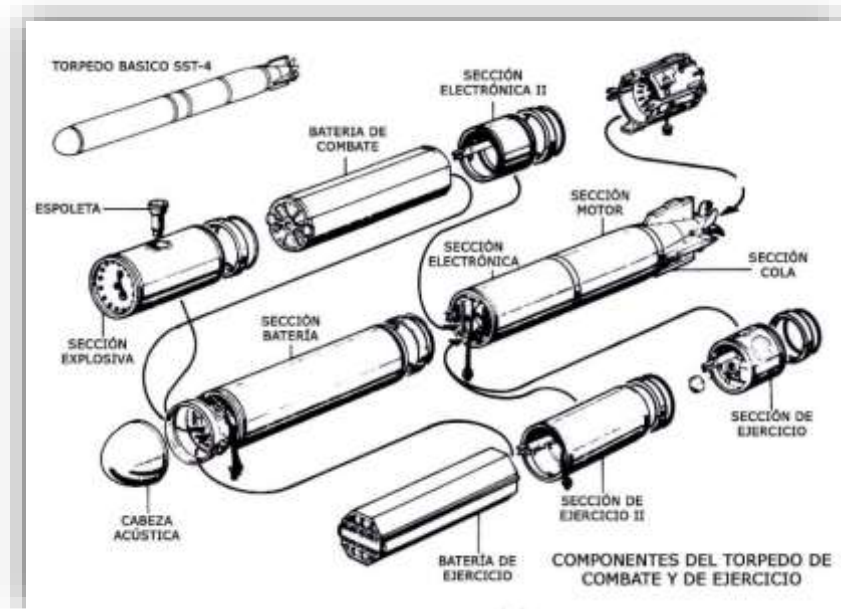


Figura 2: Partes del torpedo SST-4, versión de combate y ejercicio.

Fuente: (AEG TELEFUNKEN, Exercises Section, 1970)

a) Datos Mecánicos y Pesos

Los datos mecánicos y pesos del torpedo de combate SST-4, incluyendo la carga explosiva y la batería de combate, son las siguientes.

- Diámetro : 534 mm. +0.3/ - 0.5 mm
- Longitud total sin el casquete : 6080 mm \pm 10mm.
- Longitud total con el casquete : 6552 mm \pm 10mm.
- Peso sin el casquete : 1381.7 \pm 16.5/-14.5 kg.
- Peso con el casquete : 1469.9 kg. + 21.5/-19.5 kg.
- Desplazamiento : 1211.3 dm³
- Boyantes (agua = 1.025 kp/dm³) : -140.1 Kg +16.5/- 14.5 Kg.

b) Datos Técnicos

- Peso de la carga explosiva : 255 kg \pm 10 kg.
- Profundidad de Operación : 2 á 100 m.
- Longitud del cable de guiado : 18,4 km \pm 1 %
- Alcance del guiado : 18 km.
- Distancia de corrida en baja velocidad : 28 km.
- Distancia de corrida en alta velocidad : 12 km.
- Baja Velocidad : 23 nudos \pm 1.5
- Alta Velocidad : 34 nudos \pm 1.5
- Distancia de seguridad : 300 m.
- Tiempo de corrida a baja Velocidad : 44 minutos Min.
- Tiempo de corrida en alta Velocidad : 13 minutos Min.

c) Señales de transmisión y recepción.

Los siguientes datos son transmitidos a través del alambre de comunicación:

a. Señales de órdenes:

- **Rumbo:** 0° á 360°, codificado en binario
0.35° de graduación
- **Profundidad:** 0 a 342 pies, codificado en binario
0 a 31.5 pies: graduación 0.5 pies
32 a 342 pies: graduación 5 pies
- **Velocidad:** Torpedo Off / Baja / Alta
- **Búsqueda:** Homing ON/OFF
- **Modo acústico:** Activo / Pasivo
- **Programa Interno:** Sinusoidal (snake).
Círculo izquierdo.
Círculo derecho.
Rumbo deseado.
Rumbo programado.

b. Señales de Retorno:

- Situación (status, pulso de control)
- Distancia de Seguridad
- Cable roto
- Contacto con el blanco
- Adquisición del blanco
- Pérdida del blanco

2.2.1.2 Sistema de detección acústica.

El sistema de detección acústica del torpedo SST-4 se observa en la figura 3, ha sido diseñada para dirigir al torpedo en modo automático cuando éste haya detectado un blanco. Después de que la cabeza acústica ha detectado un blanco, transmite una orden de contacto con el blanco, por medio de la cual el torpedo abandona al rumbo ordenado y procede directo hasta que reciba una orden de dirección. Al mismo tiempo, las órdenes “contacto con el blanco” (TC) y “detección del blanco” (TD), son recibidas (cerca de 2 segundos después del contacto con el blanco) y enviadas de regreso en forma de señal, al Sistema de Control de Tiro, por medio de varias frecuencias instantáneas, a fin de informar al operador. Si se ha perdido el blanco, se hace efectivo un programa “giro alrededor” (HH o TA) que depende de varias condiciones en la operación pasiva y activa. La señal “giro alrededor”, también es enviada al Sistema de Control de Tiro.

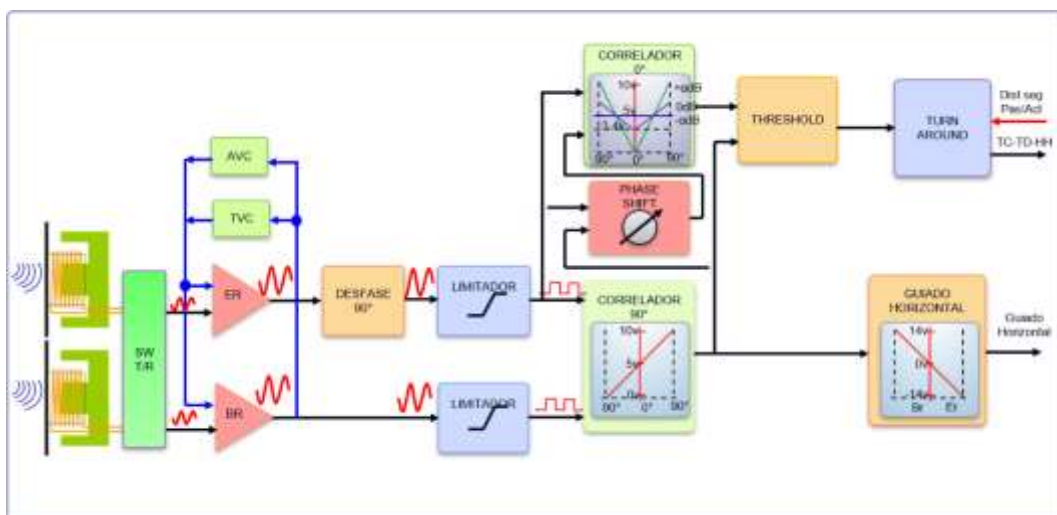


Figura 3: Sistema de detección acústica del torpedo SS-4.

Fuente: Elaboración propia.

El sistema de detección acústica está compuesto por 12 sensores acústicos dispuestos simétricamente. Los sensores son del tipo magnetostrictivo, es decir, que cuando son energizados por una señal, las placas núcleo son comprimidas de acuerdo a la presión del sonido, produciendo un voltaje de CA correspondiente. Los sensores son subdivididos en babor y estribor para la medición de ángulo de marcación en el plano horizontal. En la figura 4 se muestra el arreglo de transducer visto desde el eje longitudinal. El sonido es recibido prácticamente como un haz paralelo de ondas.

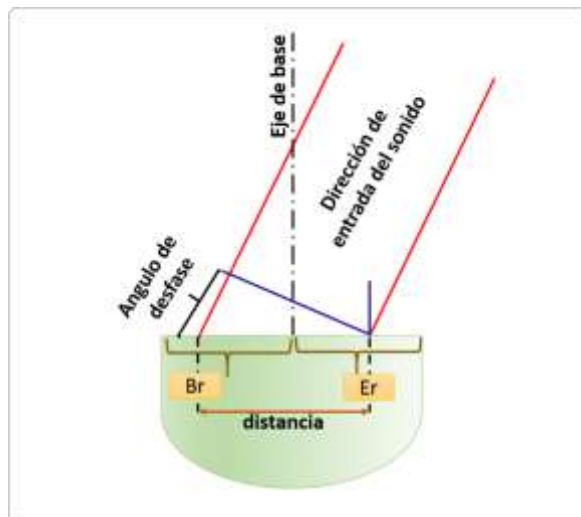


Figura 4: Arreglo esquemático del sonar del torpedo SS-4.

Fuente: Elaboración propia.

Para el cálculo de la longitud de la onda se tiene:

$\lambda = \text{Longitud de la onda}$

$c = \text{Velocidad del sonido en el agua} = 1500 \text{ m/seg}$

$f = \text{frecuencia} = 21.4 \text{ Khz}$

$$\lambda = \frac{c}{f} = \frac{1500}{21400} = 0.07 \text{ metros}$$

El ángulo de fase es computado en el bloque del sistema de medición de fase. Los amplificadores son controlados por los sistemas de C.A.V. (Control Automático de Volumen) y de C.T.V. (Control de Tiempo del Volumen). Las Salidas de los amplificadores controlables, envían las señales recibidas comprimidas en su dinámica. Sobre la base de la información de la fase, es posible determinar el ángulo entre la dirección del ruido entrante y el eje longitudinal del torpedo.

a) Contacto con el blanco

En modo pasivo mide la razón ruido a señal si es ≥ 0 db, o si en la operación activa, la razón del eco-pulso, excediera el valor de 3:1, entonces el torpedo correrá recto. La señal de contacto con el blanco es suprimida inmediatamente en la operación pasiva, cuando se interrumpe el contacto acústico y en la operación activa, en el caso de que un eco no haya excedido el límite de receptividad.

b) Detección del Blanco

En modo pasivo, si la razón ruido a señal de 0 dB se excediera por más de 2 segundos o si en la operación activa, dos ecos sucesivos excedieran la razón 3:1 del eco-pulso; La señal de detección del blanco es suprimida inmediatamente en la operación pasiva, si el contacto acústico es interrumpido y en la operación activa, 8 segundos después de que un eco no haya excedido el límite de receptividad, con tal que los siguientes ecos no excedan de nuevo, el límite de receptividad.

En las figuras 5 y 6 se muestran las condiciones para que la cabeza acústica del torpedo SST-4 tenga las condiciones de Contacto de blanco "TC" y detección de blanco "TD" en modo pasivo y modo activo.

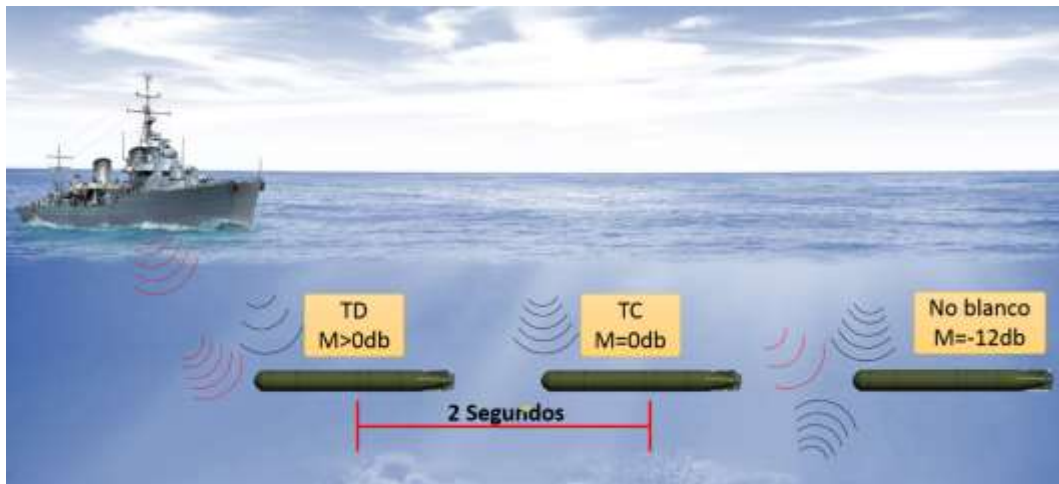


Figura 5: Condiciones para TC y TD en modo pasivo.

Fuente: Elaboración propia.

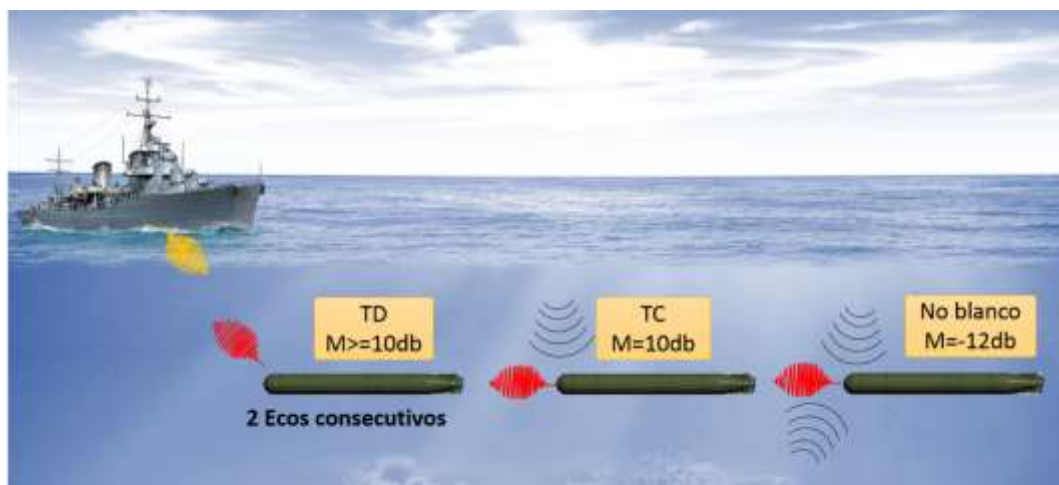


Figura 6: Condiciones para TC y TD en modo activo.

Fuente: Elaboración propia.

c) Perdida de blanco (HH o TA)

Para que se produzca perdida de blanco (HH o TA) en modo pasivo, se debe tener TC y TD; el ángulo torpedo-blanco es $> 12^\circ$ a estribor o babor; en modo activo, la distancia torpedo-blanco es < 80 mts. En la figura 7 se observa el modo de perdida de blanco en modo pasivo.

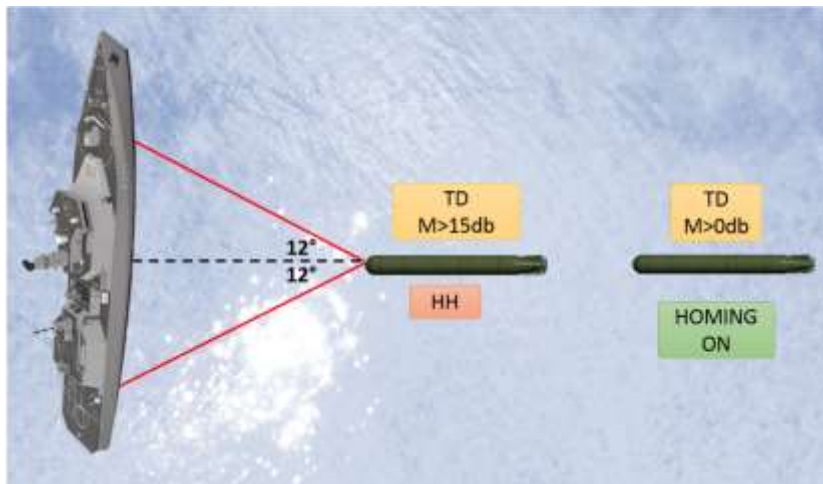


Figura 7: Condiciones para Perdida de blanco en modo pasivo.

Fuente: Elaboración propia.

d) Señal de Guiado Horizontal

Es transmitida al torpedo, es aceptada si se ha detectado al blanco.

e) Correlador de 90°

Esta señal de correlación es usada en la operación pasiva para la medición del ángulo de marcación (en conjunto con la razón ruido a señal)

f) Correlador de 0°

Esta señal de correlación es usada en la operación pasiva, para la medición de la razón ruido a señal.

2.2.1.3 Sistema de comunicación del torpedo SST-4.

El sistema de comunicación de la consola de pruebas (Test-Stand) con el torpedo SST-4, consiste en transmitir órdenes y recibir los datos actuales, el sistema de comunicación está conformado por las siguientes unidades:

- SPU-B (Unidad de procesamiento de señales, lado del buque).
- NVS-B (Transreceptor, de lado del buque).
- 1 bobina de cable de guiado, del lado del buque.
- 1 SPU-A (Unidad de procesamiento de señales, lado del torpedo).
- NVS-A (Transreceptor del lado del torpedo).
- 1 segunda bobina de cable de guiado en el torpedo.

En la figura 8 se observan los transreceptores de la consola de pruebas Test-Stand (NVS-B) y del lado del torpedo (NVS/A) conectados al terminal de cada bobina y las unidades de procesamiento de señales (SPU-B y SPU-A) conectados entre los módulos transreceptores: (NVS-B y NVS-A).

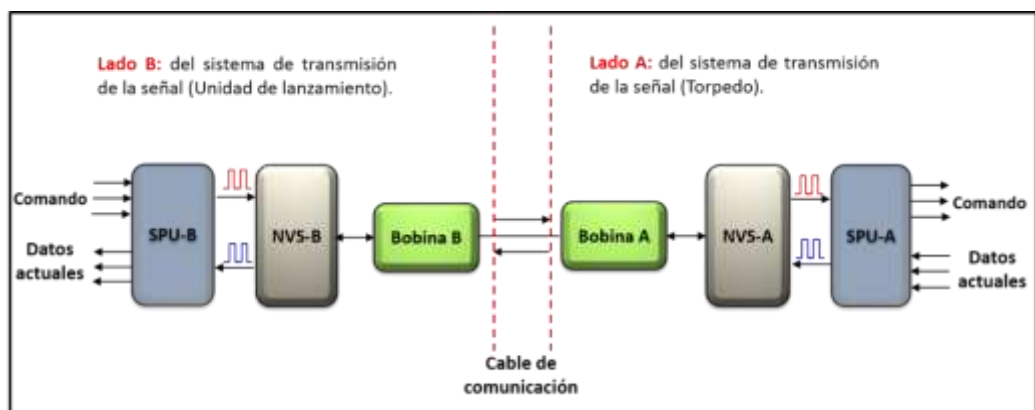


Figura 8: Sistema de comunicación del torpedo SS-4.

Fuente: Elaboración propia.

a) Telegrama de comando.

El telegrama de comando está formado por señales codificadas binarias y es transmitida a una velocidad máxima de 206 baudios (bit/segundo). Las señales recibidas son enviadas al S.P.U. como sigue:

“O” = 0 á + 0.6 V; carga: ≥ 5 m.A.

“L” = + 6.5 á + 30 V; carga: ≥ 1 m.A.

El receptor de órdenes del S.P.U. consigue, vía el transreceptor, una información continua (tren de pulsos) desde el equipo del buque (computador y sistema de comunicación del buque). La información consiste de marcas simples de corriente, que sólo pueden tener las condiciones:

L = Marca = + 12 V

O = Intervalo = 0 V

La velocidad de transmisión de la información, es decir, el número de bits por segundo es de 206 baudios. Un telegrama consiste de 8 grupos o palabras. Cada palabra tiene 8 bits, de los cuales, los bits 1 al 7, son para información, mientras que el bit 8 siempre es cero, para indicar el final de la palabra.

La señal de inicio (7 x 1) de la palabra 8, siempre es fija y efectúa la sincronización para la conversión paralelo-serie en el receptor del telegrama del S.P.U.

El convertidor serie-paralelo, proporciona siete bits de cada palabra. La operación es controlada por un reloj generador de frecuencias, el cual sincroniza en la palabra 8. En el caso de un cable roto, deberá pararse el reloj generador para que los últimos valores transmitidos permanezcan en el almacenador de salidas.

Como una protección contra errores en la transmisión de comunicación, el 8vo. bit de cada palabra es controlado; este bit es siempre cero. Si el bit 8 no fuera cero, se evitará la entrega de la palabra incorrecta a la memoria de salida; en la figura 9 se observa el telegrama de comando del torpedo SST-4.

Profundidad								Rumbo								Rumbo								programa				velocidad			
32 ft								Dirección del Rumbo								Dirección del Rumbo								snake				activo / pasivo			
160 16								180°								5.62°								circ. right				Alta vel.			
80 8								90								2.81								circ. left				Baja vel.			
40 4								45								1.4												Torp. off			
20 2								22.5								0.7															
10 1								11.25								0.35															
5 0.5																															
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
grupo 1								grupo 2								grupo 3								grupo 4							

Libre								Libre 7 x 0								Palabra de control 7 x 0								Palabra de arranque 7 x 1							
Homing on / off																															
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
grupo 5								grupo 6								grupo 7								grupo 8							

Figura 9: Telegrama de comando del torpedo SST-4.

Fuente: (AEG TELEFUNKEN, Funtional description, 1970)

b) Telegrama de retorno.

Las señales de retorno son transmitidas como una forma codificada binaria, con una velocidad máxima de 206 baudios (bit/segundo). El NVS recibe las señales del S.P.U. como sigue:

“O” = 0 á + 1.5 voltios; carga: ≤ 0.56 m.A.

“L” = + 6.5 á + 12 voltios; carga: ≤ 6.95 m.A.

Una señal de retorno de un voltaje alterno, de frecuencia y amplitud constante, es transmitida de regreso al equipo transreceptor del buque. La base de tiempo de la transmisión, es controlada por el S.P.U. en las siguientes formas:

- Sin señal
- Señal continúa
- Señal pulsada

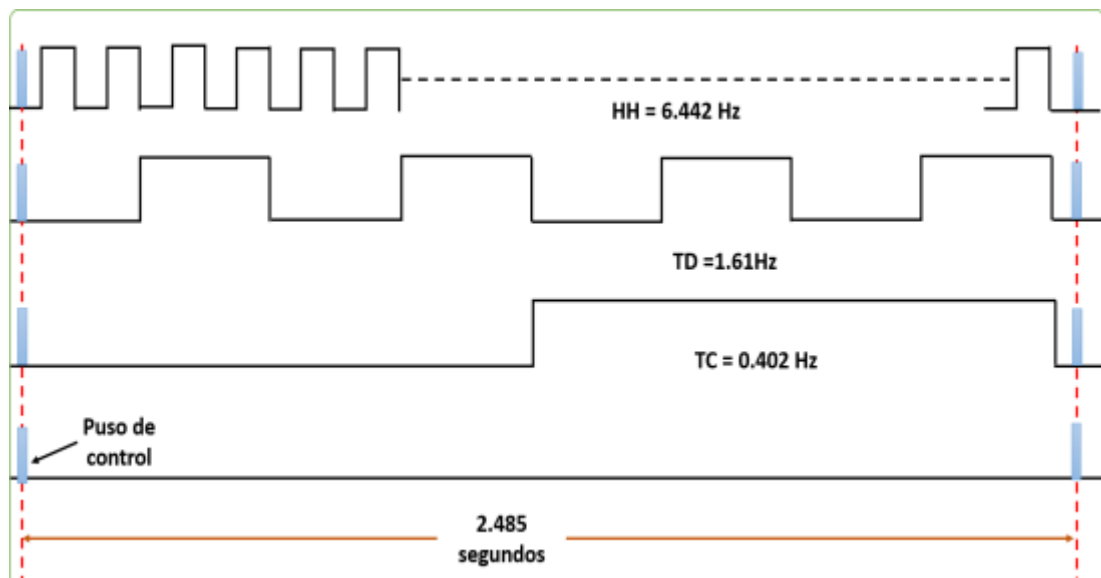


Figura 10: Telegrama Actual del torpedo SST-4.

Fuente: Elaboración propia.

c) NVS-A (Transreceptor).

El sistema de comunicación consiste de las unidades de preparación de la señal del transreceptor que se encuentra en el interior del torpedo (TELEFUNKEN, NVS-A, 1970)

El NVS-A contiene los dispositivos de transmisión y de recepción, incluyendo a los moduladores y de demoduladores para la transmisión de las señales a través del cable de guiado.

En la figura 11 se observa el módulo NVS-A instalado en la sección electrónica del torpedo SST-4, esta unidad utiliza un circuito electrónico para discriminar el telegrama de comando y el telegrama de datos actuales.



Figura 11: Modulo NVS-A del torpedo SST-4.

Fuente: Elaboración propia.

d) SPU-A (Computadora del torpedo SST-4).

El SPU-A es el computador del torpedo SST-4. es la descifradora de las órdenes que han sido transmitidas a lo largo del cable vía el transreceptor, entregando a los diferentes sistemas de control del computador del torpedo SST-4; esta unidad también realiza la preparación y codificación de las señales de retorno.

El amplificador de control de los ejes de estabilización (Rumbo, Profundidad y Balanceo), los circuitos de control y el generador de encendido (45 KHz), también se encuentran ubicados en el interior del SPU-A; las diferentes formas de operación del torpedo, de acuerdo a las diferentes condiciones y el cambio para pruebas (prueba de estabilización y acústica), son ordenadas desde el SPU-A; en la figura 12 se observa el módulo SPU-A instalado en la sección electrónica del torpedo SST-4.



Figura 12: Modulo SPU-A del torpedo SST-4.

Fuente: Elaboración propia.

2.2.1.4 Sistema de Estabilización y control del torpedo SST-4.

Las componentes del sistema de estabilización aseguran que:

- El torpedo tomará y mantendrá el rumbo y profundidad ordenados.
- Los movimientos de balanceo y cabeceo sean compensados.

El proceso completo de la estabilización es efectuado por un circuito de control; para la estabilización del torpedo se usan los siguientes valores:

- Rumbo
- Velocidad Angular del rumbo
- Angulo de Cabeceo
- Profundidad de Corrida
- Angulo de Balanceo

Los valores deseados para el rumbo y la profundidad de corrida, son producidos por el sistema de comunicación o por la cabeza acústica. El valor deseado para φ es 0° .

Los valores deseados, son graduados y mantenidos por medio de una estabilización, con tres curvas de control. Cada curva corresponde a uno de los siguientes valores de operación:

- Rumbo
- Profundidad
- Cabeceo
- Balanceo

Las componentes provistas para la estabilización, se agrupan como sigue:

- Control de Profundidad y Cabeceo.
- Control de rumbo y Balanceo.

a) Control del sistema de Profundidad y Cabeceo.

El sistema de control de profundidad y cabeceo se encuentra ubicada en la computadora del torpedo SST-4, está compuesto por un circuito de control para la profundidad, así como un circuito de control para el cabeceo, siendo el primero superpuesto al último. Los límites de estrato de 2 metros (límite de profundidad superior) y 100 metros (límite de profundidad inferior) están permanentemente fijos.

La profundidad verdadera es medida por las celdas de presión, el valor medido es comparado con la orden seleccionada para la profundidad desde el Sistema de Control de Tiro, el amplificador de control de profundidad convierte la diferencia de los dos valores en una orden de cabeceo y la transmite al amplificador de control de cabeceo. Allí el valor es comparado con el ángulo verdadero de cabeceo y luego transmitido al actuador de timón para la estabilización del torpedo en el plano vertical.

La salida del amplificador de control de cabeceo es limitada, esto da por resultado una máxima deflexión de timón de $\pm 15^\circ$ aproximadamente.

Otra orden de acción para los elevadores es provista por el integrador especial de profundidad, el que asegura que la diferencia de operación baje a cero si la profundidad deseada quedara incambiable.

Los actuadores de timón amplifican y convierten las señales dadas por el amplificador de actuadores, en deflexiones correspondientes de los timones.

Los amplificadores de control que comparan los valores deseados y verdaderos para la estabilización vertical del torpedo y proporciona señales dinámicas corregidas, para los actuadores de timón. El amplificador de actuadores se encuentra alojado en la computadora del torpedo. El receptor de órdenes alimenta a estos amplificadores con los valores deseados para el control de la profundidad; y la unidad central de giros proporciona el valor actual de cabeceo; el valor verdadero para la profundidad es provisto por las celdas de presión.

Los amplificadores de actuadores, son alimentados con los valores para la profundidad y corrección de profundidad. El factor de amplificación de los amplificadores de control, en la figura 13 se observa el sistema de control de profundidad y cabeceo del torpedo SST-4.

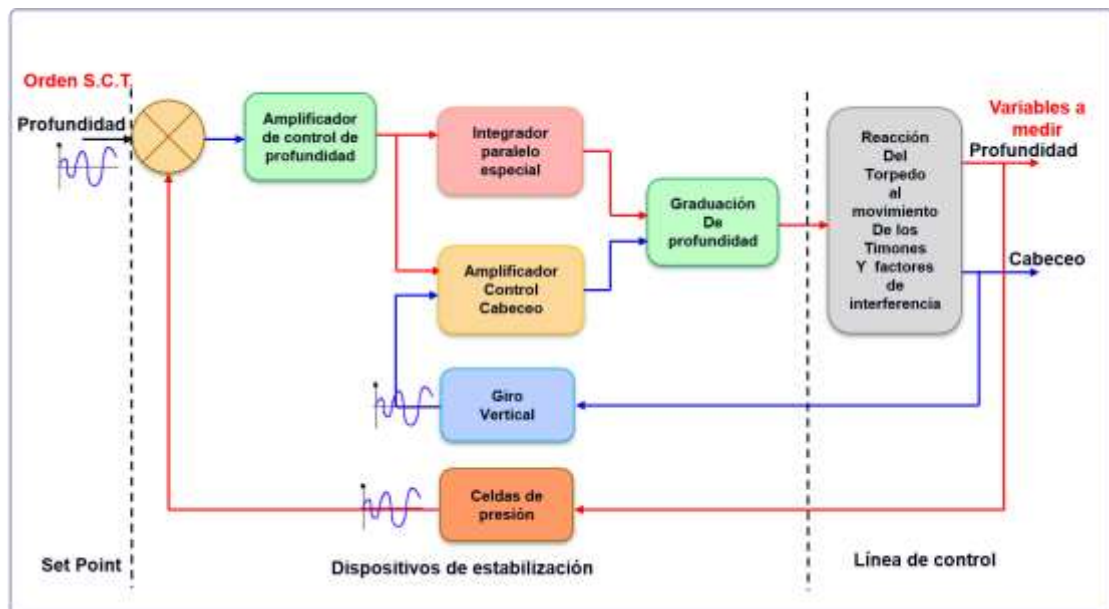


Figura 13: Control de Profundidad y Cabeceo del torpedo SS-4.

Fuente: Elaboración propia.

b) Control del sistema de Rumbo y Balanceo.

El sistema de control de rumbo y balanceo se encuentra ubicada en la computadora del torpedo SST-4, está compuesto por un circuito comparador digital (10 bits); este circuito determina la diferencia entre el rumbo verdadero y el deseado. Los siete números más bajos de estas diferencias, son convertidos por el amplificador de control del rumbo, en valores análogos, en tal forma que la diferencia análoga del ángulo del rumbo pueda variar entre $- 22.5^\circ$ y $+ 22.5^\circ$.

Los números restantes de esta diferencia, influyen un circuito lógico que verifica si el torpedo asume la nueva dirección en la ruta más corta posible, si los valores deseados fueran cambiados súbitamente por más por más de 180° .

Por otra parte, el amplificador de control de rumbo está provisto con amplificación variable de velocidad y con una limitación para la señal de salida, que da por resultado una máxima deflexión del timón del rumbo de $\pm 15^\circ$ aproximadamente.

La unidad de diferenciación digital, forma un voltaje desde el rumbo verdadero que es proporcional a la velocidad del ángulo del rumbo ($\dot{\Psi}^*$). Esta señal es llevada al amplificador de control del rumbo. Entonces se forma una curva de control para la velocidad del ángulo del rumbo, para asegurar el funcionamiento dinámico requerido de control del rumbo.

La unidad central de giro provee el ángulo verdadero de balance que es procesado en este amplificador de control y luego transmitido a los actuadores de timón. El valor deseado para el balance es siempre cero.

Los amplificadores de control comparan los valores deseados y verdaderos para la posición del torpedo y proporciona señales dinámicas corregidas, para los actuadores de timón. El receptor de órdenes alimenta a estos amplificadores con los valores deseados para el rumbo; y la unidad central de giros proporciona luego los valores verdaderos para el rumbo y balance; el valor verdadero para la profundidad es provisto por las celdas de presión; en la figura 14 se observa el sistema de control de rumbo y balanceo del torpedo SST-4.

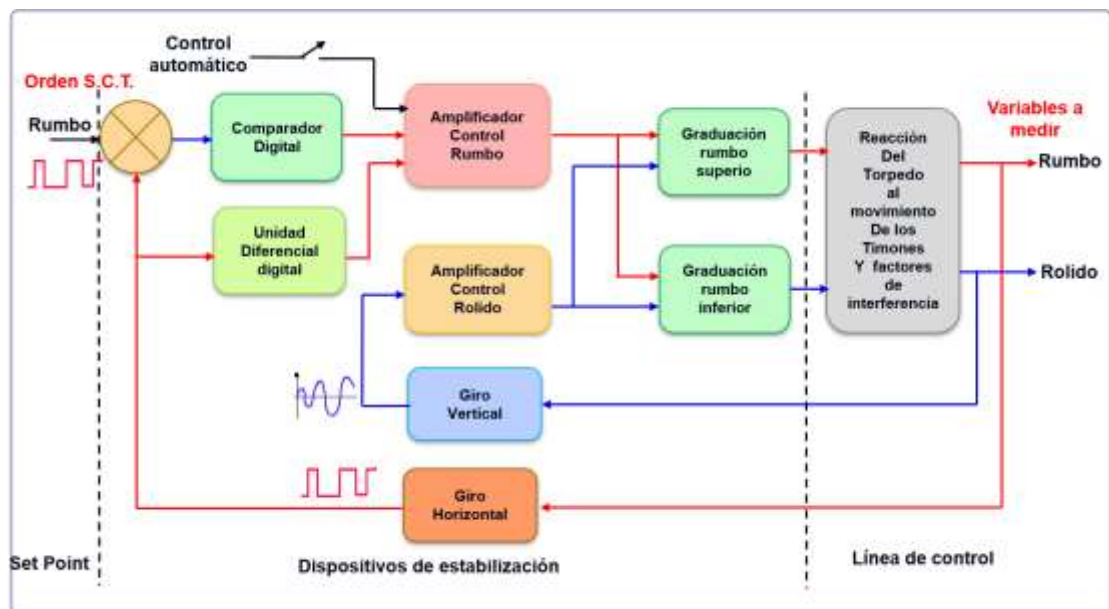


Figura 14: Control de Rumbo y Balanceo del torpedo SS-4.

Fuente: Elaboración propia.

2.2.2 Plataforma Arduino

La plataforma Arduino está basado en hardware y software libre, en cuanto al hardware incorpora un microcontrolador y el software utiliza el entorno de programación basado en lenguaje C (Arduino, 2017).

Existen diferentes modelos de placas, estas placas proveen puertos de entradas y/o salidas digitales, entradas analógicas, interfaces de comunicación serial y otros periféricos de comunicación, su costo es bastante económico comparado con otras plataformas de desarrollo y al ser libre se pueden conseguir el diseño de cada una de las tarjetas en la página oficial del fabricante. En cuanto al software, la plataforma provee de un entorno de programación basado en lenguaje C, este lenguaje es muy utilizado en diferentes plataformas debido a su gran potencialidad, además la comunidad de arduino provee una gran cantidad de librerías, el cual facilita la realización de algún proyecto, en la figura 15 se observa la placa y el entorno de programación de la plataforma Arduino.

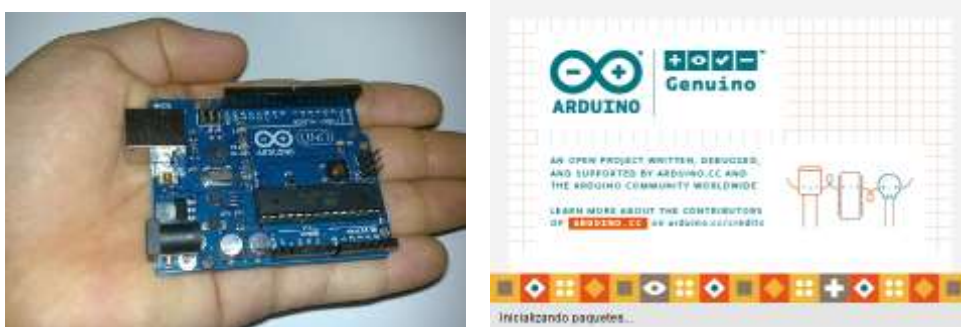


Figura 15: Hardware y software de la plataforma Arduino.

Fuente: Elaboración propia.

2.2.2.1 Características del hardware

La plataforma Arduino provee de un hardware basado en una plataforma completamente ensamblado listo para ser utilizado por los usuarios, al ser una plataforma Open Source (Hardware y Software libre), los usuarios finales pueden desarrollar sus propias placas debido a que los planos eléctricos del diseño esquemático y ruteado están disponibles para ser descargados dentro de la página oficial de Arduino, el desarrollo de estas placas utiliza como herramienta de diseño el software Eagle, que es un entorno de desarrollo de placas electrónicas.

En la figura 16 se observa el diagrama en bloques de la configuración del controlador Atmega328.

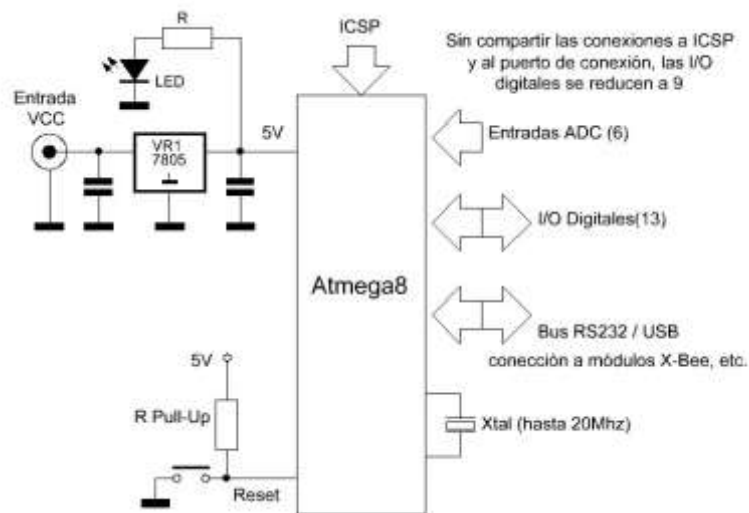


Figura 16: Diagrama de bloques del controlador Atmega.

Fuente: (Hobbytronics, 2017)

2.2.2.2 Modelos de tarjetas Arduino

Dentro de la comunidad de la plataforma Arduino existe una variedad de modelos de tarjetas electrónicas, estos modelos permiten a los usuarios tener la capacidad de disponer de una placa electrónica que tenga las características técnicas que se ajustan a sus necesidades.

Al ser tecnología abierta, estas tarjetas electrónicas han ocasionado una gran demanda a nivel global, esto ha permitido al fabricante desarrollar una variedad de modelos que son utilizados según la complejidad del proyecto a desarrollar, es necesario que el usuario conozca la dimensión del proyecto que pretende realizar, esto le permitirá seleccionar la placa adecuada para el desarrollo de su proyecto. En la figura 17 se observa los diferentes modelos de placas utilizados en la plataforma Arduino, se puede apreciar diferentes modelos que van desde los básicos hasta los más complejos.



Figura 17: Modelos de placa Arduino

Fuente: (Todo tecnología, 2017)

2.2.2.3 Arduino Mega pro-mini

La placa Arduino Mega pro-mini es una placa pequeña basada en el microcontrolador Atmega2560 es simplemente una placa totalmente compatible con la Arduino Mega, en la tabla 1 se muestran las especificaciones técnicas de la placa Mega pro-mini.

Tabla 1. Especificaciones técnicas de la placa Mega pro-mini.

Especificaciones técnicas	
Microcontrolador	Atmega 2560
Voltaje de trabajo:	5 VDC
Pines digitales Entrada/Salida:	54 (15 salida PWM)
Pines analógicos:	16 canales analógicos
Capacidad de la Flash:	256 KB
Capacidad de la SRAM:	8 KB
Capacidad de la EEPROM:	4 KB
Velocidad de ejecución del reloj:	16 MHZ

Fuente: Elaboración propia.

En la figura 18 se observa el modelo y tamaño de la placa Mega pro-mini.



Figura 18: Arduino Mega pro mini.

Fuente: Elaboración propia.

2.2.2.4 Arduino nano

La placa Arduino nano es una placa pequeña que utiliza el microcontrolador Atmega328, tiene las mismas características de funcionamiento con respecto a la tarjeta Arduino Uno, en la tabla 2 se detallan las especificaciones técnicas del microcontrolador.

Tabla 2. Especificaciones técnicas del microcontrolador Atmega328.

Especificaciones técnicas	
Microcontrolador	Atmega 328
Voltaje de trabajo:	5 VDC
Pines digitales Entrada/Salida:	14 (6 salida PWM)
Pines analógicos:	8 canales analógicos
Capacidad de la Flash:	32 KB
Capacidad de la SRAM:	2 KB
Capacidad de la EEPROM:	1 KB
Velocidad de ejecución del reloj:	16 MHZ

Fuente: Elaboración propia.

En la figura 19 se observa el modelo y tamaño de la placa Arduino nano.

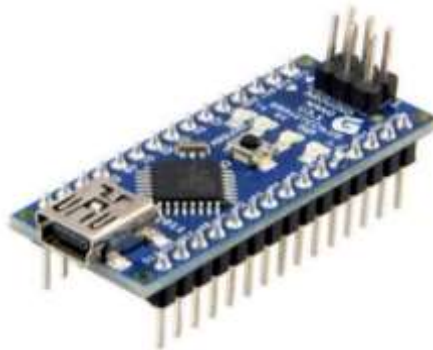


Figura 19: Arduino nano

Fuente: (Arduino, 2018)

2.2.3 Amplificadores operacionales

El amplificador operacional es un componente electrónico utilizado en aplicaciones de electrónica, puede ser configurado en diferentes modos de trabajo tales como: suma, resta, integración y diferenciación, cuenta con dos entradas y una salida, las entradas están formado por una entrada inversa y la otra no inversa, su salida tiene un factor de ganancia (G) que es multiplicada por la diferencia de sus dos entradas. (Pertence, 2008).

Las aplicaciones abarcan la industria electrónica, en la figura 20 se observa la simbología del amplificador operacional.

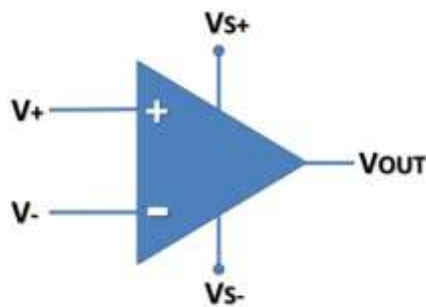


Figura 20: Simbología del Amplificador operacional.

Fuente: (Tools, 2018)

Donde:

V+: Entrada no inversa

V-. Entrada inversa

Vout: Voltaje de salida

VS+: Voltaje de entrada positivo

VS-: Voltaje de entrada negativa

2.2.3.1 Amplificador diferencial.

El amplificador en modo diferencial es un circuito que amplifica la diferencia de dos señales la no inversora menos la inversora. (Schiavon, 2010), la ventaja de este tipo de amplificador es el rechazo al ruido en modo común, conocido como circuito restador, en la figura 21 se observa la configuración del amplificador diferencial.

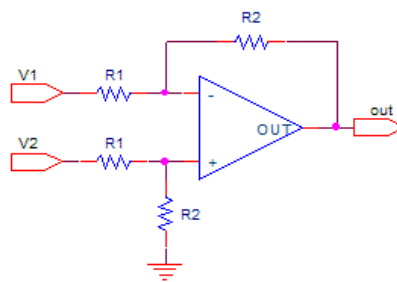


Figura 21: Amplificador operacional modo diferencial

Fuente: (Icardaba, 2018)

2.2.3.2 Amplificador Inversor

Es un circuito lineal conocido como amplificador inversor, denominada así porque la salida del amplificador se encuentra desfasada en 180° con respecto a la entrada. (Pertence, 2008), en la figura 22 se observa la configuración del circuito amplificador inversor.

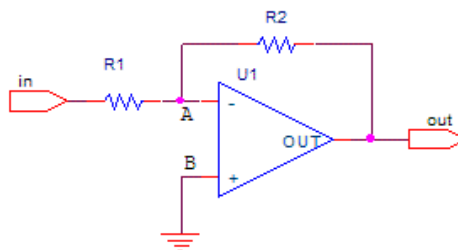


Figura 22: Amplificador inversor

Fuente: (Icardaba, 2018)

2.2.4 Optoacoplador.

Este dispositivo está compuesto en la entrada por un diodo LED y la salida por un fototransistor, cuando se presenta una señal eléctrica en la entrada del optoacoplador hace que circula una corriente a través del diodo LED, este emite un haz de luz que es recibida por la base del fototransistor, en la figura 23 se observa la simbología del circuito optoacoplador.

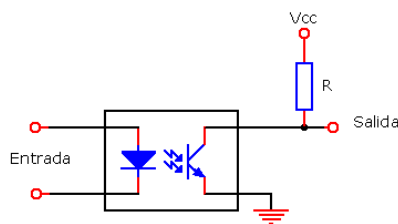


Figura 23: Circuito del optoacoplador.

Fuente: Elaboración propia.

En la tabla 3 se observa las características técnicas el optoacoplador 4N35.

Tabla 3. Características técnicas el optoacoplador 4N35.

Descripción	valor
Salida	Fototransistor
Voltaje	3550 VRMS
Corriente max. en el LED	60 mA
Voltaje max. inverso del LED	6 V
Voltaje max. colector-emisor	30 V
Corriente max. de colector	100 mA
Tiempo de respuesta	3 μ s típico
Montaje	6 pines

Fuente: Elaboración propia.

2.2.5 Regulador LM2596

El regulador LM2596 es un circuito integrado perteneciente a National Semiconductor que permite regular el voltaje de entrada de una fuente de alimentación en conmutación descendente, tiene un rango de entrada de operación de 3.5 a 40 voltios con una carga de 3.0 Amperios, el voltaje de salida es regulado por un potenciómetro de precisión, en la figura 24 se observa el módulo completo del regulador de voltaje LM2596.



Figura 24: Módulo DC-DC Converter LM2596.

Fuente: (Hetpro, 2018)

La tabla 4 muestra las especificaciones técnicas del regulador LM2596.

Tabla 4. Especificaciones técnicas del regulador LM2596.

Descripción	Especificaciones
Rango de voltaje de salida	4 a 35V
Corriente máxima de salida	3A
Eficiencia de conversión	92%
Frecuencia de switcheo	50Khz
Temperatura de operación	-40°C a 85°C (hasta 10W)
Temperatura máxima	40°C
Regulación de voltaje	±0.5%

Fuente: Elaboración propia.

2.2.6 Conversor ADC

Una señal eléctrica analógica es aquella en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. En el caso de la corriente alterna, la señal analógica incrementa su valor con signo eléctrico positivo (+) durante medio ciclo y disminuye a continuación con signo eléctrico negativo (–) en el medio ciclo siguiente.

2.2.6.1 Resolución

La resolución es la precisión de medición de una señal analógica al momento de convertir dicha señal a digital, para obtener la resolución se debe conocer el voltaje de referencia y el número de bits del conversor análogo-digital.

El número de bits permite determinar la cantidad de combinaciones posibles para obtener el valor digital. El número máximo de bits es representado por: 2^n donde n representa el número de bits.

Para obtener la resolución del conversor ADC se utiliza la siguiente fórmula:

$$Resolucion = \frac{V_{in}}{(2^n - 1)}$$

Donde:

n = Número máximo de bits.

V_{in} = Voltaje de entrada del conversor ADC.

2.2.6.2 Circuito ADS1115

El circuito ADS1115 es un circuito que convierte una señal analógica a digital (ADC), tiene una resolución de 16 bits, es utilizado para medir señales analógicas de los sensores, este dispositivo está compuesto de 4 canales de 16 bits, 15 bits son utilizados para la medición y un bit es utilizado como signo, utiliza la comunicación serial por I2C con el controlador, dispone de 4 direcciones de configuración a través de la conexión del pin ADDRESS.

Se utiliza el circuito ADS1115 para obtener mayor precisión en las lecturas de señales analógicas.

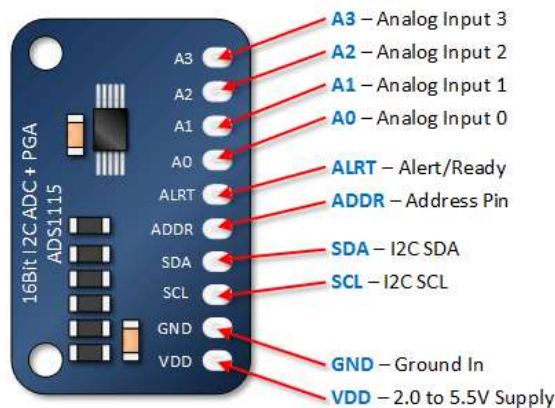


Figura 25: Pines del ADS1115.

Fuente: (HENRY'S BENCH, 2018)

El circuito ADS1115 puede trabajar en modo ended y en modo diferencial para obtener la medición de una señal. Cuando trabaja en modo ended utiliza cuatro canales ADC de 15 bits y cuando trabaja en modo diferencial utiliza dos canales ADC.

Este dispositivo se comunica con el controlador mediante la comunicación serial I2C, El pin ADDR es utilizado para determinar la configuración de la dirección I2C, en la figura 26 se observa las direcciones que pueden ser configurados.

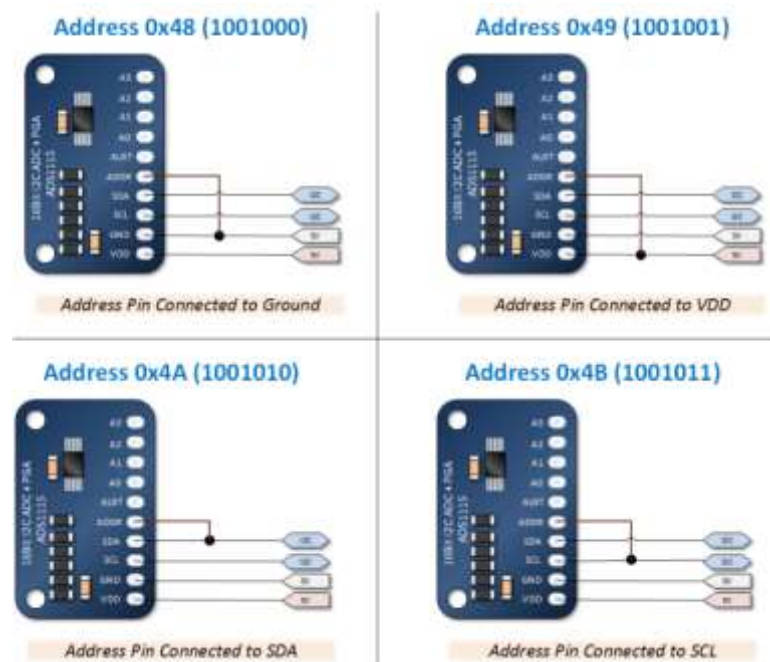


Figura 26: Configuración del ADS1115.

Fuente: (HENRY'S BENCH, 2018)

2.2.7 Visual Studio 2012

Visual Studio 2012 es una plataforma de desarrollo de programación el cual tiene una variedad de herramientas a ser utilizadas para desarrollar aplicaciones basados en lenguajes de programación orientado a objetos, por ejemplo: Servicios Web XML y ASP.NET, aplicaciones móviles y de escritorios. Dentro de los lenguajes de programación podemos encontrar los lenguajes de mayor popularidad y aceptación en el mercado como son:

Visual Basic, Visual C# y Visual C++, los cuales están incorporados dentro del mismo entorno de desarrollo integrado (IDE) (News sof, 2018).

El entorno de desarrollo de la plataforma Visual Studio se viene actualizando progresivamente, cada actualización trae herramientas más robustas que garantizan eficientemente el desarrollo de aplicaciones, mediante esta herramienta se logra interfaces de gran calidad y aceptación en el mercado competitivo, así mismo muchas empresas dedicados al desarrollo de software permiten integrar sus librerías para el desarrollo de aplicaciones, mediante el entorno de desarrollo el programador puede elegir el tipo de lenguaje de programación a utilizar, en la figura 27 se observa el logo del entorno de programación de la plataforma Visual Studio 2012.



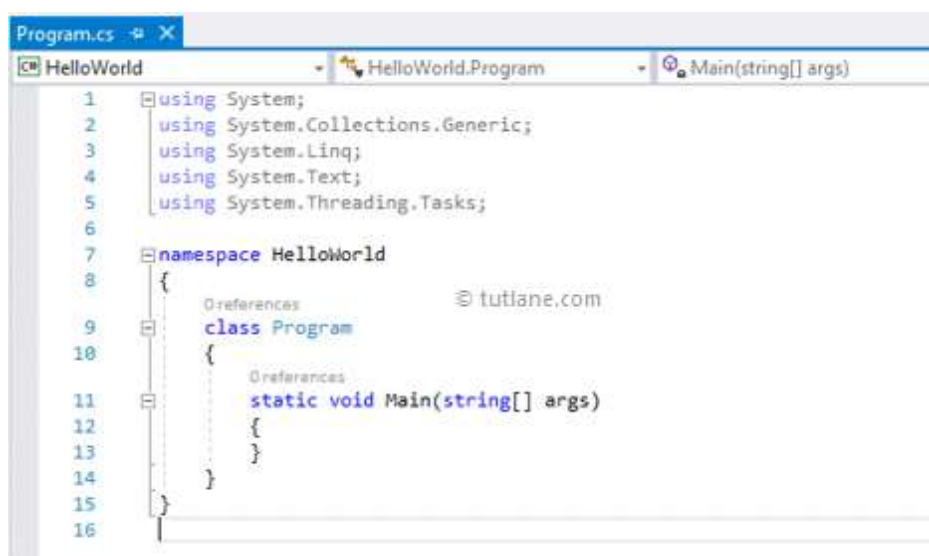
Figura 27: Logo de la plataforma Visual Studio.

Fuente: (News sof, 2018)

2.2.7.1 Visual C#

Visual C# es una herramienta de la plataforma Visual Studio, el lenguaje de programación está orientado a objetos y fue desarrollado por Microsoft, este lenguaje permite ser utilizado por los programadores con la finalidad de desarrollar aplicaciones que son ejecutadas en .NET Framework, esta tecnología permite compilar, ejecutar aplicaciones y ejecutar servicios web XML.

C# es un lenguaje sencillo, su sintaxis de edición de códigos es fácil e intuitiva, su desarrollo está basado en la familia de lenguajes C, el cual es uno de los lenguajes de programación provienen de este tipo de lenguaje, muchas de sus características han ido evolucionando y mejorando hasta llegar a la versión actual. En la figura 28 se observa el entorno de desarrollo integrado (IDE) del lenguaje de programación C# (Visual Studio, 2018).



```
Program.cs # X
HelloWorld - HelloWorld.Program - Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace HelloWorld
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13        }
14    }
15 }
16
```

Figura 28: Entorno de desarrollo C#.

Fuente: Elaboración propia.

2.2.8 Eagle Cadsoft

El software EAGLE, que significa: (Easily Applicable Graphical Layout Editor) es una herramienta de diseño para crear diagramas electrónicos y circuitos impresos. Esta herramienta nos permite crear nuestros propios esquemas de circuitos electrónicos, realizar el ruteado de las pistas en modo manual o automático obteniendo un diseño de circuito impreso. El software es intuitivo y fácil de usar, existen dos versiones del programa, una versión full mediante un pago y la otra versión es limitada por ser libre. (Electronica-electronics, 2017). La plataforma de desarrollo Eagle está formado por tres módulos, un módulo para realizar los diagramas, un módulo editor de esquemas y un módulo de ruteo, con esta herramienta las aplicaciones para el diseño de circuitos impresos y esquemas electrónicos resulta ser sencillo, en la figura 29 se observa el módulo de diagrama y ruteador del software Eagle.

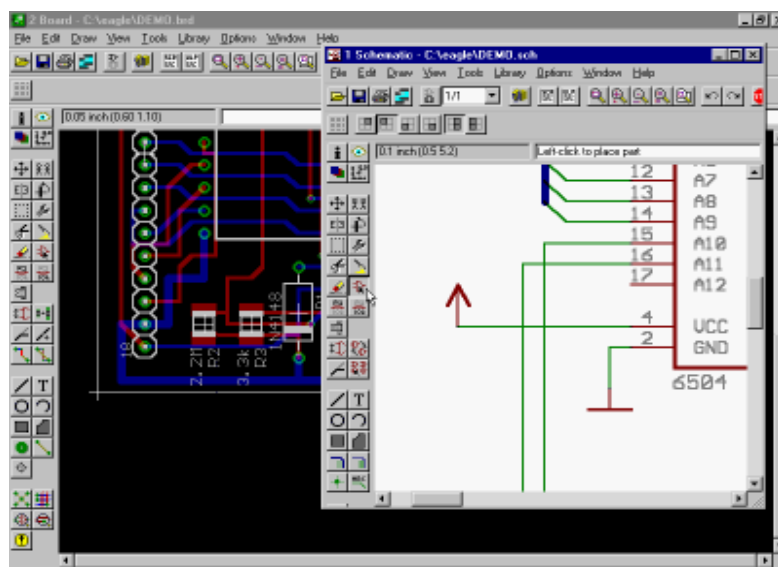


Figura 29: IDE Eagle Cadsoft, diagramador y esquema

Fuente: Elaboración propia.

CAPÍTULO III

VARIABLES E HIPOTESIS

3.1 Definición de las variables.

3.1.1 Variable dependiente.

La función del sistema de comunicación es mantener la comunicación entre el buque lanzador y el torpedo, la variable dependiente que interviene en el presente problema se define como:

- Telegrama de datos actuales

3.1.2 Variables independientes.

Las variables independientes de acuerdo al diseño del protocolo de telegrama de datos actuales del torpedo SST-4, son las señales digitales y analógicas son:

- **La Posición**, con respecto a los 3 ejes las cuales son brindadas por la unidad giroscópica del torpedo.
- **La Profundidad**, obtenido en la unidad del SPU-A del sensor de presión del torpedo.
- **La Velocidad**, obtenido del sensor magnético del motor de propulsión.
- **Señal de blanco**, con respecto a la ubicación del blanco obtenida del sonar de la cabeza acústica.
- **Modo Acústico**, obtenido del SPU-A de la Sección electrónica.
- **Modo de búsqueda**, obtenido del SPU-A de la Sección electrónica.
- **Motor de propulsión**, obtenido del SPU-A de la Sección electrónica.

3.2 Operacionalización de las variables.

En el siguiente cuadro se observa la relación de las variables de la investigación; así mismo se efectúa la dimensión y los indicadores de cada variable.

VARIABLE	DIMENSIONES	INDICADORES
Variable Independiente: Telegrama de datos actuales	Protocolo de comunicación serial del torpedo SST-4 compuesto de 8 palabras de 16 bits cada uno a una velocidad de 206 baudios.	Información de señales analógicas y digitales del torpedo SST-4.
Variable Dependiente: Posicionamiento	Respecto a los 3 ejes del torpedo; la información es obtenida de la unidad giroscópica del torpedo;	Rumbo: 0° a 360°. Rolido: -15° a +15°. Cabeceo: -50° a +50°.
Variable Dependiente: Profundidad	Señal analógica medido por el sensor de presión ubicado en la computadora del torpedo.	Pies: 0 a 342 pies.
Variable Dependiente: Velocidad	Señal de pulsos del sensor magnético obtenida del motor de propulsión.	Nudos: 0 a 34 nudos

Variable Dependiente: Señal de blanco	Señal digital con respecto a la ubicación y detección de un blanco obtenida del sonar de la cabeza acústica.	NN: No hay blanco TC: Contacto. TD: Detección. HH: Perdida.
Variable Dependiente: Modo acústico	Señal digital obtenida del telegrama de comando en el lado receptor de la computadora del torpedo de la Sección electrónica.	Pasivo: Recepción. Activo: Transmisión y recepción.
Variable Dependiente: Modo de búsqueda	Señal digital obtenida del SPU-A de la Sección electrónica, para indicar el modo de búsqueda que está efectuando el torpedo.	Homing On: Modo automático. Homing Off: Modo manual
Variable Dependiente: Motor de propulsión	Señal digital obtenida en la computadora del torpedo de la Sección electrónica.	On: Encendido Off: Apagado

3.3 Hipótesis general e hipótesis específicas.

3.3.1 Hipótesis general.

“El diseño e implementación de Telegrama de datos actuales con interfaz gráfica para el torpedo anti superficie SST-4, permitirá la ubicación y posición del torpedo con respecto al blanco durante su corrida efectiva, mejorando la operación táctica del operador del sistema de control de tiro.”

3.3.2 Hipótesis específicas.

La tarjeta electrónica de adquisición de datos permitirá leer señales digitales y analógicas de los sensores y componentes electrónicos del torpedo SST-4.

La tarjeta electrónica de transmisión del telegrama de datos actuales permitirá codificar la información obtenida de la tarjeta de adquisición de datos en un protocolo serial adaptable al sistema de comunicación.

La tarjeta electrónica de recepción en el sistema de control de tiro permitirá decodificar el telegrama de datos actuales transmitidos del torpedo.

La programación orientada a objetos de la plataforma visual Studio permitirá desarrollar aplicaciones de alta calidad y performance en el desarrollo de la interfaz gráfica para la visualización del telegrama de retorno en el sistema de control de tiro.

CAPÍTULO IV

METODOLOGÍA

4.1 Tipo de investigación.

La investigación es de tipo experimental y aplicativo (enfoque cualitativo), tecnológica y tiene como propósito aplicar el uso del conocimiento científico para llegar a solucionar la limitación del sistema de comunicación del telegrama de datos actuales del torpedo SST-4. Sus niveles de investigación son la investigación experimental el cual es utilizado durante la manipulación de las variables que están relacionadas con el sistema de comunicación. Para el diseño de la tecnología a ser aplicada

La investigación aplicada tiene el propósito de emplear los resultados obtenidos durante la investigación experimental para lograr diseñar las nuevas tecnologías a ser utilizadas durante la solución de los problemas basados en las tecnologías discontinuadas.

4.2 Diseño de la investigación.

Este trabajo de investigación se basa en el diseño de un telegrama de transmisión serial de datos actuales en el torpedo SST-4, para permitir al sistema de control de tiro tener la ubicación y posición del torpedo con respecto al blanco durante su corrida efectiva.

La primera etapa será estrictamente de investigación científica en la búsqueda del conocimiento referente al tema en cuestión. Para ello se

utilizará la bibliografía adecuada (manuales técnicos y documentos del fabricante) para definir el protocolo del telegrama de datos actuales tomando como base el protocolo serial del torpedo SUT-264.

La segunda etapa consistirá en el estudio de las señales analógicas y digitales de la unidad SPU-A de la sección electrónica y de la cabeza acústica del torpedo SST-4, los cuales serán transmitidos mediante el sistema de comunicación.

La tercera etapa consistirá en modelar matemáticamente las señales analógicas del sensor magnético, sensor de presión, balanceo, cabeceo, voltaje de control y correlador de cero grados para adaptar dichas señales al conversor análogo digital de 16 bits.

La cuarta etapa del proyecto es experimental, se diseñará la tarjeta de adquisición de datos de las señales analógicas y digitales de los sensores y componentes electrónicos del torpedo.

La quinta etapa es el estudio del sistema de comunicación para identificar los componentes electrónicos, niveles de tensión y velocidad de transmisión para adaptar el telegrama de datos actuales.

La sexta etapa del proyecto es experimental, se diseñará la tarjeta electrónica para codificar la información recibida de la tarjeta de adquisición de datos en un protocolo serial compuesto de 8 palabras de 16 bits denominado telegrama de datos actuales.

La séptima etapa es el estudio de la unidad SPU-B en el lado del sistema de control de tiro, el cual recibe el telegrama de retorno del sistema de comunicación para capturar el telegrama de datos actuales.

La octava etapa del proyecto es experimental, se diseñará la tarjeta electrónica para decodificar el telegrama de datos actuales en el sistema de control de tiro para ser enviados a la computadora.

La novena etapa consistirá en desarrollar el software que permita visualizar a través de la interfaz gráfica la información proveniente del telegrama de datos actuales.

La décima etapa servirá para comprobar el correcto funcionamiento del proyecto, esta etapa se realizará en la Estación de Armas Submarinas.

En la undécima etapa se efectuará el disparo real de un torpedo SST-4 en versión ejercicio desde una unidad submarina para dar por concluido el proyecto de investigación.

4.3 Población y muestra.

Población: Personal de las unidades submarinas y de la Estación de Armas Submarinas pertenecientes a la Marina de Guerra del Perú.

Muestra: Personal de operadores del sistema de control de tiro que laboran en las unidades submarinas y el personal de operadores que efectúan las pruebas generales al torpedo SST-4 que laboran en la Estación de Armas Submarinas, ambos pertenecientes a la Marina de Guerra del Perú.

4.4 Técnicas e instrumentos de recolección de datos.

Se reunió información del protocolo de comunicación serial de los torpedos SST-4 con el sistema de control de tiro y la plataforma de pruebas Test-Stand, así como la búsqueda de investigaciones y trabajos relacionados al proyecto usando bases de datos de artículos como IEEExplore y Scopus.

Se recopilaron informaciones de los manuales técnicos del tipo, características, niveles de tensión y formas de las señales analógicas y digitales de las señales eléctricas codificados en el telegrama de datos actuales, estas señales fueron ubicados en el SPU-A (Unidad de procesamiento de señales) de la sección electrónica del torpedo y el uso de técnicas y desarrollo de sistemas electrónicos para su adquisición con un controlador.

Se realizaron mediciones de las señales digitales en los módulos de transmisión del torpedo para adaptar los niveles de tensión y corriente del telegrama de datos actuales.

Se realizaron mediciones de las señales en el módulo de recepción para adaptar los niveles de tensión y corriente del telegrama de datos actuales para adaptarlos al sistema de pruebas.

Durante el desarrollo y pruebas del sistema de comunicación se obtuvo la información del telegrama de datos actuales en una interfaz gráfica desarrollado en la computadora.

4.5 Procedimiento de recolección de datos.

El estudio y análisis de recolección de datos, se basa en capturar las señales digitales y analógicas de los sensores y unidades electrónicas del torpedo SST-4, en la figura 30 se observa el modelo del procesamiento de la recolección de datos para obtener el telegrama de datos actuales, el estudio y análisis de las señales digitales y analógicas se realizan en la unidad de procesamiento de señales (SPU-A) de la sección electrónica del torpedo SST-4.

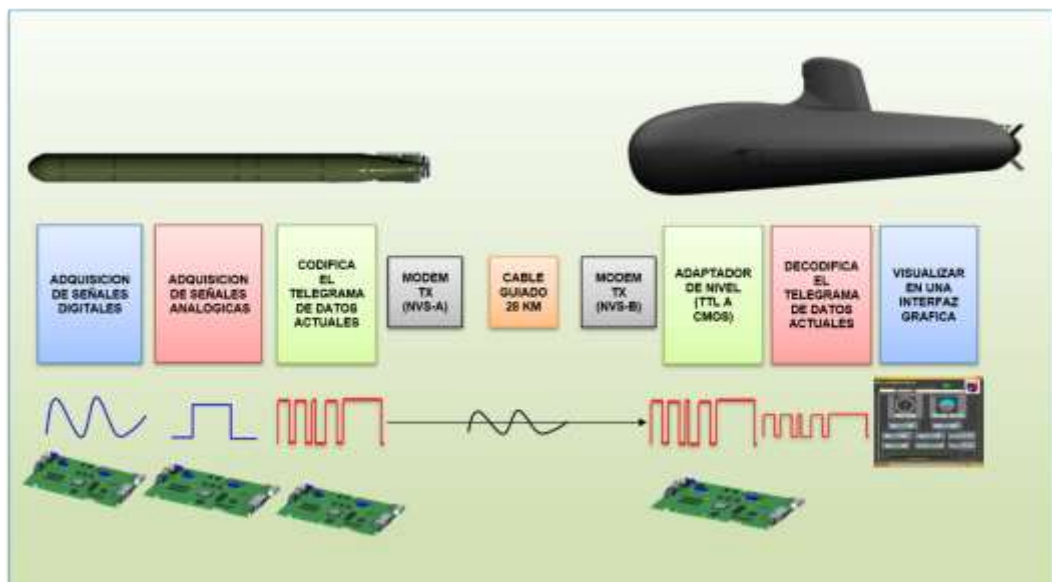


Figura 30: Etapas de procesamiento del telegrama de datos actuales.

Fuente: Elaboración propia.

El resultado de la codificación de las señales digitales y analógica son transmitidos a la unidad submarina por medio del cable de guiado, los modem del lado del torpedo y del submarino están diseñados para transmitir el telegrama serial a una velocidad de 206 baudios.

4.5.1 Señales digitales del torpedo SST-4.

En la computadora del torpedo SST-4 se ubica en la sección electrónica, en esta unidad se realiza el procesamiento de las señales digitales, dentro de las señales digitales utilizados en el telegrama de datos actuales se encuentra la señal de Homing que es utilizada para indicar si el torpedo SST-4 se encuentra en modo de guiado manual o automático, la señal distancia de seguridad indica si el torpedo paso los 300 metros desde el punto de lanzamiento, la señal de detección es informado por la cabeza acústica para saber si el torpedo tiene algún blanco, la señal modo acústico es transmitido para conocer si la cabeza acústica se encuentra en modo pasivo o activo, la señal de motor indica si el motor esta encendido o desconectado y la señal de rumbo de 10 bits es proporcionado por el giróscopo horizontal. En la siguiente tabla se detallan las señales digitales a ser implementados en el desarrollo del telegrama de datos actuales del torpedo SST-4.

Tabla 5. Señales digitales del telegrama de datos actuales.

N°	Señales digitales
1	Homing: On, Off
2	Distancia seguridad: Cubierta
3	Detección: No blanco, TC, TD HH
4	Modo Acústico: Pasivo, Activo
5	Motor: On-Off
6	Rumbo

Fuente: Elaboración propia.

En la figura 31 se observa las señales digitales que son utilizados para diseñar la tarjeta de adquisición de señales digitales.

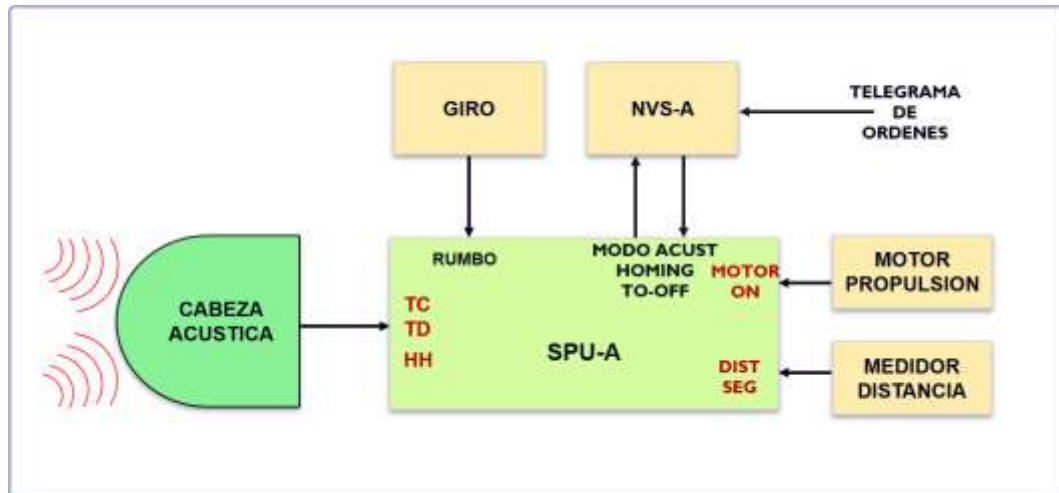


Figura 31: Señales digitales en el (SPU-A).

Fuente: Elaboración propia.

4.5.1.1 Modo de guiado.

El torpedo SST-4 puede ser guiado en modo manual o automático, esta señal puede ser HOMING OFF (Modo manual: guiado por las ordenes de rumbo del sistema de control de tiro) o HOMING ON (modo automático: guiado por su cabeza acústica cuando tiene blanco), en la tabla 6 se observa los valores medidos en la tarjeta electrónica número 125 de la unidad de procesamiento de señales SPU-A de la sección electrónica.

Tabla 6. Señal modo de guiado.

Modo de guiado	Nro. De tarjeta	Pin	1 lógico	0 lógico
HOMING ON	125	A12-A15		0 VDC
HOMING OFF	125	A12-A15	12 VDC	

Fuente: Elaboración propia.

4.5.1.2 Distancia de seguridad.

El torpedo SST-4 tiene una distancia de seguridad de 300 metros, esta distancia es para proteger al submarino después de efectuar el lanzamiento, el torpedo informa esta señal al sistema de control de tiro mediante una señal digital de un bit, en la tabla 7 se observa los valores medidos en la tarjeta electrónica número 133 de la unidad de procesamiento de señales SPU-A de la sección electrónica del torpedo SST-4.

Tabla 7. Señal de la distancia de seguridad.

Distancia seguridad	Nro. de tarjeta	Pin	1 lógico	0 lógico
Cubierta	133	A12-A15	--	0 VDC
No cubierta	133	A12-A15	12 VDC	---

Fuente: Elaboración propia.

4.5.1.3 Detección de blanco.

El estudio de las señales de detección de blanco es producido por el sonar de la cabeza acústica del torpedo SST-4, el sonar es conocido como transducer, está compuesto por un arreglo de 14 bobinas del tipo magnetostrictivos que son utilizados para capturar las señales acústicas producidas por un buque de superficie, en la figura número 32 se puede observar las condiciones de las señales de blanco para producir las señales de detección que serán informados al sistema de control de tiro.

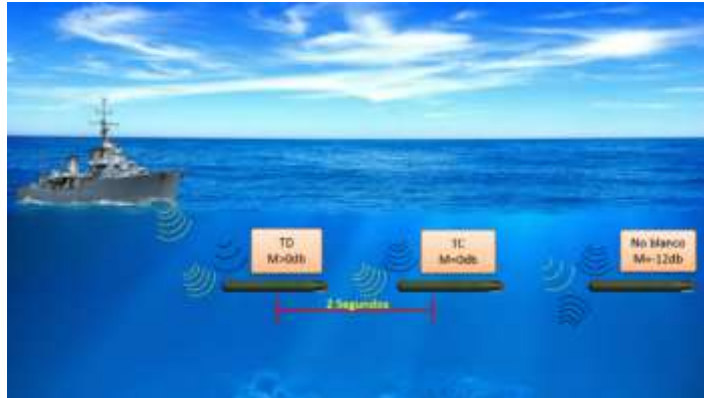


Figura 32: Modo de detección acústica.

Fuente: Elaboración propia.

La información TC representa la señal de contacto con el Blanco, la información de TD corresponde a la señal de detección del Blanco, la información de HH corresponde a la señal de pérdida de blanco, cada señal es de 1 bit, en la tabla 8 se observa los valores medidos en la tarjeta electrónica G062 de la cabeza acústica del torpedo SST-4.

Tabla 8. Señales de detección de blanco TC, TD y HH.

Señal de Blanco	Nro. De tarjeta	Pin	1 lógico	0 lógico
NO TC	G062	19	24 VDC	--
TC	G062	19	--	0 VDC
NO TD	G062	17	24 VDC	--
TD	G062	17	--	0 VDC
NO HH	G062	2	24 VDC	--
HH	G062	2	--	0 VDC

Fuente: Elaboración propia.

4.5.1.4 Modo Acústico.

La cabeza acústica del torpedo SST-4 tiene dos modos de funcionamiento Pasivo y Activo, en modo pasivo el sonar recibe señales acústicas, en modo activo transmite el pulso acústico, en la tabla 9 se observa los valores del modo acústico en la unidad de procesamiento de señales SPU-A.

Tabla 9. Señal modo acústico.

Modo acústico	Nro. De tarjeta	Pin	1 lógico	0 lógico
PASIVO	125	A12-A15	12 VDC	
ACTIVO	125	A12-A15		0 VDC

Fuente: Elaboración propia.

4.5.1.5 Motor ON.

El motor del torpedo SST-4 es alimentado por la batería de propulsión a través del motor contactor del distribuidor principal, cuando el motor es alimentado en el SPU-A se genera la señal Motor ON, esta información se obtiene en la tarjeta electrónica número 126 de la unidad de procesamiento de señales SPU-A, en la tabla número 10 se observa la señal Motor On.

Tabla 10. Señal de Motor On.

Motor ON	Nro. De tarjeta	Pin	1 lógico	0 lógico
MOTOR ON	126	A10-A15	12 VDC	
MOTOR OFF	126	A10-A15		0 VDC

Fuente: Elaboración propia.

4.5.1.6 Rumbo.

La señal de rumbo es utilizada por el torpedo SST-4 para ser guiado durante su desplazamiento, esta señal es producida por el giróscopo horizontal y está conformado por 10 bits, las señales son digitales y llegan específicamente a las tarjetas electrónicas número 141 y 142 de la unidad de procesamiento de señales (SPU-A), cada bit tiene un peso específico, los valores de los 10 bits de rumbo con el valor por cada bit se observan en la tabla número 11.

Tabla 11. Señales digitales de las señales de rumbo.

Rumbo	Nro. De tarjeta	Pines	1 lógico	0 lógico
180°	141	A8-A15	12 V.	0 V.
90°	141	A9-A15	12 V.	0 V.
45°	141	A10-A15	12 V.	0 V.
22.5°	141	A11-A15	12 V.	0 V.
11.25°	141	A12-A15	12 V.	0 V.
5.62°	142	A8-A15	12 V.	0 V.
2.81°	142	A9-A15	12 V.	0 V.
1.40°	142	A10-A15	12 V.	0 V.
0.7°	142	A11-A15	12 V.	0 V.
0.35°	142	A12-A15	12 V.	0 V.

Fuente: Elaboración propia.

4.5.2 Señales analógicas del torpedo SST-4.

En la computadora del torpedo se ubican las señales analógicas del telegrama de datos actuales, el giro vertical entrega las señales de cabeceo y balanceo, la cabeza acústica las señales correlador de 0° y distancia al blanco y las celdas de presión la profundidad, en la siguiente tabla 12 se detallan las señales analógicas del telegrama de datos actuales.

Tabla 12. Señales analógicas del telegrama de datos actuales.

N°	Señales Analógicas
1	Cabeceo
2	Balanceo
3	Distancia al blanco
4	Correlador de 0°
5	Profundidad

Fuente: Elaboración propia.

En la figura 33 se observa las señales analógicas que son utilizados para el diseño e implementación de la tarjeta de adquisición.

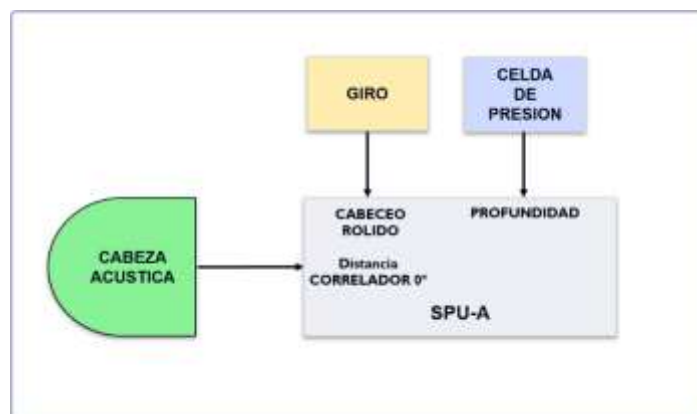


Figura 33: Señales analógicas en el (SPU-A).

Fuente: Elaboración propia.

4.5.2.1 Cabeceo.

La señal analógica de cabeceo es utilizada para medir el ángulo de inclinación de proa con respecto a la popa del torpedo, cada vez que el torpedo realiza cambios de profundidad el ángulo de cabeceo varía constantemente, esta señal es proporcionada por el giróscopo vertical, en la tabla 13 se observa los voltajes del ángulo de cabeceo según el manual de operación, el ángulo de medida se encuentra en el rango de -50° a $+50^{\circ}$.

Tabla 13. Voltajes del ángulo de cabeceo.

Valor medido (Voltios)	Tolerancia (Voltios)	Angulo de Cabeceo
5.69	± 0.18	50°
4.55	± 0.18	40°
3.41	± 0.18	30°
2.27	± 0.18	20°
1.14	± 0.18	10°
0.00	± 0.18	0°
-1.14	± 0.18	-10°
-2.27	± 0.18	-20°
-3.41	± 0.18	-30°
-4.55	± 0.18	-40°
-5.69	± 0.18	-50°

Fuente: Elaboración propia.

El gir6scopo vertical entrega voltaje y no valor angular, para hallar el valor angular deseado es necesario convertir el voltaje medido en grados, seg6n la tabla anterior, el valor m6ximo de cabeceo es 50° equivalente a 5.69 voltios y el valor m6nimo de cabeceo -50° corresponde a -5.69 voltios, para obtener los valores angulares de cabeceo a partir de los voltajes obtenidos es necesario utilizar la ecuaci6n de la recta para obtener el resultado deseado, en la figura 34 se observa el diagrama de la ecuaci6n de la recta utilizado para obtener la formula general del 6ngulo de cabeceo.

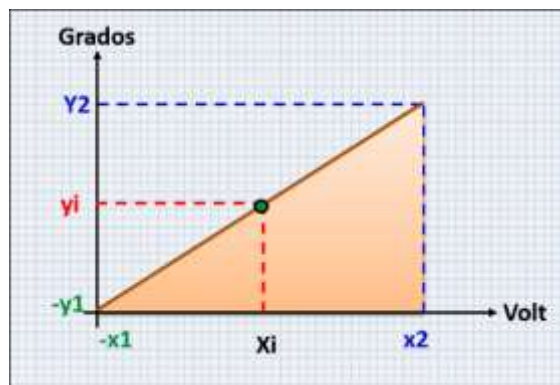


Figura 34: Conversi6n de voltaje a 6ngulo de cabeceo

Fuente: Elaboraci6n propia.

Donde:

Y_2 = 6ngulo de salida m6ximo.

Y_i = 6ngulo de salida.

$-Y_1$ = 6ngulo de salida m6nimo.

X_2 = Voltaje de entrada m6ximo.

X_i = Voltaje de entrada (V_{in}).

$-X_1$ = Voltaje de entrada m6nimo.

La fórmula de la ecuación de la recta para hallar el ángulo de salida (Y_i) se muestra a continuación:

$$Y_i = \left(\frac{Y_2 - Y_1}{X_2 - X_1} \right) (X_i - X_1) + Y_1 \dots\dots\dots (1)$$

Reemplazando los valores del ángulo de cabeceo en la fórmula (1) se obtiene:

$$Y_i = \left(\frac{50 - (-50)}{5.69 - (-5.69)} \right) (V_{in} - (-5.69)) + (-50)$$

Se desarrolla la ecuación para hallar la fórmula de cabeceo:

$$C_{ab} = \left(\frac{100}{11.38} \right) (V_{in} + 5.69) - 50$$

$$C_{ab} = \left(\frac{100V_{in} + (100 * 5.69)}{11.38} \right) - 50$$

$$C_{ab} = \left(\frac{100V_{in} + 569}{11.38} \right) - 50$$

$$C_{ab} = \left(\frac{100V_{in}}{11.38} + \frac{569}{11.38} \right) - 50$$

$$C_{ab} = (8.787346V_{in} + 50) - 50$$

La fórmula para hallar el valor del ángulo de cabeceo es:

$$C_{ab} = 8.787346 * V_{in} \dots\dots\dots (2)$$

4.5.2.2 Balanceo.

El giro vertical mide el ángulo de balanceo de babor a estribor con respecto al torpedo durante su desplazamiento, el valor del ángulo de balanceo varía constantemente durante el desplazamiento del torpedo, esto se debe a las corrientes del mar y a los cambios de rumbo que son ordenados desde el sistema de control de tiro de la unidad submarina, en la siguiente tabla se observa los voltajes del ángulo de balanceo según el manual de operación, el ángulo de medida se encuentra en el rango de -10° a $+10^{\circ}$.

Tabla 14. Voltajes del ángulo de rolido.

Valor medido (Voltios)	Tolerancia (Voltios)	Angulo de Rolido
10.00	± 1.00	10°
8.00	± 1.00	8°
6.00	± 1.00	6°
4.00	± 1.00	4°
2.00	± 1.00	2°
0.00	± 1.00	0°
-2.00	± 1.00	-2°
-4.00	± 1.00	-4°
-6.00	± 1.00	-6°
-8.00	± 1.00	-8°
-10.00	± 1.00	-10°

Fuente: Elaboración propia.

Según la tabla anterior, el valor máximo de rizado es 10° equivalente a 10.00 voltios y el valor mínimo es de -10° que corresponde a -10.00 voltios, los valores observados indican que el valor angular es lineal con respecto al voltaje de entrada, la fórmula para hallar el valor de rizado se muestra a continuación.

$$Rol = Vin \dots \dots \dots (3)$$

4.5.2.3 Distancia en activo.

La señal de distancia es proporcionada por la cabeza acústica del torpedo cuando se encuentra trabajando en modo activo, en este modo el torpedo transmite el pulso acústico, la tarjeta G051 de la cabeza acústica mide la distancia al blanco, el funcionamiento de medición de la distancia en el modo activo se observa en la figura 35, la cabeza acústica envía un pulso de transmisión que recorre una distancia "x", cuando el pulso impacta con el blanco un buque de superficie este rebota y produce un eco que recorre la misma distancia "x".

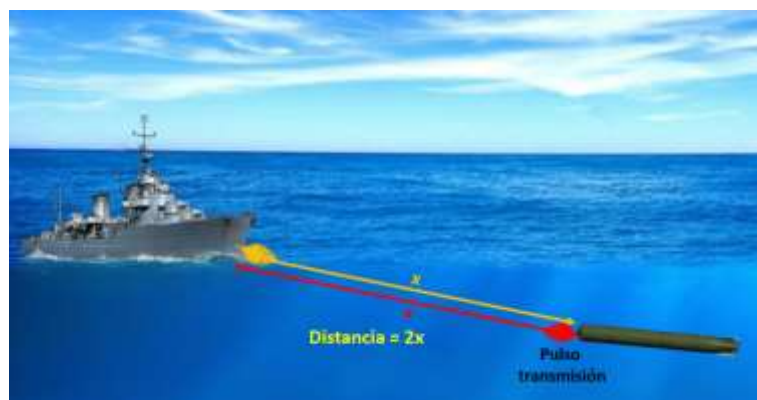


Figura 35: Recorrido del pulso en modo activo.

Fuente: Elaboración propia.

Para calcular la distancia en modo activo se utiliza la siguiente fórmula:

$$Dist = (C * t) \dots\dots\dots (4)$$

Donde:

Dist = Distancia en metros.

C = Velocidad del sonido en el mar = 1479(m/s).

t = Tiempo = 0.208 * Vin.

Si el pulso recorre dos veces la distancia “x”, entonces de la fórmula (8):

$$2 * Dist = (C * t) \quad Dist = \frac{C * t}{2}$$

Reemplazando los valores de velocidad “C” y tiempo “t”:

$$Dist = \frac{1479 * 0.208 * Vin}{2} \dots\dots\dots (5)$$

Aplicando la formula (5) se realizan los cálculos para hallar la distancia en metros, en la tabla 15 se observa los valores obtenidos de la distancia.

Tabla 15. Valores calculados de la distancia al blanco.

Voltaje entrada (VDC)	Velocidad sonido mar (m/s)	Distancia (metros)
5.079	1479	781,25
2.54	1479	390,63
1.27	1479	195,31
0.63	1479	97.656
0.32	1479	48.828
0.158	1479	24.414
0.079	1479	12.207

Fuente: Elaboración propia.

4.5.2.4 Correlador de 0°.

El correlador de 0° mide la relación de nivel de la señal del blanco con respecto al ruido, los valores obtenidos son medidos en la tarjeta electrónica G052 de la cabeza acústica del torpedo SST-4 cuando trabaja en modo pasivo, mide la relación señal/ruido del blanco, estando en modo pasivo, en la tabla 16 se observa los valores de la relación señal/ruido de un blanco.

Tabla 16. Valores de la relación señal/ruido.

Señal/Ruido (db)	Valor (voltios)	Tolerancia (voltios)
21	0.00	+/- 0.60
9	1.50	+/- 0.15
3	2.75	+/- 0.27
0	3.40	+/- 0.34
-1	5.12	+/- 0.60

Fuente: Elaboración propia.

4.5.2.5 Profundidad.

La profundidad actual del torpedo SST-4 es medido por las celdas de presión, una de las celdas es utilizada para medir valores de baja profundidad que están en el rango de 0 a 32 pies y la otra celda de presión es utilizado para medir alta profundidad que están en el rango de 32 pies a 342 pies, los voltajes de salida de las celdas de presión están en el rango de voltajes de 0 voltios a +12 voltios, , estos voltajes llegan a la unidad de procesamiento de señales (SPU-A).

Para obtener la información de la señal de profundidad se utilizó la tarjeta electrónica número 159 de la unidad de procesamiento de señales SPU-A, en la tabla número 17 se observan las características técnicas de la celda de presión los voltajes con respecto a la profundidad medido en pies.

Tabla 17. Valores de la profundidad actual.

Valor medido (Voltios)	Tolerancia (Voltios)	Profundida d (Pies)
0	± 0.1	0
0.016	± 0.1	1
0.032	± 0.1	2
0.048	± 0.1	3
0.064	± 0.1	4
0.512	± 0.1	32
5.472	± 0.1	342

Fuente: Elaboración propia.

Para hallar el valor de la profundidad actual es necesario convertir el voltaje medido en pies, según la tabla anterior, el valor máximo de profundidad es 342 pies equivalente a 5.472 voltios y el valor mínimo de profundidad es 0 pies corresponde a 0.00 voltios, se realiza la ecuación de la recta para linealizar los valores de salida en pies, en la figura 36 se observa el diagrama de la ecuación de la recta utilizado para obtener la formula general de la profundidad actual medido en pies.

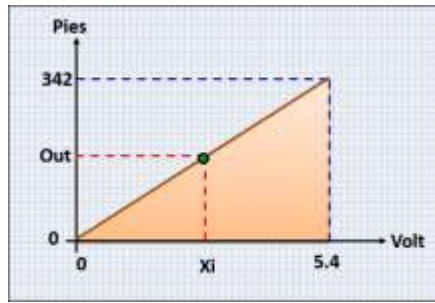


Figura 36: Conversión de voltaje a pies.

Fuente: Elaboración propia.

Donde:

342 = Valor máximo de profundidad en pies.

Out = Salida de profundidad en pies.

5.4 = Voltaje de entrada máximo.

Xi = Voltaje de entrada (V_{in}).

Se utiliza la fórmula de la ecuación de la recta de la formula (1) y se reemplazan los valores de la figura 28 y se obtiene:

$$Prof = \left(\frac{342 - 0}{5.4 - 0} \right) (V_{in} - 0) + 0$$

Se desarrolla la ecuación para hallar la fórmula de profundidad:

$$Prof = \left(\frac{342}{5.4} \right) (V_{in})$$

La fórmula para hallar el valor de profundidad en pies es:

$$Prof = 63.3333 * V_{in} \dots\dots\dots (6)$$

4.5.3 Diseño del DAQ de señales digitales.

4.5.3.1 Hardware del DAQ de señales digitales.

Los niveles de tensión de las señales digitales son procesadas en la computadora del lado del torpedo, emplean niveles de tensión tipo CMOS que varían de 0 voltios a 12 voltios, es necesario adaptar estos niveles de tensión para que puedan ser leídos correctamente por el microcontrolador, este emplea puertos de propósitos generales que son configurados como entrada, los niveles de tensión de lectura de los puertos del microcontrolador están comprendidos de 0 voltios a 5 voltios, el optoacoplador es un dispositivo electrónico utilizado para la adaptación de niveles de tensión, en la figura 37 se observa la configuración del optoacoplador para acoplar las señales digitales de 12 voltios proveniente de la unidad de giro de rumbo a 5 voltios para ser leídos por el microcontrolador.

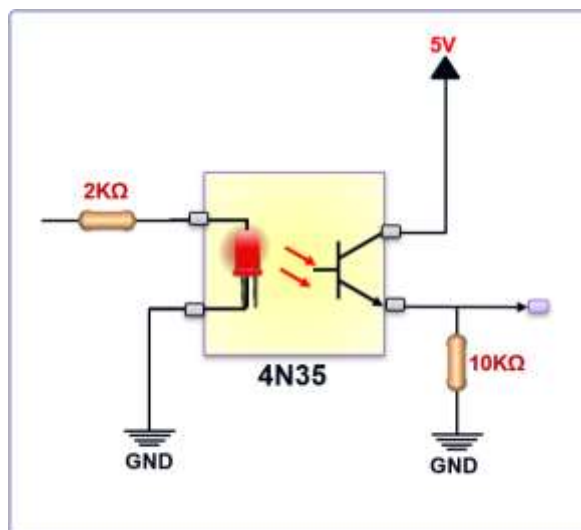


Figura 37: Circuito optoacoplador.

Fuente: Elaboración propia.

En este ejemplo se utiliza una línea de las señales digitales, la entrada del optoacoplador se conecta en serie con una resistencia para limitar la corriente del diodo led interno, si la línea de entrada tiene una señal de 12 voltios la resistencia reduce la corriente que circula en el diodo y emite un haz de luz que incide en la base del fototransistor, se genera una corriente de colector a emisor actuando como un interruptor electrónico; el microcontrolador recibe 5 voltios.

El controlador Atmega 328 es utilizado para las lecturas de las señales digitales, este dispositivo permite la lectura de las señales digitales de los diferentes sistemas de control de la computadora del torpedo, así mismo el microcontrolador envía la información recibida a la etapa de codificación mediante el puerto serial, se utiliza el circuito integrado Max232 para transmitir y recibir información serial, en la figura 38 se observa el diseño del circuito esquemático de la tarjeta de adquisición de señales digitales.

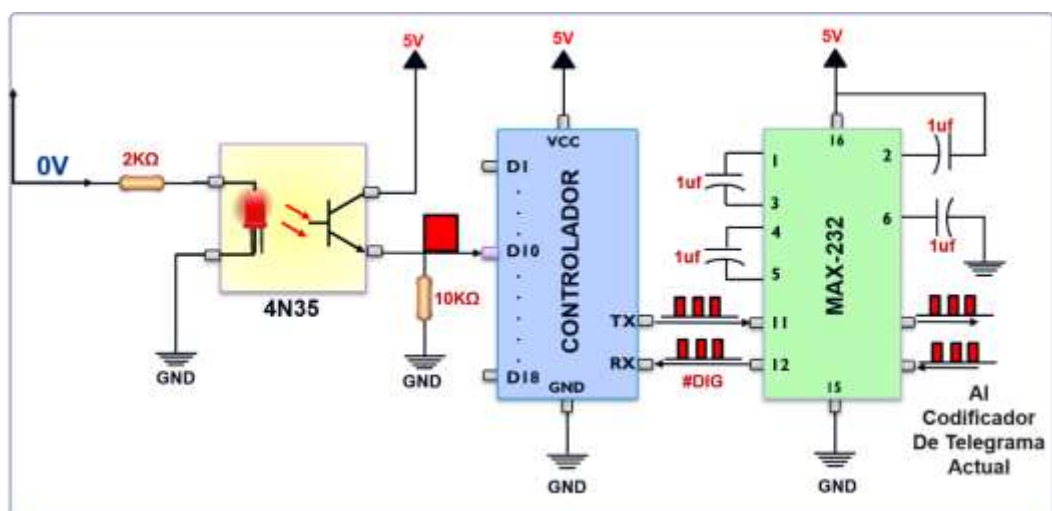


Figura 38: Circuito de lectura de las señales digitales.

Fuente: Elaboración propia.

4.5.3.2 Software del DAQ de señales digitales.

El algoritmo a desarrollar en la tarjeta de adquisición de las señales digitales está basado en el lenguaje C, este tipo de lenguaje emplea funciones específicas en cada etapa del procesamiento de la información, en la figura 39 se observa el diagrama de flujo del programa principal del controlador de la tarjeta DAQ Digital, en la primera parte del diagrama de flujo se declara las variables, la etapa de configuración se ejecuta cuando se enciende el controlador, se configura los puertos de entrada que serán utilizados para realizar las lecturas de las señales digitales y se configura la comunicación serial a 19200 baudios.

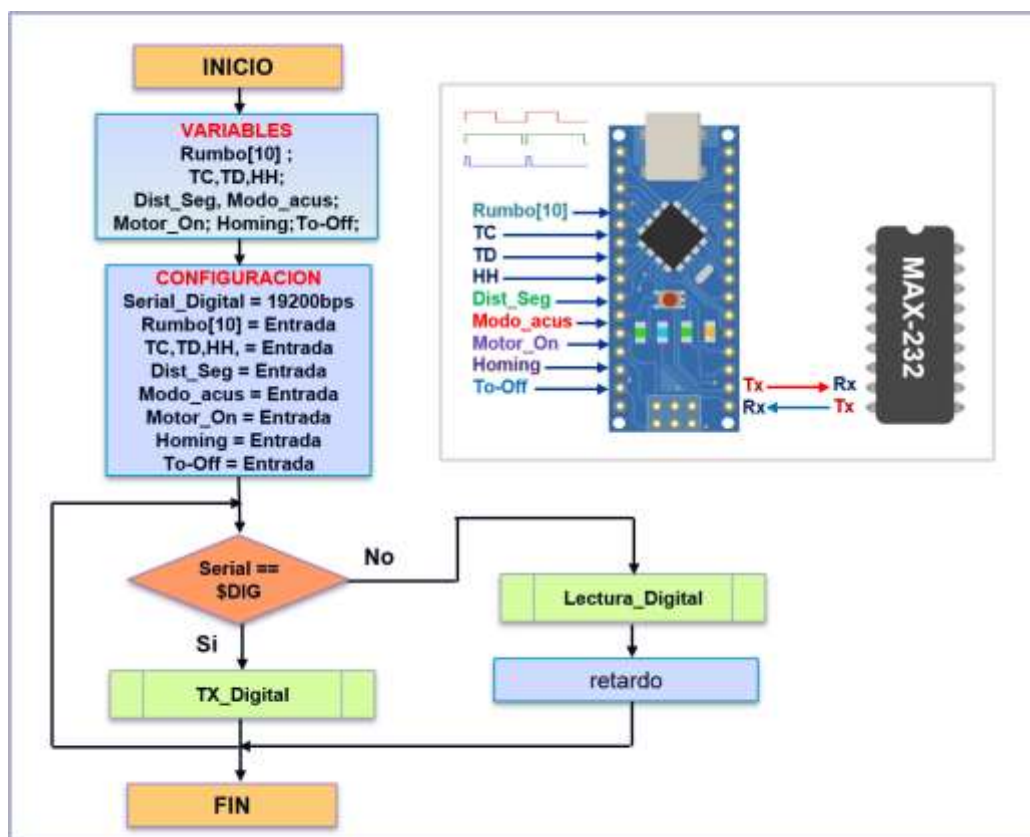


Figura 39: Diagrama de flujo del programa del algoritmo Digital.

Fuente: Elaboración propia.

Luego de la etapa de configuración se ejecuta el programa principal en un ciclo repetitivo, en esta etapa el controlador está a la espera de recibir el dato "\$DIG" por el puerto serial, mediante el algoritmo condicional se verifica si ha recibido datos por el puerto serial y que el dato recibido tenga el valor "\$DIG", en caso sea falso es decir no ha recibido datos o el dato recibido no es el correcto el programa ejecuta nuevamente la función "Lectura_Digital" almacenando los valores en un arreglo de datos para ser transmitidos, luego viene un pequeño retardo y el ciclo vuelve a repetirse, cuando el controlador de la tarjeta de Registro envía el comando "\$DIG" por el puerto serial la condicional serial verifica el si el dato recibido es el correcto y si se cumple la condición verdadero este ejecuta la función "TX_Digital", esta función transmite al controlador de registro toda la información digital previamente empaquetado en un arreglo de datos para su procesamiento, una vez transmitido los datos el programa vuelve a ejecutar el ciclo repetitivo verificando si ha recibido datos por el puerto serial.

En la figura 40 se observa cómo se realizó el diagrama de flujo del algoritmo de la función "Lectura_Digital", esta función tiene un inicio y un fin, para leer la información disponible en los puertos digitales del controlador se ejecutan funciones para cada señal digital a ser leído, dentro de las funciones digitales de lectura tenemos: "Leer_Motor_On" esta función lee la información para verificar si el motor esta encendido o apagado, "Leer_Dist_Cub" esta función lee la información para verificar si la distancia

de seguridad ha sido cubierta por el vehículo submarino, esta distancia equivale a 300 metros, “Leer_Rumbo” esta función lee la información proveniente del giro horizontal mediante 10 bits, “Leer_Homing” esta función lee la información para verificar si el vehículo se encuentra en modo de control manual o en modo automático, “Leer_Modo_Acust” esta función lee la información para verificar si el sonar de detección se encuentra en modo pasivo o activo, “Leer_Deteccion” esta función lee la información obtenida del sonar para verificar si ha detectado la señal de un blanco, la última función “Leer_Motor_On” verifica si el vehículo recibió la orden de apagado.



Figura 40: Diagrama de flujo de la función Lectura-Digital.

Fuente: Elaboración propia.

En la figura 41 se observa cómo se realizó el diagrama de flujo de las funciones, “Leer_Motor_On” y “Leer_Dist_Cub”, ambas funciones son

parecidas, el código solamente lee un bit, la función “Leer_Motor_On” inicializa la variable Motor_On a cero, se lee el valor digital del puerto correspondiente y se compara si el valor leído es uno, si la condición es verdad la variable Motor_On es igual a uno, si es falso permanece en cero y termina la función, la función “Leer_Dist_Cub” es similar, utiliza la variable Dist_cub para almacenar el valor leído del puerto específico.



Figura 41: Diagrama de flujo de la función “Leer: Motor_On y Dist_Cub”.

Fuente: Elaboración propia.

En la figura 42 se observa se observa cómo se realizó el diagrama de flujo de las funciones, “Leer_Homing” y “Leer_Modo_Acust” estas funciones tienen la misma estructura de las funciones “Leer_Motor_On” y “Leer_Dist_Cub”.

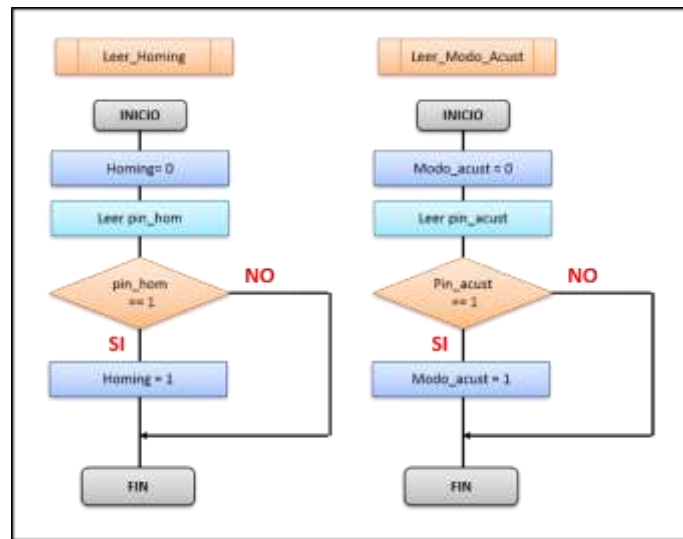


Figura 42: Diagrama de flujo de la función “Leer: Homing, Modo_Acust”

Fuente: Elaboración propia.

La señal de rumbo está compuesto por 10 bits, cada bit tiene un peso específico, el algoritmo de lectura de esta función se observa en la figura 43, se inicializa la variable rumbo a cero, se lee el valor digital del puerto del Bit_0 y se compara si el valor leído, si la condición es verdad a la variable Rumbo es igual a 1 que, en caso la condición resulta ser falso la variable rumbo permanece con el valor de inicio.

La función continua con la lectura del puerto que corresponde al Bit_1 y se repite hasta completar los 10 bits de faltantes, por último, la variable rumbo termina con el valor del rumbo real del torpedo.

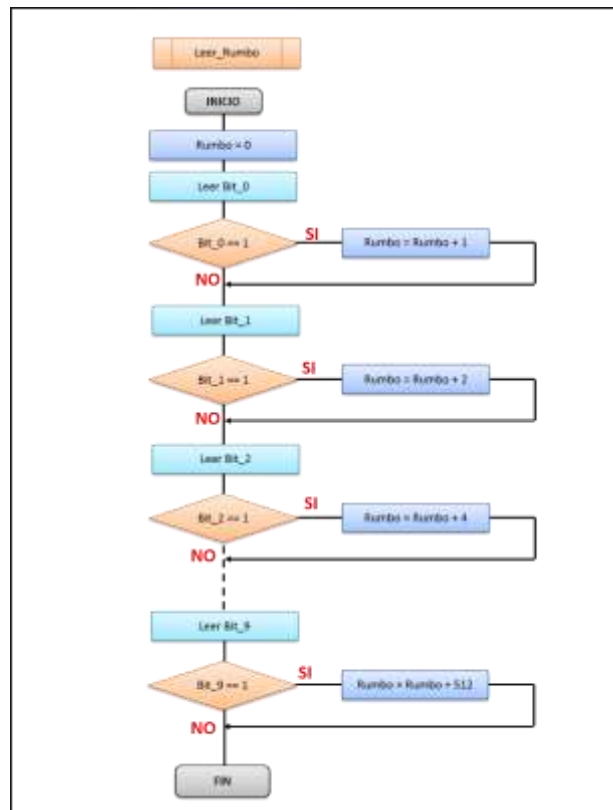


Figura 43: Diagrama de flujo de la función “Leer_Rumbo”.

Fuente: Elaboración propia.

La cabeza acústica procesa la información de detección de blanco y envía tres señales de “TC” significa contacto, “TD” significa detección y la señal “HH” que significa perdida, la función “Leer_Deteccion” se observa en la figura 44, esta función lee cada uno de estos bits, se inicializa la variable Detección a cero, se lee el valor del puerto de la señal “TC” y se compara, se verifica si el valor leído es uno lógico, si es verdad, la variable Detección es igual 1, primer peso del bit menos significativo, en caso la condición resulta falso, la variable Detección permanece con el valor de inicio, luego se lee el valor de la señal “TD” verificando el valor leído para añadir el peso

2 en caso se cumple la condición y por último se lee la señal “HH” que corresponde al peso 3 en caso se cumpla la condición de verdad.

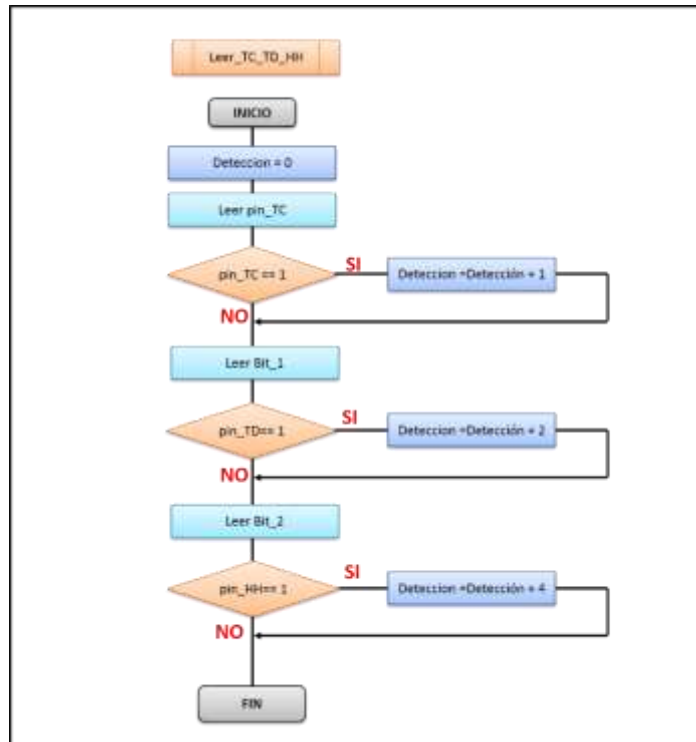


Figura 44: Diagrama de flujo de la función “Leer_TC_TD_HH”.

Fuente: Elaboración propia.

4.5.4 Diseño del DAQ de señales analógicas.

4.5.4.1 Hardware del DAQ de señales digitales.

Las señales analógicas de los sistemas de control de cabeceo y rolo generan voltajes de tensión en el rango de -12 voltios a +12 voltios, las señales analógicas de distancia, correlador de 0° y profundidad generan voltajes de tensión en el rango de 0 voltios a +12 voltios; las salidas de tensión de los sensores son lineales de acuerdo a las mediciones efectuadas (ver tabla 5).

El análisis y diseño del circuito electrónico de acondicionamiento de señal de las señales de cabeceo y balanceo están determinados por la ecuación de la recta. Para el acondicionamiento de los voltajes de cabeceo y balanceo se emplea el amplificador operacional en modo diferencial, la configuración del circuito en este modo se observa en la figura 45.

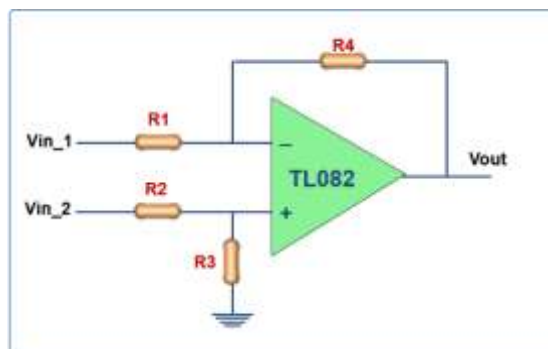


Figura 45: Amplificador diferencial.

Fuente: Elaboración propia.

Donde:

Vin1= Voltaje de entrada 1.

Vin2= Voltaje de entrada 2.

Vout = Voltaje de salida.

R1, R2, R3 Y R4 = Resistencias.

Cálculos de las resistencias.

Colocar un voltaje de referencia común a las entradas inversora y no inversora, en la figura 46 se observa el voltaje de referencia.

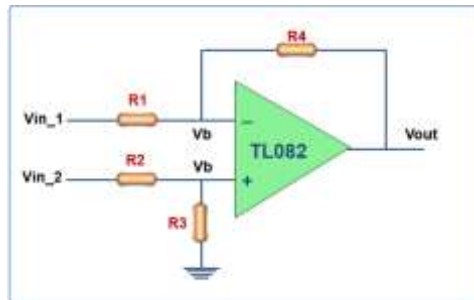


Figura 46: Amplificador diferencial con voltaje de referencia.

Fuente: Elaboración propia.

Donde:

V_b = Voltaje de referencia.

Se asignan las corrientes que circularán por las resistencias configurado en modo diferencial como se observa en la figura 47.

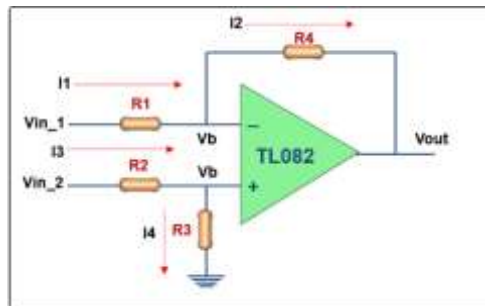


Figura 47: Diagrama de corrientes en el amplificador diferencial.

Fuente: Elaboración propia.

Donde:

I_1 = Corriente 1.

I_2 = Corriente 2.

I_3 = Corriente 3.

I_4 = Corriente 4.

La corriente I1 que circula en el resistor R1 está conectado a la entrada Vin1 y el voltaje de referencia Vb, mediante la ley de ohm se obtiene:

$$I1 = \frac{V_{in1} - V_b}{R1} \dots\dots\dots (7)$$

La corriente I2 circula por la resistencia R2; está conectado a la salida Vout y el voltaje de referencia Vb, mediante la ley de ohm se obtiene:

$$I2 = \frac{V_b - V_{out}}{R2} \dots\dots\dots (8)$$

La corriente I3 en la resistencia R3 está conectado a la tensión Vin2 y el voltaje de referencia Vb, mediante la ley de ohm se obtiene:

$$I3 = \frac{V_{in2} - V_b}{R3} \dots\dots\dots (9)$$

La corriente I4 circula por la resistencia R4; está conectado con el voltaje de referencia Vb y tierra, mediante la ley de ohm se obtiene:

$$I4 = \frac{V_b - 0}{R4} \dots\dots\dots (10)$$

Se igualan las corrientes I1 = I2

$$\frac{V_{in1} - V_b}{R1} = \frac{V_b - V_{out}}{R2}$$

$$\frac{V_{in1}}{R1} - \frac{V_b}{R1} = \frac{V_b}{R2} - \frac{V_{out}}{R2}$$

$$\frac{V_b}{R2} + \frac{V_b}{R1} = \frac{V_{in1}}{R1} + \frac{V_{out}}{R2}$$

$$V_b \left(\frac{1}{R_2} + \frac{1}{R_1} \right) = \frac{V_{in1}}{R_1} + \frac{V_{out}}{R_2}$$

Se obtiene la siguiente fórmula:

$$V_b \left(\frac{R_1 + R_2}{R_1 * R_2} \right) = \frac{V_{in1}}{R_1} + \frac{V_{out}}{R_2} \dots \dots \dots (11)$$

Se igualan las corrientes I3 = I4.

$$\frac{V_{in2} - V_b}{R_3} = \frac{V_b - 0}{R_4}$$

$$\frac{V_{in2}}{R_3} = \frac{V_b}{R_4} + \frac{V_b}{R_3}$$

$$\frac{V_{in2}}{R_3} = V_b \left(\frac{1}{R_4} + \frac{1}{R_3} \right)$$

Se obtiene la siguiente fórmula:

$$V_b = \frac{V_{in2}}{R_3} * \left(\frac{R_3 * R_4}{R_3 + R_4} \right) \dots \dots \dots (12)$$

Se substituye Vb de la fórmula (12) en la formula (11):

$$\frac{V_{in2}}{R_3} * \left(\frac{R_3 * R_4}{R_3 + R_4} \right) * \left(\frac{R_1 + R_2}{R_1 * R_2} \right) = \frac{V_{in1}}{R_1} + \frac{V_{out}}{R_2}$$

Se igualan las resistencias R1=R3 y R2=R4.

$$\frac{V_{in2}}{R_1} = \frac{V_{in1}}{R_1} + \frac{V_{out}}{R_2}$$

Se obtiene la fórmula para hallar el voltaje de salida en modo diferencial:

$$V_{out} = \frac{R_2}{R_1} (V_{in2} - V_{in1}) \dots \dots \dots (13)$$

Con la fórmula (16); $R1=4.8k$ y $R2=1k$, $V_{in2}=-12$ voltios y V_{in1} = señal de entrada; en la tabla 18 se observa el voltaje de salida (V_{out}).

Tabla 18. Valores de conversión de voltajes +/-12 voltios.

VIN1 (Vdc)	VIN2 (Vdc)	R1 (K Ω)	R2 (K Ω)	VOUT (Vdc)
12.00	-12.00	4.8	1	-5.00
11.00	-12.00	4.8	1	-4.79
10.00	-12.00	4.8	1	-4.58
9.00	-12.00	4.8	1	-4.38
8.00	-12.00	4.8	1	-4.17
7.00	-12.00	4.8	1	-3.96
6.00	-12.00	4.8	1	-3.75
5.00	-12.00	4.8	1	-3.54
4.00	-12.00	4.8	1	-3.33
3.00	-12.00	4.8	1	-3.13
2.00	-12.00	4.8	1	-2.92
1.00	-12.00	4.8	1	-2.71
0.00	-12.00	4.8	1	-2.50
-1.00	-12.00	4.8	1	-2.29
-2.00	-12.00	4.8	1	-2.08
-3.00	-12.00	4.8	1	-1.88
-4.00	-12.00	4.8	1	-1.67
-5.00	-12.00	4.8	1	-1.46
-6.00	-12.00	4.8	1	-1.25
-7.00	-12.00	4.8	1	-1.04
-8.00	-12.00	4.8	1	-0.83
-9.00	-12.00	4.8	1	-0.63
-10.00	-12.00	4.8	1	-0.42
-12.00	-12.00	4.8	1	0.00

Fuente: Elaboración propia.

En la tabla 18 se obtuvo el voltaje de salida (V_{out}) que está en el rango de 0 a -5 voltios, para convertir el voltaje negativo a positivo, en la figura 48 se observa el amplificador inversor para convertir la señal a positivo.

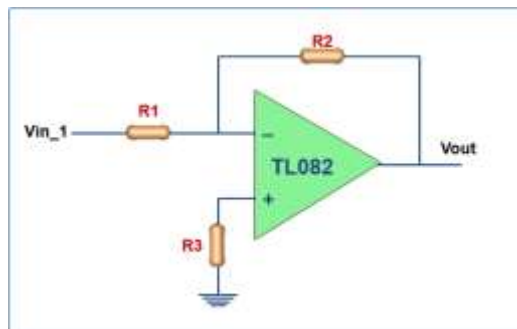


Figura 48: Amplificador inversor.

Fuente: Elaboración propia.

Donde:

V_{in} = Señal de entrada. V_{out} = Señal de salida.

$R1 = R2 =$ Resistencias.

Cálculo de las resistencias $R1$ y $R2$.

Colocar un voltaje de referencia 0 voltios a la entrada inversora del amplificador, en la figura 49 se muestra el voltaje de referencia.

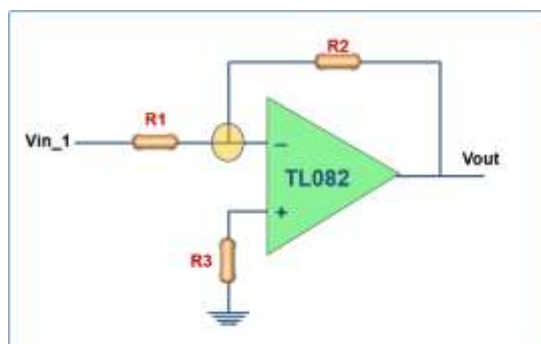


Figura 49: Amplificador inversor con referencia 0 voltios.

Fuente: Elaboración propia.

Asignar las corrientes para que circulen por las resistencias configurado en modo inversor, en la figura 50 se visualiza la configuración.

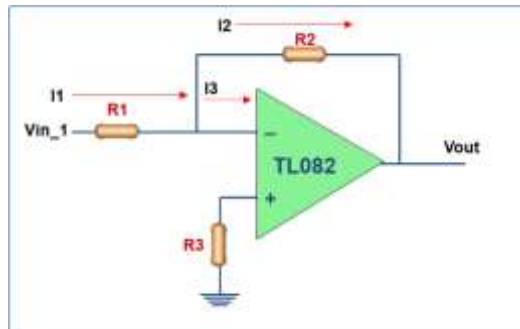


Figura 50: Diagrama de corriente en el amplificador inversor.

Fuente: Elaboración propia.

La corriente I1 en R1 está conectado con la tensión de entrada Vin1 y el voltaje de referencia es 0 voltios, utilizando la ley de ohm y se obtiene:

$$I1 = \frac{V_{in1}-0}{R1} \dots\dots\dots (47)$$

Se analizar la corriente I2 que circula por R2 que está conectado con la salida Vout y el voltaje de referencia 0 voltios, se utiliza la ley de ohm para obtener la siguiente fórmula.

$$I2 = \frac{0-V_{out}}{R2} \dots\dots\dots (15)$$

Se igualan las corrientes I1 = I2

$$\frac{V_{in1} - 0}{R1} = \frac{0 - V_{out}}{R2} \qquad \frac{V_{in1}}{R1} = \frac{-V_{out}}{R2}$$

Se obtiene la fórmula del amplificador inversor:

$$V_{out} = -V_{in} \left(\frac{R2}{R1} \right) \dots\dots\dots (16)$$

Con la fórmula (16) del amplificador inversor se obtiene: $R1=R2=10K$, en la tabla 19 se observa los cálculos del voltaje de salida (V_{out}).

Tabla 19. Conversión de voltajes negativos a positivos.

V_{in}	R1	R2	V_{OUT}
-5.00	10	10	5.00
-4.50	10	10	4.50
-4.00	10	10	4.00
-3.50	10	10	3.50
-3.00	10	10	3.00
-2.50	10	10	2.50
-2.00	10	10	2.00
-1.50	10	10	1.50
-1.00	10	10	1.00
-0.50	10	10	0.50
0.00	10	10	0.00

Fuente: Elaboración propia.

En la figura 51 se observa el diseño esquemático del acondicionador de nivel de tensión de la señal de cabeceo y roldo.

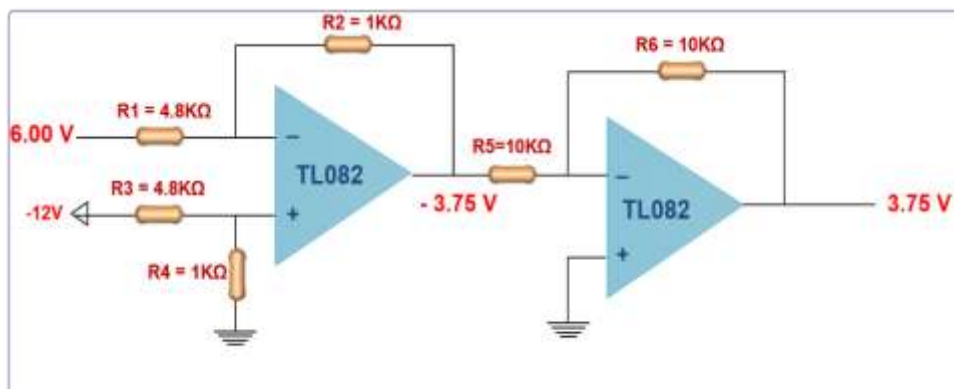


Figura 51: Circuito acondicionador de nivel de cabeceo y roldo.

Fuente: Elaboración propia.

Los voltajes de la celda de distancia en activo, correlador de 0°, voltaje de control y profundidad tienen niveles de tensión positivos de 0 a 12 voltios, con la fórmula (16) se calcula los valores $R1=2.4K$ y $R2=1k$, en la tabla 20 se observa el voltaje de salida (V_{out}) aplicando la fórmula (19).

Tabla 20. Conversión de voltajes negativos a positivos.

V_{in}	R1	R2	VOUT
12.00	2.4	1	-5.00
11.00	2.4	1	-4.58
10.00	2.4	1	-4.17
9.00	2.4	1	-3.75
8.00	2.4	1	-3.33
7.00	2.4	1	-2.92
6.00	2.4	1	-2.50
5.00	2.4	1	-2.08
4.00	2.4	11	-18.33
3.00	2.4	1	-1.25
2.00	2.4	1	-0.83
1.00	2.4	1	-0.42
0.00	2.4	1	0.00

Fuente: Elaboración propia.

En la tabla anterior se observa el voltaje de salida (V_{out}) está en el rango negativo, para convertir el voltaje positivo se vuelve a utilizar la fórmula (19) del amplificador inversor, se igualan $R1=R2=10K$, se utiliza la fórmula del amplificador inversor y se obtiene los valores mostrados en la tabla 19.

En la figura 52 se observa el circuito esquemático del acondicionador de nivel de tensión de la señal de profundidad.

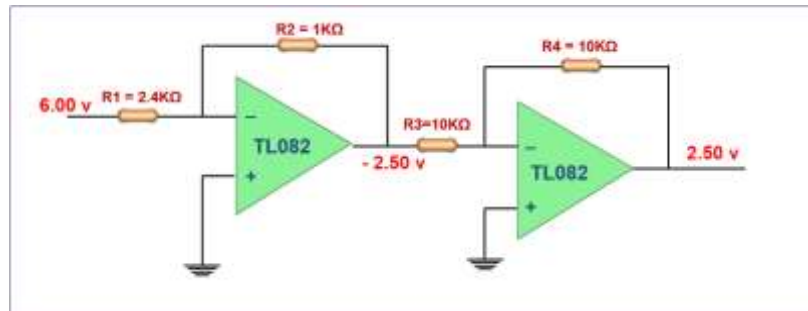


Figura 52: Circuito acondicionador de la señal profundidad.

Fuente: Elaboración propia.

Los voltajes de los amplificadores operacionales son conectados a los conversores análogo digital ADS1115, el controlador recibe la información de la variación de voltaje de las señales analógicas, calcula los valores y transmite la información por el puerto serial cuando la información es solicitada por el codificador del telegrama de datos actuales. En la figura 53 se observa el diseño del circuito DAQ Digital.

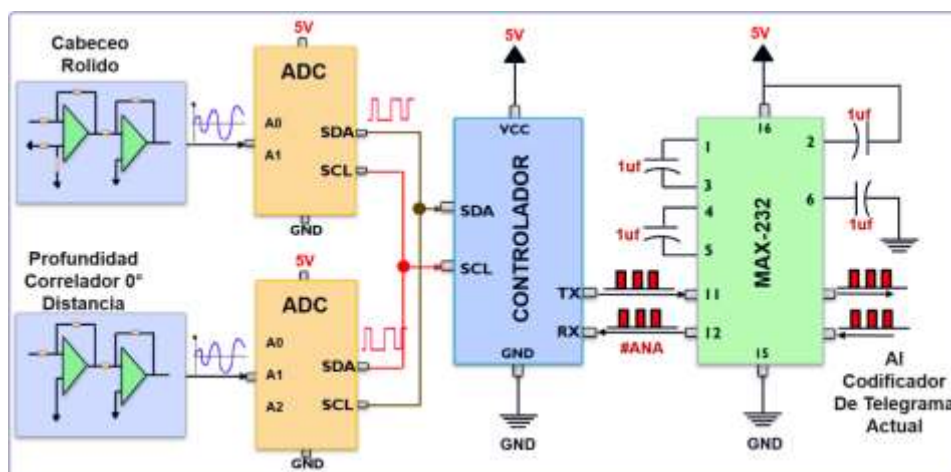


Figura 53: Circuito DAQ Analógico.

Fuente: Elaboración propia.

4.5.4.2 Software del DAQ de señales Analógicas.

Para desarrollar el algoritmo de las señales analógicas se utiliza el controlador Atmega 328, en la figura 54 se observa el diseño del diagrama de flujo del programa, se declaran las variables, se configura el puerto serial a 19200 baudios y se inicializan los conversores análogos digital.

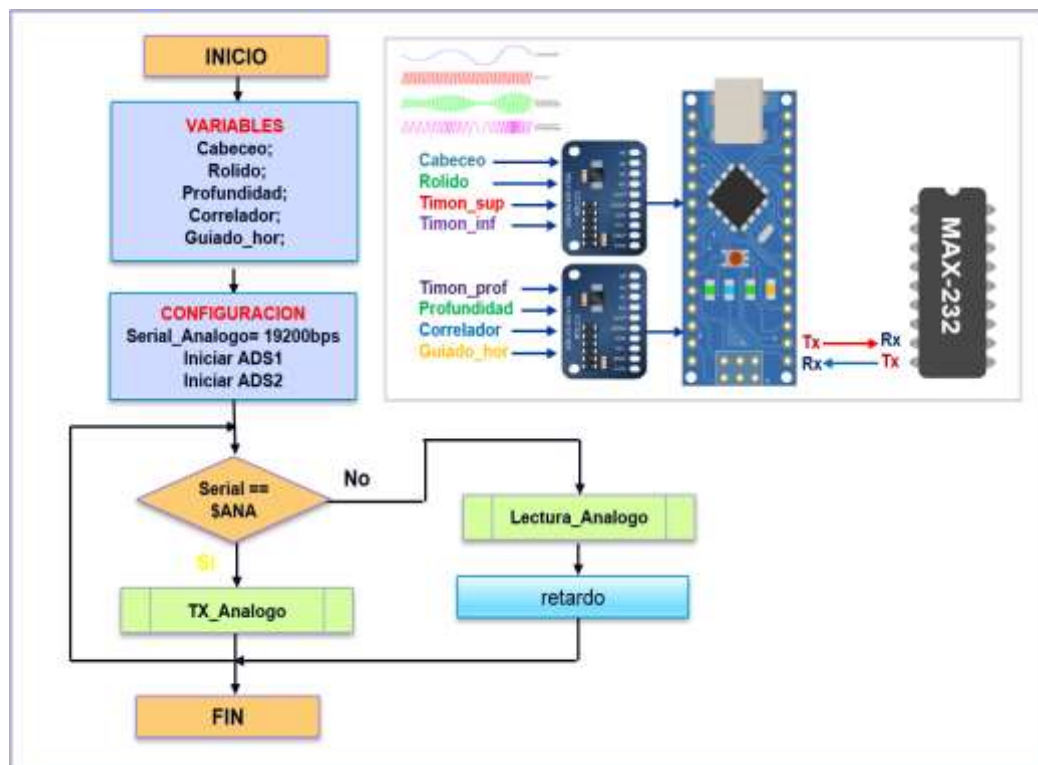


Figura 54: Diagrama de flujo del programa de señales analógicas.

Fuente: Elaboración propia.

El programa principal verifica si hay dato serial y que corresponde al dato "\$ANA" si es falso se llama a la función "Leer_Analogo" para leer la información de las señales analógicas, si es verdadero se ejecuta la función "TX_Analogo" esta función transmite los valores de las señales analógicas al codificador de Telegramas Actual.

La función “Leer_Analog” contiene 5 funciones para realizar las lecturas individualmente en cada canal, cada una de estas funciones difieren en las fórmulas de conversión de voltaje a los valores deseados que fueron descritos en el análisis de las señales analógicas, en la figura 55 se observa el diseño del diagrama de flujo de la función de lecturas analógicas.



Figura 55: Diagrama de flujo de la función “Lectura_Analogo”.

Fuente: Elaboración propia.

El controlador convierte los valores recibidos en voltajes de 0 a 5 voltios, luego en voltajes de 0 a 12 voltios o de -12 a + 12 voltios, con el valor obtenido se realiza el cálculo del valor deseado según el tipo de señal.

Con la fórmula (1) de la ecuación de la recta, se reemplazan los valores.

$$V_{in} = \left(\frac{6.144 - 0}{32767 - 0} \right) (A_{dc} - 0) + 0$$

$$V_{in} = \left(\frac{A_{dc} * 6.144}{32767} \right)$$

Se desarrolla la ecuación para hallar la fórmula de conversión ADC.

$$V_{in} = \frac{(A_{dc} * 0.1875)}{1000} \dots\dots\dots (17)$$

Para hallar el voltaje de -12 voltios a +12 voltios se utilizan la formula (1) y se reemplazan los valores en la ecuación de la recta:

$$V_{out} = \left(\frac{12 - (-12)}{5 - 0} \right) (V_{in} - 0) + (-12)$$

Se obtiene la siguiente formula de voltajes -12 a +12 voltios.

$$V_{out} = \left(\frac{24}{5} \right) (V_{in}) - 12 \dots\dots\dots (18)$$

Para hallar el voltaje de 0 a +12 voltios se utilizan la formula (1) y se reemplazan los valores en la ecuación de la recta:

$$V_{out} = \left(\frac{12 - 0}{5 - 0} \right) (V_{in} - 0) + 0$$

Se obtiene la siguiente formula de voltajes 0 a 12 voltios.

$$V_{out} = \left(\frac{12}{5} \right) (V_{in}) \dots\dots\dots (19)$$

La señal de cabeceo utiliza la función “Leer_Cabeceo”, en la figura 56 se observa el diagrama de flujo de esta función, se inicia las variables “volt” y “cont”, luego se efectúa 10 lecturas del canal analógico y se obtiene el valor promedio, para la conversión de voltajes se utiliza las siguientes formulas:

$$V_{in} = \frac{(A_{dc} * 0.1875)}{1000} \dots\dots\dots (17)$$

$$V_{out} = \left(\frac{24}{5}\right) (V_{in}) - 12 \dots\dots\dots (18)$$

$$C_{ab} = 8.787346 * V_{in} \dots\dots\dots (2)$$

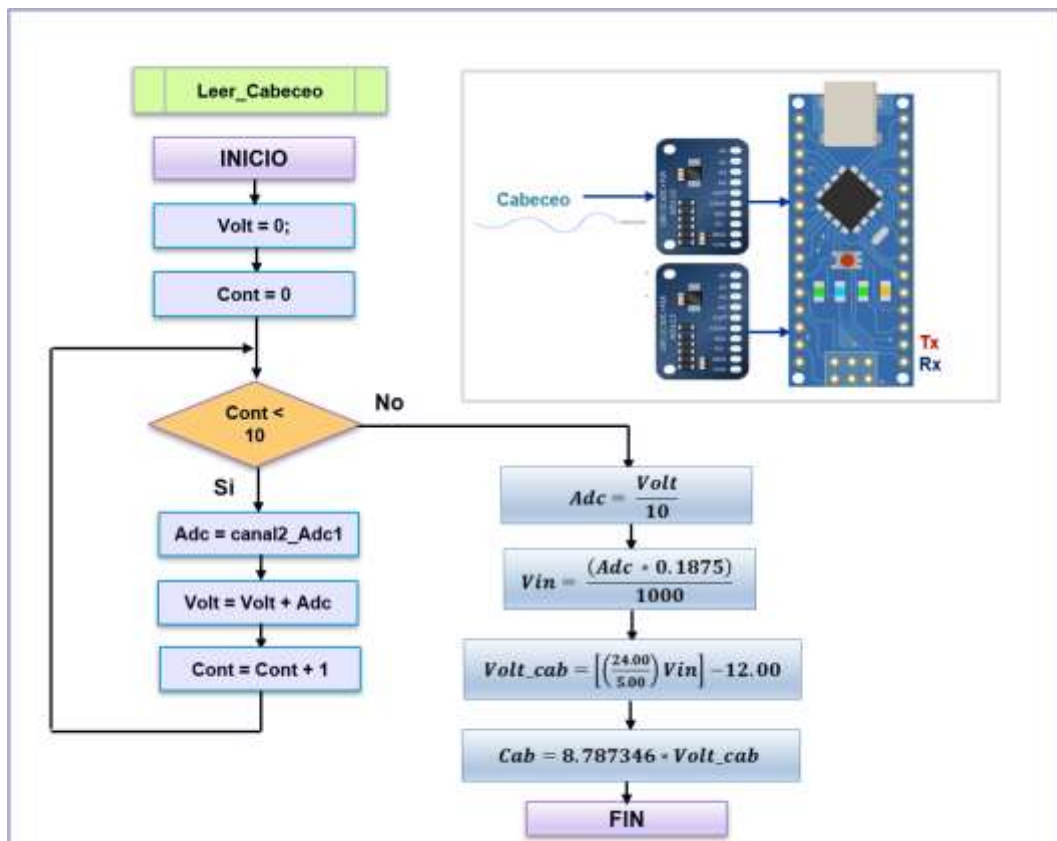


Figura 56: Diagrama de flujo de la función “Leer_Cabeceo”.

Fuente: Elaboración propia.

Para La señal balanceo se utiliza la función “Leer_Rolido”, en la figura 57 se observa el diagrama de flujo de esta función, es similar a la función de cabeceo, el canal analógico de lectura del ADC es el canal 2, para obtener el valor deseado de balanceo se utilizan las siguientes formulas.

$$V_{in} = \frac{(Adc * 0.1875)}{1000} \dots\dots\dots (17)$$

$$V_{out} = \left(\frac{24}{5}\right) (V_{in}) - 12 \dots\dots\dots (18)$$

$$Rol = V_{in} \dots\dots\dots (3)$$

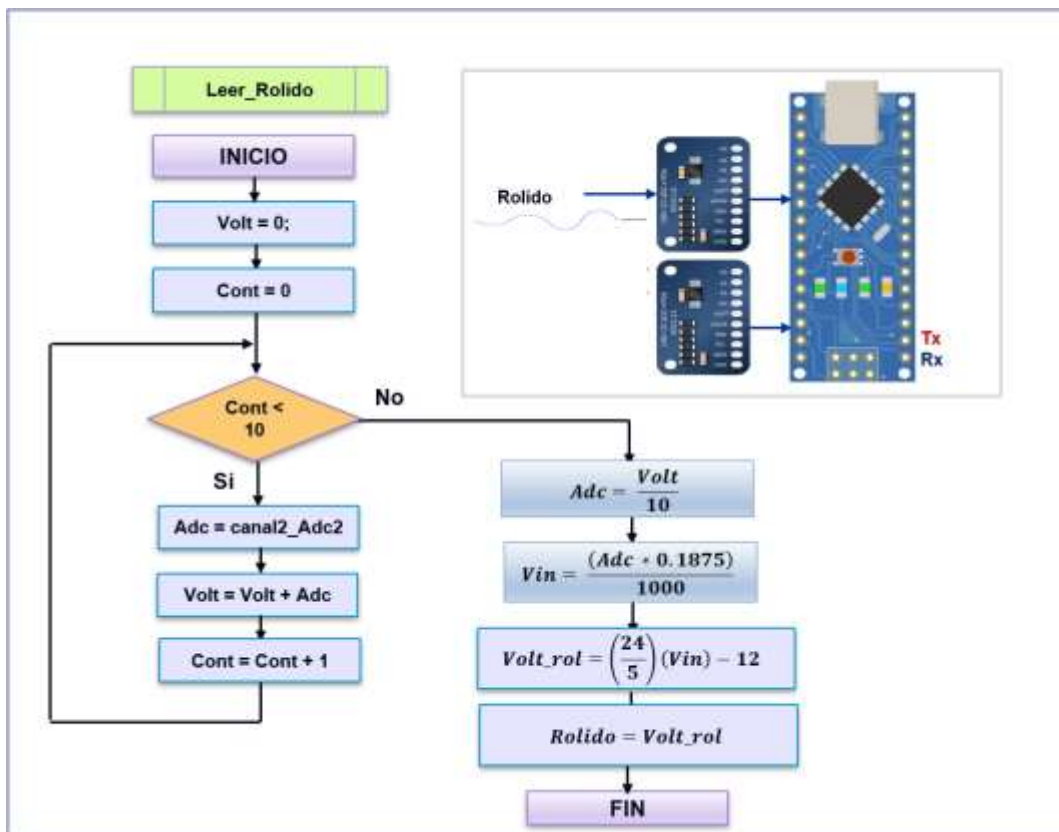


Figura 57: Diagrama de flujo de la función “Leer_Rolido”.

Fuente: Elaboración propia.

La función “Leer_Profundidad”, es utilizado para obtener la profundidad actual del torpedo, en la figura 58 se observa el diagrama de flujo de esta función, para obtener el valor deseado de profundidad en pies se utilizan las siguientes formulas.

$$V_{in} = \frac{(Adc * 0.1875)}{1000} \dots\dots\dots (17)$$

$$V_{out} = \left(\frac{24}{5}\right) * (V_{in}) \dots\dots\dots (19)$$

$$Prof = 63.3333 * V_{in} \dots\dots\dots (6)$$

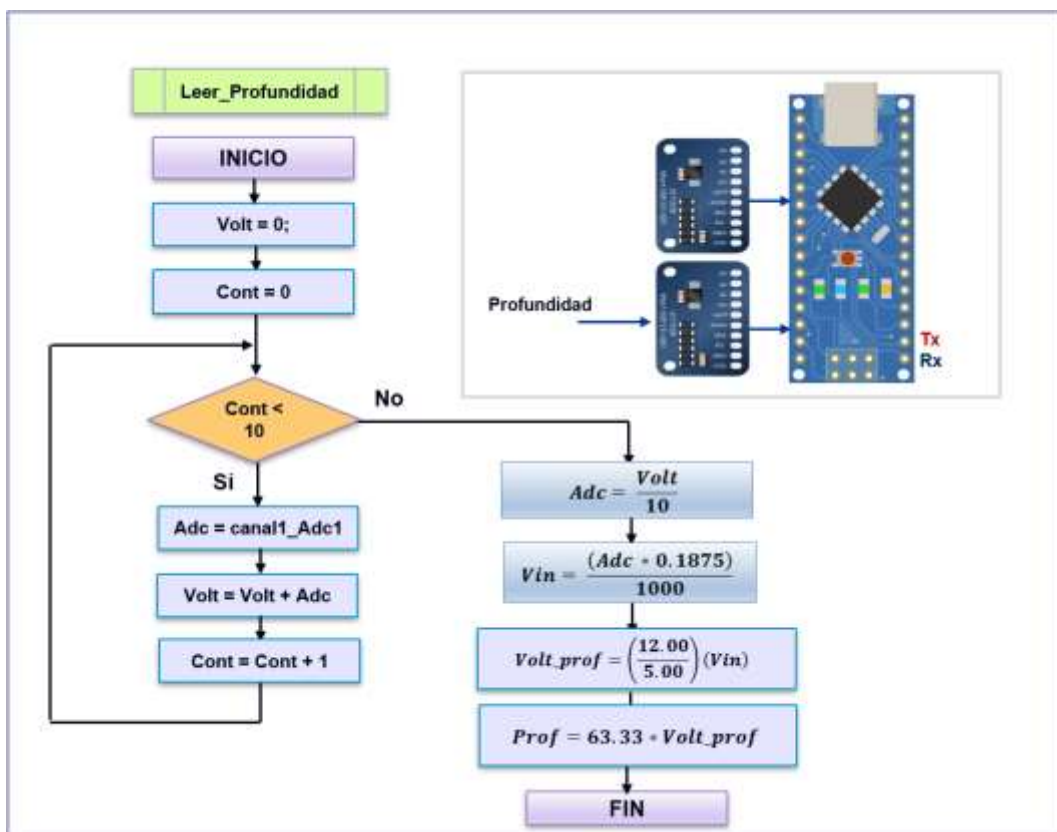


Figura 58: Diagrama de flujo de la función “Leer_Profundidad”.

Fuente: Elaboración propia.

Para obtener el valor del correlador de 0° se utiliza la función “Leer_Correlador”, en la figura 59 se observa el diagrama de flujo de esta función es similar a la función de profundidad, para obtener el valor deseado del correlador de 0° se utilizan las siguientes formulas.

$$V_{in} = \frac{(Adc * 0.1875)}{1000} \dots\dots\dots (17)$$

$$V_{out} = \left(\frac{24}{5}\right) * (V_{in}) \dots\dots\dots (19)$$

$$Corr = V_{out} \dots\dots\dots (3)$$

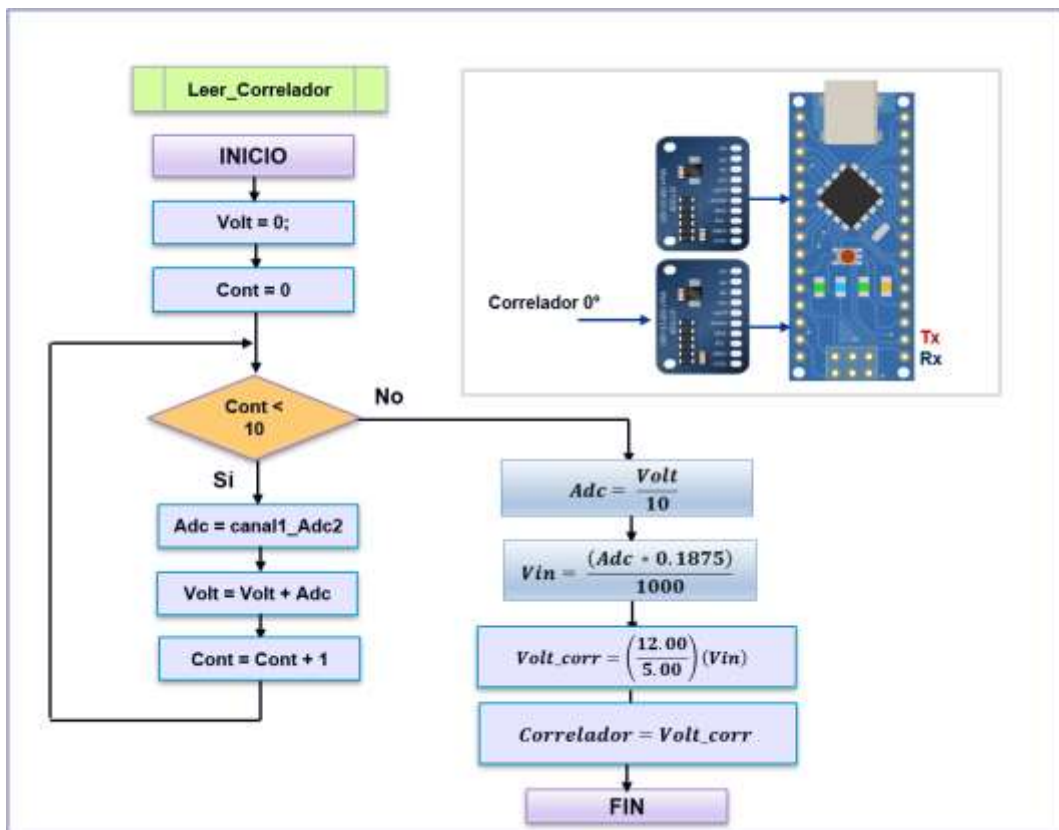


Figura 59: Diagrama de flujo de la función “Leer_Correlador”.

Fuente: Elaboración propia.

Para obtener la distancia se utiliza la función “Leer_Distancia”, en la figura 60 se observa el diagrama de flujo, para obtener el valor deseado de distancia al blanco en modo activo se utilizan las siguientes formulas.

$$V_{in} = \frac{(Adc * 0.1875)}{1000} \dots\dots\dots (17)$$

$$V_{out} = \left(\frac{24}{5}\right) * (V_{in}) \dots\dots\dots (19)$$

Formula (5): Distancia en activo (metros).

$$Dist = \frac{1479 * 0.208 * V_{in}}{2} \dots\dots\dots (9)$$

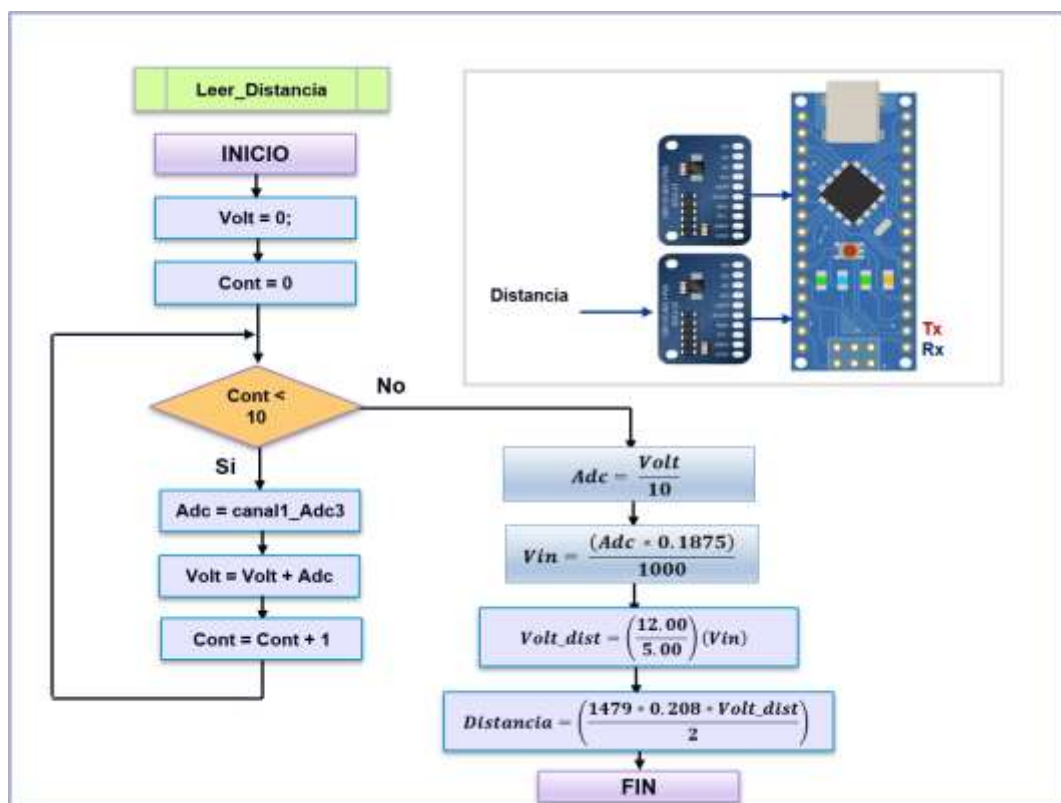


Figura 60: Diagrama de flujo de la función “Leer_Distancia”.

Fuente: Elaboración propia.

4.5.5 Diseño del Codificador de Telegrama Actual.

4.5.5.1 Hardware del Codificador de Telegrama Actual.

El hardware del Codificador de Telegrama Actual se basa en el uso del microcontrolador Atmega2560, este tipo de microcontrolador posee 4 UARTs (puertos seriales por hardware), se empleó 2 puertos seriales, el primer puerto serial es utilizado para la comunicación serial con el hardware DAQ digital, el segundo puerto serial es empleado para la comunicación con el hardware DAQ Analógico; ambos puertos seriales son empleados por el controlador para solicitar información de las señales digitales y analógicas, para establecer una comunicación estable se utiliza el circuito MAX-232, este dispositivo permite la conexión necesaria para lograr comunicación entre el puerto serial con funcionamiento en base a señales de nivel TTL/CMOS; además este tipo de circuito integrado posee dos conversores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL. El puerto TX es la señal de transmisión de datos, RX es la de recepción.

El controlador Atmega2560 transmite un valor de cabecera para solicitar la información de cada una de las tarjetas de adquisición de datos digitales y analógicas; la cabecera serial "DIG" es utilizada para solicitar datos digitales; la cabecera serial "ANA" es utilizada para solicitar datos analógicos; ambas solicitudes se efectúan durante la etapa de sincronización de la octava palabra del telegrama de datos actuales.

Una vez obtenido la información de las señales digitales y analógicas, el controlador codifica la información en un arreglo de 8 palabras de 16 bits, cada palabra está compuesto con información de los sensores, en cada palabra se utilizan bits de paridad y cantidad de unos lógicos, esto con la finalidad de comprobar que no existan pérdidas durante la transmisión de toda la información.

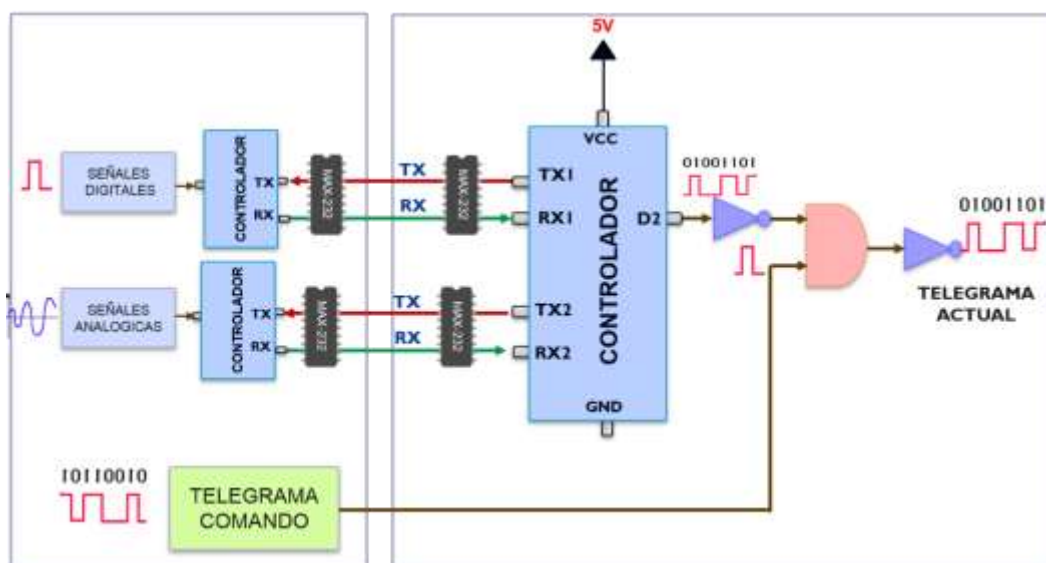


Figura 61: Esquema del Codificador de Telegrama Actual.

Fuente: Elaboración propia.

La salida del telegrama de datos actuales se produce por el puerto digital “D2” del controlador, esta señal es invertida y amplificada a la compuerta lógica AND que evalúa la correcta recepción del telegrama de comando; la salida nuevamente es invertida y acoplada al módulo transmisor NVS-B para transmitir el telegrama de datos actuales al Sistema de Control de Tiro.

4.5.5.2 Software del Codificador de Telegrama Actual.

Para desarrollar el algoritmo del Codificador de Telegrama Actual se utiliza el controlador Atmega2560, este controlador dispone de 4 puertos seriales, el puerto serial 1 y 2 son utilizados para la comunicación con las tarjetas de adquisición de datos, en la figura 62 se observa el diagrama de flujo del programa, la primera etapa de la programación es la configuración, en esta etapa se configuran los puertos seriales a 19200 baudios, se inicializa las variable Flag=1; el puerto de salida de telegrama denominado Bit se pone en estado 1 lógico y se activa la interrupción del timer 1; la segunda etapa es el programa principal para fabricar el telegrama de datos actuales.

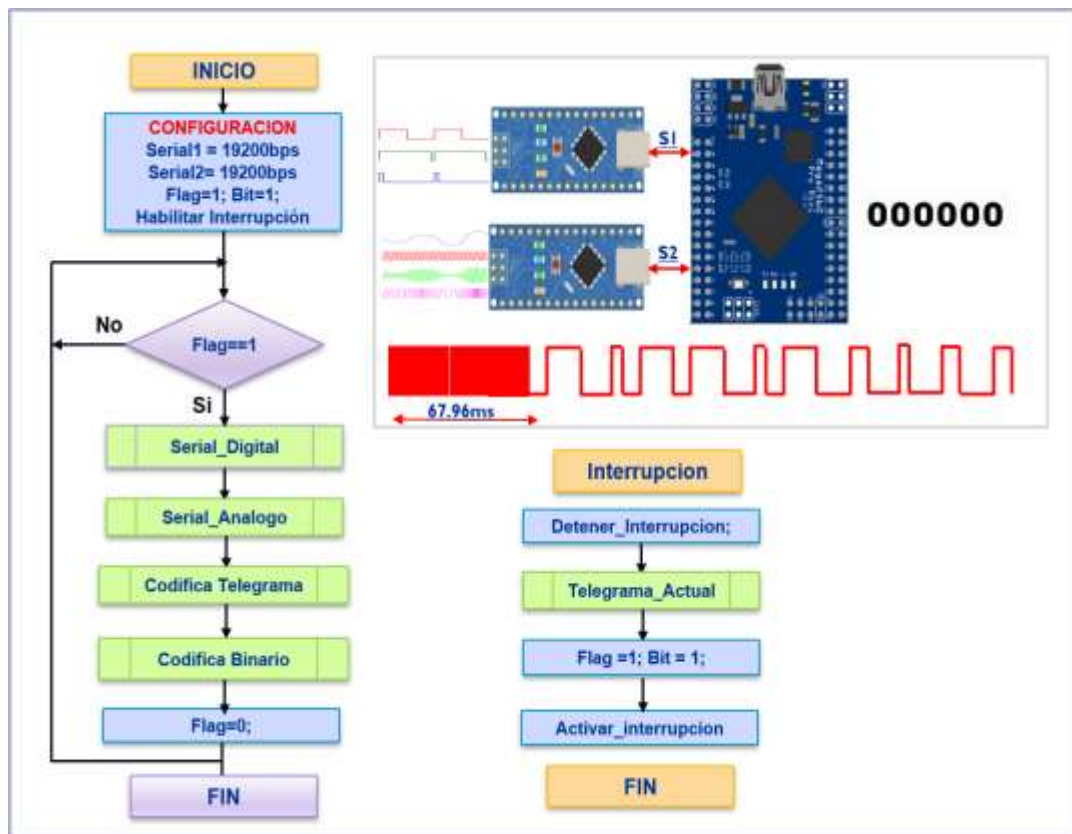


Figura 62: Diagrama de flujo del Codificador de Telegrama Actual.

Fuente: Elaboración propia.

El programa principal compara la variable Flag; si el resultado es verdadero; se ejecutan 4 funciones; la función "Serial_Digital", solicita información de datos digitales; la función "Serial_Analogo", solicita información de datos analógicos; la función "Codifica_Telegrama" prepara los telegramas y la función "Codifica_Binario" prepara en binario las 7 palabras para ser transmitidos; finalmente se pone a cero la variable flag; se retorna al inicio del programa y la condicional resulta ser falso; esta secuencia se ejecuta hasta que ocurra la interrupción por timer; la interrupción ocurre durante la palabra de sincronismo que dura 67.96 milisegundos.

La interrupción ejecuta la función "Telegrama_Actual"; por el puerto "D2" del controlador se transmite la información del telegrama actual, cada bit es transmitido a 206 baudios para ser adaptados posteriormente al sistema de comunicación del torpedo SST-4; en la figura 63 se observa el telegrama de datos actuales, las palabras del 1 al 7 contienen la información de las señales digitales y analógicas codificadas, la palabra 8 es la señal de sincronismo.

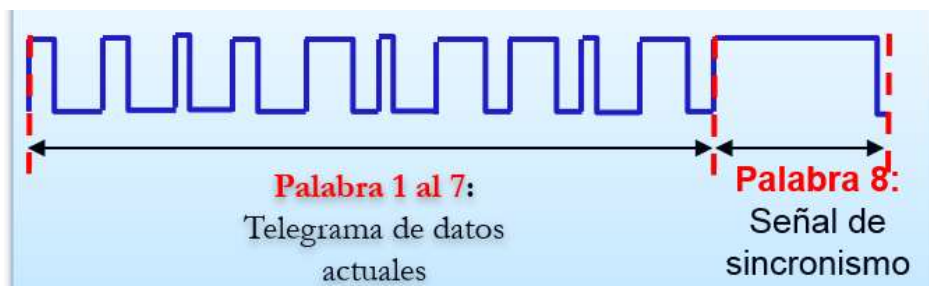


Figura 63: Diseño del telegrama de datos actuales.

Fuente: Elaboración propia.

4.5.6 Diseño del Decodificador de Telegrama Actual.

4.5.6.1 Hardware del Decodificador de Telegrama Actual.

El módulo NVS-B del Sistema de Control de Tiro, recibe el telegrama de datos actuales que son transmitidos desde la computadora del torpedo SST-4, los datos llegan en un protocolo serial de 8 palabras con niveles de tensión CMOS, el dispositivo optoacoplador adapta la señal tipo CMOS a nivel TTL para ser leído por el puerto digital "D2" del controlador Atmega328; para la captura de la información del telegrama de datos actuales el controlador se sincroniza con la palabra de sincronismo, la técnica de sincronización es por medición del ancho de pulso que equivale a 67.96 milisegundos; en la figura 64 se observa el diseño esquemático de la etapa del decodificador de telegrama actual con la interfaz gráfica.

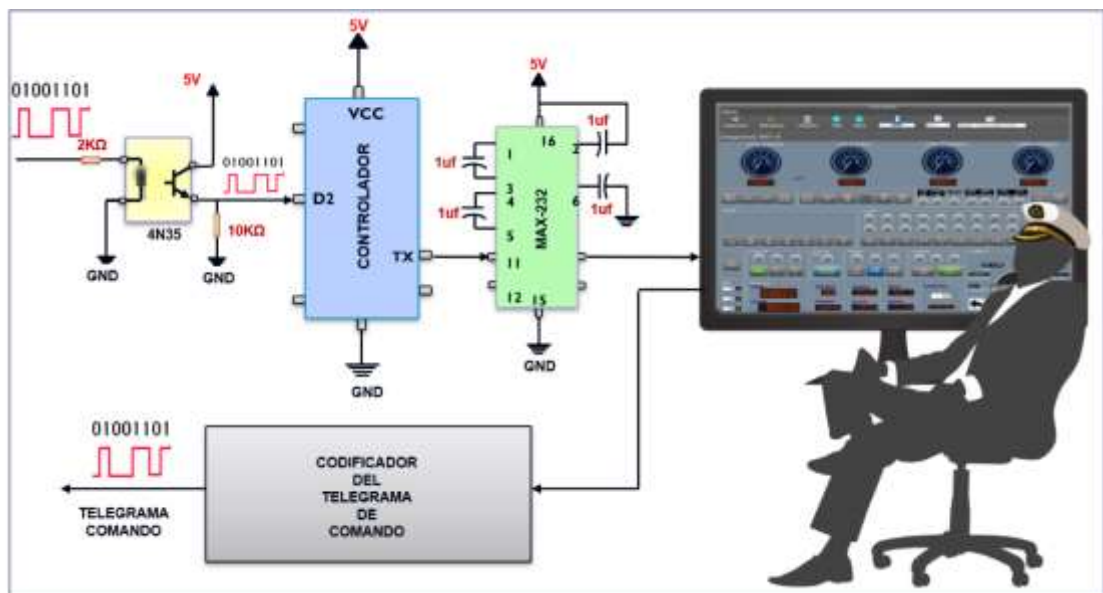


Figura 64: Esquema del Decodificador de Telegrama Actual.

Fuente: Elaboración propia.

El controlador tiene las características técnicas necesarias para sincronizar el telegrama de datos actuales mediante interrupción por flanco de subida, el cual mide el ancho de pulso de la señal en estado lógico 1; luego se efectúa la lectura de información de las palabras 1 al 7 para ser decodificados de acuerdo al diseño del telegrama de datos actuales; la técnica de decodificación consiste en capturar cada bit que llega por el puerto digital; estos se almacenan en 7 variables que corresponden a las 7 palabras para ser convertidos en valores enteros. El resultado obtenido es transmitido al computador del Sistema de Control de Tiro por el puerto serial a una velocidad de 19200 baudios, el circuito MAX-232 es utilizado para adaptar los niveles de tensión TTL a MOS; posteriormente con el conversor RS-232 a USB la información es transmitida a la computadora; en la figura 65 se observa la etapa de adaptación de la comunicación serial.

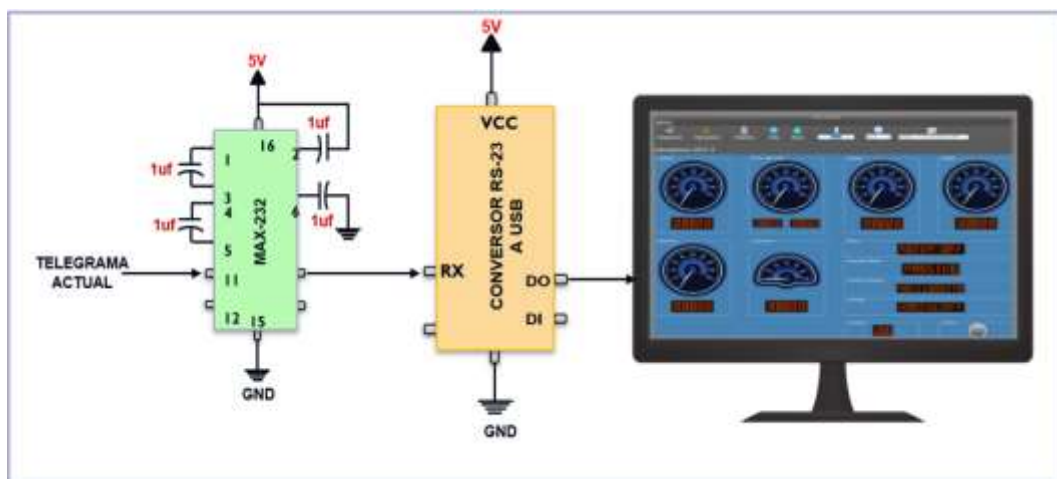


Figura 65: Esquema de conversión serial TTL a USB.

Fuente: Elaboración propia.

4.5.6.2 Software del Decodificador de Telegrama Actual.

El algoritmo del Decodificador de Telegrama Actual utiliza el controlador Atmega328, este controlador dispone de 1 puerto serial para transmitir al computador de la interfaz gráfica el resultado obtenido de la decodificación del telegrama de datos actuales; en la figura 66 se observa el diagrama de flujo del programa, la primera etapa de la programación es la configuración, en esta etapa se configuran: el puerto serial a 19200 baudios, el puerto de entrada para la lectura del telegrama actual y se inicializa el temporizador.

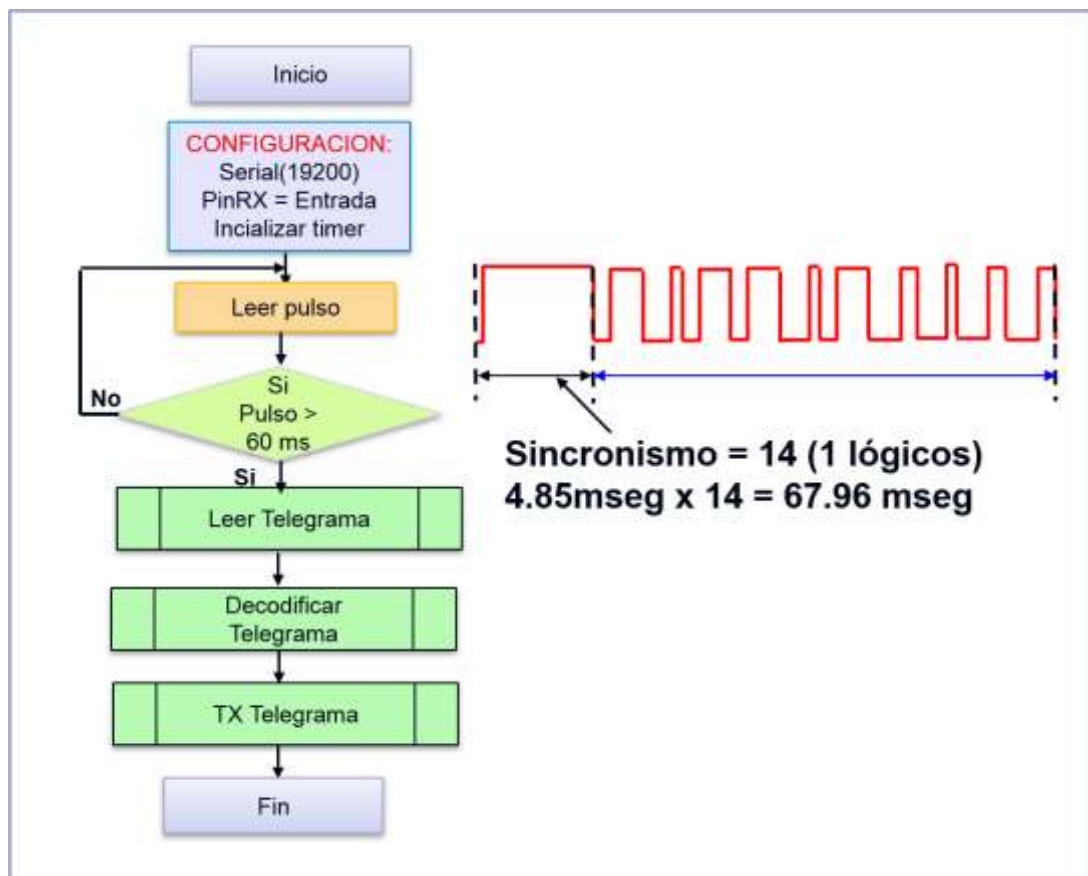


Figura 66: Programa del Decodificador del Telegrama Actual.

Fuente: Elaboración propia.

El programa principal lee el ancho de pulso del telegrama actual, en caso el ancho del pulso es mayor a 60 milisegundos que corresponde a la palabra 8 de sincronismo se ejecuta 3 funciones: la primera función “Leer_Telegrama” captura la información serial de las palabras 1 al 7; la segunda función “Decodificar_Telegrama” convierte los datos leídos de cada palabra de 16 bits en valores enteros; la última función transmite por el puerto serial los datos convertido en enteros para su visualización en la interfaz gráfica.

En la figura 67 se observa el diagrama de flujo de la función de lectura del telegrama actual; en esta función se ejecutan 7 contadores de 16 repeticiones; cada contador almacena cada 4.85 milisegundos los bits del telegrama de datos actuales.

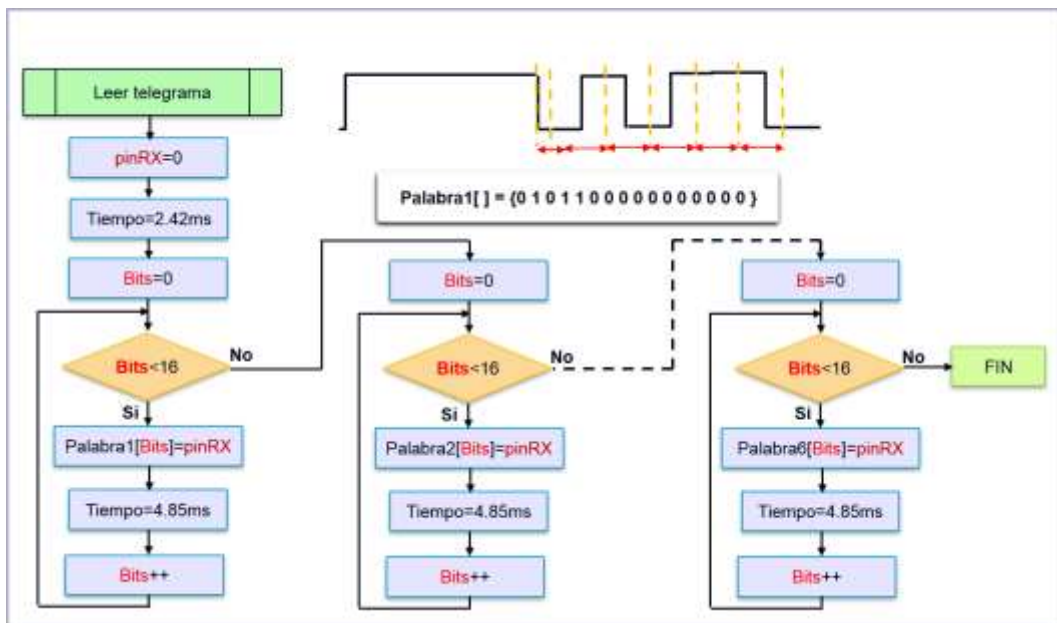


Figura 67: Función de lectura del telegrama de datos actuales.

Fuente: Elaboración propia.

4.5.8 Diseño y desarrollo de la interfaz gráfica.

Para la implementación de la interfaz gráfica se tuvo en cuenta diversos factores necesarios en todo desarrollo tecnológico con respecto a la elaboración de un proyecto de alta envergadura. Debe destacarse el hecho que para desarrollar una interfaz existen en la actualidad una variedad de lenguajes de programación, cada uno con características propias y ventajas singulares que llevan a cuestionarse sobre la mejor alternativa para el proyecto en curso.

Para la correcta elección del lenguaje y la plataforma en la que se desarrollará la interfaz se tomó en cuenta los lenguajes de programación más utilizados, de mayor soporte y con una comunidad grande de desarrolladores. Este criterio se toma por el hecho de que se necesita dotar de portabilidad a nivel de actualización y mejoras al programa en cuestión y debe realizarse con las herramientas adecuadas que permitan la aceleración y practicidad de dicha actividad. En tal sentido, hasta noviembre de 2019, la empresa TIOBE, dedicada al rubro de la calidad de software y tendencias a nivel mundial, en su top 5 mensual de lenguajes de programación, coloca la siguiente lista (Tiobe,2018).

1. Java
2. C
3. Python
4. C++
5. C#

Indudablemente ante la existencia de diversidad de dispositivos Android, el lenguaje de programación Java es el más utilizado en la actualidad. Le siguen los lenguajes C y Python que tienen un potencial bastante elevado en programas de tipo descriptivo. En la cuarta posición se tiene al lenguaje de programación C++ con todo su despliegue gráfico y funcionalidades notorias. Sin embargo, los lenguajes de programación mencionados, si bien es cierto tienen la capacidad de lograr una interfaz gráfica bastante apreciable y manejable, no se comparan con todas las ventajas que ofrece el lenguaje C# para la elaboración de programas e interfaces gráficas para la plataforma de Windows. Es por ello, que ante todo lo descrito, se hace la elección del lenguaje C# para el desarrollo total del software de interfaz para la modernización del Test Stand.

C# es un lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes lo cual lo dota de ventajas sobre otros similares. Se utiliza un lenguaje estandarizado para Microsoft por el hecho de que en la actualidad el mayor número de sistemas operativos utilizados por usuarios a nivel mundial son desarrollados por dicha empresa. La siguiente figura explica mejor dicha situación:

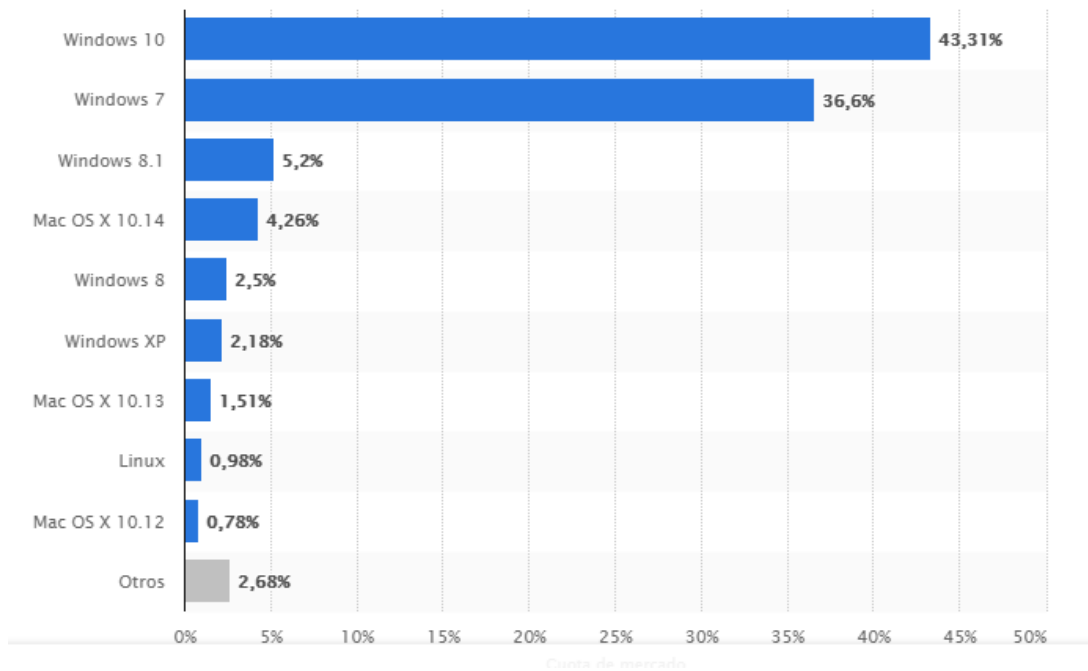


Figura 68: Cuota del mercado mundial de S.O. en el 2019

Fuente: (Statista.com, 2019)

Como se puede apreciar, en la actualidad los últimos 3 sistemas operativos desarrollados por Microsoft encabezan la lista con notoria diferencia versus sus adversarios. Dicha tendencia es bastante homóloga en nuestro país y, de forma más específica y con respecto al objeto del estudio en cuestión, en la Marina de Guerra del Perú la mayor parte de computadoras tienen dichos sistemas operativos, reservando los de Linux y derivados para las áreas de información, servidores y similares.

A nivel ingenieril y de desarrollo tecnológico, la mayor parte de herramientas de software está diseñadas y optimizadas para sistemas operativos de Windows y versiones. Un claro ejemplo de ello son las herramientas desarrolladas por la empresa Autodesk cuya mayoría de

opciones de herramientas de desarrollo está dirigida a Microsoft. Es por todos estos puntos y tendencias, que se selecciona como lenguaje de programación C# para el desarrollo de la interfaz gráfica y plataforma destino Windows 10 Pro de Microsoft.

Así mismo, como entorno de desarrollo integrado óptimo para C# se hace uso del Microsoft Visual Studio. Así mismo, el .NET Framework que se utilizará será el 4.5. En tal sentido, basta con hacer uso de Microsoft Visual Studio 2012 al ser la primera IDE en soportar dicha versión de Framework.

Por último, se destaca las ventajas de C# con respecto a sus competidores y similares más cercanos con respecto al proyecto en cuestión. Al tratarse de un proyecto con un nivel de confidencialidad bastante elevado, es recomendable el uso de plataformas y lenguajes cerrados que no sean necesariamente de libre uso o denominados open source manipulables por personas no autorizadas. En tal sentido, el lenguaje y el programa en sí dota de tal nivel de seguridad necesario ante cualquier tipo de amenaza externa. Por otro lado, el lenguaje de programación C# tiene la totalidad de librerías implementadas para el control de los periféricos de una computadora. Es decir, tiene el control sobre los puertos USB, comunicaciones con el computador, puerto y comunicaciones vía ethernet, etc. En este punto se destaca que para la comunicación de la interfaz gráfica con las tarjetas de adquisición de datos y tarjetas de comunicación con el torpedo se hará uso de la comunicación serial. En este sentido, el lenguaje C# presenta una gama de características que aseguran la

transmisión y recepción de información por todos los medios mencionados. Así mismo, presenta ventajas técnicas como el uso de memoria optimizado, declaración legible de variables, sencillez en su programación, elevado nivel de lenguaje orientado a objetos, etc.

En el proceso de gestación del presente proyecto se partió con una premisa fundamental con respecto a utilizar tecnología vigente en la modernización de un equipo que data de los años 70. Indudablemente parte del proceso de modernización correspondía a mantener la filosofía del equipo en cuestión, la funcionalidad y algunos aspectos importantes.

Todo ello por el hecho de que por años el personal ha sido capacitado en el control de esa consola. Sin embargo, realizar nuevamente una consola, como la que se puede visualizar en la figura posterior y que corresponde al sistema de control principal del Test Stand, conllevaba a estructurar nuevamente botoneras, indicadores, llaves, etc., que, con el tiempo y debido al uso constante al que está sometido este equipo, iba a terminar deteriorándose. Así mismo, debe considerarse la proximidad geográfica de la dependencia al mar y, por ende, los problemas de salinidad, humedad y deterioro debido a este. Por último, si se necesitaban modificar funciones, agregar capacidades, o realizar reparaciones, es sabido que un sistema físico completamente presenta mayores dificultades, tiempo y gasto.



Figura 70: Consola de pruebas Test-Stand.

Fuente: Elaboración propia

Ante lo expuesto, se llegó a la conclusión de que la mejor alternativa de modernización no debía basarse en un sistema de hardware si no en una plataforma de software, completamente funcional, ergonómica a nivel de manejo y que mantenga la fisionomía, funcionalidad y respuesta que el sistema original.

En tal sentido, se optó por desarrollar una interfaz gráfica en el lenguaje C# en la IDE de Microsoft Visual Studio 2012, que se comunicara con las tarjetas electrónicas de adquisición de datos y de control descritas también en el presente informe.

Para ello se comenzó el proceso de abstracción de la interfaz como modelo similar a la plataforma principal de control. En dicho aspecto era necesario considerar el número de botones e indicadores visuales. Así mismo, que estos tengan una proporción adecuada en la totalidad de la pantalla para asegurar que el usuario pueda identificar de manera precisa cada botón sin confundirlo con otro y asegurando una comunicación exitosa con el CPU del torpedo.

En tal sentido, se optó por realizar una ventana dentro del software que mantuviera el orden y disposición de dichos botones e indicadores por el hecho de que el personal ya se encuentra adaptado a dicho sistema y, por un proceso simple de cognición y repetición pueda adaptarse de manera más precisa y rápida al software y equipo modernizado.

4.6 Procesamiento estadístico y análisis de los datos.

Para realizar el procesamiento y análisis de los datos obtenidos se empleará el uso de tablas o cuadros en hojas de cálculo. Además, se tiene que realizar las descripciones de las mediciones que hemos obtenido los cuales serán contrastados con los objetivos de nuestro trabajo y a su vez deben de ser verificados con la hipótesis de nuestra investigación.

Para realizar el análisis de la información, durante el desarrollo y pruebas del sistema, los datos son comparados mediante gráficos realizados en hojas de cálculo donde se pueden generar conclusiones del comportamiento del sistema desarrollado.

CAPÍTULO V

RESULTADOS

5.1 Resultados.

5.1.1 Tarjeta de adquisición de señales digitales.

Se desarrolló y confecciono la tarjeta electrónica de adquisición de datos de las señales digitales, el hardware reúne todos los componentes electrónicos necesarios para el acondicionamiento de los niveles de voltaje de las señales digitales presentes en la unidad de procesamiento de señales (SPU-A), estas señales son adaptados a los puertos digitales del controlador.

Con la ayuda del osciloscopio en la figura 70 se observa el comportamiento de la etapa de conversión de niveles de las señales digitales, se coloca una entrada digital (Color amarillo) con nivel de tensión de 12 voltios y se obtiene una salida (Color verde) con un nivel de tensión de 5 voltios.

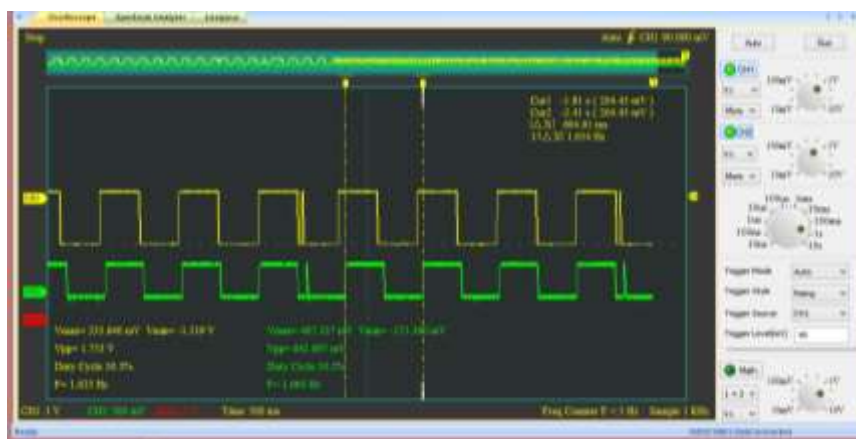


Figura 71: Vista en osciloscopio de la adaptación de la señal digital.

Fuente: Elaboración propia.

La tarjeta electrónica denominada tarjeta controladora teretor DAQ Digital se observa en la figura 71, resta tarjeta recibe las señales eléctricas digitales de rumbo, modo acústico, contacto en niveles de tensión de +12 voltios como “1” lógico, y cero voltios como “0” lógico, estas señales son acondicionados con un circuito opto-acoplador para producir señales digitales con niveles de tensión de 5 voltios para un “1” lógico, el controlador mediante sus puertos de entrada se encarga de leer la información digital para ser almacenados en su memoria interna, la información obtenida es empaquetado en un telegrama serial para ser transmitidos por el puerto serial a 9600 baudios al controlador de la tarjeta electrónica codificadora del telegrama de datos actuales.

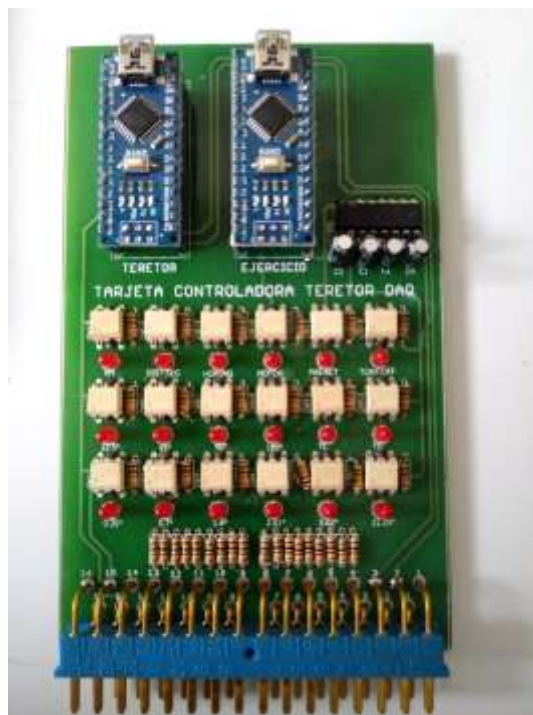


Figura 72: Tarjeta de adquisición de datos digitales.

Fuente: Elaboración propia.

5.1.2 Tarjeta de adquisición de datos de señales analógicas.

La segunda tarjeta electrónica se observa en la figura 72, se denomina teretor DAQ Analógica, recibe las señales eléctricas analógicas de profundidad, balanceo, cabeceo y distancia del contacto con niveles continuos de tensión positiva y negativa, estas señales son acondicionadas con niveles de tensión continuo de cero a 5 voltios para ser procesados con el conversor Análogo a digital (ADC), se utiliza el conversor ADS 1115 por tener una resolución de 16 bits, el conversor ADC utiliza la comunicación serial de tipo síncrona I2C para entregar el valor obtenido al controlador.

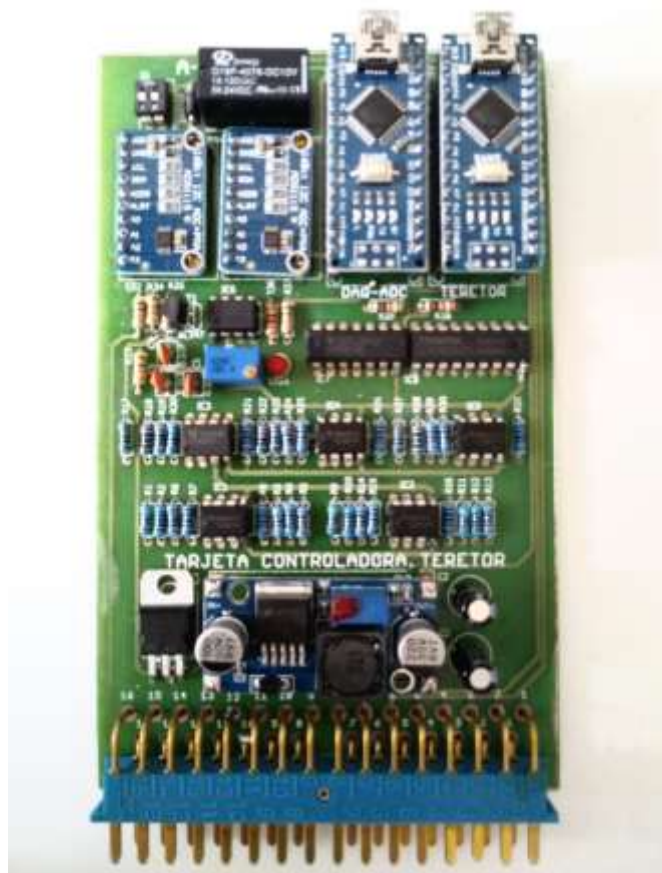


Figura 73: Tarjeta de adquisición de datos analógicos.

Fuente: Elaboración propia.

5.1.3 Tarjeta codificador del telegrama de datos actuales.

La tercera tarjeta electrónica codificador del telegrama de datos actuales recibe las señales digitales y analógicas y los codifica en un telegrama de 8 palabras, cada palabra este compuesto por 16 bits. En la figura 73 se observa la tarjeta electrónicas Codificadora del telegrama de datos actuales de los sistemas de control del torpedo SST-4.



Figura 74: Tarjeta Codificadora del telegrama de datos actuales.

Fuente: Elaboración propia.

5.1.4 Tarjeta decodificador del telegrama de datos actuales.

La última tarjeta electrónica recibe el nombre de tarjeta Decodificadora del telegrama serial proveniente del torpedo SST-4, su función es decodificar el telegrama de datos actuales y transmitir la información obtenida a la interfaz gráfica para su posterior análisis y verificación de las respuestas de los sistemas de control a efectuarse en la computadora del torpedo SST-4. En la figura 74 se observa la tarjeta electrónica decodificadora del telegrama de datos actuales para ser instalado en la plataforma de pruebas Test-Stand.



Figura 75: Tarjeta Decodificadora del telegrama de datos actuales.

Fuente: Elaboración propia.

5.1.5 Interfaz gráfica.

Para el desarrollo de la interfaz gráfica se desarrollaron diversas ventanas con funcionalidades definidas que en conjunto suman al sistema completo. Dichas ventanas serán descritas en el proceso documento y el cómo fueron diseñadas. Sin embargo, el funcionamiento específico de cada una se encuentra detallado en el manual de usuario de la interfaz.

En las pantallas se desarrollaron la pantalla principal de control similar a la que tenía la plataforma original del sistema. Por otro lado, se tiene a la ventana de los indicadores donde se desarrollaron instrumentos de medición e indicadores visuales donde se puede apreciar el estado de las variables del torpedo. A todo ello, se le agregó un nivel de seguridad de acceso por el hecho de que se trata de un activo bastante crítico que sólo puede ser manipulado por personal capacitado, se desarrolló un logo que ahora forma parte de la filosofía del sistema en general y de las pantallas de presentación.



Figura 76: Logo del software del Test-stand SST-4

Fuente: Elaboración propia.

Al ser un programa completo cuenta con diversas ventanas que mantendrán una interacción con el usuario de forma que se adapte a la necesidad o requerimiento inmediato. El siguiente diagrama describe en formal general las ventanas con las que cuenta el software.

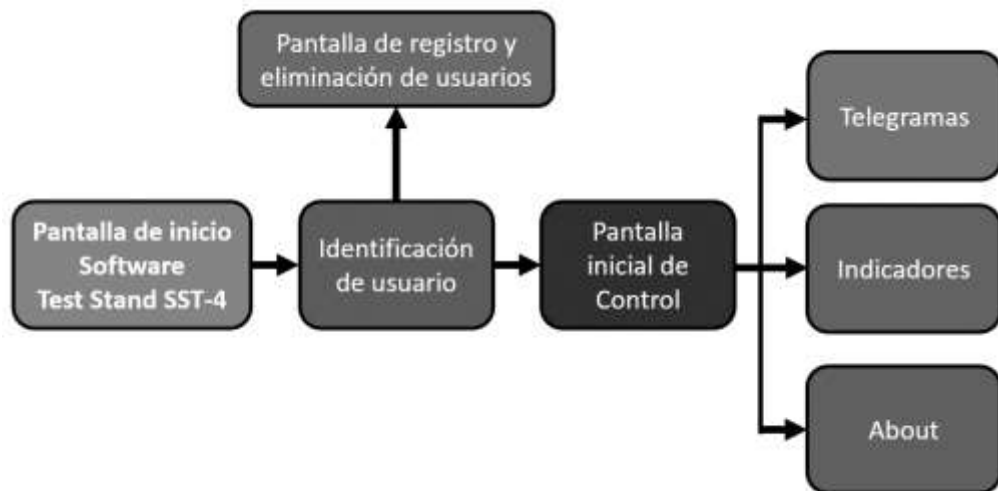


Figura 77: Diagrama general del funcionamiento del software.

Fuente: Elaboración propia.

Sin embargo, en el presente informe primero se describirá el desarrollo de las pantallas principales y, luego las de control de acceso de usuarios y las secundarias.

Pantalla inicial de control

Es desde esta pantalla que se puede interactuar entre una ventana u otra. Así mismo, la información que se envía y se recibe no se pierde por la interacción entre ventanas. Lo cual asegura la integrada y disponibilidad de la información a requerimiento del usuario.

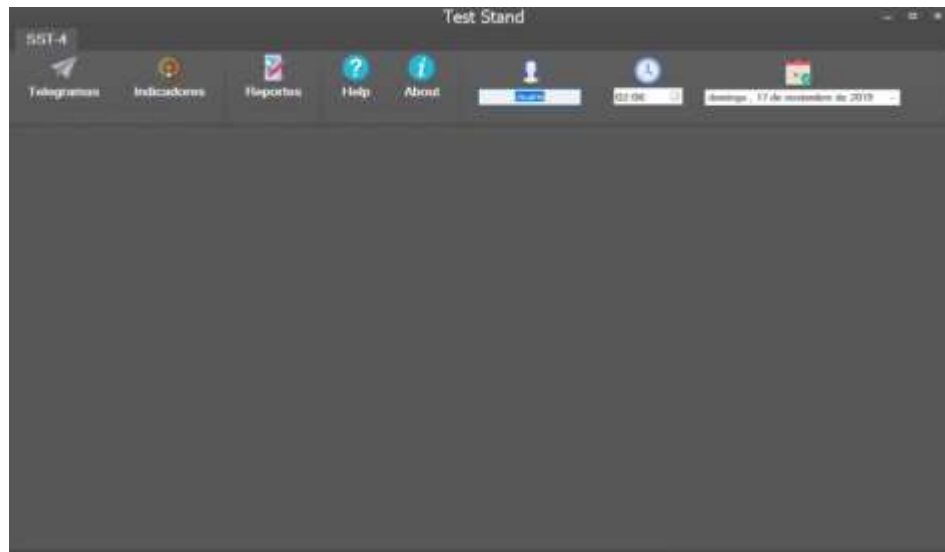


Figura 78: Ventana inicial de control.

Fuente: Elaboración propia.

Mediante los botones se puede navegar en diversas ventanas. Los siguientes botones son los encargados de presentar la interacción de dichas ventanas.



Figura 79: Botones de ventanas de control.

Fuente: Elaboración propia.

Así mismo, se agregó un botón de "Reportes", que será implementado en la siguiente etapa del proyecto. Un botón de "Help" que será implementado, también, en una posible siguiente etapa y un botón "About" que presenta una ventana con la información de los responsables del proyecto en cuestión.

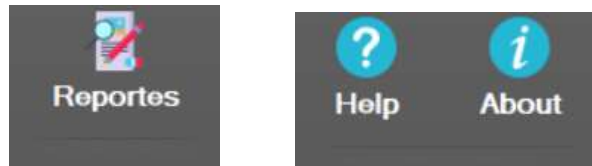


Figura 80: Botones de reporte, ayuda e información.

Fuente: Elaboración propia.

Por otro lado, se tiene en esta misma pantalla la información del usuario registrado que está haciendo uso de la plataforma. Es decir, la interfaz tiene la capacidad de visualizar el usuario en cuestión para un control de accesos y de seguridad. Así mismo, presenta la hora actual y la fecha para efecto de reportes e informes.



Figura 81: Información de usuario y del entorno

Fuente: Elaboración propia.

Por último, se tiene el área donde se interactuará con las ventanas de control.



Figura 82: Panel de sub-Ventanas de control

Fuente: Elaboración propia.

Como resumen de la pantalla en cuestión se presenta la siguiente tabla que reúne la totalidad de objetos utilizados, llámase botones, indicadores visuales, objetos de información, entre otros.

Tabla. Resumen de elementos de la pantalla inicial de control.

Tipo de elemento	Nombre de elemento
Botón de un click	Telegramas
Botón de un click	Indicadores
Botón de un click	Reportes
Botón de un click	Help
Botón de un click	About
Botón de un click	Nombre de usuario
Botón de un click	Hora actual
Botón de un click	Fecha actual
Caja de texto	Minimizar la pantalla
Caja de texto	Restaurar la pantalla
Caja de fecha	Cerrar la aplicación

Fuente: Elaboración propia.

Ventana de Telegramas

Se desarrolló la ventana de telegramas donde se interactúa con el torpedo mediante los telegramas de comando y las señales de activación de relays de control. Así mismo, se recibe información del torpedo para poder ser visualizada en los indicadores digitales de tipo foco que permitan aseverar que la información enviada está siendo recibida y procesada por el CPU del torpedo. Así mismo, dicha ventana es la encargada de llevar el diseño original de la plataforma desarrollada en los años 70. Para visualizar que ello se ha logrado se mostrarán ambas plataformas en la siguiente imagen:



Figura 83: Plataforma original vs interfaz gráfica.

Fuente: Elaboración propia.

Como se puede apreciar se desarrolló la interfaz a imagen de su predecesora. Se mantienen los instrumentos de medición, indicadores visuales, orden de botoneras y llaves de encendido. Algunas partes se han depurado y otras aún no tienen una funcionalidad definida por formar parte de la segunda etapa del proyecto.

A continuación, se presenta la interfaz gráfica en una pantalla más grande.



Figura 84: Ventana de Telegramas. Interfaz Test Stand SST-4

Fuente: Elaboración propia.

Como se puede apreciar cada parte se encuentra segmentada de acuerdo a su función como la presentaba en su plataforma el sistema original. La funcionalidad detalla de cada parte se presenta en el manual de usuario, que se encuentra como anexo del presente informe.

En las siguientes líneas se hará una comparación de resultados entre la plataforma original y las partes del software que representan la modernización de cada una de las etapas para poder presentar una idea clara sobre lo que se ha modernizado y la funcionalidad que ha mantenido.

Instrumentos de medición de los timones

Para la parte de los timones se utilizó instrumentos de medición de tipo aguja que permiten tener una percepción visual correcta con respecto a la señal que representan. El sistema original contaba con instrumentos físicos de medición. Se consideró dejar dichos elementos en la posición en la que se encontraban inicialmente, parte superior de la Ventana de telegramas.

ANTES



AHORA



Figura 85: Modernización de los instrumentos de medición de los timones.

Fuente: Elaboración propia.

En esta primera etapa se tiene sólo los timones sin embargo no tiene una funcionalidad asignada de momento. En siguientes etapas se implementará dicha funcionalidad de la mano de las tarjetas electrónicas correspondientes.

Botones de control

El sistema original contaba con botoneras que con el paso de los años se deterioraron considerablemente por el uso y por las condiciones climatológicas donde el sistema ha sido instalado. Para la implementación de los botones se consideran diversas etapas o grupos dentro de los botones de control. Los primeros indicadores que se encuentran en la parte superior izquierda de la Ventana de telegramas son los siguientes:

- BATT SIMU
- ASS
- EXP
- RUD AN-IN

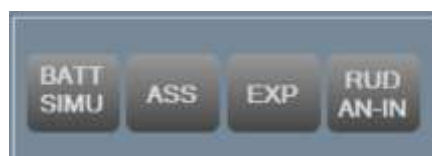
ANTES



AHORA



OFF



ON



Figura 86: Modernización de los botones del sistema. Primer grupo.

Fuente: Elaboración propia.

En esta segunda etapa o grupo de botones se encuentran los relacionados a las siguientes funcionalidades:

- BATT HEAT
- SURFACE
- SIM ON
- SHOT
- COMM
- TIME REF

ANTES



AHORA



OFF

ON



Figura 87: Modernización de los botones del sistema. Segundo grupo.

Fuente: Elaboración propia.

En esta tercera etapa o grupo de botones se encuentran los relacionados a las siguientes funcionalidades:

- I RELAY
- SAFE DIST

ANTES



AHORA



OFF



ON



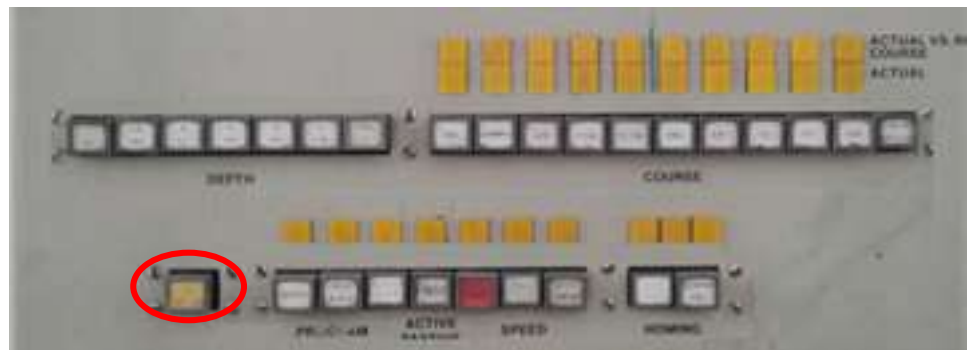
Figura 88: Modernización de los botones del sistema. Tercer grupo.

Fuente: Elaboración propia.

En esta cuarta etapa o grupo de botones se encuentra el relacionado a la siguiente funcionalidad:

- IN ACTION

ANTES



AHORA



OFF



ON



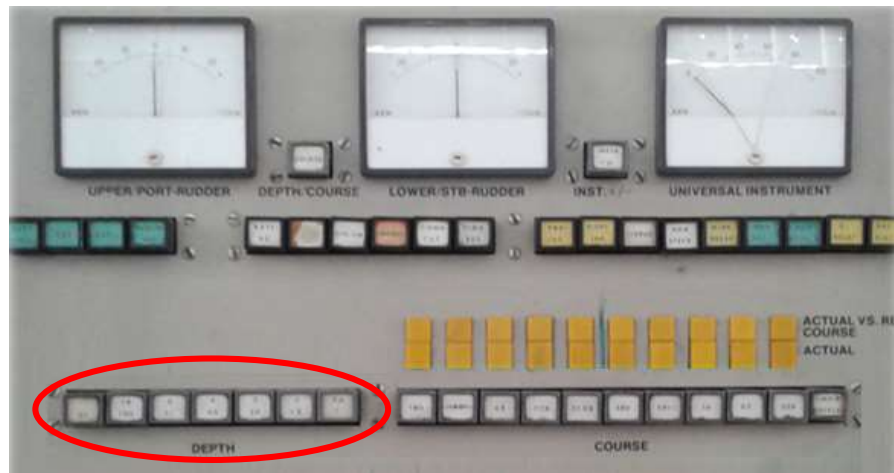
Figura 89: Modernización de los botones del sistema. Cuarto grupo.

Fuente: Elaboración propia.

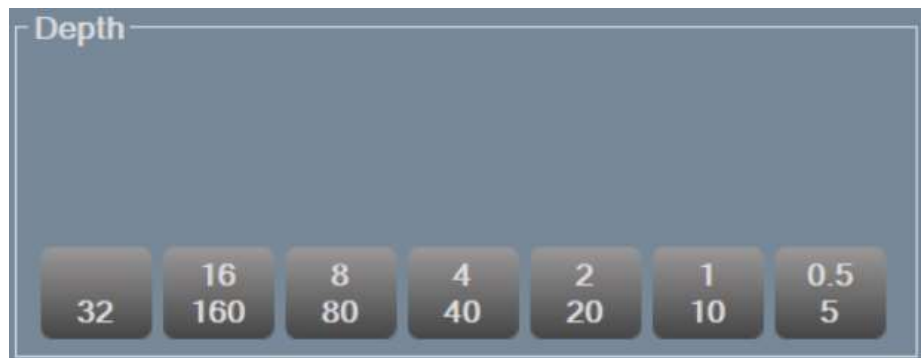
Botones de profundidad

En esta etapa o grupo de botones se encuentran los relacionados a la profundidad que se le asigna al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado al torpedo.

ANTES



AHORA



INDICADOR DIGITAL DEL VALOR ASIGNADO



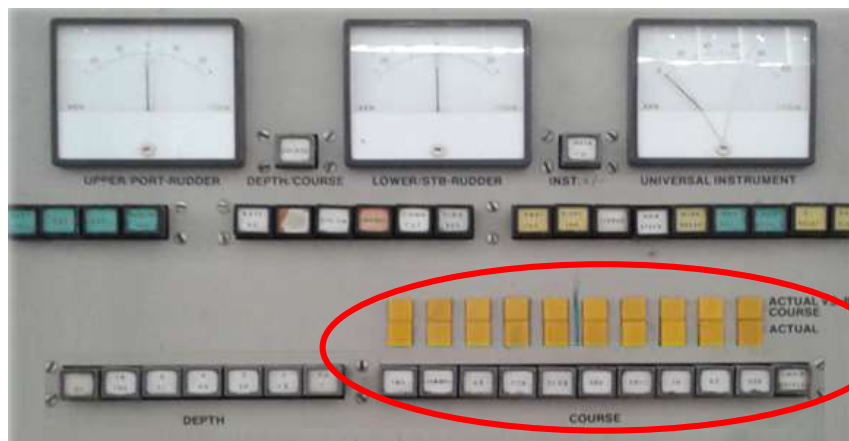
Figura 90: Modernización de los botones de profundidad.

Fuente: Elaboración propia.

Botones e indicadores de rumbo

En esta etapa de botones se encuentran los relacionados al curso que se le asigna al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado al torpedo. Así mismo, se mantuvo el formato de tener los botones en la parte posterior e inmediatamente en la parte superior del grupo los indicadores digitales con respecto a los valores actuales de rumbo, valores enviados por el torpedo para aseverar que la orden ha sido correctamente interpretada.

ANTES



AHORA



INDICADOR DIGITAL DEL VALOR ASIGNADO



Figura 91: Modernización de los botones de rumbo.

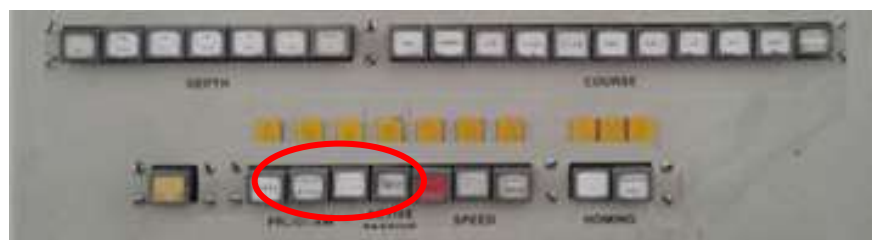
Fuente: Elaboración propia.

Botones de program

En esta etapa de botones se encuentran los relacionados al programa que se le asigna al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado al torpedo en el área de "Summary". Se tiene tres tipos de programas para el torpedo en cuestión.

- SNAKE
- CIRCLE LEFT
- CIRCLE RIGHT

ANTES



AHORA



Figura 92: Modernización de los botones de program.

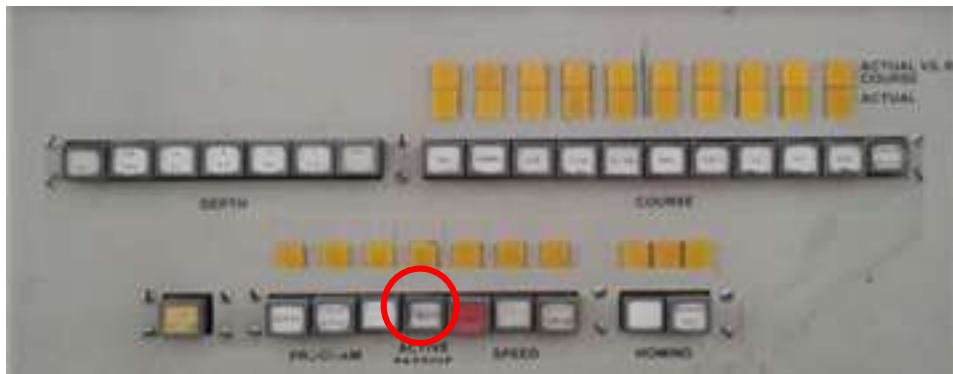
Fuente: Elaboración propia.

Botón de modo acústico

En esta etapa se encuentra el botón que asigna el modo acústico al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado en el área de "Summary". Se tiene dos modos acústicos para el torpedo en cuestión.

- PASSIVE/ACTIVE

ANTES



AHORA



Figura 93: Modernización de los botones del modo acústico.

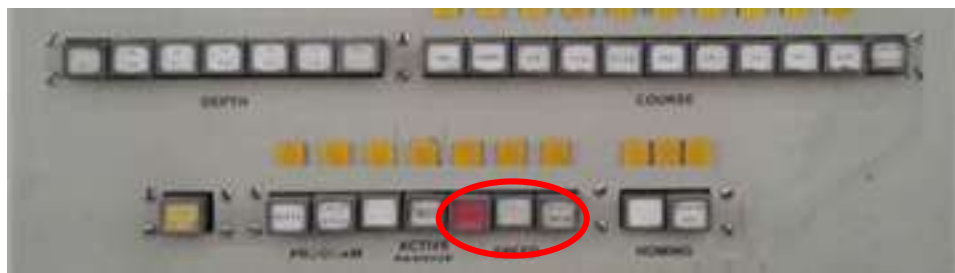
Fuente: Elaboración propia.

Botones de velocidad

En esta etapa de botones se encuentran los relacionados a la velocidad que se le asigna al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado al torpedo en el área de “Summary”. Se tiene tres tipos de programas modos de velocidad para el torpedo en cuestión.

- TO OFF
- LOW SPEED
- HIGH SPEED

ANTES



AHORA



Figura 94: Modernización de los botones de velocidad.

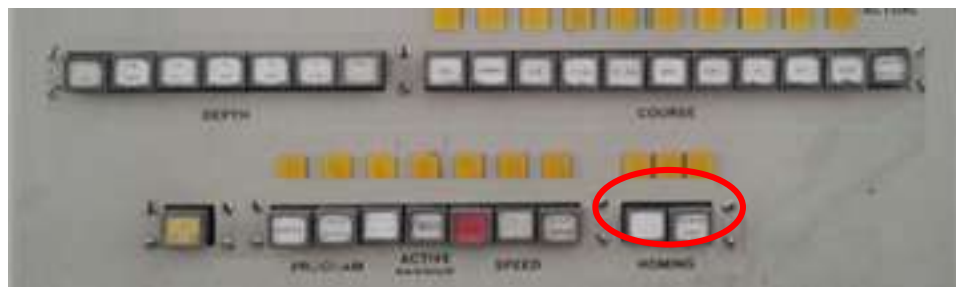
Fuente: Elaboración propia.

Botones de homing

En esta etapa o grupo de botones se encuentran los relacionados al modo de homing que le asigna al torpedo. Se agregó un indicador digital donde se puede apreciar al valor asignado al torpedo en el área de “Summary”. Se tiene dos modos de homing para el torpedo en cuestión.

- HOMING ON
- HOMING OFF

ANTES



AHORA



Figura 95: Modernización de los botones de Homing.

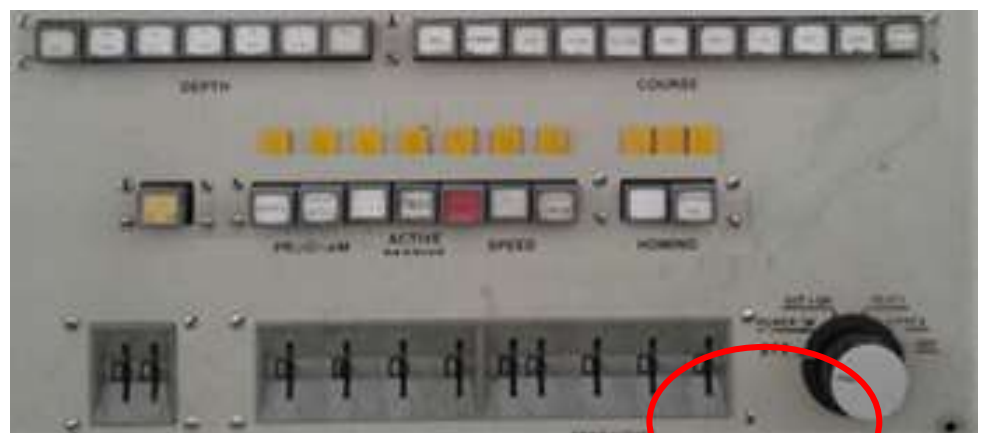
Fuente: Elaboración propia.

Llave de encendido

La llave de encendido emula la llave física original con la cual se tenían 6 estados con los que se manejaba el protocolo de encendido de prueba del torpedo. Los estados son los siguientes:

- OFF
- POWER ON
- SST-4 ON
- TEST I
- TEST II
- READY

ANTES



AHORA



Figura 96: Modernización de la llave de encendido.

Fuente: Elaboración propia.

Summary

Sin bien es cierto, esta etapa no la tenía el tablero de control original, permite tener una visión de todos los datos enviados al torpedo para tener un control de este.

En el área de summary podemos encontrar el estado de las siguientes señales o comandos de control:

- Depth
- Course
- Sim Batt
- Program
- Homing
- Acoustic mode
- Mode
- Speed

Por último, se tiene el modo de comunicación del torpedo. Existen dos modos habilitados:

- Palabra de control y Teretor



Figura 97: Summary

Fuente: Elaboración propia.

Ventana de Indicadores

Con los telegramas actuales implementados en el presente proyecto se puede tener instrumentos e indicadores digitales donde se aprecie el estado actual de las señales y valores del torpedo para poder aseverar que los datos enviados en la ventana de telegramas han sido enviados de manera correcta e interpretados por el torpedo.

En la siguiente figura se observa la interfaz gráfica para visualizar los datos del telegrama de retorno.



Figura 98: Ventana de indicadores

Fuente: Elaboración propia.

Los instrumentos e indicadores reciben la información con respecto al estado actual de las variables del torpedo por medio de comunicación serial a una velocidad de 19200 baudios con telegramas de construcción e interpretación propia por el software y por las etapas de hardware del presente proyecto. La tasa de actualización es de 1 segundo, no hay pérdida de datos por buffer de memoria ni por el tiempo prolongado del ciclo de adquisición.



Figura 99: Instrumentos e indicadores de rumbo, rolo y cabeceo

Fuente: Elaboración propia.



Figura 100: Instrumentos e indicadores de velocidad y profundidad.

Fuente: Elaboración propia.



Figura 101: Instrumentos e indicadores de distancia y correlador.

Fuente: Elaboración propia.



Figura 102: Indicadores digitales.

Fuente: Elaboración propia.

Así mismo, como medida de seguridad con respecto a la correcta recepción de datos se implementó un indicador de tipo led que parpadea cada vez que llega un dato, es decir cada un segundo con respecto a la tasa de recepción.

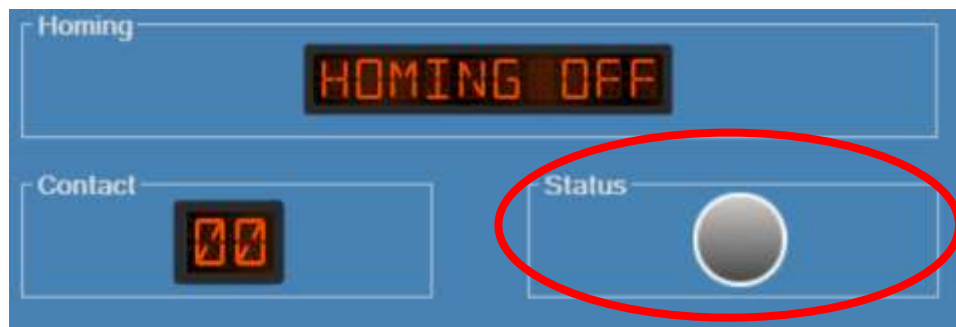


Figura 103: Indicador Status.

Fuente: Elaboración propia.

Como resumen de la pantalla en cuestión se presenta la siguiente tabla que reúne la totalidad de objetos utilizados, llámase botones, indicadores visuales, objetos de información, entre otros.

Control de acceso y registro de usuarios

El software Test Stand SST-4 controla un activo crítico de la Marina de Guerra del Perú. Para que un usuario controle las funcionalidades de un torpedo debe pasar por un proceso bastante extenso de capacitación para lograr copar los conocimientos necesarios y así manejar de manera correcta el torpedo SST-4 en su estación de pruebas. Es por ello, que se ha realizado un control de acceso de usuarios, para que sólo usuarios capacitados y/o autorizados tengan acceso a la plataforma de software.

Estas pantallas se despliegan al inicio del programa y no permiten el acceso hasta colocar las credenciales adecuadas. De igual forma, se tiene una ventana de registro donde se puede agregar, verificar y eliminar usuarios. A esta ventana de registro sólo se accede con una contraseña especial que se le otorgará al encargado o responsable de la línea o personal que considere necesario el jefe de la dependencia. Para ingresar al programa deberá colocarse el nombre de usuario, que viene a ser el CIP, y la contraseña asociada. Una vez se haya ingresado dicha información se deberá hacer click en el botón ingresar. En ese momento se enviará las credenciales al sistema de control para que puedan ser visualizados en la pantalla principal y saber que usuario está registrado.

A continuación, se muestran las pantallas de acceso y registro de usuarios.

La imagen muestra una pantalla de acceso con un fondo negro. En la parte superior central hay un ícono de un usuario dentro de un círculo azul. Debajo del ícono, el texto "TEST STAND SST-4" está subrayado en blanco. En la parte inferior, hay dos campos de entrada de texto blancos: el superior está etiquetado como "Usuario" y el inferior como "Contraseña". Debajo de estos campos hay un botón rectangular azul con el texto "INGRESAR" en blanco y un ícono de una cerradura a la izquierda.

Figura 104: Control de acceso a la interfaz gráfica.

Fuente: Elaboración propia.



Figura 105: Ingreso correcto a la interfaz gráfica.

Fuente: Elaboración propia.

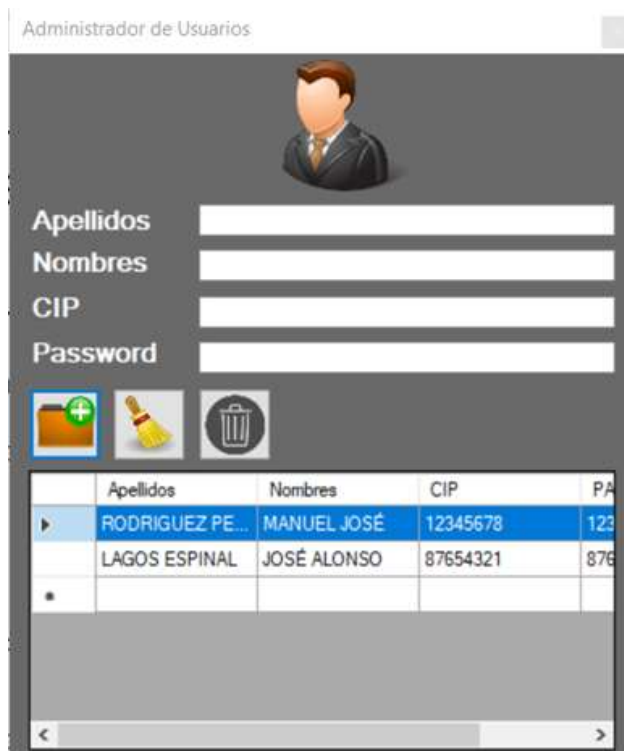


Figura 106: Administrador de usuario de la interfaz gráfica.

Fuente: Elaboración propia.

CAPÍTULO VI

DISCUSIÓN DE RESULTADOS

6.1 Contratación de hipótesis con los resultados.

La tarjeta electrónica de adquisición de datos permitió la lectura de las señales digitales y analógicas de los sensores de los sistemas de control de estabilización y búsqueda en la computadora del torpedo SST-4.

En la figura 106 se observa el procesamiento de lectura de las señales digitales y analógicas que son codificados en un telegrama serial y transmitidos hacia la tarjeta codificadora; este decodifica el telegrama de datos actuales para ser visualizados en la interfaz gráfica y validar las ordenes de los sistemas de control del torpedo SST-4.



Figura 107: Telegrama de datos Actuales instalado.

Fuente: Elaboración propia.

Para verificar si la palabra 8 del telegrama de datos actuales corresponde al tiempo de sincronización; se utilizó un osciloscopio digital para verificar el pulso de la señal de sincronismo; el resultado obtenido se muestra en la figura 107, donde se puede apreciar que la palabra de sincronismo (color amarillo) tiene un tiempo de 68 milisegundos; el tiempo se encuentra en el rango de sincronismo; se observa el tiempo de solicitud de datos digitales y analógicos (color verde); el tiempo aproximado es de 24 milisegundos.



Figura 108: Señal de sincronismo del telegrama de datos actuales.

Fuente: Elaboración propia.

En la figura anterior se puede determinar la correcta sincronización del telegrama de datos actuales que corresponde a 14 bits; así mismo dentro de esta señal el controlador del codificador de telegrama se sincroniza con las tarjetas de adquisición de las señales digitales y analógicas de los sistemas de control para empaquetar y preparar el telegrama de datos actuales.

En el lado de la plataforma de pruebas Test-Stand se recibió correctamente el telegrama de datos actuales del torpedo SST-4, dicha señal fue decodificada y enviada al a la interfaz gráfica; en la figura 108 se observa la instalación de la tarjeta decodificadora del telegrama de datos actuales.



Figura 109: Tarjeta Decodificadora instalado en el Test-stand.

Fuente: Elaboración propia.

En el lado de la interfaz gráfica se logró validar la información de los sensores de los sistemas de control de la computadora del torpedo SST-4; así como modificar las ordenes de los sistemas de control; en la figura 109 se observa la computadora de la interfaz gráfica instalado en el sistema de pruebas Test-Stand.



Figura 110: Pruebas del torpedo SST-4 en el Test-Stand.

Fuente: Elaboración propia.

CAPÍTULO VII

CONCLUSIONES

La implementación del software y hardware en el telegrama de datos actuales se caracterizó porque las pruebas realizadas comprobaron el funcionamiento del sistema de comunicación, el cual garantiza el cumplimiento de los requerimientos establecidos en la implementación el Telegrama de datos actuales con interfaz gráfica para el torpedo SST-4, este proyecto permite a las unidades submarinas mejorar el control y guiado del torpedo, además permite efectuar las pruebas generales a los torpedos SST-4.

Se desarrolló dos tarjetas electrónicas para adquirir las señales digitales y analógicas de los sensores y unidades electrónicas del torpedo SST-4, el diseño de separar las señales digitales y analógicas responde a una estructura de diseño modular, cada tarjeta captura las señales, almacena la información y transmite el resultado obtenido a la tarjeta de control cada vez que es requerido mediante la comunicación serial.

Se desarrollo una tarjeta electrónica que se comunica con las tarjetas de adquisición de señales digitales y analógicas mediante comunicación serial, esta tarjeta tiene la capacidad de recibir la información digital y analógica para ser codificados en un protocolo de 8 palabras compuesto por 16 bits.

Se desarrolló una tarjeta PCB con los Estándares y medidas adecuados para ser instalado en la unidad SPU-B de la consola de pruebas Test – Stand, el hardware reúne todos los componentes necesarios para comunicarse con la interfaz gráfica y a la vez decodificar el telegrama de datos actuales de torpedo SST-4.

La implementación de la interfaz gráfica (GUI), en la plataforma de Visual Studio en lenguaje de programación C#, incidió positivamente en la modernización de la plataforma de pruebas (Test-Stand), mediante esta interfaz el operador visualizar la información de los torpedos SST-4, utilizando una interfaz gráfica familiar y mucho más potente que la plataforma de pruebas original.

CAPÍTULO VIII

RECOMENDACIONES

Actualmente el proyecto ha sido instalado en un solo torpedo SST-4, los componentes tecnológicos utilizados en este proyecto son de tecnología actual y se pueden encontrar en el mercado local, se recomienda implementar el prototipo desarrollado en los torpedos SST-4 que no cuentan con un sistema de telegrama de datos actuales.

Toda evolución tecnológica, por lo frecuente, conlleva a los usuarios adaptarse a los nuevos sistemas dentro de las instituciones, ya que la modernización conlleva a la automatización y a disminuir el tiempo de trabajo. Es por ello que se propone a la estación de armas submarinas capacitar a todo el personal en el uso de esta nueva herramienta tecnológica.

El sistema de comunicación del telegrama de datos actuales es una parte de las pruebas generales que se realizan en la plataforma Test-Stand de la estación de armas submarinas, se recomienda a la estación de armas submarinas continuar con la investigación e integrar el proyecto desarrollado con la finalidad de reemplazar la plataforma de pruebas en un sistema completamente moderno.

CAPÍTULO IX

REFERENCIAS BIBLIOGRÁFICAS

AEG TELEFUNKEN. "Manual Cabeza Acústica". 1392263-10. Alemania, 1970, pp. 5-10.

AEG TELEFUNKEN. "Manual Sección Electrónica". 1392263-12. Alemania, 1970, pp. 10-20.

AEG TELEFUNKEN. "Manual Test-Stand". 1392263-12. Alemania, 1970, pp. 45.

De la Horra, K. "Sistemas de adquisición de datos basados en la plataforma Arduino: aplicaciones a Matlab, Simulink y Android". Universidad Calos III de Madrid, España: (tesis de pregrado), 2013.

Núñez J. "Diseño e integración de un sistema de adquisición de datos mediante el uso de Arduino y Raspberry-pi". Universidad Nacional autónoma de México. México: (tesis de pregrado). 2014.

Sharp, J. "Microsoft Visual C# 2013 - Step by Step". Estados Unidos: 2015.

Espinoza, O. "Sistema de combate "KALPA". (Proyecto de Investigación y tecnología para la Marina de Guerra del Perú). Perú: 2015.

AEG TELEFUNKEN. "Descripción funcional". 1392263-01. Alemania, 1970, pp. 2.

AEG TELEFUNKEN. "Descripción funcional". 1392263-01. Alemania, 1970, pp. 4.

AEG TELEFUNKEN. “Descripción funcional”. 1392263-01. Alemania, 1970, pp. 6.

AEG TELEFUNKEN. “Descripción funcional”. 1392263-01. Alemania, 1970, pp. 7.

AEG TELEFUNKEN. “Descripción funcional”. 1392263-01. Alemania, 1970, pp. 10.

AEG TELEFUNKEN. “Sistema de comunicación”. 1392263-13. Alemania, 1970, pp. 25.

AEG TELEFUNKEN. “Manual de funcionamiento NVS-A”. 1392263-18. Alemania, 1970, pp. 3-9.

AEG TELEFUNKEN. “Manual de funcionamiento SPU-A”. 1392263-19. Alemania, 1970, pp. 5-62.

Arduino. “Introducción a la plataforma Arduino”. [En línea] 22 de Abril del 2017. Recuperado de: <https://arduino.cc/en/Guide/Introduction>.

Hobbytronics. “Microcontrolador ATMEGA 328”. [En línea] 23 de Abril del 2017. Recuperado de: <https://hobbytronics.co.uk/manufacturer-atmel>

Todo Tecnología. “Modelos de placa Arduino”. [En línea] 26 de Abril del 2017. Recuperado de: <https://Todotecnologia-eso.blogspot.com.es>

Arduino. “Tarjeta Arduino Nano”. [En línea] 20 de Junio del 2018. Recuperado de: <https://arduino.cc/en/Main/ArduinoNano>

Pertence, A. "Amplificadores operacionales y filtros activos" Editorial McGraw-Hill, Madrid, España: 2008.

Tools. "Amplificador" [En línea] 8 de Setiembre del 2018. Recuperado de: <http://electrontools.com/Home/WP/2017/04/04/amplificador-operacional-como-comparador/>

Pertence, A. "Amplificadores Operacionales y Filtros Activos" Editorial McGraw-Hill, Madrid, España: 2008.

Icardab. "Opams" [En línea] 22 de Agosto del 2018. Recuperado de: <http://www.lcardaba.com/articles/opamps.html>

Hetpro. "LM2596S" [En línea] 22 de Agosto del 2018. Recuperado de: <http://hetpro-store.com/regulador-de-voltaje-step-down-lm2596s/>

HENRY'S BENCH. "ADS1115". [En línea] 7 de Setiembre del 2018. Recuperado en: <http://henrysbench.capnfatz.com/henrys-bench/arduino-voltage-measurements/arduino-ads1115-module-getting-started-tutorial/>

News sof. "Visual Studio 2012". [En línea] 22 de Agosto del 2018. Recuperado de: <https://news.softpedia.com/news/Visual-Studio-2010-RTM-139703.shtml>

Microsoft. "Lenguaje de programación Visual Studio C#". [En línea] 4 de Noviembre del 2018. Recuperado de: [https://msdn.microsoft.com/es-pe/library/aa287558\(v=vs.71\).aspx](https://msdn.microsoft.com/es-pe/library/aa287558(v=vs.71).aspx)

Electronica-electronics. "Eagle cadsoft". [En línea] 2 de Abril del 2018. Recuperado de: <http://electronica-electronics.com/software/eagle.html>

Atlas Elektronik. “Manual SUT-264”. Alemania, 2010, pp. 3-55.

Tiobe. “Top 5 de lenguajes de programación”. [En línea] 12 de Abril del 2018. Recuperado de: <https://www.tiobe.com/tiobe-index/>

Statista. “Cuota del mercado mundial de S.O.”. [En línea] 18 de Abril del 2018. Recuperado de: <https://es.statista.com/estadisticas/576870/cuota-de-mercado-mundial-de-los-sistemas-operativos/>

ANEXOS

Anexo 1: Programa de la tarjeta DAQ digital

//Entradas digitales

```
int pin_rumb0 = 13; //Bit de Rumbo=0.35°
int pin_rumb1 = 16; //Bit de Rumbo=0.7°
int pin_rumb2 = 19; //Bit de Rumbo=1.4°
int pin_rumb3 = 10; //Bit de Rumbo=2.8°
int pin_rumb4 = 2; //Bit de Rumbo=5.625°
int pin_rumb5 = 5; //Bit de Rumbo=11.25°
int pin_rumb6 = 14; //Bit de Rumbo=22.5°
int pin_rumb7 = 17; //Bit de Rumbo=45°
int pin_rumb8 = 12; //Bit de Rumbo=90°
int pin_rumb9 = 9; //Bit de Rumbo=180°
int pin_tc = 3; //Contacto de blanco TC=1
int pin_td = 6; //Deteccion de blanco TD=1
int pin_hh = 15; //Perdida de blanco HH=1
int pin_dist = 18; //Distancia cubierta Si=0, No=1
int pin_homing = 11; //Homing OFF=1; ON=0
int pin_motor = 8; //Motor On=1; Off=0
int pin_acust = 4; //Modo acustico Pasivo=1; Activo=0
int pin_to_off = 7; //Torpedo Off = 1
```

//Variables de lectura y transmision

```
int Rumbo = 0;
int Deteccion;
int Homing;
int Dist_Cub;
int Modo_Acust;
int Motor_On;
int To_Off;
```

```

int TC= 0;
int TD = 0;
int HH = 0;
int estado;
byte data_update_event=1;//New data flag

void setup()
{
  Serial.begin(19200);
  for(int x=2;x<20;x++){ pinMode(x,INPUT);}
}

void loop()
{
  while(Serial.available() > 0)
  {
    TX_Digital();
  }
  Lectura_Digital();
}

void Lectura_Digital()
{
  Leer_Rumbo();
  Leer_Deteccion();
  Leer_Dist_Cub();
  Leer_Homing();
  Leer_Motor_ON();
  Leer_Modo_Acust();
  Leer_TO_Off();
}

```

```

void Leer_Rumbo()
{
Rumbo=0;
estado=digitalRead(pin_rumb0); //Rumbo 0.35°
if(estado == HIGH){Rumbo=Rumbo+1;}
estado=digitalRead(pin_rumb1); //Rumbo 0.7°
if(estado == HIGH){Rumbo=Rumbo+2;}
estado=digitalRead(pin_rumb2); //Rumbo 1.4°
if(estado == HIGH){Rumbo=Rumbo+4;}
estado=digitalRead(pin_rumb3); //Rumbo 2.8°
if(estado == HIGH){Rumbo=Rumbo+8;}
estado=digitalRead(pin_rumb4); //Rumbo 5.625°
if(estado == HIGH){Rumbo=Rumbo+16;}
estado=digitalRead(pin_rumb5); //Rumbo 11.25°
if(estado == HIGH){Rumbo=Rumbo+32;}
estado=digitalRead(pin_rumb6); //Rumbo 22.5°
if(estado == HIGH){Rumbo=Rumbo+64;}
estado=digitalRead(pin_rumb7); //Rumbo 45°
if(estado == HIGH){Rumbo=Rumbo+128;}
estado=digitalRead(pin_rumb8); //Rumbo 90°
if(estado == HIGH){Rumbo=Rumbo+256;}
estado=digitalRead(pin_rumb9); //Rumbo 180°
if(estado == HIGH){Rumbo=Rumbo+512;}
}

```

```

void Leer_Deteccion()
{
Deteccion=0;
estado=digitalRead(pin_tc);//TC
if(estado == HIGH){Deteccion=Deteccion+1;}else{Deteccion=Deteccion;}
estado=digitalRead(pin_td);//TD
}

```



```

if(estado == HIGH){Deteccion=Deteccion+2;}else{Deteccion=Deteccion;}
estado=digitalRead(pin_hh);//HH
if(estado == HIGH){Deteccion=Deteccion+4;}else{Deteccion=Deteccion;}
}

```

```

void Leer_Dist_Cub()
{ Dist_Cub=digitalRead(pin_dist); } // CUBIERTA=0; NO CUBIERTA =1
}

```

```

void Leer_Homing()
{ Homing=digitalRead(pin_homing);} // Homing ON=0 ---> Homing OFF=1

```

```

void Leer_Motor_ON()
{ Motor_On=digitalRead(pin_motor);}

```

```

void Leer_Modo_Acust()
{ Modo_Acust=digitalRead(pin_acust);}
}

```

```

void Leer_TO_Off()
{ To_Off = digitalRead(pin_to_off);}

```

```

void TX_Digital()
{
  Serial.print("$DIG:");
  Serial.print("A");Serial.print(Rumbo);Serial.print(",");
  Serial.print("B");Serial.print(Homing);Serial.print(",");
  Serial.print("C");Serial.print(Dist_Cub);Serial.print(",");
  Serial.print("D");Serial.print(Deteccion);Serial.print(",");
  Serial.print("E");Serial.print(Modo_Acust);Serial.print(",");
  Serial.print("F");Serial.print(Motor_On);Serial.print("\n");
}

```

Anexo 2: Programa de la tarjeta DAQ Analógico

```
#include <Adafruit_ADS1015.h>
Adafruit_ADS1115 ads1(0x48);
Adafruit_ADS1115 ads2(0x49);
const float multiplier = 0.1875F;

//Variables de lectura
float Cabeceo = 0.00;
float Rolido = 0.00;
float Profundidad = 0.00;
float Correlador= 0.00;
float Guiado_hor = 0.00;
float Velocidad= 0.00;
float rpm;
float nudos;

//Variables de voltajes
float volt_cab = 0.00;
float volt_rol = 0.00;
float volt_prof = 0.00;
float volt_corr = 0.00;
float volt_guiad = 0.00;

//Variables auxiliares
int cont=0;
int16_t adc;
int16_t volt;
float Vin=0.00;
int led=12;
byte data_update_event=1;//New data flag
```

```

void setup()
{
  pinMode(led,OUTPUT);
  Serial.begin(19200);
  ads1.begin();
  ads2.begin();
}

void loop()
{
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Cabeceo();
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Rolido();
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Profundidad();
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Correlador();
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Guiado_Hor();
  decode(); if((data_update_event&0x01)==0x01)
{ TX_Analog();data_update_event=0;}
  Leer_Velocidad();
}

```

```

void Leer_Profundidad()
{
  adc=ads1.readADC_SingleEnded(0);
  Vin = (adc * 0.1875)/1000;
  volt_prof= ((12.00/5.00)*Vin);
  Profundidad = 63.3333 * volt_prof;
}

```

```

void Leer_Correlador()
{
  adc=ads1.readADC_SingleEnded(1);
  Vin = (adc * 0.1875)/1000;
  Correlador = ((12.00/5.00)*Vin);
}

```

```

void Leer_Cabeceo()
{
  adc=ads1.readADC_SingleEnded(2);
  Vin = (adc * 0.1875)/1000;
  volt_cab = ((24.00/5.00)*Vin)-12.00;
  Cabeceo=8.787346*volt_cab;
  //Serial.print(volt_cab);Serial.print(",");Serial.println(Cabeceo);
}

```

```

void Leer_Guiado_Hor()
{
  adc=ads2.readADC_SingleEnded(0);
  Vin = (adc * 0.1875)/1000;
  Guiado_hor = ((12.00/5.00)*Vin);
}

```

```

void Leer_Rolido()
{
  adc=ads2.readADC_SingleEnded(1);
  Vin = (adc * 0.1875)/1000;
  volt_rol = ((24.00/5.00)*Vin)-12.00;
  Rolido = volt_rol;
}

void Leer_Velocidad()
{
  Velocidad=23.50;
}

void TX_Analog()
{
  Serial.print("$ANA:");
  Serial.print("A");Serial.print(Cabeceo);Serial.print(",");
  Serial.print("B");Serial.print(Rolido);Serial.print(",");
  Serial.print("C");Serial.print(Profundidad);Serial.print(",");
  Serial.print("D");Serial.print(Correlador);Serial.print(",");
  Serial.print("E");Serial.print(Guiado_hor);Serial.print(",");
  Serial.print("F");Serial.print(Velocidad);Serial.print("\n");
  digitalWrite(led,0);
}

```

Anexo 3: Programa Codificador de Telegrama Actual

```
#include <TimerOne.h>
//#include <TimerTrhee.h>
int Palabra_1[16]={0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1};//Cabeceo y roldo
int Cab[10];
int Rol[10];
int Palabra1_P1;
int Palabra1_P2;
int Palabra1_P3;
int Palabra_2[16]={0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1};
int Palabra_3[16];//Velocidad y homing
int Veloc[10];
int Palabra3_P1;
int Palabra3_P2;
int Palabra3_P3;
int Palabra_4[16];//Distancia mts, dist-seg, deteccion d-e blanco
int Distancia[10];
int Palabra4_P1;
int Palabra4_P2;
int Palabra4_P3;
int Palabra_5[16];//Correlador, pasivo/activo
int Corr[10];
int Palabra5_P1;
int Palabra5_P2;
int Palabra5_P3;
int Palabra_6[16];//profundidad, motor
int Prof[10];
int Palabra6_P1;
int Palabra6_P2;
int Palabra6_P3;
```

```

int Palabra_7[16];//rumbo
int Rumb[10];
int Palabra7_P1;
int Palabra7_P2;
int Palabra7_P3;
int Telegrama[112];

// Variables de lectura digital
int estado=0;
int rumbo=180;
int deteccion=1;
int dista_seg=0;
int homing=0;
int motor=0;
int modo_acus=1;
int to_off=0;

// Variables de lectura analogo
float cabeceo = 12.00;
float rovido = 5.00;
float distancia = 3.00;
float profundidad = 12.00;
float correlador = 4.00;
float guiado_hor = 5.00;
float velocidad = 23.00;

// Variables de monitoreo
int led_dig = 9;
int led_ana = 11;
int led_tel = 13;
int led_mon = 53;

```

```

int temp=0;
int signo=0;
int valor;
int flag=1;
int x, Y;
int Bit = 2;
int Baudios_206 = 4854;
byte data_update_event1=1;//New data flag
byte data_update_event2=1;//New data flag

void setup()
{
  pinMode(Bit, OUTPUT);
  pinMode(led_dig,OUTPUT);
  pinMode(led_ana, OUTPUT);
  pinMode(led_tel, OUTPUT);
  pinMode(led_mon, OUTPUT);
  Serial.begin(19200);
  Serial1.begin(19200);
  Serial2.begin(19200);
  flag=1; digitalWrite(Bit,1);
  cli();
  TCCR1A = 0;// set entire TCCR1A register to 0
  TCCR1B = 0;// same for TCCR1B
  TCNT1 = 0;//initialize counter value to 0
  OCR1A = 9722;// = (16*10^6) / (13.7334*1024) - 1 (must be <65536)
  TCCR1B |= (1 << WGM12);
  TCCR1B |= (1 << CS12) | (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);
  sei();
}

```



```

ISR(TIMER1_COMPA_vect)
{
  cli(); Telegrama_206_Baudios();
  digitalWrite(Bit,HIGH); flag=1; sei();
}

void loop()
{
  if(flag==1)
  {
    digitalWrite(led_mon,1); Serial_Digital(); Serial_Analogo();
    Decode_Telegrama(); Decode_Binario(); digitalWrite(led_mon,0);
    flag=0;
  }
}

void Serial_Digital(void)
{
  Serial2.print("#DIG"); Serial2.print("\n");
  temp=0; data_update_event2=0;
  for(temp=0; temp<20; temp++)
  {
    decode_Digital();
    if((data_update_event2&0x01)==0x01)
    { digitalWrite(led_dig,1);
      data_update_event2=0; temp=100; }
    delay(1);
  }
  digitalWrite(led_dig,0);
}

```

```

void Serial_Analogo(void)
{
  data_update_event1=0; Serial1.print("#ANA"); Serial1.print("\n");
  for(temp=0; temp<20; temp++)
  {
    decode_Analogo();
    if((data_update_event1&0x01)==0x01)
    {
      digitalWrite(led_ana,1);delay(2);
      data_update_event1=0; temp=100;
    }
    delay(1);
  }
  digitalWrite(led_ana,0);
}

```

```

void Decode_Binario()
{
  y=0; for(x=0; x<=15; x++) {Telegrama[x]=Palabra_1[y]; y++;}
  y=0; for(x=16; x<=31; x++) {Telegrama[x]=Palabra_2[y]; y++;}
  y=0; for(x=32; x<=47; x++) {Telegrama[x]=Palabra_3[y]; y++;}
  y=0; for(x=48; x<=63; x++) {Telegrama[x]=Palabra_4[y]; y++;}
  y=0; for(x=64; x<=79; x++) {Telegrama[x]=Palabra_5[y]; y++;}
  y=0; for(x=80; x<=95; x++) {Telegrama[x]=Palabra_6[y]; y++;}
  y=0; for(x=96; x<=111; x++){Telegrama[x]=Palabra_7[y]; y++;}
}

```

```

void Telegrama_206_Baudios()
{
  digitalWrite(led_tel,1);//Indicador encendido de salida de telegrama
  digitalWrite(Bit,LOW);delayMicroseconds(Baudios_206);
}

```

```

for(x=0; x<=111; x++)
{
digitalWrite(Bit,Telegrama[x]);delayMicroseconds(Baudios_206);}
digitalWrite(Bit,LOW);delayMicroseconds(Baudios_206);
digitalWrite(led_tel,0);
}

void decode_Digital(void)
{
    static int counter=0;
    while(Serial2.available() > 0)
    {
        digitalWrite(led_dig,1);
        buffer[counter]=Serial2.read();
        Serial.print(buffer[counter]);
        if(buffer[counter]=='\n')
        {
if (strcmp (buffer,"$DIG:",5) == 0)
        {
            rumbo = Search_Int(buffer,"A");
            homing = Search_Int(buffer,"B");
            dista_seg = Search_Int(buffer,"C");
            deteccion = Search_Int(buffer,"D");
            modo_acus = Search_Int(buffer,"E");
            motor = Search_Int(buffer,"F");
            data_update_event2|=0x01;
        }
        for(int a=0; a<=counter; a++){buffer[a]=0;}counter=0;
        }
        else{counter++;} }
}

```

```

void decode_Analogo(void)
{
    static int counter=0;
    while(Serial1.available() > 0)
    {
        buffer[counter]=Serial1.read();
        if(buffer[counter]=='\n')
        {
            if (strncmp (buffer,"$ANA:",5) == 0)
            {
                cabeceo = Search_Float(buffer,"A");
                roldo = Search_Float(buffer,"B");
                profundidad = Search_Float(buffer,"C");
                correlador = Search_Float(buffer,"D");
                distancia = Search_Float(buffer,"E");
                velocidad = Search_Float(buffer,"F");
                data_update_event1|=0x01;
            }
            for(int a=0; a<=counter; a++){buffer[a]=0;}
            counter=0; //Resetear el counter.
        }
        else{counter++;}
    }
}

```

```

void Decode_Telegrama()
{
    Palab_1(); Palab_2(); Palab_3(); Palab_4();
    Palab_5(); Palab_6(); Palab_7();
}

```

```

void Palab_1()
{
    valor=cabeceo; valor=rolido;
    if(cabeceo>=0)
    {
        for(x=0; x<10; x++) {Cab[x]=bitRead(valor, x);}Cab[6]=0;}
    else{valor=valor*-1;signo=1;
        for(x=0; x<10; x++) {Cab[x]=bitRead(valor, x);}Cab[6]=1;}
    if(rolido>=0)
    {for(x=0; x<10; x++) {Rol[x]=bitRead(valor, x);}Rol[4]=0;}
    else{valor=valor*-1;signo=1;
        for(x=0; x<10; x++) {Rol[x]=bitRead(valor, x);}Rol[4]=1;}
    y=0; for(x=6; x>=0; x--) {Palabra_1[x]=Cab[y];y++;}
    y=0; for(x=12; x>=8; x--) {Palabra_1[x]=Rol[y];y++;}
    Palabra1_P1=0; Palabra1_P2=0;
    for(x=0; x<=6; x++) {Palabra1_P1=Palabra_1[x]+(Palabra1_P1*1);}
    switch (Palabra1_P1)
    {
        case 0: case 2: case 4: case 6: Palabra_1[7]=1; break;
        case 1: case 3: case 5: case 7: Palabra_1[7]=0; break;}
    for(x=0; x<=12; x++) {Palabra1_P2=Palabra_1[x]+(Palabra1_P2*1);}
    switch (Palabra1_P2)
    {
        case 0: case 3: case 6: case 9: case 12:
        Palabra_1[13]=0; Palabra_1[14]=0; break;
        case 1: case 4: case 7: case 10: case 13:
        Palabra_1[13]=0; Palabra_1[14]=1; break;
        case 2: case 5: case 8: case 11: case 14:
        Palabra_1[13]=1; Palabra_1[14]=0; break;
    }
    Palabra1_P3=0; //Paridad 3=0; bit 8 al bit 14

```

```

for(x=8; x<=14; x++) {Palabra1_P3=Palabra_1[x]+(Palabra1_P3*1);}
switch (Palabra1_P3)
{case 0: case 2: case 4: case 6: Palabra_1[15]=1; break;
case 1: case 3: case 5: case 7: Palabra_1[15]=0; break;}
}

void Palab_3()
{
valor=velocidad;
for(x=0; x<10; x++) {Veloc[x]=bitRead(valor, x);}
for(x=6; x>=0; x--) {Palabra_3[x]=Veloc[y];y++;}
Palabra_3[8] = Veloc[1]; Palabra_3[9] = Veloc[0];
Palabra_3[10] = 0; Palabra_3[11] = 0; //dato vacio
estado=digitalRead(18);
if (estado==HIGH) {Palabra_3[12]=1;} else {Palabra_3[12]=0;}
Palabra3_P1=0; //Paridad 1=0; bit 0 al bit 6
for(x=0; x<=6; x++) {
Palabra3_P1=Palabra_3[x]+(Palabra3_P1*1);}
switch (Palabra3_P1)
{case 0: case 2: case 4: case 6: Palabra_3[7]=1; break;
case 1: case 3: case 5: case 7: Palabra_3[7]=0; break;}
Palabra3_P2=0; //Paridad 2=0; bit 0 al bit 12
for(x=0; x<=12; x++) {Palabra3_P2=Palabra_3[x]+(Palabra3_P2*1);}
switch (Palabra3_P2)
{case 0: case 3: case 6: case 9: case 12: Palabra_3[13]=0;
Palabra_3[14]=0; break;
case 1: case 4: case 7: case 10: case 13: Palabra_3[13]=0;
Palabra_3[14]=1; break;
case 2: case 5: case 8: case 11: case 14: Palabra_3[13]=1;
Palabra_3[14]=0; break;}
Palabra3_P3=0;

```

```

for(x=8; x<=14; x++) {Palabra3_P3=Palabra_3[x]+(Palabra3_P3*1);}
switch (Palabra3_P3)
{case 0: case 2: case 4: case 6: Palabra_3[15]=1; break;
case 1: case 3: case 5: case 7: Palabra_3[15]=0; break;}
}

void Palab_4()
{
valor=distancia;
for(x=0; x<10; x++) {Distancia[x]=bitRead(valor, x);}
for(x=6; x>=0; x--) {Palabra_4[x]=Distancia[y];y++;}
Palabra_4[9] = 0; //dato vacio
if (dista_seg==1) {Palabra_4[8]=1;}else {Palabra_4[8]=0;}
switch(deteccion)
{
case 0: Palabra_4[10]=1;Palabra_4[11]=1;Palabra_4[12]=1;break;
case 1: Palabra_4[10]=0;Palabra_4[11]=1;Palabra_4[12]=1;break;//tc
case 2: Palabra_4[10]=0;Palabra_4[11]=0;Palabra_4[12]=1;break;//td
case 3: Palabra_4[10]=1;Palabra_4[11]=1;Palabra_4[12]=0;break;//hh
default: break;
}
Palabra4_P1=0; //Paridad 1=0; bit 0 al bit 6
for(x=0; x<=6; x++) {Palabra4_P1=Palabra_4[x]+(Palabra4_P1*1);}
switch (Palabra4_P1)
{case 0: case 2: case 4: case 6: Palabra_4[7]=1; break;
case 1: case 3: case 5: case 7: Palabra_4[7]=0; break;}
Palabra4_P2=0; //Paridad 2=0; bit 0 al bit 12
for(x=0; x<=12; x++) {Palabra4_P2=Palabra_4[x]+(Palabra4_P2*1);}
switch (Palabra4_P2)
{case 0: case 3: case 6: case 9: case 12:
Palabra_4[13]=0; Palabra_4[14]=0; break;

```

```

    case 1: case 4: case 7: case 10: case 13:
Palabra_4[13]=0; Palabra_4[14]=1; break;
    case 2: case 5: case 8: case 11: case 14:
Palabra_4[13]=1; Palabra_4[14]=0; break;}

```

```

Palabra4_P3=0; //Paridad 3=0; bit 8 al bit 14
for(x=8; x<=14; x++) {Palabra4_P3=Palabra_4[x]+(Palabra4_P3*1);}
switch (Palabra4_P3)
{case 0: case 2: case 4: case 6: Palabra_4[15]=1; break;
case 1: case 3: case 5: case 7: Palabra_4[15]=0; break;}
}

```

```

void Palab_5()
{
valor=correlador;
for(x=0; x<10; x++) {Corr[x]=bitRead(valor, x);}
y=0;
for(x=6; x>=0; x--) {Palabra_5[x]=Corr[y];y++;}
Palabra_5[8] = 0;
Palabra_5[9] = 0; //dato vacio
Palabra_5[10] = 0;
Palabra_5[11] = 0; //dato vacio
if (modo_acus==1) {Palabra_5[12]=1;}
else {Palabra_5[12]=0;}
Palabra5_P1=0; //Paridad 1=0; bit 0 al bit 6
for(x=0; x<=6; x++) {Palabra5_P1=Palabra_5[x]+(Palabra5_P1*1);}
switch (Palabra5_P1)
{case 0: case 2: case 4: case 6: Palabra_5[7]=1; break
case 1: case 3: case 5: case 7: Palabra_5[7]=0; break;}
Palabra5_P2=0; //Paridad 2=0; bit 0 al bit 12
for(x=0; x<=12; x++) {Palabra5_P2=Palabra_5[x]+(Palabra5_P2*1);}

```



```

switch (Palabra5_P2)
{case 0: case 3: case 6: case 9: case 12:
Palabra_5[13]=0; Palabra_5[14]=0; break;
case 1: case 4: case 7: case 10: case 13:
Palabra_5[13]=0; Palabra_5[14]=1; break;
case 2: case 5: case 8: case 11: case 14:
Palabra_5[13]=1; Palabra_5[14]=0; break;}

Palabra5_P3=0; //Paridad 3=0; bit 8 al bit 14
for(x=8; x<=14; x++) {Palabra5_P3=Palabra_5[x]+(Palabra5_P3*1);}
switch (Palabra5_P3)
{
case 0: case 2: case 4: case 6: Palabra_5[15]=1; break;
case 1: case 3: case 5: case 7: Palabra_5[15]=0; break;
}
}

void Palab_6()
{
valor=profundidad;
for(x=0; x<10; x++) {Prof[x]=bitRead(valor, x);}
y=3;
for(x=6; x>=0; x--) {Palabra_6[x]=Prof[y];y++;} //carga los 6 primeros bits
de profund en la palabra6
Palabra_6[8] = Prof[2]; Palabra_6[9] = Prof[1];
Palabra_6[10] = Prof[0]; Palabra_6[11] = 0;
if (motor==1) {Palabra_6[12]=1;}
else {Palabra_6[12]=0;}
Palabra6_P1=0; //Paridad 1=0; bit 0 al bit 6
for(x=0; x<=6; x++) {Palabra6_P1=Palabra_6[x]+(Palabra6_P1*1);}
switch (Palabra6_P1)

```

```

{
  case 0: case 2: case 4: case 6: Palabra_6[7]=1; break;
  case 1: case 3: case 5: case 7: Palabra_6[7]=0; break;
}
Palabra6_P2=0; //Paridad 2=0; bit 0 al bit 12
for(x=0; x<=12; x++) {Palabra6_P2=Palabra_6[x]+(Palabra6_P2*1);}
switch (Palabra6_P2)
{case 0: case 3: case 6: case 9: case 12:
Palabra_6[13]=0; Palabra_6[14]=0; break;
  case 1: case 4: case 7: case 10: case 13:
Palabra_6[13]=0; Palabra_6[14]=1; break;
  case 2: case 5: case 8: case 11: case 14:
Palabra_6[13]=1; Palabra_6[14]=0; break;}

Palabra6_P3=0; //Paridad 3=0; bit 8 al bit 14
for(x=8; x<=14; x++) {Palabra6_P3=Palabra_6[x]+(Palabra6_P3*1);}
switch (Palabra6_P3)
{
  case 0: case 2: case 4: case 6: Palabra_6[15]=1; break;
  case 1: case 3: case 5: case 7: Palabra_6[15]=0; break;
}
}

void Palab_7()
{
  valor=rumbo;
  for(x=0; x<10; x++) {Rumb[x]=bitRead(valor, x);}
  y=3;
  for(x=6; x>=0; x--) {Palabra_7[x]=Rumb[y];y++;}
  Palabra_7[8] = Rumb[2]; Palabra_7[9] = Rumb[1];
  Palabra_7[10] = Rumb[0]; Palabra_7[11]=0;
}

```

```

Palabra_7[12]=0; Palabra7_P1=0;
for(x=0; x<=6; x++) {Palabra7_P1=Palabra_7[x]+(Palabra7_P1*1);}
switch (Palabra7_P1)
{
case 0: case 2: case 4: case 6: Palabra_7[7]=1; break;
case 1: case 3: case 5: case 7: Palabra_7[7]=0; break;
}
Palabra7_P2=0; //Paridad 2=0; bit 0 al bit 12
for(x=0; x<=12; x++) {Palabra7_P2=Palabra_7[x]+(Palabra7_P2*1);}
switch (Palabra7_P2)
{
case 0: case 3: case 6: case 9: case 12: Palabra_7[13]=0;
Palabra_7[14]=0; break;
case 1: case 4: case 7: case 10: case 13: Palabra_7[13]=0;
Palabra_7[14]=1; break;
case 2: case 5: case 8: case 11: case 14: Palabra_7[13]=1;
Palabra_7[14]=0; break;
}

Palabra7_P3=0; //Paridad 3=0; bit 8 al bit 14
for(x=8; x<=14; x++) {Palabra7_P3=Palabra_7[x]+(Palabra7_P3*1);}
switch (Palabra7_P3)
{
case 0: case 2: case 4: case 6: Palabra_7[15]=1; break;
case 1: case 3: case 5: case 7: Palabra_7[15]=0; break;
}
}

```

Anexo 4: Programa Decodificador de Telegrama Actual

```
int cont=0; //contador de los 5 pulsos de sincronizacion
int rx = 2; //pin de lectura del telegrama de 206 baudios
int monitor=13; //led indicador de transmision telegrama
unsigned long pulso; // variable de ancho pulso de sincronizacion
unsigned long sincro[5]; //lecturade 5 de pulso de sincronizacion
unsigned long promedio=0; //promedio de los 5 de pulso
long time = 0; //tiempo de sincronizacion del telegrama
```

```
////VECTORES DEL TELEGRAMA
```

```
int Bits;
int Palabra1[16];
int Palabra2[16];
int Palabra3[16];
int Palabra4[16];
int Palabra5[16];
int Palabra6[16];
int Palabra7[16];
```

```
long word1=0;
long word2=0;
long word3=0;
long word4=0;
long word5=0;
long word6=0;
long word7=0;
long multiplicador = 1;
int x=0;
```

```
///VARIABLES DE DATOS DECODIFICADOS DE LAS PALABRAS
```

```
float cabeceo=0.00; //PALABRA 1
```

```

float roldo=0.00;    //PALABRA 1
float velocidad=0.00; //PALABRA 3
int homing = 0;      //PALABRA 3
float dist_act = 0.00; //PALABRA 4
int dist_seg = 0;    //PALABRA 4
int contacto = 0;    //PALABRA 4
float correlador = 0.00; //PALABRA 5
int modo_acustico = 0; //PALABRA 5
float profundidad =0.00; //PALABRA 6
int motor = 0;      //PALABRA 6
float rumbo = 0.00; //PALABRA 7

void setup()
{
  pinMode(rx, INPUT);
  pinMode(monitor, OUTPUT);
  Serial.begin(19200);
  time = 4850;
}

void loop()
{
  pulso = pulseIn(rx, HIGH);
  if(pulso > 60000)
  {
    Leer_Telegrama();
    Decodificar_Telegrama();
    Transmitir_Telegrama();
  }
}

```

```

//////////////////////////////////LECTURA DEL TELEGRAMA ACTUAL//////////////////////////////////
void Leer_Telegrama(void)
{
  int led=0;
  delayMicroseconds(time/2);//2420

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra1[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra2[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra3[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra4[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra5[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra6[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}

  for(Bits=0;Bits<=15;Bits++){delayMicroseconds(time);Palabra7[Bits]=digital
  IRead(rx);led=!led;digitalWrite(monitor,led);}
  digitalWrite(monitor, LOW);
}

```

```

Void Decodificar_Telegrama()
{
word1=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 1
for(x=0; x<=15; x++) {word1=(Palabra1[x]*multiplicador) + word1;
multiplicador=multiplicador*2;}
word2=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 2
for(x=0; x<=15; x++) {word2=(Palabra2[x]*multiplicador) + word2;
multiplicador=multiplicador*2;}
word3=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 3
for(x=0; x<=15; x++) {word3=(Palabra3[x]*multiplicador) + word3;
multiplicador=multiplicador*2;}
word4=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 4
for(x=0; x<=15; x++) {word4=(Palabra4[x]*multiplicador) + word4;
multiplicador=multiplicador*2;}
word5=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 5
for(x=0; x<=15; x++) {word5=(Palabra5[x]*multiplicador) + word5;
multiplicador=multiplicador*2;}
word6=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 6
for(x=0; x<=15; x++) {word6=(Palabra6[x]*multiplicador) + word6;
multiplicador=multiplicador*2;}
word7=0; multiplicador = 1; // Conversion de binario a Decimal de la
Palabra 7
for(x=0; x<=15; x++) {word7=(Palabra7[x]*multiplicador) + word7;
multiplicador=multiplicador*2;}
}

```

```
void Transmitir_Telegrama()
{
  Serial.print("$TEL:");
  Serial.print(word1); Serial.print(",");
  Serial.print(word2); Serial.print(",");
  Serial.print(word3); Serial.print(",");
  Serial.print(word4); Serial.print(",");
  Serial.print(word5); Serial.print(",");
  Serial.print(word6); Serial.print(",");
  Serial.print(word7); Serial.print('\n');
}
```