

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN**

Diseño e implementación de un sistema de comunicaciones bidireccional entre el GPS de un helicóptero BELL 430 hacia una estación base mediante un bus de datos y un módulo LTE.

PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN TELECOMUNICACIONES

Autores:

Freddy Isaac Mora Oñate

Anggie Carolina Rugel Díaz

GUAYAQUIL-ECUADOR

Año: 2022

DEDICATORIA

El presente trabajo se lo dedico en primer lugar a Dios, por bendecirme con una excelente salud para poder seguir adelante con mis estudios y bendiciones a lo largo de mi vida.

A mis padres Carlos Rugel y Mariuxi Díaz, por apoyarme desde siempre con su tiempo, de dedicación y sacrificio para que pueda cumplir una de mis anheladas metas que es ser profesional y ser su orgullo.

A mi hermana Pamela Rugel, por ayudarme en momentos en que la necesitaba y a mi hermano Jean Carlos Rugel ya que es uno de los motores para no rendirme y en conjunto con mi hermana ser un ejemplo para él.

A mis abuelos maternos Plácido Díaz y Elena Chiang, por llenarme de amor y estar conmigo en todo momento.

Anggie Carolina Rugel Díaz.

DEDICATORIA

A mi familia y a los buenos amigos que hice en el camino.

Freddy Isaac Mora Oñate.

AGRADECIMIENTOS

En estos años he tenido el placer de compartir con compañeros que se han convertido en amigos y demás personas que han aportado en mi crecimiento profesional y sobre todo el personal.

A mis padres, familiares y a Salim mi compañero incondicional por estar siempre conmigo y brindarme sus sabios consejos para ser una persona buena, amable, responsable, honesta, ya que para *“ser un gran profesional hay que ser una excelente persona”*

Al PhD Francisco Novillo y a la PhD Patricia Chávez por su paciencia, consejos para el desarrollo del proyecto integrador y enseñanzas valiosas necesarias para mi crecimiento como profesional.

Finalmente quiero expresar mi más profunda gratitud a la Escuela Superior Politécnica Superior del Litoral, a todos los profesores de la facultad de FIEC en especial al MSc. Eduardo Chancay y al Ingeniero Aristóteles Amat por sus enseñanzas, paciencia, amistad y por compartir sus conocimientos para poder como profesional.

Anggie Carolina Rugel Díaz.

Quiero agradecer a Dios, por la maravillosa oportunidad de la vida, por mantenerme fuerte, saludable y poner en mi camino gente muy buena.

A mis padres, Freddy Mora Alay y Dolly Oñate Macías, por sus infinitas muestras de amor hacia mí, por apoyarme de manera incondicional en mis estudios, por estar siempre que los necesité, porque cada uno de mis pasos llevan la huella imborrable de su trabajo en mí.

A la PhD Patricia Chávez Burbano y Al PhD Francisco Novillo Parales, por el apoyo, dirección, paciencia, consejos y por su irrestricta predisposición para el desarrollo de este proyecto.

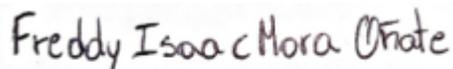
Al Sr. CPCB-IG José Peñafiel Fernández, al Sr. TNNV-IG Wellington Guanopatin Matute y al Sr. SGOS-ET-AV Jhonny Cacao, por la confianza depositada en mí y por el apoyo logístico desinteresado.

A mi alma máter, la Escuela Superior Politécnica del Litoral, con su personal académico, administrativo y de servicios generales, por la cálida acogida, por hacer de la universidad un hogar para todos, con una mención especial para PhD César Martín Moreno, MSc Eduardo Chancay Rojas y MSc Nathaly Sánchez Chan.

Freddy Isaac Mora Oñate.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Fredy Issac Mora Oñate y Angie Carolina Rugel Díaz damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Fredy Isaac Mora Oñate



Angie Carolina Rugel Díaz

EVALUADORES

Ph.D. Francisco Vicente Novillo Parales

PROFESOR DE LA MATERIA

PhD. Patricia Ximena Chávez Burbano

PROFESOR TUTOR

RESUMEN

En la actualidad, salvaguardar la seguridad de la soberanía marítima del país es esencial para evitar el paso de flotas pesqueras o embarcaciones que puedan afectar a la estabilidad de los ecuatorianos.

Siendo imprescindible la participación de una entidad pública del Ecuador para realizar actividades de patrullaje en zonas de difícil acceso, detectando de esta forma contactos de interés en el área para el adecuado seguimiento y notificación.

En el transporte aéreo en donde se realiza la vigilancia de la costa ecuatoriana, posee un sistema de comunicaciones que impide la transmisión y recepción de los datos del objetivo detectado en tiempo real, debido a que no logran obtener un radioenlace adecuado, a causa de los obstáculos que impiden una efectiva comunicación entre el helicóptero y la base.

Por este motivo, se propuso la implementación y el diseño de un sistema de comunicaciones bidireccional con el objetivo de obtener, procesar y almacenar los datos del GPS mediante un bus de datos (RTX SPA429) siendo un dispositivo programable para la conversión de datos del protocolo ARINC 429 a uno convencional como es el USB, mediante el lenguaje de programación c++ a través de la interfaz Visual en el sistema operativo Windows.

Adicional, para fines de seguridad y respaldo, se almacenaron los datos extraídos en un archivo de texto plano para ser enviado a la nube a través de un módulo LTE para que el personal de administración de la base central pueda acceder a los datos en tiempo real a través de la tecnología Long Term Evolution para finalmente ser respaldado en un servidor local.

ABSTRACT

Currently, safeguarding the security of the country's maritime sovereignty is essential to prevent the passage of fishing fleets or vessels that may affect the stability of Ecuadorians.

The participation of one public entity of Ecuador is essential to carry out patrolling activities in areas of difficult access, thus detecting contacts of interest in the area for adequate follow-up and notification.

In the air transport where the surveillance of the Ecuadorian coast is carried out, it has a communications system that prevents the transmission and reception of the data of the target detected in real time, because they fail to obtain an adequate link, due to the obstacles that prevent effective communication between the helicopter and the base.

For this reason, the implementation and design of a bidirectional communications system was proposed with the aim of obtaining, processing, and storing GPS data through a data bus (RTX SPA429) being a programmable device for the conversion of data from the ARINC 429 protocol to a conventional one such as USB, through the c++ programming language through of Visual interface in the Windows operating system.

Additionally, for security and backup purposes, the extracted data was stored in a plain text file to be sent to the cloud via an LTE module so that the central base management staff can access the data in real time through Long Term Evolution technology to finally be backed up on a local server.

ÍNDICE GENERAL

CAPÍTULO 1	14
1. Introducción	14
1.1 Definición del Problema	15
1.2 Justificación del Problema	16
1.3 Objetivos.....	17
1.4 Propuesta de Solución	17
1.5 Metodología	18
CAPÍTULO 2	20
2. Marco Teórico	20
2.1 Comunicación Serial	20
2.2 Protocolo de Comunicación USB	21
2.3 Protocolo ARINC 429.....	21
2.4 Formato de la palabra ARINC 429	21
2.5 Velocidad de transmisión.....	24
CAPÍTULO 3	26
3. Diseño de la solución	26
3.1 Descripción del sistema	28
3.2 Planteamiento inicial.....	31
3.3 Funcionamiento del sistema	31
CAPÍTULO 4	34
4. Resultados y análisis.....	34
4.1. Implementación del sistema de comunicación	34
4.2 Pruebas de Campo.....	36
CAPÍTULO 5	44
5. Conclusiones y recomendaciones	44
5.1. Conclusiones	44
5.2. Recomendaciones.....	45
BIBLIOGRAFÍA.....	46
ANEXOS.....	48

ÍNDICE DE TABLAS

Tabla 2-1. Matriz de Signo/Estado de la codificación de datos BCD y BNR24

ÍNDICE DE FIGURAS

<i>Figura 1-1. Diagrama de bloques del Proyecto</i>	18
<i>Figura 2-1. Formato de la palabra ARINC 429</i>	22
<i>Figura 2-2. Cifras de etiqueta</i>	22
<i>Figura 2-3. Datos codificados en BNR</i>	23
<i>Figura 2-4. Datos codificados con BCD</i>	23
<i>Figura 3-1. Helicóptero Bell 430 diseño 2D AutoCAD</i>	26
<i>Figura 3-2. Helicóptero Bel 430 tamaño real</i>	27
<i>Figura 3-3. Circuito de GPS de helicóptero Bell 430</i>	28
<i>Figura 3-4. Sistema de comunicaciones</i>	31
<i>Figura 3-5. Estructura general del código fuente de la aplicación</i>	32
<i>Figura 4-1. Serial de comunicación hacia el GPS</i>	35
<i>Figura 4-2. Conector serial macho</i>	35
<i>Figura 4-3. Circuito Armado</i>	36
<i>Figura 4-4. Pruebas de Campo</i>	37
<i>Figura 4-5. GPS del Helicóptero</i>	38
<i>Figura 4-6. Datos de latitud y longitud decodificados</i>	39
<i>Figura 4-7. Coordenadas gráficas del GPS</i>	40
<i>Figura 4-8. Coordenadas gráficas decodificadas</i>	40
<i>Figura 4-9. Distancia entre las dos coordenadas</i>	41
<i>Figura 4-10. Datos en .txt</i>	42

ÍNDICE DE ANEXO

<i>Anexo 1. Función decode</i>	48
<i>Anexo 2. Función label</i>	49
<i>Anexo 3. Función sign</i>	50

CAPÍTULO 1

1. Introducción

En los últimos años en el país una entidad pública ha detectado varias flotas pesqueras realizando actividades ilegales cercanas a la costa ecuatoriana y demás embarcaciones de actividades sospechosas. Estos objetivos de interés son notificados a una base central en tierra.

Sin embargo, para realizar esta acción muchas veces las aeronaves tienen que realizar varios procesos para poder enviar la información debido a que usualmente el patrullaje se realiza en zonas donde no se logra un radioenlace correcto entre el helicóptero y la antena receptora de la central.

Esta situación ocasiona que los datos que se requieren enviar sobre los contactos de interés no sean en tiempo real debido a que, el sistema de comunicación que posee el helicóptero BELL 430 está descontinuado. Adicionalmente, provoca que el patrullaje no sea efectivo ya que, al no tener una buena línea de vista para enviar la información los pilotos abandonan la zona y se alejan del objetivo, lo que implica un mayor consumo de combustible debido a la trayectoria recorrida y pérdida de las embarcaciones sospechosas.

Por este motivo, este proyecto consiste en la renovación de tecnología implementando un sistema de comunicación bidireccional. Como primera fase se obtendrán los datos del GPS, el cual trabaja con el protocolo ARINC 429 [1]. Estos datos serán transformados para su envío por un puerto USB mediante el RTX SPA 429 [2], permitiendo una mayor facilidad del manejo de datos para agregar y obtener las etiquetas que se requieren.

Los datos procesados, se almacenarán en una computadora y a su vez se guardarán los archivos generados en la nube a través de un módulo LTE para

transmitir y recibir datos en tiempo real, y siendo una tecnología implementada en muchos servicios actuales [3], respaldando los datos en un servidor local.

1.1 Definición del Problema

El Estado Ecuatoriano tiene como principal objetivo generar estrategias para garantizar el bienestar de su población, protegiendo su territorio frente a posibles amenazas que atenten contra la estabilidad e intereses de sus habitantes. Siendo imprescindible la participación de la entidad pública, para defender los espacios marítimos con el fin de contribuir con la defensa y seguridad regional.

Las aeronaves de este organismo mantienen vigilancia y patrullaje continuo en el país detectando en varias ocasiones la presencia de flotas que usualmente están conformadas por embarcaciones pesqueras, de suministro, buques, entre otros, que atentan contra la integridad de la vida marina y humana. Sin embargo, las distintas aeronaves realizan usualmente las labores de patrullaje en zonas que son de difícil acceso lo que impide un correcto monitoreo, detección y envío de datos sobre las embarcación o contactos de interés que realicen actividades que no se encuentren dentro del margen legal.

Entre las aeronaves se encuentra el helicóptero Bell 430, el cual posee sistemas responsables de captar datos sobre la navegación, curso, velocidad, entre otros, de distintos contactos de interés durante sus labores cotidianas de patrullaje. Estos datos necesitan ser enviados a una estación base en tierra o a un buque que pueda brindar soporte en una posible eventualidad. No obstante, el sistema actual de comunicaciones presenta limitantes al momento de transmitir y recibir dichos datos, ya que no cuenta con una adecuada línea de vista para realizar esta actividad. Obligando al helicóptero BELL 430 al tener que acercarse hacia la Costa lo que deriva en un incremento de costos operativos en combustible, pérdida de visión con el contacto de interés y una notable disminución en la eficiencia de sus labores.

1.2 Justificación del Problema

En el presente proyecto se realizará el manejo de datos del protocolo de comunicación aviónico ARINC 429, el cual permite la transmisión de flujo de información entre varios dispositivos. Los datos obtenidos que trabajan con este protocolo se los obtiene del GPS que se encuentra en helicóptero Bell 430 con el fin de procesar dichos datos y convertirlos a un tipo de protocolo más convencional como el USB.

Esto se debe realizar como la fase inicial de la implementación de un sistema paralelo de comunicación, ya que el actual sistema presenta inconvenientes para transmitir y recibir los datos hacia una estación base, debido por la tecnología obsoleta con la que trabaja actualmente el helicóptero además de la falta de línea de vista debido a obstáculos no deseados en ciertas ocasiones cuando se realiza la actividad de patrullaje. Estos sucesos obligan al operador del helicóptero a desplazarse hasta lograr un radioenlace adecuado para la correcta comunicación bidireccional y que puedan ser enviadas a la base al detectar contactos de interés que afecten al bienestar de la población ecuatoriana.

Además, como consecuencia de no renovar esta tecnología se ha producido un incremento en los costos de operación. Existe a un mayor consumo de combustible ya que el helicóptero recorre un mayor trayecto al detectar a los contactos de interés y después para alejarse y poder encontrar una correcta posición y altura para tener una buena línea de vista. Adicionalmente, se tiene una limitación en la autonomía de la aeronave, lo que implica intervalos de patrullaje más cortos y como consecuencia más grave la pérdida de seguimiento de los posibles objetivos.

Es por esto, que el desarrollo de este proyecto como fase inicial permitirá obtener los datos adquiridos en el GPS que trabajan con el protocolo ARINC 429 para

ser tratados y convertidos a USB para después poder almacenarlos en la nube para finalmente proyectarlos en pantalla.

Como beneficiario de este proyecto se tiene una entidad pública del Ecuador, que obtendrá como resultado la extracción de los datos de GPS para realizar el mismo método utilizando en radar como fase final del proyecto.

1.3 Objetivos

Este proyecto tiene como objetivo general ‘Diseñar e implementar un sistema de comunicaciones bidireccional entre el GPS de un helicóptero BELL 430 hacia una estación base mediante un bus de datos y un módulo LTE’. Para conseguir propósito se proponen y detallan 4 objetivos específicos.

- Visualizar las etiquetas ARINC 429 en una PC mediante una herramienta de panel gráfico.
- Implementar herramientas que permitan la lectura y escritura de datos en el GPS.
- Convertir los datos del protocolo ARINC 429 a un protocolo convencional.
- Transmitir los datos mediante un módulo LTE hacia la nube y almacenarlos en un servidor local.

1.4 Propuesta de Solución

En la propuesta se plantea como una posible solución a los problemas identificados, emplear mecanismos para lograr un mejor rendimiento del sistema de comunicaciones actual. Tales como modificar la potencia de salida de la señal del dispositivo transmisor, logrando así que la potencia recibida no sea menor a la sensibilidad del dispositivo, y de manera paralela, poder aumentar la altura de las antenas de la estación base.

Así mismo se propone como segunda solución, la renovación de tecnología con un sistema de comunicación paralelo, para lo cual, como fase previa se recomienda realizar la obtención, procesamiento y almacenamiento de los datos proporcionados por el GPS de un helicóptero BELL 430 mediante los canales que manejan el protocolo ARINC 429. Estos datos se transformarán a un protocolo diferente, en este caso puntual al protocolo USB, para ser enviados a través de un puerto de salida de la PC hacia un módulo LTE, mismo que cargará los archivos en la nube, para luego ser almacenados de manera local en una computadora que cumplirá las funciones de un servidor.

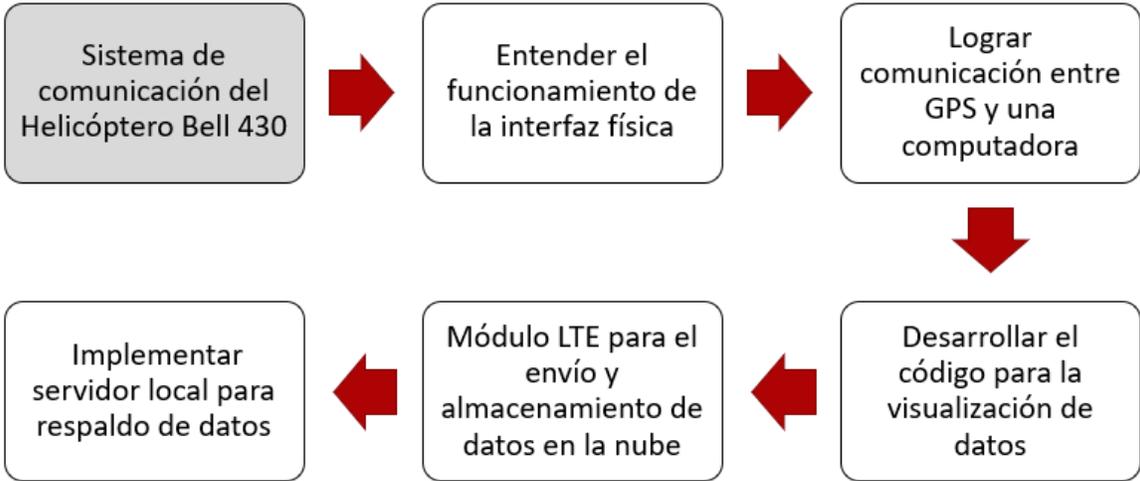


FIGURA 1-1. DIAGRAMA DE BLOQUES DEL PROYECTO

1.5 Metodología

Inicialmente, en fases previas a la implementación de este proyecto, se realizó la investigación para definir el proceso de adquisición de datos del radar del Helicóptero Bell 430, con la finalidad de lograr la forma óptima y viable para acceder a los datos antes mencionados. Se concluyó que se podía acceder a dicha información mediante interfaces de salida que manejan protocolos aviónicos, considerando de acuerdo con las especificaciones de los equipos, como el óptimo el protocolo ARINC 429.

Luego se realizó la identificación de interfaces físicas que cumplan con los requerimientos para manejar el protocolo ARINC 429, de manera que este dispositivo pueda realizar la conversión de los datos del protocolo antes mencionado a un protocolo más convencional, como USB, para una mayor facilidad en el manejo de los datos.

Adicionalmente, se implementó el desarrollo de un programa de código basado en lenguaje C mediante la herramienta de desarrollador del bus de datos, una API con las funciones necesarias para la transmisión y recepción de datos del GPS al microcontrolador.

Posteriormente, siguiendo con el cumplimiento de los objetivos planteado, se creó un panel gráfico que permita observar los datos de las etiquetas ARINC 429 generadas en el GPS del helicóptero Bell 430.

Seguidamente, se enviarán los datos almacenados y procesados en el computador, por el puerto USB hacia el módulo LTE, para ser almacenados en la nube.

Finalmente, se realizó la implementación de un servidor local que realice el respaldo de los datos almacenados en la nube.

CAPÍTULO 2

2. Marco Teórico

En este capítulo se especificarán fundamentos teóricos que son importante para entender la propuesta de solución.

2.1 Comunicación Serial

Entre los dispositivos para realizar la conexión se emplea un puerto serial mediante la interfaz RS-232 desarrollada por la EIA (Electronic Industries Association) la cual establece una norma para el intercambio de datos entre un equipo terminal de datos y uno de comunicación [4]. Siendo un medio para la transmisión de palabras que están compuesta por números binarios de forma continua se transmite por un solo canal alcanzando mayores distancias. [5]

Por otra parte, a diferencia de la comunicación serial también se puede realizar la comunicación paralela, lo que implica que los datos no se transfieren en un solo canal, sino que se realiza la emisión de bits por canales separados.

Además, que en la comunicación serial se transmiten bit por bit lo que indica que se enviar bytes de información mientras que en la comunicación paralela se realiza la emisión de un byte completo, siendo una comunicación más rápida. [6]

Es por eso que estándar de la IEEE 488.1-2003 establece que los sistemas para interconectar aparatos electrónicos programables como no programables con otros dispositivos se debe aplicar el sistema de instrumentación [7] de la IEEE donde se establece que para la comunicación en paralelo se debe de utilizar un cable que posee un largo de al menos a 20 metros [8], mientras que para la comunicación serial puede llegar a medir hasta 1200 metros aproximadamente, lo que permite que esta comunicación sea de mayor alcance.

2.2 Protocolo de Comunicación USB

El protocolo de comunicación serial USB o en sus siglas en inglés Universal serial Bus, es una interfaz empleada para la transmisión de datos en la que intervienen el controlador anfitrión y los dispositivos, donde el primer componente tiene el control. Por otra parte, los dispositivos conectados escuchan al bus y se encuentran en espera de que el anfitrión, les solicite o envíe información, es decir, es un bus serie direccional, iniciando de esta forma la comunicación. [9]

2.3 Protocolo ARINC 429

ARINC 429 o formalmente conocido como Sistema de Transferencia de Información Digital MARK 33, es el protocolo que define los estándares de la industria de los medios de transporte aéreo, ya que regula la transmisión de datos digitales [10] como la estructuras de las palabras, características eléctricas y lenguajes necesarios para poder establecer la comunicación sobre el bus de datos aviónico.

2.4 Formato de la palabra ARINC 429

La palabra ARINC 429 tiene un formato que está constituida por 32 bits que se dividen en 5 bytes para formar 8 dígitos hexadecimales que se transmiten desde la fuente que en este caso en el GPS. Como se muestra en la Figura 2-1, los 32 bits se dividen en 5 campos que son:

- Etiqueta.
- Identificador Origen/Destino (SDI).
- Datos.
- Matriz de Signo/Estado (SSM).
- Paridad (P).

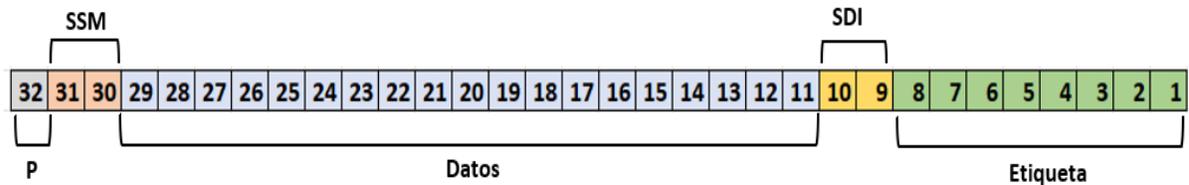


FIGURA 2-1. FORMATO DE LA PALABRA ARINC 429

2.4.1 Etiqueta

Los bits de 1 al 8 contienen información relevante sobre la etiqueta del Identificador de Información que indica el tipo de dato a procesar. La etiqueta se debe expresar en codificación octal dando como resultado 3 cifras [11]. La primera cifra se forma con los bits 1 y 2, para la segunda y tercera cifra se obtiene con 3 dígitos binarios, siendo la cifra 2 conformada por los bits 3, 4 y 5, y la cifra 3 por los bits 6, 7 y 8, como se muestra en la Figura 2-2.

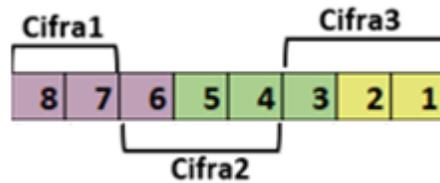


FIGURA 2-2. CIFRAS DE ETIQUETA

2.4.2 Identificador Origen/Destino (SDI)

De los 32 bits de la palabra ARINC 429, los bits 9 y 10 son exclusivamente reservados para reconocer el dispositivo que actúa como fuente o el equipo receptor en donde se va a transmitir las palabras. Codificando los bits SDI en las entradas y salidas según corresponda, por lo general, el código que contienen es el “00”. [12] Adicional, los bits 9 y 10 también pueden ser empleados para obtener una mayor resolución.

2.4.3 Datos

Los datos están conformados desde el bit 11 hasta el bit 29, donde se encuentra la información que aprovechará el destinatario, en donde se puede emplear dos tipos de

codificación que es BNR (codificación en binario) y BCD (decimal codificado en binario), esto dependerá del tipo de dato a enviar [13]. A continuación, se detallan los esquemas de cada codificación que son:

- Codificación en binario.** – Los bits de información se codifican en binario y en complemento de 2, donde a cada bit se le establece un valor con respecto al anterior pero dividido para dos. Como se observa en la Figura 2-3, los bits de datos BNR son desde el 23 hasta el bit 29, lo que indica que el número codificado solo usa 7 bits, mientras que los sobrantes son de relleno, siendo el bit de la izquierda el más significativo, lo que indica el signo del número, siendo éste positivo o negativo.

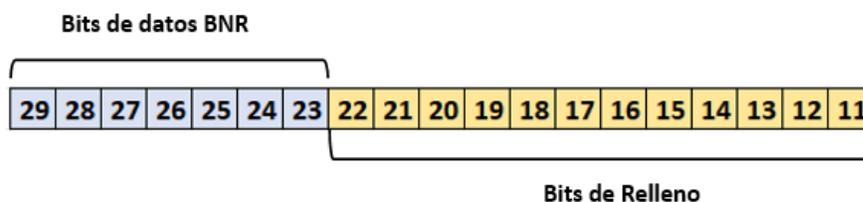


FIGURA 2-3. DATOS CODIFICADOS EN BNR

- Decimal codificado en binario.** – La codificación BDC utiliza 4 bits para representar a cada dígito, dando un máximo de 5 dígitos, como se visualiza en la Figura 2-4. Sin embargo, solo el dígito 1 estará formado por 3 bits, teniendo como valor máximo de 7, sin embargo, si excede el valor los bits 27, 28 y 29 se rellenan de 0, y no tendrá valor por lo que el dígito 2 se convertirá en el MSB.

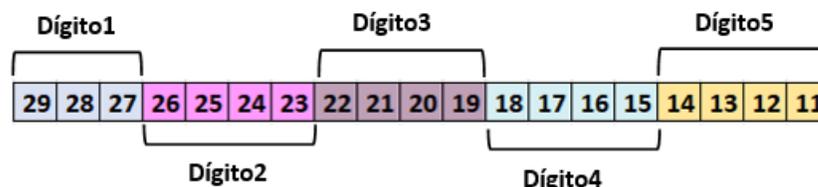


FIGURA 2-4. DATOS CODIFICADOS CON BCD

2.4.4 Matriz de Estado

La matriz de estado va a depender de la etiqueta y del tipo de codificación empleada en los datos ya sea BNR y BCD, variando el significado de la combinación de los bits en cada codificación como se muestra en la Tabla 2-1.

SSM	BCD	Significado	SSM	BNR	Significado
0	0	Mas, Norte, Este, Derecha, A, Arriba	0	0	Advertencia de falla
0	1	Sin Datos procesados	0	1	Sin datos procesados
1	0	Test Funcional	1	0	Test Funcional
1	1	Menos, Sur, Oeste, Izquierda, De, Abajo	1	1	Operación Normal

TABLA 2-1. MATRIZ DE SIGNO/ESTADO DE LA CODIFICACIÓN DE DATOS BCD Y BNR

2.4.5 Paridad

El bit 32 de paridad tiene como fin detectar errores más no corregirlo, lo que indicará si la palabra del protocolo 429, es impar o par dependiendo del total de 1 de los 31 bits restantes. Si la cantidad de 1 es impar por ende la paridad transmitida será impar, como consecuencia, se enviará un bit 0. Sin embargo, si la palabra transmitida es par, se emitirá un 1 como bit 32.

2.5 Velocidad de transmisión

La velocidad de transferencia es la cantidad de datos que se trasladan de un medio a otro, teniendo como unidad el Baudio no el bit en este caso, ya que el baudio es la unidad de

la velocidad de señalización. Esto implica que con esta unidad de medición se determina la existencia o ausencia de pulsos en un canal.

La velocidad en baudios, se la obtiene del inverso de la duración del impulso, el cual debe ser el más corto, donde su unidad está dada en segundos.

Cabe recalcar que las unidades de bit por segundo y baudios no son iguales, ya que representan diferentes situaciones, sin embargo, estas unidades significan lo mismo solo en el caso específico donde no hay variación en el ancho de los impulsos indicando que todos los pulsos son de información. [14]

CAPÍTULO 3

3. Diseño de la solución

El diseño del prototipo de solución se lo realizó debido a que el sistema que tiene incorporado el helicóptero BELL 430 de una entidad pública entidad pública del país, es obsoleto. Debido a que posee muchas limitantes al realizar uno de los principales objetivos que es el de transmitir y recibir datos de los contactos de interés u objetivos en tiempo real.

Por lo que el sistema diseñado como parte de la primera fase del proyecto, se centra en la renovación de tecnología con un sistema de comunicación que funcione de forma paralela al sistema convencional para poder enviar y recibir la información detectada del objetivo hacia la estación base y asegurando la integridad y bienestar del país.

En la Figura 3-1, se puede observar el interior del helicóptero BELL 430 diseñado en 2D, donde se implementará el sistema paralelo en la parte delantera de la cabina del piloto (panel de control).

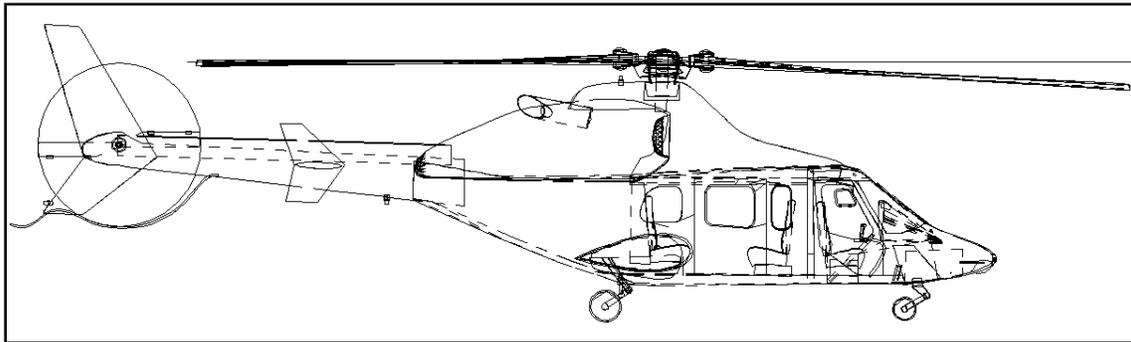


FIGURA 3-1. HELICÓPTERO BELL 430 DISEÑO 2D AUTOCAD

En la Figura 3-2, se observa el helicóptero en tamaño real, siendo el vehículo aéreo utilizado para realizar las actividades de patrullaje de forma continua en las periferias del país.



FIGURA 3-2. HELICÓPTERO BEL 430 TAMAÑO REAL

Siendo de gran importancia la implementación del sistema propuesta ya que en la actualidad el sistema implementado no permite la emisión y recepción de datos de forma adecuada en zonas de difícil acceso.

Como consecuencia se tiene pérdida considerable de combustible debido a que al realizar las actividades de vigilancia aérea y al tratar de enviar los datos sobre el contacto de interés el helicóptero debe de tener una adecuada línea de vista (LOS) y en muchas zonas no es posible por lo cual debe de realizar un mayor recorrido hasta lograr conexión con la estación base, incrementado de esta forma gastos en combustible.

Adicional, se tiene como consecuencia la pérdida de seguimiento del contacto de interés ya que al realizar las actividades correspondientes al patrullaje para establecer conexión en la base a tierra se pierde la visibilidad del objetivo.

Por otro lado, la suma de las consecuencias anteriormente mencionadas genera una disminución en la eficiencia de las labores de vigilancia realizadas por la entidad pública ya que, el tiempo designado para el patrullaje es menor al requerido evitando la detección de flotas pesqueras y demás objetivos.

Siendo el diseño propuesto con nueva tecnología esencial para poder realizar la obtención, procesamiento y almacenamiento de datos que como primera fase los proporcionará el GPS del medio de transporte aéreo para poder enviarlos hacia la estación base utilizando una nueva tecnología como LTE y no mediante UVHF.

3.1 Descripción del sistema

Para el diseño de la solución, se ha utilizado como dispositivo principal el GPS del helicóptero BELL 430, como se muestra en la Figura 3-3, del cual se obtiene los datos e información requerida para ser emitida hacia la estación base. Sin embargo, para poder extraer los datos se incorporó un bus datos que va conectado al GPS mediante en puerto RS232 utilizando los pines 46 y 47.

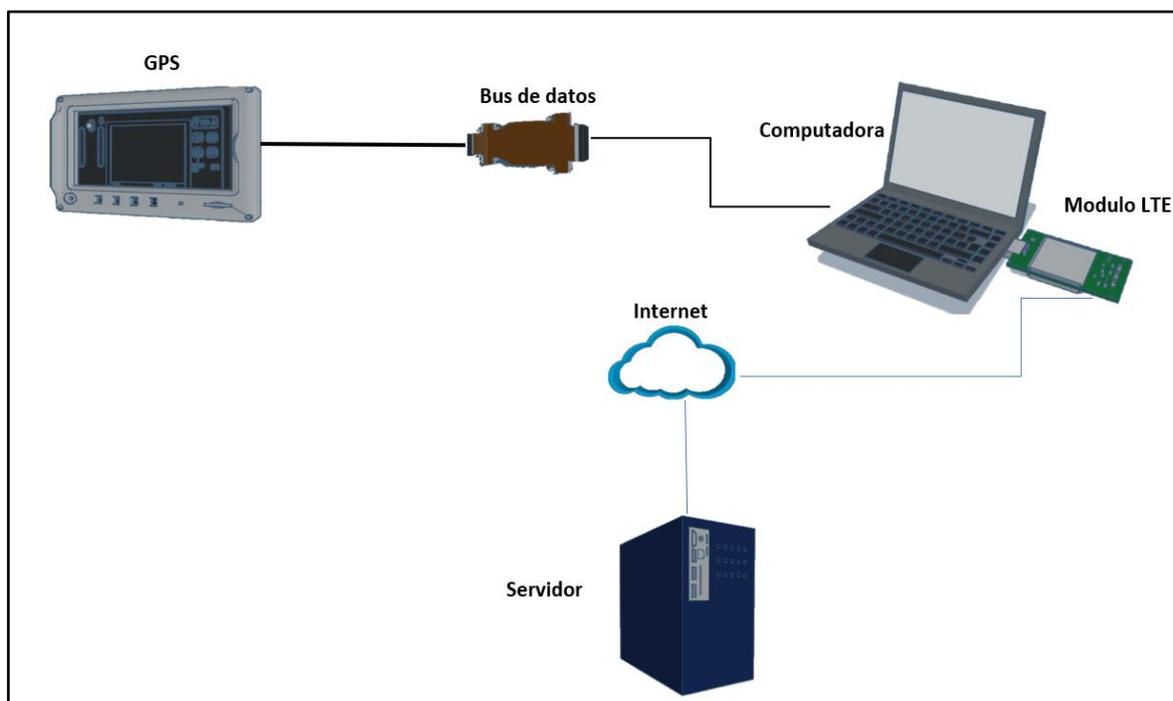


FIGURA 3-3. CIRCUITO DE GPS DE HELICÓPTERO BELL 430

Debido a que está configurado para 2 canales de transmisión y recepción ARINC 429, por lo que se conectan el canal de transmisión 0 al canal de recepción 0 y el canal de

transmisión 1 al de recepción 1. Activando en el GPS el canal ARINC 429 ya que es protocolo aviónico que ayuda a la transmisión de datos digitales donde puede variar la velocidad de transmisión en alta (100 Kbps) y baja (12,5 kbps). Siendo empleada para el proyecto una velocidad de 12,5 Kbps para el flujo de los datos.

Activación del sistema reconociendo el canal de transmisión, así como parámetros sobre la velocidad, se los configuró en el bus de datos empleando el lenguaje de programación de baja categoría que es utiliza como sistema operativo Windows siendo este Microsoft Visual (C++). En el código se utilizó una biblioteca, la cual contiene 3 archivos como APISPA.dll, APISPA.lib y APISPA.h ya que, contienen funciones de gran importancia que ayuda a inicializar el bus de datos y para la obtención de la información del canal ARINC 429.

Los datos recibidos desde el sistema de posicionamiento global del medio de transporte aéreo tienen un formato de 32 bits que se divide en 5 campos fundamentales; tratándolos y segmentándolos para obtener la información correspondiente sin errores.

El primer campo es la etiqueta que contiene los 8 bits menos significativos de toda la palabra ARINC, lo que significa que son los primeros bits en transmitirse por el canal. A su vez, estos bits se dividen en 3 cifras, la primera menos significativa contine solo 2 dígitos mientras que la segunda y tercera cifra incluyen 3 bits cada una. Con el fin de codificar cada cifra a un dígito de forma octal desde el menos significativo hasta el más significativo. Para obtener de esta forma la etiqueta que es la que proporciona información del tipo de dato que se está recibiendo, como longitud, latitud, meridiano de Greenwich, entre otros de acuerdo con la numeración de la etiqueta.

Adicional, se obtienen los bits 9 y 10 los cuales ayudan a detectar desde que dispositivo ha sido enviada la información o el destino, de la misma forma que la etiqueta su codificación es octal.

Los bits que contiene la información numérica de los datos e información sobre coordenadas dependiendo de cada etiqueta correspondiente, se extraen de los bits 11 hasta el 29, siendo el bit 29 quien brinda información en el caso de la etiqueta de latitud y longitud si la coordenada es hacia el norte, sur, este, oeste, más o menos dependiendo si el bit 29 es 1 o 0, además que ayuda a detectar el tipo de codificación que se debe de emplear para los datos.

Una vez detectada la numeración en octal de la etiqueta y el bit 29, se procede a extraer los datos digitales del 11 al 28, donde si el bit 29 es 1 se realiza la codificación BNR que consiste en codificar en binario y realizar el complemento a 2, obteniendo los datos binarios para proceder a convertirlo en valor número dependiendo de la etiqueta, en el caso de ser longitud y latitud se procede a sumar los bits 1 de acuerdo con las posiciones, y a multiplicar a su escala correspondiente que es 180. Obteniendo datos en grados, y por lo consiguiente, convertir la parte decimal a minutos multiplicando por 60, consiguiendo de esta manera el formato de la posición en grados y en minutos.

En caso de que el bit 29 sea positivo, no se realiza la codificación BNR, sino que se emplea la codificación BCD que consiste en 5 dígitos para poder realizar la codificación y obtener los datos decodificados.

Para finalmente almacenar los datos en la nube y enviarlos mediante un módulo que usa la tecnología LTE que permite tener conectividad para guardar datos en internet, además de poder resguardarlos en un servidor local en la base central

3.2 Planteamiento inicial

Se obtendrán datos de etiquetas del protocolo ARINC 429 de un GPS Garmin 400, las cuales se recibirán por medio de una interfaz de bus de datos capaz de transcribir los datos recibidos del GPS a un protocolo más convencional como USB. Esta información será procesada en un computador dotado con una aplicación que decodifique la información recibida, la muestre en pantalla y la almacene en un archivo de texto en unidades de ingeniería. Esta información será enviada mediante un módulo LTE a la nube y guardada en el servidor.

3.3 Funcionamiento del sistema

La Figura 3-4, muestra el funcionamiento general del sistema de comunicaciones, que consta de varias etapas bien definidas. Primero se observa la fuente de etiquetas ARINC 429, un GPS Garmin 400 (de dónde se obtendrán los datos), luego se observa un bus de datos, responsable de la transferencia de datos entre el GPS y la computadora.

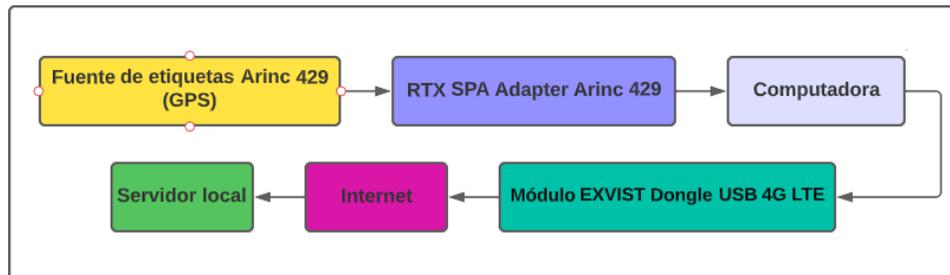


FIGURA 3-4. SISTEMA DE COMUNICACIONES

El bus de datos se inicializará mediante el software descrito desde la computadora con capacidad de procesar y decodificar la información, para luego almacenarla en un archivo de texto. Los datos almacenados se enviarán a la nube mediante un módulo LTE para luego ser respaldados en un servidor local.

La Figura 3-5, muestra la estructura general del código fuente de la aplicación implementada para el procesamiento de datos ARINC 429 del GPS Garmin 400. Primero se incluyen las librerías necesarias para las operaciones requeridas para que el programa funcione de manera correcta.

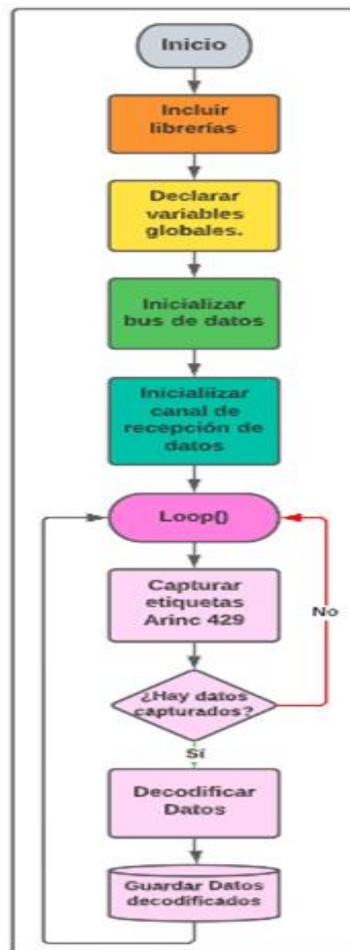


FIGURA 3-5. ESTRUCTURA GENERAL DEL CÓDIGO FUENTE DE LA APLICACIÓN

Luego se declaran las variables globales necesarias para el uso de funciones. Posterior a eso se inicializa el bus de datos, definiendo el puerto de comunicaciones que se utilizará y luego de esto, inicializar el canal de recepción en el que se recibirán los datos. Se ingresa en un loop dónde se capturarán etiquetas ARINC 429.

Si la captura de datos es efectiva, se procede a decodificar la información presente en la palabra ARINC 429, para luego ser almacenada en un archivo de texto, lista para ser transmitida por el módulo LTE hacia internet.

CAPÍTULO 4

4. Resultados y análisis

Para verificar el funcionamiento de la toma de datos, así como la decodificación, se realizaron pruebas en dos distintos escenarios: en el primer escenario obtuvieron los datos de posición (Latitud y longitud) con el GPS estático mientras que en el segundo escenario se validó la detección en tiempo real de las coordenadas con el sistema de posicionamiento global en movimiento.

Las pruebas realizadas en los dos escenarios son de gran importancia para constatar que se pueden detectar datos mientras el helicóptero no se encuentre en constante movimiento y cuando ya se esté desplazando en las costas obteniendo los datos del protocolo ARINC 429 codificándolos y decodificándolos de forma óptica para después transmitirlos hacia la nube y realizando un respaldo de estos.

4.1. Implementación del sistema de comunicación

En primera instancia, se realizó un primer circuito de prueba, con el GPS del helicóptero que es alimentado por una fuente que proporciona un voltaje de 27 [V] para energizarlo.

Cabe recalcar que, para la conexión y comunicación de los dos dispositivos, es necesario conectar de forma correcta los canales de transmisión y recepción de cada uno. Para el GPS se emplean los pines 46 y 47, mientras que en el bus de datos se utilizan los pines 1 y 2 como se muestra en la Figura 4-1. Conectando el pin 46 del GPS al pin 1 del bus de datos y en pin 47 al pin 2 del bus de datos que son los canales 0 de transmisión y recepción.

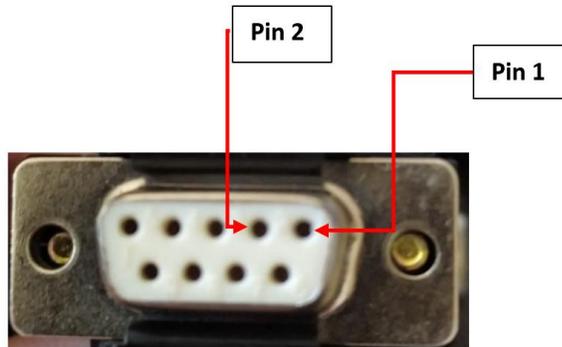


FIGURA 4-1. SERIAL DE COMUNICACIÓN HACIA EL GPS

Adicionalmente, como se requiere conectar el RTX SPA429 a una computadora, se utilizó un adaptador para convertir el protocolo RS-232 que se enlaza con el conector macho del SPA que se muestra en la Figura 4-2, a uno convencional que es el USB, además de que, las señales se encuentran en el rango de voltaje que maneja el serial es de -10[V] a +10 [V], adaptándolo a rango en que trabaja la computadora con el puerto USB.



FIGURA 4-2. CONECTOR SERIAL MACHO

El bus de datos es conectado a una computadora para tener acceso a la programación al software del dispositivo y realizar la programación pertinente basada en el lenguaje c++, en el entorno de Windows mediante el software Visual, debido a que las librerías para inicializar el RTX SPA429, están establecidas con la librería windows.h así como las APIs.

Logrando de esta forma que se encienda el bus de datos de forma interna y que el GPS al detectar que hay un dispositivo conectado al canal ARINC429, envíe datos y comenzar con el procesamiento de los datos realizando la codificación y decodificando los

bits para conocer las etiquetas e imprimirlas por pantalla a través de una computadora para visualizar los datos obtenidos del GPS en tiempo real.

Y a su vez generando un archivo de texto plano en formato .txt, el cual contiene las coordenadas adquiridas para enviarlo a un servidor web empleando tecnología LTE que está conectado a la computadora, como se muestra en el Figura 4-3.



FIGURA 4-3. CIRCUITO ARMADO

4.2 Pruebas de Campo

Para comprobar la veracidad del funcionamiento del circuito propuesto para el sistema de comunicaciones, se procedieron a realizar varias pruebas de campo aplicando diferentes escenarios donde el sistema esta estático y en movimiento, además de realizarlas en diferentes lugares.

Se realizó una simulación con el sistema estático en el laboratorio de electrónica de la entidad pública, con el fin de lograr la obtención de los datos y validar si la información recibida era constante y no tenía variaciones ya que, el sistema no estaba en movimiento.



FIGURA 4-4. PRUEBAS DE CAMPO

En la Figura 4-5, se observa de cerca la pantalla del GPS, en donde se visualizan varias barras de color verde que representan los satélites detectados por la antena del GPS que son en total 8 satélites, siendo como requisito mínimo que se detecten 3 satélites para que el dispositivo de posicionamiento engancharse y el status se encuentre en 3D Navigation, para así transmitir datos, como la posición que se observa en la pantalla marcando como latitud: S 02° 09.77' y en longitud: W 79° 53.39'.



FIGURA 4-5. GPS DEL HELICÓPTERO

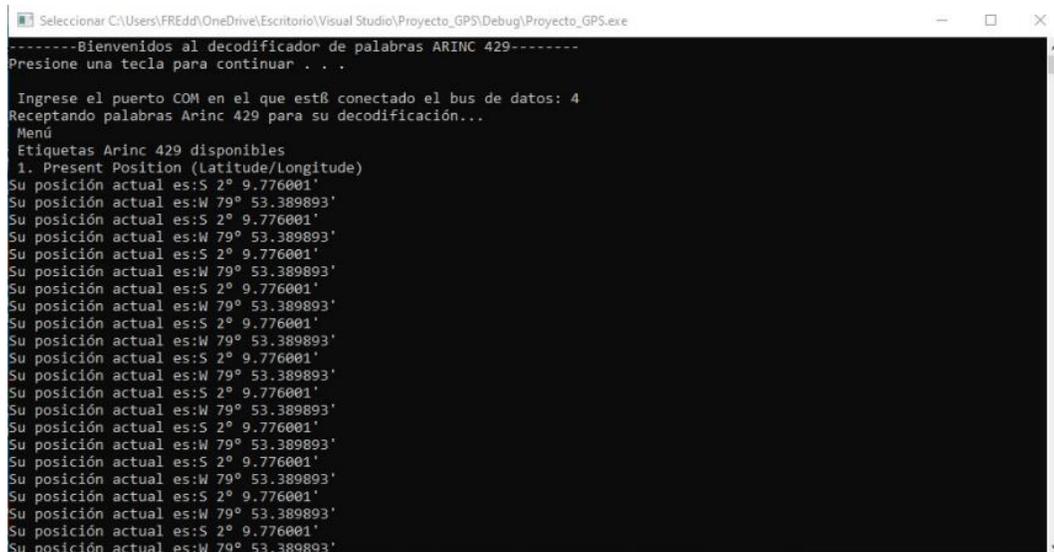
Se encendió el sistema del bus de datos para el tratamiento de la información, ingresando por consola la numeración del puerto del puerto COM, ya que es un parámetro necesario en la función de para inicializar el puerto RS-232, que además tiene configurada la velocidad en unidades de baudios siendo esta de 115.2 Kbaudios.

Por lo consiguiente, se corrió el programa realizado en Visual, que a través de diversas librerías empleadas activar diversos parámetros necesarios, se realiza la detección de la palabra ARINC 429 de 32 bits. En donde se procesan los 8 bits más significativos y codificándolos a numeración octal, ya que es el formato requerido y que de acuerdo con el GPS se tienen en total 255 etiquetas.

En este caso, se obtuvieron datos de latitud y longitud, lo que implica que las etiquetas detectadas son 310 y 311 respectivamente. Además de analizar el bit 29 el cual se puede fue 1 indicando que la coordenada de latitud está dirigida hacia el Sur mientras que la de longitud hacia el Oeste, lo que además decide el tipo de codificación a emplear para lo

datos, siendo en esta simulación BNR obteniendo de esta forma el valor en grados y minutos de cada etiqueta.

En la Figura 4-6, se observan los datos por consola de latitud y longitud, brindando la posición actual del GPS que es S 2° 9.776001' y W 79° 53.389893', en donde se han recolectado la mayor cantidad de decimales para una mejor precisión. Adicional, se observa que los valores fueron constantes debido a que se mantuvo el sistema en la misma posición, lo que indica que la simulación en el primer escenario fue realizada con éxito.



```
Seleccionar C:\Users\FREdd\OneDrive\Escritorio\Visual Studio\Proyecto_GPS\Debug\Proyecto_GPS.exe
-----Bienvenidos al decodificador de palabras ARINC 429-----
Presione una tecla para continuar . . .

Ingrese el puerto COM en el que est  conectado el bus de datos: 4
Receptando palabras Arinc 429 para su decodificaci n...
Men 
Etiquetas Arinc 429 disponibles
1. Present Position (Latitude/Longitude)
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
Su posici n actual es:S 2  9.776001'
Su posici n actual es:W 79  53.389893'
```

FIGURA 4-6. DATOS DE LATITUD Y LONGITUD DECODIFICADOS

Mediante una plataforma de Geolocalizaci n, Google Earth, se ingres  la coordenada de posici n brindada por el GPS, como se muestra en la Figura 4-7, en donde se valida el lugar correcto en donde se est  realizando la prueba.



FIGURA 4-7. COORDENADAS GRÁFICAS DEL GPS

Después de realizar el proceso de obtención y decodificación de datos, se procedió a ingresar las coordenadas en la plataforma de Geolocalización, como se muestra en la Figura 4-8, con el fin de constatar de que los datos adquiridos sean los más cercanos a los brindados por el GPS. A pesar, que los datos conseguidos a través del sistema de comunicaciones implementados tienen mayor cantidad de decimales en las coordenadas, se muestra que se grafica muy cerca a la posición proporcionada por el GPS.



FIGURA 4-8. COORDENADAS GRÁFICAS DECODIFICADAS

En la Figura 4-9, se puede observar las dos coordenadas que se observan en el GPS y las coordenadas que se decodificaron, en donde se puede observar que la diferencia entre los dos puntos es de aproximadamente 10.92 m, esta diferencia se da ya que en la programación realizada se está trabajando con todos los decimales para tener más precisión en los cálculos. La distancia entre los puntos es significativa, sin embargo, el grado de precisión del GPS por el canal ARINC 429 es más alto que lo presentado en pantalla.

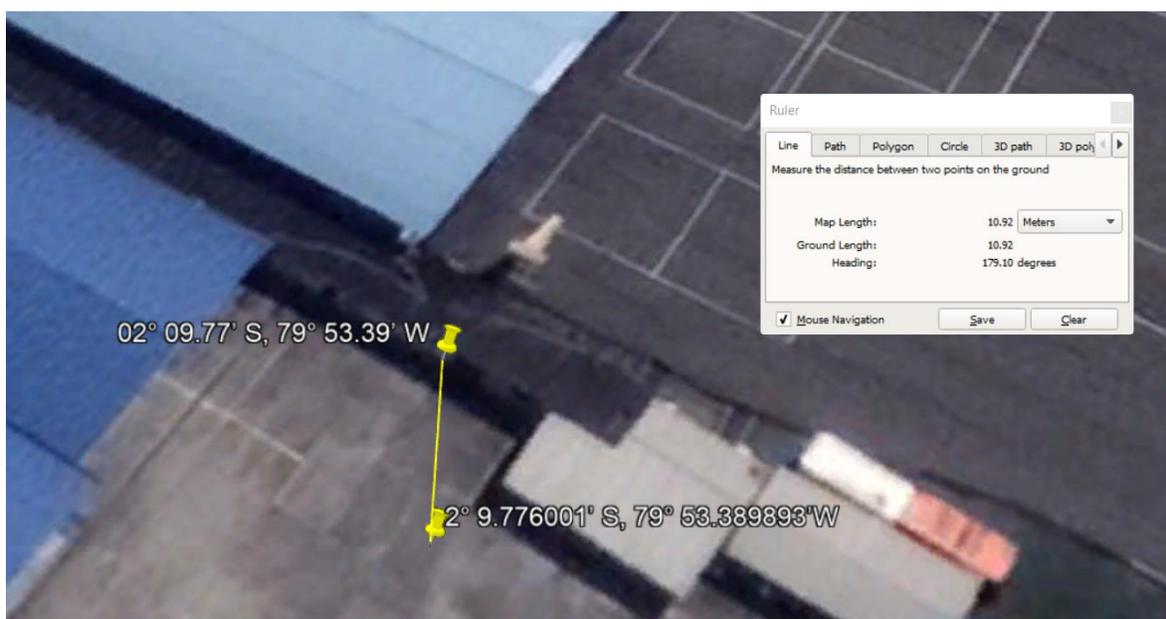


FIGURA 4-9. DISTANCIA ENTRE LAS DOS COORDENADAS

Finalmente, se observa en la Figura 4-10, que se almacenaron los datos obtenidos del GPS en un archivo de texto plano (.txt) que son de latitud y longitud en unidades de grados y minutos, además de observar la dirección de las coordenadas siendo Sur y Oeste respectivamente, lo cual es correcto de acuerdo con la ubicación geográfica del país.

Los datos de posición que se guardaron en el archivo .txt, son exactamente los mismos y repetitivos, porque el sistema estuvo estático cuando se realizó la prueba y por ende el GPS

no mostró ni detectó otras coordenadas. Por otra parte, el archivo fue subido a la nube y respaldado en un servidor.

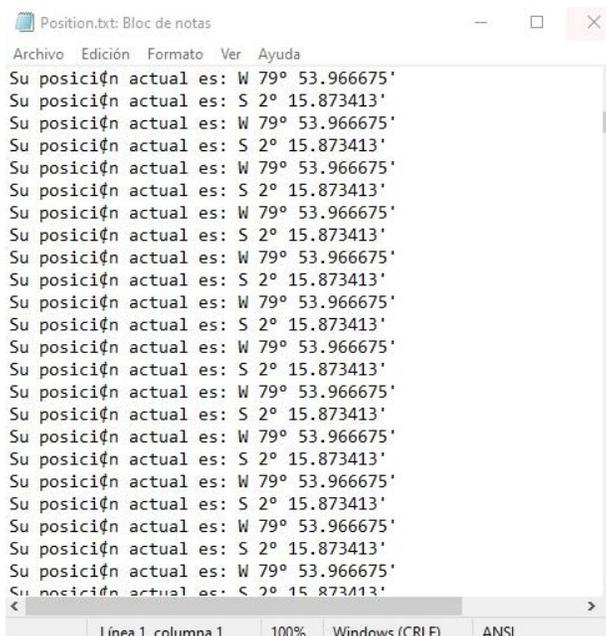


FIGURA 4-10. DATOS EN .TXT

A continuación, se realiza la estimación de errores de los datos obtenidos de manera visual en el GPS y los procesados mediante la decodificación de datos en el canal ARINC 429, ya que existe una diferencia entre los datos como se observa de manera gráfica en la Figura 4-9.

El protocolo ARINC 429 en sus especificaciones técnicas, detalla un valor de resolución para cada etiqueta. En el presente caso, se tratan etiquetas de latitud y longitud, donde la resolución correspondiente es de 0.000172° de acuerdo con el tipo de etiqueta que se obtuvo. [12]

Si se toman en cuenta los datos obtenidos de manera visual y los datos de la información decodificada desde el canal ARINC 429 de una etiqueta de latitud, se puede

calcular el error absoluto como la diferencia positiva entre el valor real (del panel gráfico del GPS) y el valor estimado (decodificado), como se observa a continuación.

$$\textit{Error absoluto} = |\textit{Valor real} - \textit{Valor estimado}|$$

$$\textit{Error absoluto} = |2.16283333^\circ - 2.16293335^\circ|$$

$$\textit{Error absoluto} = 1.0002 * 10^{-4} \text{ }^\circ$$

$$\textit{Error absoluto} = 0.00010002^\circ$$

El valor de resolución de las etiquetas es superior al error absoluto generado producto de la decodificación.

$$\textit{Resolución} > \textit{Error Absoluto}$$

$$0.000172 \text{ }^\circ > 0.00010002 \text{ }^\circ$$

CAPÍTULO 5

5. Conclusiones y recomendaciones

Luego del análisis de resultados realizados en el capítulo anterior, se pueden realizar conclusiones respecto al rendimiento, diseño y aplicaciones del proyecto.

Las conclusiones aquí expuestas tienen sustento en ensayos realizados en pruebas de laboratorio y en campo, las cuales ayudaron a entender de mejor manera el comportamiento del prototipo presentado en los dos distintos escenarios.

Así también, se realizan recomendaciones como retroalimentación a las limitantes presentadas al momento de la implementación del proyecto, con el objetivo de lograr una mejora continua de las interfaces físicas y la optimización de los procesos realizados.

5.1. Conclusiones

El proyecto implementado, en su fase inicial, es un avance significativo en el desarrollo de tecnologías orientadas a comunicaciones aviónicas, ya que como se observa en la Figura 4-6, se logra decodificar la información del protocolo ARINC 429, y el acceso a esta información ayudará a solucionar la problemática planteada en este proyecto.

El prototipo fue sometido a pruebas de campo, dónde se recolectaron mediciones de posición. Estas fueron comparadas con la posición presentada en el GPS por pantalla, como se observa en la Figura 4-9. Se puede apreciar que el error absoluto de una etiqueta de latitud es de 0.00010002° , cuyo valor es menor a la resolución de la etiqueta en el protocolo ARINC 429 que demuestra la alta fiabilidad de la decodificación de los datos.

El manejo de la información obtenida desde el GPS y la presentación por pantalla se deben realizar en fases separadas, ya que, para efectos de visualización, se debe incluir un

retardo en la impresión de datos, mientras se recolectan en tiempo real, esto luego de cierto tiempo generará un desfase en la información recolectada y la presentada en pantalla.

Las etiquetas seleccionadas en el desarrollo de la aplicación de decodificación de datos muestran la posición actual del GPS en latitud y longitud, las mismas fueron escogidas para probar el entendimiento del protocolo ARINC 429 comparando con la información presentada en la pantalla del GPS.

5.2. Recomendaciones

Se recomienda realizar en el desarrollo del programa encargado de decodificar los datos, la optimización de los procesos, ya que así se evitaría que el programa demande más requisitos de hardware que los óptimamente necesarios para realizar su función.

Se debe garantizar una correcta cobertura de señal al GPS, ya que, al tener mala cobertura, deja de actualizar las etiquetas con la nueva información y procede a enviar etiquetas sin datos.

Es importante que los cables empleados en las conexiones del prototipo para el transporte de datos tengan la impedancia correcta, ya que, una impedancia incorrecta generaría errores de bits y, por lo tanto, fallos en el sistema de comunicaciones.

BIBLIOGRAFÍA

- [1] G. Ltd., «GPS 400 Pilot's Guide and Reference,» Garmin International, Inc., Olathe, Kansas 66062 USA, 2009.
- [2] R. S. Inc, «RTX Systems,» RTX Systems Inc, [En línea]. Available: <http://www.rtxsystems.com/SPA.html>. [Último acceso: 1 Junio 2022].
- [3] J. A. Fernandez, «4G LTE, La Nueva Era de las Comunicaciones Tácticas,» Dialnet, 2015.
- [4] L. A. B. B. & E. P. B. Bassante, «Diseño e Implementacion de una Red I2C para Instrumentacion Industrial,» Latacunga, 2005.
- [5] N. G. G. Luna, «Sistema Didactico para comunicacion serial entre dos computadores personales,» Quito, 1992.
- [6] «Desarrollo de una interfaz para la visualizaci[on y adquisicion de datos provenientes de la ECU a través de OBD-II mediante un dispositivo de comunicación serial y del analizador de gases Qrotech 6000,» Cuenca, 2016.
- [7] I. SA, «IEEE STANDARDS ASSOCIATION,» IEEE SA, 16 Junio 1998. [En línea]. Available: <https://standards.ieee.org/ieee/488.1/715/>. [Último acceso: 21 Junio 2022].
- [8] I. S. Board, «IEEE Standard Digital Interface for Programmable Instrumentation,» The Institute of Electrical and Electronics Engineers, Inc, New York, NY 10017, 1988.
- [9] L. S. D. M. Ismael Potolicchio, «Comunicación USB entre un microcontrolador y un dispositivo móvil para telemetría: Primer Parte,» Departamento de Electrónica Facultad de Ingeniería Nacional del Comahue, Buenos Aires.
- [10] I. RTX Systems, «ARINC 429 Serial Port Adapter,» RTX Systems, Inc., Melville, NY 11747, 2014.
- [11] F. J. V. García-Donas, «Diseño e Implementación de un simulador del protocolo ARINC 429 sobre UDP,» Universidad de Sevilla, Sevilla, 2015.
- [12] A. E. E. COMMITTEE, «Mark 33 Digital Information Transfer System (DITS) Part 1 Funtional description, electrical, interface, label assignments and word formats,» AERONAUTICAL RADIO, INC, Annapolis, Maryland , 2004.
- [13] A. I. Gutiérrez, «Sistema aviónico de instrumentación para adquisición de datos y gestión de buses ARINC,» Universidad Politécnica de Catalunya, Catalunya, 2007.
- [14] L. F. C. C, «Modem de 200 Baudios.- Modulador,» Quito, 1980.

- [15] Microsoft, «Visual Studio Code,» 7 Febrero 2022. [En línea]. Available: <https://code.visualstudio.com/docs>. [Último acceso: 21 Junio 2022].
- [16] A. B. Jiménez, «Desarrollo de un Software de Medición del Estrés par aun Dispositivo Foto-pletismógrafo Basado en el Protocolo USB 2.0,» Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México,D.F., 2010.

ANEXOS

```
Decode.cpp  x Proyecto_GPS.cpp
Proyecto_GPS  (Ámbito global)
1  #include <iostream>
2  #include <bitset>
3  #include <cmath>
4
5  // Declaración de variables de retorno
6  double mult;
7  // Prototipo de Función Decode
8  double Decode(long int Data, int signo);
9
10 // Definición de Función Decode
11
12 double Decode(long int Data, int signo) {
13     if (signo < 0) {
14         int databits = (Data >> 10) & 0x3FFFF; // Segmenta la palabra del bit 28 al bit 11.
15         int complemento1 = ~databits; // Complemento a 1 de los datos a decodificar.
16         int complemento2 = complemento1 + 1; // Complemento a 2 de los datos a decodificar.
17
18         int what_bit_i_am_testing = 18; // Contador
19
20         double suma = 0;
21
22         while (what_bit_i_am_testing > 0) {
23             if (complemento2 & 0x01) { // Último bit #1 entra al if
24                 suma += (1 / pow(2, what_bit_i_am_testing));
25             }
26             what_bit_i_am_testing--;
27             complemento2 = complemento2 >> 1; //desplazamiento de 1
28         }
29
30         mult = suma * 180;
31     }
32
33     if (signo > 0) {
34         int databits = (Data >> 10) & 0x3FFFF; // Segmenta la palabra del bit 28 al bit 11.
35
36         int what_bit_i_am_testing = 18; // Contador
37
38         double suma = 0;
39
40         while (what_bit_i_am_testing > 0) {
41             if (databits & 0x01) { // Último bit #1 entra al if
42                 suma += (1 / pow(2, what_bit_i_am_testing));
43             }
44             what_bit_i_am_testing--;
45             databits = databits >> 1; //desplazamiento de 1
46         }
47
48         mult = suma * 180;
49     }
50
51     double grados = mult;
52     return grados;
53 }
```

ANEXO 1. FUNCIÓN DECODE

```
Label.cpp  Proyecto_GPS.cpp
Proyecto_GPS (Ámbito global)
1  #include <iostream>
2  #include "APISPA.H"
3  #include <bitset>
4
5
6  int etiqueta;
7
8  // Prototipo de la función Label
9  int Label(int Word);
10
11 // Definición de la función Label
12
13 int Label(int Word) {
14     if ((Word & 0xff) == 200){
15         etiqueta = 1;
16     }
17     if ((Word & 0xff) == 201) {
18         etiqueta = 2;
19     }
20     int label = etiqueta;
21
22     return label;
23 }
```

ANEXO 2. FUNCIÓN LABEL

```
Sign.cpp x Proyecto_GPS.cpp
Proyecto_GPS (Ámbito global)
1 #include <iostream>
2 #include "APISPA.H"
3 #include <bitset>
4
5 // Declaración de variables de retorno
6 int signo;
7
8 // Prototipo de Función Label
9
10 int Sign(int Word, int label);
11
12 // Definición de Función Label
13
14 int Sign(int Word, int label) {
15     if (label == 1) {
16         if (((Word >> 28) & 0x1) == 1) {
17             signo = -1; // S:Sur
18         }
19         else {
20             signo = 1; // N:Norte
21         }
22     }
23     if (label == 2){
24         if (((Word >> 28) & 0x1) == 1) {
25             signo = -2; // W:Oeste
26         }
27         else {
28             signo = 2; // E:Este
29         }
30     }
31     int salida = signo;
32     return salida;
33 }
```

ANEXO 3. FUNCIÓN SIGN