

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA ELECTRICA



**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA OPTIMIZADO PARA  
LA TRANSMISIÓN DE IMÁGENES USANDO RADIOS HF**

PRESENTADO POR:

**ANA MARIELA MARAVILLA FIGUEROA**

PARA OPTAR AL TITULO DE:

**INGENIERA ELECTRICISTA**

CIUDAD UNIVERSITARIA, MAYO 2021

**UNIVERSIDAD DE EL SALVADOR**

RECTOR:

**MSC. ROGER ARMANDO ARIAS ALVARADO**

SECRETARIO GENERAL:

**ING. FRANCISCO ANTONIO ALARCON**

**FACULTAD DE INGENIERIA Y ARQUITECTURA**

DECANO:

**PhD. EDGAR ARMANDO PEÑA FIGUEROA**

SECRETARIO:

**ING. JULIO ALBERTO PORTILLO**

**ESCUELA DE INGENIERÍA ELÉCTRICA**

DIRECTOR:

**ING. ARMANDO MARTÍNEZ CALDERÓN**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERA ELECTRICISTA**

Título:

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA OPTIMIZADO  
PARA LA TRANSMISIÓN DE IMÁGENES USANDO RADIOS HF**

Presentado por:

**ANA MARIELA MARAVILLA FIGUEROA**

Trabajo de Graduación Aprobado por:

Docente Asesor:

**ING. WERNER DAVID MELÉNDEZ VALLE**

SAN SALVADOR, MAYO 2021

Trabajo de Graduación Aprobado por:

Docente Asesor:

**ING. WERNER DAVID MELÉNDEZ VALLE**

## NOTA Y DEFENSA FINAL

En esta fecha, Jueves 25 de febrero de 2021, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 5:00 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Armando Martínez Calderón  
Director

  
Firma


  
Firma



2. MSc. José Wilber Calderón Urrutia  
Secretario

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:


- ING. WERNER DAVID MELÉNDEZ VALLE  
(Docente Asesor)

  
Firma

- DR. CARLOS OSMIN POCASANGRE JIMENEZ

  
Firma

- ING. WALTER LEOPOLDO ZELAYA CHICAS

  
Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA OPTIMIZADO PARA LA TRANSMISIÓN DE IMÁGENES USANDO RADIOS HF

A cargo de los Bachilleres:

- AMAYA DÍAZ, ORGE ALBERTO

- MARAVILLA FIGUEROA, ANA MARIELA

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.4

( OCHO PUNTO CUATRO )

## **AGRADECIMIENTOS**

Primeramente, agradezco a Dios por haberme dado fuerza y voluntad para continuar en todo este proceso de aprendizaje.

A mi asesor Ing. Werner Meléndez, quien con sus conocimientos y apoyo me guío durante este proyecto.

También a mis padres y hermanos por todos sus esfuerzos, cariño y apoyo, que quienes con su amor y paciencia me han permitido cumplir esta meta.

Por último, a mis amigos y compañeros por todos los buenos momentos y el apoyo mutuo para vencer los obstáculos durante la carrera.

# ÍNDICE

<b>AGRADECIMIENTOS</b> .....	<b>2</b>
<b>OBJETIVOS</b> .....	<b>1</b>
<b>ALCANCES</b> .....	<b>2</b>
<b>ANTECEDENTES</b> .....	<b>3</b>
<b>JUSTIFICACIÓN</b> .....	<b>4</b>
<b>CAPÍTULO 1 - RADIO COMUNICACIÓN HF</b> .....	<b>5</b>
Introducción .....	5
1.1 Funcionamiento general .....	5
1.2 Planteamiento del problema .....	6
1.3 Espectro radioeléctrico .....	7
1.4 Radio enlace en HF .....	8
1.5 Alternativas Evaluadas .....	9
1.6 Sistemas APRS .....	9
1.6.1 Protocolo AX.25 .....	9
1.6.2 TNC .....	10
1.7 SSTV .....	10
1.7.1 Funcionamiento del SSTV .....	11
1.7.2 Modos SSTV y velocidad de línea .....	11
1.8 Tipos de modulación .....	12
<b>CONCLUSIÓN</b> .....	<b>14</b>
<b>GLOSARIO</b> .....	<b>15</b>
<b>BIBLIOGRAFÍA</b> .....	<b>16</b>
<b>CAPÍTULO 2 - EQUIPO PARA APRS y SSTV</b> .....	<b>17</b>
2.1 INTRODUCCIÓN .....	17
2.2 Selección de HARDWARE para APRS y SSTV .....	17
2.2.1 Transceptor Kenwood TS-50 .....	17
2.2.2 Tuner Kenwood AT-50 .....	18
2.2.3 Raspberry Pi .....	19
2.2.4 Interfaz entre Raspberry y Transceptor .....	19
2.2.5 Control PTT .....	20
2.3 SOFTWARE para APRS .....	22
2.3.1 TNC Direwolf .....	22
2.3.2 configuración Direwolf .....	22
2.3.3 configuración de dispositivo de sonido .....	23

2.4 Xastir .....	24
2.4.1 configuración Xastir .....	24
2.4.2 gen_packets .....	26
2.5 Software para SSTV .....	26
2.5.1 Codificador SSTV PYSSTV .....	26
2.5.2 Decodificador QSSTV .....	27
2.6 Prueba de modos SSTV .....	28
<b>BIBLIOGRAFÍA .....</b>	<b>32</b>
<b>CAPÍTULO 3 - FUNCIONAMIENTO DEL PROYECTO .....</b>	<b>33</b>
3.1 Introducción .....	33
3.2 Raspberry como punto de acceso WIFI .....	33
3.3 Servidor APACHE y PHP .....	35
3.4 Aplicación Android .....	36
3.5 Transmisión de los datos obtenidos. ....	39
3.6 Recepción de datos .....	41
CONCLUSIONES .....	44
<b>BIBLIOGRAFÍA .....</b>	<b>46</b>
<b>ANEXOS .....</b>	<b>47</b>



## ÍNDICE DE FIGURAS

Figura 1- Esquema de estación para comunicación por radio.....	6
Figura 2 - TNC-PI .....	10
Figura 3 - Radio Kenwood HF TS-50.....	18
Figura 4 - Tuner Kenwood AT-50 .....	18
Figura 5 - Raspberry pi 3 Model B .....	19
Figura 6 – SoundCard USB.....	19
Figura 7 - PINOUT de conector MIC de Kenwood TS-50S.....	20
Figura 8 - Circuito control PTT.....	21
Figura 9 - Archivo Direwolf.conf .....	22
Figura 10 - configuración. asoundrc .....	23
Figura 11 - Interfaz de Xastir indicando estación de radio UES2.....	24
Figura 12 - Seleccionando TNC para Xastir. ....	25
Figura 13 - Configuración de estación en Xastir.....	25
Figura 14 - Software QSSTV recibiendo una imagen en Robot36 .....	27
Figura 15 – configuración inicial RasAp Webgui.....	33
Figura 16 – Interfaz de configuración de RasAP .....	34
Figura 17 – señal wifi disponible desde Raspberry.....	34
Figura 18 – Pantalla inicial de aplicación de prueba.....	36
Figura 19 - Formulación de la aplicación.....	37
Figura 20 – botones habilitados en la aplicación al tener datos almacenados.....	38
Figura 21 – Fotografía recibida en el servidor de la Raspberry. ....	38
Figura 22 – Datos recibidos del formulario.....	39
Figura 23 – Esquema prueba final.....	39
Figura 24 – Formulario recibido. ....	42
Figura 25 – Fotografía recibida por QSSTV .....	43

## ÍNDICE DE TABLAS

Tabla 1 - Radio Espectro. ....	7
Tabla 2 - Clasificación según longitud de Onda - HF .....	8
Tabla 3 - Frecuencias normalmente utilizadas para SSTV .....	11
Tabla 4 - Algunos modos SSTV comúnmente utilizados. ....	12
Tabla 5 - comparación de formatos SSTV .....	29

## **OBJETIVOS**

General:

- ❖ Proponer un sistema de comunicación basado en radios de HF, con capacidad de transmitir imágenes eficientemente.

Específicos:

- ❖ Analizar alternativas disponibles en la actualidad, que puedan ser aplicadas al presente trabajo.
- ❖ Valorar la construcción y/o adaptación de interfaces que optimicen la transmisión de imágenes a través de canales de comunicación limitados.
- ❖ Proponer la forma en la cual se podrían adaptar, las diferentes opciones encontradas, a los radios de HF disponibles en la EIE.

## ALCANCES

- ❖ Analizar diferentes protocolos y algoritmos asociados a la transmisión de imágenes en medios limitados en ancho de banda y a partir de ello, seleccionar los más adecuados para el presente trabajo, tomando en cuenta los recursos disponibles y criterios de bajo costo.
- ❖ Construir un sistema el cual incluirá un nodo o terminal móvil, que demuestre el funcionamiento del sistema a proponer.
- ❖ Integrar a la transmisión de la imagen, información referente a la posición GPS (Global Positioning System) desde el punto del cual se envía.

## **ANTECEDENTES**

Los canales de comunicación que utilizan radios de HF, se enfocan principalmente en la transmisión de voz, siendo ineficientes como medio de envío de imágenes, no obstante, no se puede ignorar que sus características de propagación, traducidas en su gran cobertura (hasta cientos de kilómetros en algunos casos y bajo ciertas circunstancias), los vuelve una opción importante en situaciones de emergencia. Para contrarrestar la ineficiencia antes mencionada, actualmente es posible encontrar diferentes alternativas (tanto de hardware como de software), que vale la pena estudiar e implementar, para tratar de adaptarlas al entorno de El Salvador, con la intención de mejorar la eficiencia de transmisión de imágenes tanto como sea posible, manteniendo el enfoque de bajo costo.

## **JUSTIFICACIÓN**

La orografía de El Salvador dificulta las comunicaciones vía radio que utilizan frecuencias por encima de 100MHz, volviendo a este tipo de sistemas excesivamente caros (por la infraestructura que se debe montar para cubrir todo el país); sin embargo, las frecuencias de HF se ven afectadas en menor grado por esta condición (obstáculos), a tal punto que con un par de radios se podría cubrir casi la totalidad del territorio nacional sin necesidad de repetidor, de ahí su importancia como medio de transmisión de información en diferentes condiciones, en especial cuando otras alternativas colapsan o fallan

# CAPÍTULO 1 - RADIO COMUNICACIÓN HF

## **Introducción**

La transmisión de información a través de radios de Alta Frecuencia o HF (High Frequency), podría considerarse un tanto discontinuada u obsoleta en nuestros días, si consideramos alternativas como las redes móviles o enlaces de fibra óptica; sin embargo, siguen siendo un medio confiable y robusto, especialmente cuando los medios tradicionales fallan o colapsan por cualquier causa. Aunado a lo anterior, los radios de HF ofrecen un radio de cobertura difícilmente igualable por otros medios, además de ser portátiles, de tal forma que se convierten en una alternativa importante en casos de emergencia por desastres naturales. El diseño de este tipo de radios se enfoca en la transmisión de voz o datos a baja velocidad; no obstante, en la actualidad existen alternativas tanto de protocolos de transmisión como de algoritmos de compresión que podrían mejorar la eficiencia de la transmisión de imágenes a través de este tipo de medios.

## **1.1 Funcionamiento general**

Hoy en día es común la comunicación inalámbrica por medio de la telefonía móvil y el internet, y podría considerarse como obsoleta la transmisión HF, pero lo cierto es que aún es utilizada para fines investigativos, radio afición y recopilación de datos. Además, que un medio de comunicación importante en zonas donde la comunicación telefónica o de internet no logra llegar.

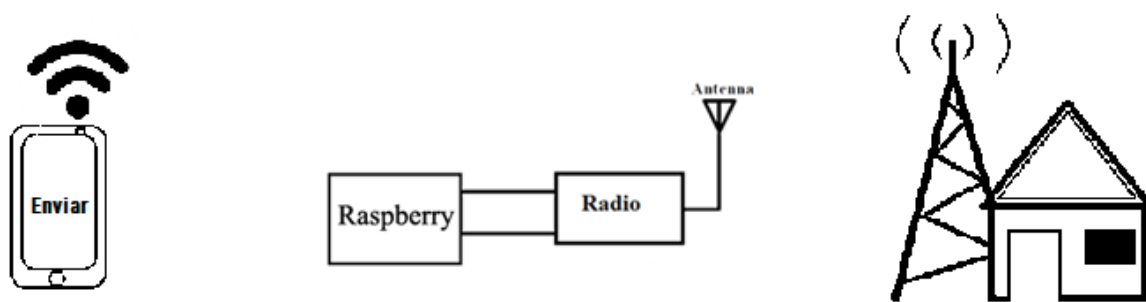
La comunicación HF nos permite transmitir y recibir datos de forma inalámbrica por medio de ondas de radio que usan como medio de transmisión el espacio libre, es capaz de recorrer grandes distancias, incluso es posible a nivel mundial.

## 1.2 Planteamiento del problema

Usualmente en los procesos posteriores a un desastre natural, se incluyen mecanismos de evaluación de la infraestructura asociada a la zona afectada, tomando esto en consideración, se tomó la iniciativa de diseñar un sistema que utilizando el GPS de un celular Android el cual su función será el tomar los datos de posición, los datos de un formulario llenado con información de la estructura y el tomar una fotografía, con la función de que estos datos luego sean enviadas a una estación de radio móvil y este las reenvíe vía radio a una estación base.

Actualmente en El Salvador ingenieros informáticos han diseñado una aplicación llamada SAGIT el cual obtiene la información GPS y de formulario en el celular, y estudiantes de la escuela de ingeniería eléctrica realizará la tarea de diseñar el envío de datos hacia la base móvil y de ahí a la base fija que funcionaría como centro de monitoreo. Este proyecto brindaría una alternativa adicional de comunicación en situaciones de emergencia en el país.

En la figura 1 se muestra el esquema general de comunicación por radio, el equipo está compuesto por el usuario que envía los datos vía WIFI desde el celular hacia la Raspberry, la Raspberry procesara la información y la adaptara su formato para que la radio pueda transmitirla. Este esquema es un ejemplo básico y económico con la que se puede establecer comunicación a través de software para radiocomunicación con la que se puede enviar o recibir datos de la radio.



*Figura 1- Esquema de estación para comunicación por radio.*

En El Salvador se requiere que todo equipo de telecomunicaciones utilizado siga las recomendaciones o estándares de la Unión Internacional de Telecomunicaciones u otras organizaciones internacionales reconocidas y para hacer uso de las frecuencias de radio es necesario tener una licencia en caso contrario la transmisión realizada es ilegal.



Existen clasificaciones internacionales para el uso de las bandas de frecuencias, pero en cada país puede ligeramente variar su uso según se crea más conveniente, y no generar interferencia entre señales. Las más utilizadas para radio son frecuencias HF, VHF y UHF.

### 1.3 Espectro radioeléctrico

La distribución del conjunto de frecuencias radioeléctricas utilizado para las telecomunicaciones se le conoce como espectro radioeléctrico, el cual el Estado se encarga de administrar para los diferentes servicios de telecomunicación en cómo y quién puede usarlas.

Existe una clasificación internacional, pero en cada país puede variar ligeramente los usos según sus necesidades. Las bandas se clasifican:

<b>VLF:</b> <i>Very low Frequency</i>	(3 kHz - 30 kHz), muy baja frecuencia comúnmente utilizado para radionavegación y servicios telefónicos marítimos.
<b>LF:</b> <i>Low Frequency</i>	(30 kHz - 300 kHz), baja frecuencia con características similares a VLF es también utilizado para radionavegación y radiodifusión.
<b>MF:</b> <i>Medium Frequency</i>	(300 kHz - 3 MHz), media frecuencia utilizado comúnmente para radiodifusión en AM.
<b>HF:</b> <i>High Frequency</i>	(3 MHz - 30 MHz), alta frecuencia comúnmente utilizado para radioaficionados y radiodifusión.
<b>VHF:</b> <i>Very High Frequency</i>	(30 MHz – 300 MHz), muy alta frecuencia generalmente utilizada para radioaficionado, televisión abierta, radionavegación.
<b>UHF:</b> <i>Ultra High Frequency</i>	(300 MHz – 3000 MHz), ultra alta frecuencia generalmente utilizada para radioaficionados, telefonía fija y móvil, televisión abierta.
<b>SHF:</b> <i>Super High Frequency</i>	(3 GHz – 30 GHz), súper alta frecuencia generalmente utilizada para comunicación satelital y radio terrestre.
<b>EHF:</b> <i>Extremely High Frequency</i>	(30 GHz – 300 GHz), extremadamente alta frecuencia, son poco utilizadas debido al alto costo del equipo especializado para estas frecuencias.

*Tabla 1 - Radio Espectro.*

## 1.4 Radio enlace en HF

Desde los 3 MHz a los 30 MHz, se le conoce también como onda corta, con longitudes de onda entre 100 m y 10 m. Comúnmente utilizados para aplicaciones por radioaficionados y radiodifusión ya que sus propiedades de propagación permiten comunicación a largas distancias con poca potencia.

Ondas HF se propagan por reflexión en capas de la ionosfera de la Tierra, gracias a ello logra propagarse permitiendo la comunicación incluso en diferentes países del mundo, pero debido a que las capas de la ionósfera dependen también de la radiación solar existe cierta inestabilidad en la señal HF, por lo que tendrá variaciones según la estación del año y la hora del día que se realiza la transmisión.

La radiación solar afecta de diferente forma a las bandas en que se transmite. Para HF tenemos las siguientes:

<b>Banda</b>	<b>Características</b>
80 metros	Su rendimiento es mayor durante la noche, durante el día presenta ruido, pero puede alcanzar distancias hasta de 1000 km.
40 metros	Es una banda con un gran alcance y muy utilizada en especial en horas nocturnas cuando su rendimiento es mayor.
20 metros	Suele ser utilizada durante las 24 horas ya que permite una comunicación a grandes distancias ideales para las comunicaciones a el mundo.
15 metros	Es una banda utilizada durante el día en especial en las horas de máxima actividad solar. Depende de la época del año o de la actividad solar del día, ya que si este es mínimo la banda permanece cerrada.
10 metros	Utilizada únicamente de día, depende de los ciclos solares alcanzando grandes distancias en máxima actividad solar y cerrada en actividad mínima.

*Tabla 2 - Clasificación según longitud de Onda - HF*

## **1.5 Alternativas Evaluadas**

Se propone realizar un sistema que transmita de forma remota con una estación móvil donde el usuario se trasporta hacia infraestructura a evaluar y con su teléfono celular tomar los datos y una fotografía del lugar y desde una aplicación Android instalado en el teléfono celular los envía a un servidor apache en la Raspberry pi de la estación móvil y está realizará el proceso para enviarla al transmisor HF para que llegue a un servidor de la estación base que es una estación fija. Los datos se enviarán en paquetes APRS utilizando el protocolo AX.25 y las imágenes utilizando el método SSTV en Robot36.

- APRS
- SSTV

## **1.6 Sistemas APRS**

APRS de las siglas en inglés Automatic Packet Reporting System se traduce como Sistema Automático de Informe de Paquetes, es un sistema comúnmente utilizado por los radioaficionados que les permite enviar coordenadas GPS, clima, mensajes de texto. Los datos APRS pueden visualizarse en un mapa utilizando un software diseñado para ello.

El sistema APRS utiliza el protocolo AX.25 y necesita de un TNC para la transmisión de la Raspberry hacia la radio.

### **1.6.1 Protocolo AX.25**

Es un protocolo de enlace para establecer conexiones y transferir datos entre canales de comunicaciones para uso de los radioaficionados.

El protocolo AX.25 ofrece servicios en canales de radio dúplex y full dúplex, y puede establecer sin problemas enlaces de punto a punto como de punto a multipunto, o si tiene que pasar por repetidores, así como obtener una conexión fiable ya que tiene la capacidad de recuperación de enlace después de presentar errores de transmisión antes de perder la conexión. Este protocolo maneja el transmisor y el receptor, asignando un indicativo al paquete de cada uno del protocolo.

### 1.6.2 TNC

TNC (Controlador de nodo terminal), su función es convertir las señales binarias del software a señales de audio que la radio pueda transmitir.

Con APRS se pretende enviar mensajes con los datos de texto provenientes de la aplicación, ya que la Raspberry Pi es muy utilizada por los radioaficionados se encuentra en el mercado un hardware TNC especialmente para la Raspberry, con la que en un principio se trabajó en este proyecto para el envío de datos.



*Figura 2 - TNC-PI*

Se consideró la posibilidad de transmitir imágenes utilizando APRS codificando la imagen a base64 y así transmitirla como texto, las pruebas realizadas determinaron que esto no fue factible ya que haría un uso inadecuado del ancho de banda, una transmisión de una imagen necesitaría de mucho tiempo para completarse, en las pruebas realizadas una imagen pequeña de 186 x 173 duraba unos 10 minutos, y una imagen un poco más grande de 320 x 240 podría durar hasta 40 minutos. Por lo tanto, se investigó otros métodos disponibles.

### 1.7 SSTV

Slow Scan Television (SSTV) o Televisión de barrido lento, es un método diseñado para la transmisión de imágenes por radio, estas pueden ser en blanco y negro o a color. SSTV transmite únicamente imágenes estáticas utilizando unos 3 kHz de ancho de banda.

Ya que utilizan frecuencias de voz es utilizada para transmisión en HF, aunque también es posible en algunas frecuencias VHF. Es un método lento que necesita desde unos segundos a unos minutos para transmitir una imagen, normalmente se transmite usando la Banda Lateral Única (SSB), aunque también es posible en modulación FM.

HF (SSB)	3.845 MHz
	7.040 MHz
	14.230 MHz
	14.233 MHz
	21.340 MHz
	28.680 MHz
VHF (FM)	144,500 MHz

*Tabla 3 - Frecuencias normalmente utilizadas para SSTV*

### **1.7.1 Funcionamiento del SSTV**

SSTV es un modo análogo de transmisión, la imagen se compone de líneas escaneadas para ser transmitidas por ondas analógicas, no por píxeles digitales ni con valores de bits, por lo que no utiliza una velocidad de baudios para la transmisión. Hay parámetros de temporización por línea que para transmitirlo lo hace por medio de tonos de audio a través de una tarjeta de sonido a una frecuencia de muestreo.

La transmisión se produce por un escaneo de izquierda a derecha de líneas horizontales, enviando los componentes de color por separado tras cada línea, esta codificación varía entre los diferentes modos. Existen aproximadamente unos 60 modos en el cual pueden transmitir la imagen con parámetros diferentes, diferente tiempo, diferentes dimensiones, a color o blanco y negro.

### **1.7.2 Modos SSTV y velocidad de línea**

La velocidad de transmisión se clasifica por líneas por minuto (lpm), es decir la cantidad de líneas que transfieren en un minuto, esta velocidad depende del modo seleccionado para transmitir.

Existen alrededor de 70 modos para SSTV, a continuación, presentamos las propiedades de los modos más comúnmente utilizados en la transmisión SSTV:

Modo	Color or Black & White	Duración (s)	Resolución (líneas x columnas)	lpm
MartinM1	color	114	256x320	134.395
MartinM2	color	58	259x320	264.553
ScottieS1	color	110	256x320	140.115
ScottieS2	color	71	256x320	216.067
Robot36	color	36	240x320	400.000
Robot72	color	72	240x320	200.000
RobotBW8	B&W	8	160x120	900.000
RobotBW24	B&W	24	320x240	300.000
PasokonP3	color	203	496x640	146.565
PasokonP5	color	305	496x640	97.710

*Tabla 4 - Algunos modos SSTV comúnmente utilizados.*

### **Consideraciones para transmitir en SSTV**

Para transmitir SSTV no se necesita TNC así que no será factible hacer uso de la TNC-pi que se mencionó anteriormente y se buscaron otras alternativas de TNC para el APRS.

Generalmente se utiliza SSTV para transmisión en HF, con modulación AM en la Banda Lateral Única (SSB), aunque también es posible en FM y en algunas frecuencias VHF en FM.

### **1.8 Tipos de modulación**

En la modulación se utilizan dos señales, una es la señal que contiene la información o señal moduladora y la segunda es la señal portadora la cuál es una señal de gran importancia ya que es la podemos manipular según nuestras necesidades de tal manera que se evite lo más posible la perdida de información. Los tipos de modulación más conocidas son AM (modulación por amplitud) y FM (modulación por frecuencia) utilizadas para radiotransmisión.

## **Modulación AM**

La modulación AM fue el primer método para la radiotransmisión, llamado modulación por amplitud o amplitud modulada (AM), una onda modulada se puede descomponer en tres partes: una de frecuencia portadora y dos bandas laterales que una es el resultado de sumar y la otra de restar la frecuencia de la señal moduladora y de la portadora.

Por lo tanto, cada señal portadora, y bandas laterales consumen un porcentaje de la potencia, por ello para un mejor aprovechamiento o ahorro de la potencia, la señal AM evoluciona en otros tipos de modulación como:

Modulación en banda lateral única (SSB), recomendada para SSTV, es una señal con la portadora y una de las bandas laterales suprimidas ya sea la superior o la inferior.

- Banda lateral superior (USB), modulación con la que se realizaron pruebas.

- Banda lateral inferior (LSB).

## **Modulación FM**

Frecuencia Modulada (FM) es el proceso de modulación en el que varía la frecuencia de la onda portadora, es la técnica de modulación más popular en nuestros días.

Con FM se obtiene un mejor aprovechamiento del ancho de banda que en AM con menos potencia, generalmente se utiliza en transmisiones analógicas, pero también es posible transmitir datos digitales por medio del desplazamiento de onda de frecuencia.

## CONCLUSIÓN

- La TNC puede ser hardware o en software, en nuestro caso haremos uso del software, lo que presenta un ahorro económico y de espacio.
- La transmisión HF puede tener desventajas en disponibilidad comparado con otros medios de comunicación por su dependencia a las condiciones atmosféricas, sin embargo, aun así, es una buena opción al ser un sistema robusto que logra vencer obstáculos que se encuentran en la trayectoria de la transmisión, alcanzando largas distancias disminuyendo la necesidad de repetidores.



## GLOSARIO

**Análogo:** *señal*. Es un tipo de señal que tiene una variación continua, representadas por una función matemática en la que sus variables son la amplitud y el periodo en función del tiempo.

**Android:** *sistema operativo*. Es un sistema operativo de código abierto para dispositivos móviles con pantalla táctil, es basado en Linux.

**Frecuencia:** En un evento periódico es el número de repeticiones por una unidad de tiempo.

**Inalámbrico:** *Dispositivo*. Que no necesita cables para establecer comunicación con otro dispositivo, lo que permite hacer transferencia de datos y recibirlas a distancia de la base, por medio de ondas de radio.

**Parámetros:** Elemento o característica con la cual se define o se clasifica para identificar o examinar un tema o sistema.

**Radioafición:** Es un servicio de la radiocomunicación definido por las reglas de la Unión Internacional de Telecomunicaciones, en la cual los radioaficionados deben demostrar conocimientos de teoría y práctica de radio para obtener permiso para el acceso a bandas de frecuencias específicas para radioaficionados y ser utilizadas en intercomunicación e investigaciones técnicas.

**Radiodifusión:** Es el servicio de emitir señales de radio que están destinadas a uso público.

**Remoto:** Que cierto sujeto y objeto se encuentra muy lejos.

## BIBLIOGRAFÍA

- [1] William A. Beech, Douglas E. Nielsen, Jack Taylor (1994). *AX.25 Link Access Protocol for Amateur Packet Radio*.
- [2] Sitio web: Radiopaquete  
[https://javiervalcarce.eu/amsat/radiopaquete.htm#Nivel\\_de\\_enlace](https://javiervalcarce.eu/amsat/radiopaquete.htm#Nivel_de_enlace)
- [3] sstv-handbook.com  
<https://www.sstv-handbook.com/download/sstv-handbook.pdf>
- [4] Libro de texto: Jose M. Hernando Rabanos, 1º ED. *Transmisión por Radio*.
- [5] Sitio web: Superintendencia General de Electricidad y Telecomunicaciones  
[www.siget.gob.sv](http://www.siget.gob.sv)
- [6] Ética y procedimientos operativos para radioaficionados  
<https://www.lu4aa.org/wp/wp-content/uploads/2018/05/iaru-etica-y-procedimientos-operativos-para-radioaficionados.pdf>
- [7] Dirección General de protección civil y emergencias – España : Propagación HF  
<http://www.proteccioncivil.es/catalogo/carpeta02/carpeta24/vademecum17/vdm039.htm#03909a>

# **CAPÍTULO 2 - EQUIPO PARA APRS y SSTV**

## **2.1 INTRODUCCIÓN**

En el mercado podemos encontrar diferentes opciones para realizar la transmisión SSTV, softwares gratuitos para Windows y Linux, aunque también ya hay software disponible para Android. Así como también encontramos con diferentes interfaces para que desde el software de la computadora se acoplen a la radio y realizar la transmisión.

## **2.2 Selección de HARDWARE para APRS y SSTV**

Se toma en cuenta que en el mercado se encuentran a la venta equipos para radio comunicación cada vez más avanzados a lo largo de los años, con mayor capacidad de potencia de transmisión entre otras funciones, sin embargo, para este proyecto se hará uso del equipo disponible en la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador.

### **2.2.1 Transceptor Kenwood TS-50**

Este modelo trabaja con frecuencias HF, tiene una potencia máxima de salida de 100 W, puede ajustarse a 50 W y 10 W. Con un buen rendimiento de transmisión/recepción con los modos de modulación integrados: LSB, USB, CW, AM y FM.

Con una banda de operación para radio afición de 160 m – 10 m y una cobertura generalmente de 500 kHz y 30 MHz. Posee un punto de interceptación avanzado (AIP) con el que mejora el rendimiento de recepción disminuyendo el nivel de ruido, con una facilidad de uso en su menú de ajustes incluso desde su micrófono.

Con un sintetizador digital directo (DDS) con control lógico por pasos, es capaz de sintonizar desde pasos de 5 Hz hasta un modo de avance rápido. Así el operador también tiene la opción de ajustar el torque: Alto para uso móvil, y Bajo para estaciones fijas.



*Figura 3 - Radio Kenwood HF TS-50*

### **2.2.2 Tuner Kenwood AT-50**

Sintonizador de antena externo opcional para la Radio Kenwood TS-50, en este caso lo usaremos ya que nos brinda una sintonización precisa hasta en los bordes de la banda, ya que adapta la impedancia de la línea de transmisión hasta un valor adecuado de impedancia para el equipo de radio.

Al iniciar presionando la tecla AT-TUNE en la radio TS-50, los datos de la banda se transfieren automáticamente al sintonizador por lo que no es necesario realizar ajustes directamente con el Tuner.



*Figura 4 - Tuner Kenwood AT-50*

### 2.2.3 Raspberry Pi

La Raspberry tendrá la función de ejecutar el software que codificara los datos que llegaran al transceptor, además de hacer la función de punto de wifi (sin internet) y de esta forma el celular Android pueda conectarse enviar los datos recopilados hacia un servidor dentro de la Raspberry



*Figura 5 - Raspberry pi 3 Model B*

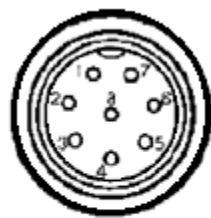
### 2.2.4 Interfaz entre Raspberry y Transceptor

Ya que no sé hará uso del TNC-PI deberemos diseñar nuestra interfaz para que la radio reciba las señales de audio, para ello necesitamos una tarjeta de sonido para la Raspberry, esta puede ser una USB que cumpla con estas funciones de speaker y micrófono.



*Figura 6 – SoundCard USB*

La interfaz permitirá a las señales de audio que salgan del speaker de la tarjeta de sonido de la Raspberry entre al pin TXD del transceptor (ver figura 7) y viceversa, el audio del transceptor (RXD) entre por el micrófono de la tarjeta de sonido.



Visto desde  
el frente

No.	Denominación del Pin
1	TXD
2	PTT
3	MIC (DOWN)
4	MIC (UP)
5	MIC 8V
6	RXD
7	MIC (GND)
8	GND

Figura 7 - PINOUT de conector MIC de Kenwood TS-50S. Ref: Manual Kenwood TS-50

### 2.2.5 Control PTT

PTT siglas en inglés de Push to Talk (pulsar para hablar) es un método de comunicación halfduplex en el que consiste en que al presionar un botón podemos transmitir nuestra voz en el micrófono de la radio y al soltar el botón dejamos libre para recibir.

Para en caso de transmisión APRS la TNC-PI puede tener la función de activar y desactivar el PTT el que consiste en realizar un corto circuito con el pin PTT de la transceptor y tierra durante el tiempo de la transmisión. En este caso se utilizarán los pines GPIO de la Raspberry para el control del PTT, el pin 25 de la Raspberry irá a la base de un transistor que al activarse cerrará el circuito en que dejará fluir 5V del pin 4 para activar un relé que realizará el corto del PTT del transceptor.

Para el control de los pines GPIO utilizamos un código Python que viene por defecto en la Raspberry pi, con el que establecemos que pin se quiere utilizar como salida:

```
import RPi.GPIO as GPIO;
GPIO.setmode(GPIO.BCM);
GPIO.setup(25, GPIO.OUT);
GPIO.output(25, True);
```

Mientras el pin permanezca activo estará a 3.3V, para apagarlo se utiliza el código siguiente:

```
GPIO.output(25, False)
```

La activación del PTT debe estar sincronizado con la salida de audio de la tarjeta de sonido, el PTT debe activarse unos segundos antes de la salida de audio entre 25ms y 100 ms y este debe desactivarse unos segundos después de que la salida de audio haya terminado, entre 10 ms y 20 ms.

Se muestra el circuito del control PTT (Ver figura 8), básicamente cuando el pin GPIO25 se active el relé cerrará el switch del PTT provocando la circulación de corriente desde el pin 2 del transceptor (ver figura 7) a tierra, mientras que el audio se transfiere por las líneas desde el speaker de la soundcard hasta el mic del transceptor.

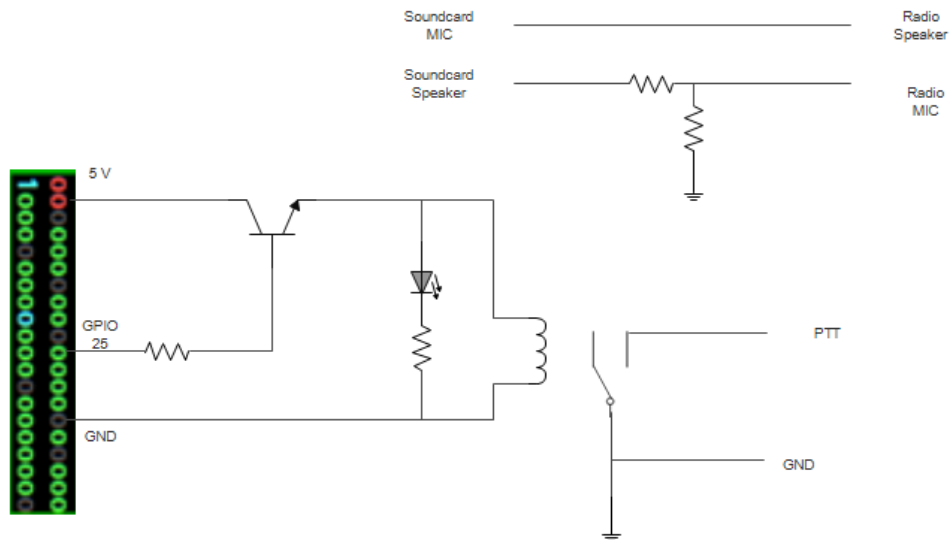


Figura 8 - Circuito control PTT.

## 2.3 SOFTWARE para APRS

### 2.3.1 TNC Direwolf

La TNC se encarga de codificar y decodificar los datos generados por Xastir en APRS, las TNC pueden ser hardware que se conectan directamente al radio, pero también los hay virtuales como es el caso del software Direwolf. Este software puede ser utilizado en Windows y Linux.

Direwolf es un software TNC de paquetes AX.25 que funciona con la tarjeta de sonido de la computadora utilizando ALSA, permite configurar la velocidad en baudios que representa la cantidad de símbolos por segundo en una transmisión digital siendo las más comunes 300 y 1200 baudios utilizadas para la transmisión APRS, esta configuración dependerá del transceptor a utilizar.

### 2.3.2 configuración Direwolf

La configuración de Direwolf se realiza en el archivo direwolf.conf que se encuentre en la carpeta /home/pi al abrir el archivo con el editor de preferencia se encontrará con las opciones de configuración que ofrece Direwolf, la mayoría se encuentra como comentarios con un # al inicio de la línea, por lo tanto, para que un cambio tenga efecto se debe remover el numeral antes de guardar los cambios.

```
#####
#                                     #
#           FIRST AUDIO DEVICE PROPERTIES           #
#           (Channel 0 + 1 if in stereo)           #
#                                     #
#####

#
# Many people will simply use the default sound device.
# Some might want to use an alternative device by choosing it here.
#
# Linux ALSA is complicated. See User Guide for discussion.
# To use something other than the default, generally use plughw
# and a card number reported by "arecord -l" command. Example:

# ADEVICE plughw:1,0
# ADEVICE plughw:1,0

# Starting with version 1.0, you can also use "-" or "stdin" to
# pipe stdout from some other application such as a software defined
# radio. You can also specify "UDP:" and an optional port for input.
# Something different must be specified for output.

# ADEVICE - plughw:1,0
# ADEVICE UDP:7355 default
```

Figura 9 - Archivo Direwolf.conf



Entre las configuraciones importantes para su uso está la tarjeta de sonido -ADEVICE plughw:1,0 - ya que se utiliza ALSA la tarjeta de sonido por defecto no se podrá compartir con otros dispositivos, para ello se realizó configuración en la tarjeta de sonido que se mostrará más adelante. También en direwolf.conf se configurará los baudios a 300 que son los que comúnmente se utilizan para transmisión HF, el control PTT nos da la opción para activarlo usando un pin de Raspberry en nuestro caso el pin 25.

### 2.3.3 configuración de dispositivo de sonido.

Para poder compartir el dispositivo de sonido entre más de una aplicación haremos uso del plugin dmix y dsnoop para la salida y entrada de sonido respectivamente, para ello editamos el archivo “.asoundrc” que se encuentra en /home/pi con el editor de texto de preferencia, y conociendo nuestro dispositivo agregamos lo siguiente:

```
pcm.dmixed {
    type dmix
    ipc_key 1024
    ipc_key_add_uid 0
    slave.pcm "hw:1,0"
}
pcm.dsnooped {
    type dsnoop
    ipc_key 1025
    slave.pcm "hw:1,0"
}

pcm.asymed {
    type asym
    playback.pcm "dmixed"
    capture.pcm "dsnooped"
}
|
pcm.usb {
    type plug
    slave.pcm "asymed"
}
```

Figura 10 - configuración. asoundrc

En este caso nuestro dispositivo es “hw:1,0” pero para utilizar los plugin dmix y dsnoop utilizaremos el pcm “usb” como dispositivo, por lo tanto, en direwolf la configuración de dispositivo será: ADEVICE usb.

## 2.4 Xastir

Xastir es un software de código abierto compatible principalmente con sistemas operativos base Linux. Xastir está dedicado a trabajar con APRS con una interfaz que nos proporciona un mapa mundial en el que nos permite marcar nuestra ubicación y la de otras radios cuando recibe sus datos de coordenadas.

El software envía los datos APRS a la TNC para que este después de convertirlos a señales binarias son enviados a la radio, igual el proceso inverso. Además de nuestra posición podemos mandar mensajes de texto o email si se cuenta con acceso a internet y realizar alertas sobre el clima.

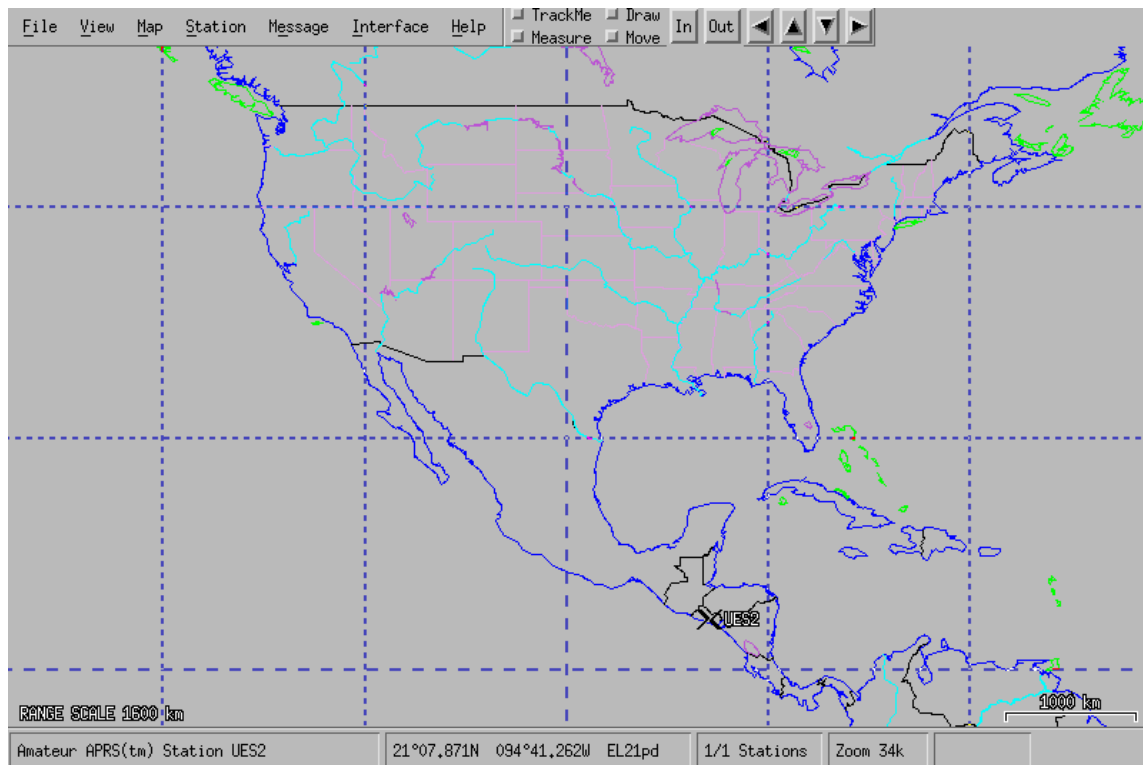


Figura 11 - Interfaz de Xastir indicando estación de radio UES2

### 2.4.1 configuración Xastir

Con el equipo y software mencionado se dispone de dos estaciones de radio como la que se muestra en la Figura 1, una es utilizada como transmisor y otra como receptor y viceversa.

En las Raspberry se deberá configurar la TNC en Xastir para que el software pueda reconocerlo. En la interfaz de Xastir se da clic en barra de menú a Interface>Interface control, se abre una ventana y seleccionamos el botón “Add”, se abre otra ventana donde se muestra

un listado de TNC y otros equipos, en nuestro caso seleccionamos Networked AGWPE. Después de seleccionarlo se abre otra ventana más donde podemos realizar las configuraciones necesarias:

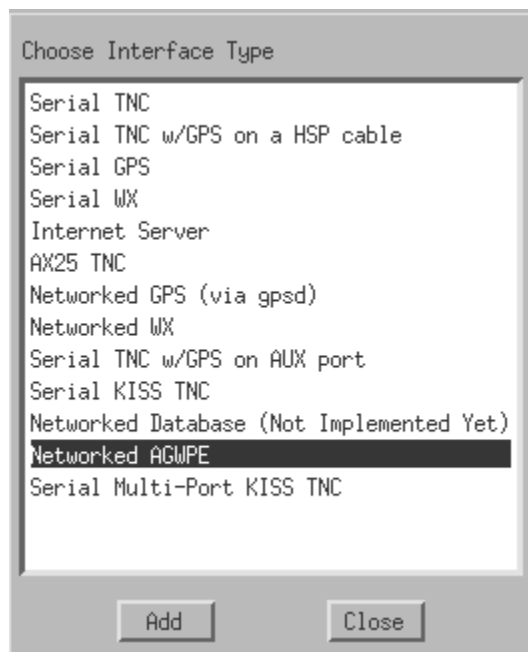


Figura 12 - Seleccionando TNC para Xastir.

Luego configuramos nuestra estación, con el Callsign podemos identificarnos en el mapa, y utilizar el callsign de otras estaciones para podemos comunicarnos directamente.

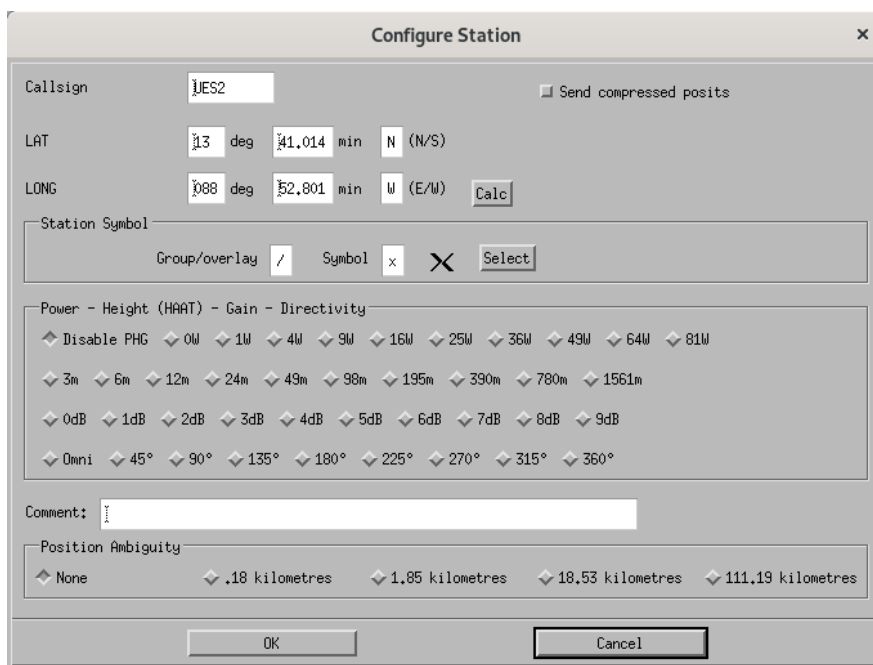


Figura 13 - Configuración de estación en Xastir.

Xastir nos proporciona un número para callsing, con el siguiente comando:

*Callpass < callsing >*

Este número es necesario para poder enviar datos desde la consola, para ellos también debemos conocer el puerto 2023 de Xastir podemos observar las acciones de Xastir desde consola, y decidir que queremos ver. Para enviar mensajes desde la consola utilizamos la línea de código:

*xastir\_udp\_client localhost2023 < callsing > < passcode > -to\_rf -to\_inet*

#### **2.4.2 gen\_packets**

Direwolf incluye una aplicación para pruebas llamada gen\_packets, esta aplicación convierte texto en paquetes AX.25 desde la consola de comandos, esta aplicación crea un archivo de audio WAV con los parámetros que se necesiten, por ejemplo:

MYCALL>WORLD: TEXTO | gen\_packets -a 50 -o namefile.wav -B 300 -

Se observa que podemos configurar nuestro callsing y el texto que queremos codificar, -a 75 representa la amplitud de la señal este se encuentra en un rango entre 0 y 200% que por defecto codifica en un 50%, con -o namefile.wav creamos el archivo de audio WAV con el texto codificado, con -B baudios, establecemos el parámetro de baudios en 300, 1200 o 9600.

### **2.5 Software para SSTV**

Actualmente existen diversos programas disponibles para SSTV, y aunque en su mayoría son gratuitos o de software libre, el sistema operativo en el cual se utilizan puede no serlo.

#### **2.5.1 Codificador SSTV PYSSTV**

PYSSTV es un código en python creada por Andrés Veres-Szentkirályi con licencia MIT, este código genera archivos de sonido WAV codificados por un modo SSTV seleccionado, este puede codificar imágenes PNG, JPEG, GIF, entre otras.

Algunos de los modos disponibles para codificar con este código son los siguientes: MartinM1, MartinM2, ScottieS1, ScottieS2, Robot36, PasokonP3, PasokonP5, PasokonP7, PD90, PD120, PD160, PD180, PD240, Robot8BW, Robot24BW.

Para este proyecto se decidió utilizar Robot36 ya que transmite imágenes a color a un tiempo y tamaño de imagen (320x240) aceptable, es además de ser uno de los modos más utilizados habiendo incluso algunos software SSTV dedicados únicamente a Robot36.

### 2.5.2 Decodificador QSSTV

QSSTV es un software para transmitir y recibir SSTV en Linux, en este caso lo utilizaremos únicamente para recibir, se debe seleccionar la tarjeta de audio a utilizar y el driver ALSA o PULSE AUDIO, en este caso utilizaremos ALSA al igual que la TNC Direwolf, paralelamente se debe realizar ajustes en el volumen este no debe ser tan alto que pueda dañar el transceptor o que genere interferencia ni tan bajo que dificulte la recepción de la señal.

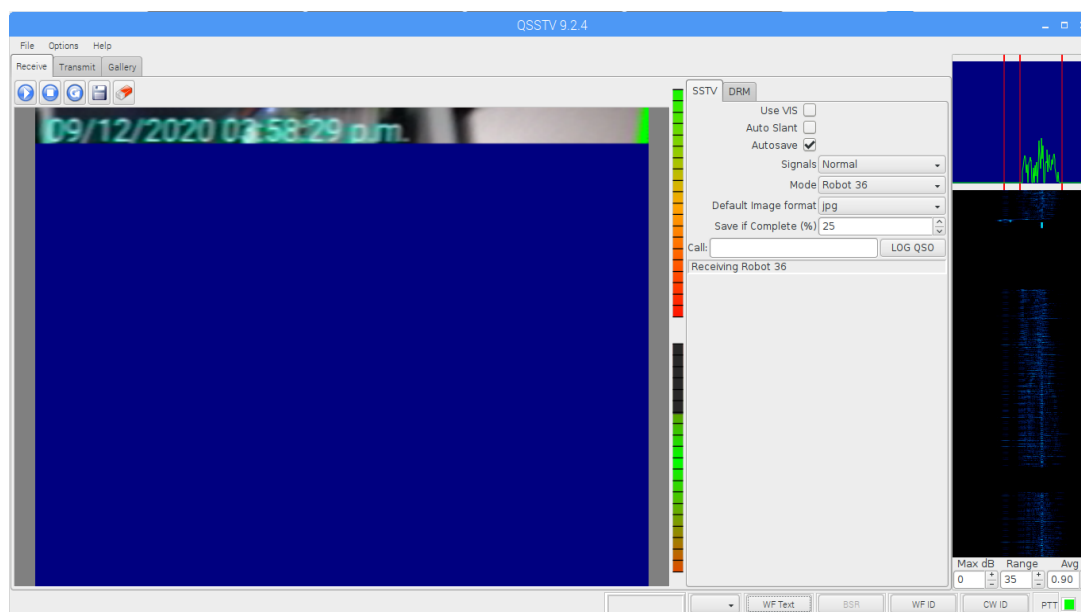

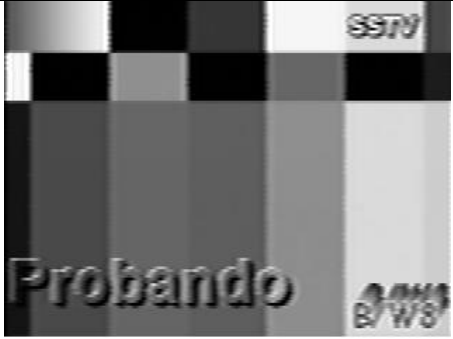






Figura 14 - Software QSSTV recibiendo una imagen en Robot36

## 2.6 Prueba de modos SSTV

Entre los modos más conocidos tenemos Martin1, Martin2, Robot36, Robot72, Scottie1 y Scottie2. Buscamos un modo en el cual haya una calidad y tiempo aceptable. Se realizó pruebas con estos formatos en condiciones sin pérdida de transmisión, se obtuvo los siguientes resultados:

Mode	Original a codificar	Recibida decodificada	Tiempo
B/W8			8s
Martin 1			122s
Martin 2			106s

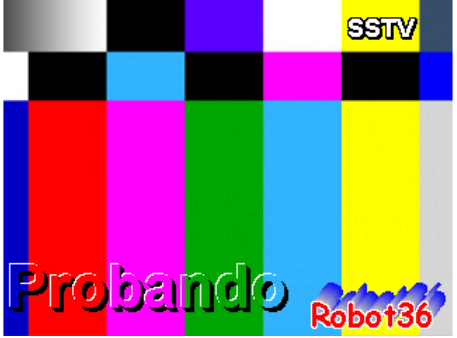

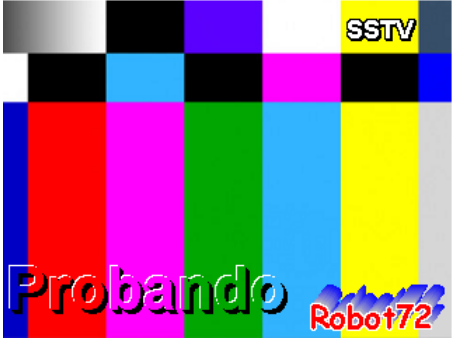





Robot36			38s
Robot72			75s
Scottie1			112s
Scottie2			78s

Tabla 5 - comparación de formatos SSTV

La calidad de la imagen disminuye durante el proceso de codificar y decodificar, sin embargo, las imágenes conservan la claridad. Para este proyecto elegiremos la Robot36, que de los modos listados a color es la que tiene la codificación más rápida.

## CONCLUSIÓN

- Para la transmisión de SSTV y APRS no es necesario invertir un alto costo, ya que en nuestros días no dependemos de equipos extra especializados, ya que en su lugar podemos obtener buenos resultados utilizando software libre, conectando la radio al PC con una tarjeta de sonido.
- Existen alrededor de 70 de modos SSTV, sin embargo, no todos están disponibles en todos los softwares.
- El tiempo de codificación y transmisión de SSTV puede depender del software o código que se utiliza.



## GLOSARIO

**Código:** el código es un sistema de reglas o elementos que se combinan siguiendo un orden y estos deben ser semánticamente interpretables, para que permita intercambiar información entre dos o más puntos, siempre y cuando conozcan el mismo código.

**Codificar:** Es el procesamiento de información a un sistema de reglas definido, de tal modo que no pueda ser interpretada por ningún otro que no conozca el sistema de reglas utilizado.

**Decodificar:** Es el proceso inverso a codificar, en el cual luego de recibir información codificada se realiza el proceso para interpretar y obtener la información a su forma original.

**Hardware:** Hace referencia a la parte física de todo equipo tecnológico o informático.

**Sistema operativo:** Es un conjunto de programas que son capaces de manejar el hardware de una computadora y así como almacenamiento de información y los diferentes periféricos o recursos del sistema.

**Software:** Hace referencia a la parte lógica de un dispositivo, es la parte del sistema no físico, programas con la que es posible realizar tareas específicas.

**Software libre:** Hace referencia a que un software es de uso libre para el público, tanto como para ejecutarlo, distribuirlo, estudiarlo, modificarlo o mejorarlo.

## BIBLIOGRAFÍA

- [1] Sitio web: github.com  
<https://github.com>
  
- [2] Sitio web: PYSSTV  
<https://pypi.org/project/PySSTV/>
  
- [3] Sitio web: Xastir APRS Program  
[https://xastir.org/index.php/Main\\_Page](https://xastir.org/index.php/Main_Page)
  
- [4] sstv-handbook.com  
<https://www.sstv-handbook.com/download/sstv-handbook.pdf>
  
- [5] Direwolf User guide  
<https://raw.githubusercontent.com/wb2osz/direwolf/master/doc/User-Guide.pdf>
  
- [6] Direwolf Raspberry Pi Packet TNC  
<https://github.com/wb2osz/direwolf/blob/master/doc/Raspberry-Pi-APRS.pdf>
  
- [7] Sitio web: Raspberry  
[www.raspberrypi.org/forums](http://www.raspberrypi.org/forums)
  
- [8] Sitio web: SSTV Site  
<http://www.g0hwc.com>
  
- [9] Sitio web: QSSTV Manual  
<http://users.telenet.be/on4qz/qsstv/manual/>
  
- [10] Manual de instrucciones Kenwood TS 50S  
<http://www.hamanuals.com/MMans/mants50s.pdf>

# CAPÍTULO 3 - FUNCIONAMIENTO DEL PROYECTO

## 3.1 Introducción

La Geolocalización de fotografías nos ayuda a tener documentado las coordenadas del lugar en donde se ha tomado la foto, el cual puede ser utilizado en diferentes aplicaciones como en una emergencia siendo este el caso se debe poder utilizar sin necesidad de internet.

Tomando en cuenta una examinación de daños después de una emergencia en la que los servicios de telefonía e internet no estén disponibles y sea necesario que un inspector de daños se presente a un lugar de desastre, es importante que el equipo con el que se movilice en persona sea ligero como un teléfono celular, y disponga el equipo para transmitir dentro de un vehículo.

Se hará una breve explicación del hardware y software con el que se realizará el proyecto, este consiste en una Raspberry Pi que contiene el software para codificar los archivos tanto como SSTV y APRS, este último utilizará a Xastir y una TNC virtual como Direwolf que codificará los datos para ser enviados a través de la tarjeta de sonido USB hacia la radio transmisor HF.

## 3.2 Raspberry como punto de acceso WIFI

Se configuro a la Raspberry Pi 3 model B como punto de acceso WIFI utilizando el asistente de código abierto RasAp Webgui. Por defecto RaspAP inicia con la siguiente configuración:

- IP address: 10.3.141.1
  - Username: admin
  - Password: secret
  
- DHCP range: 10.3.141.50 to 10.3.141.255
- SSID: rasp`i`-webgui
- Password: ChangeMe

*Figura 15 – configuración inicial RasAp Webgui*

Estos valores pueden ser cambiados si así se desean, para ellos abrimos un navegador web donde introducimos la IP address o bien puede colocarse localhost, con el puerto 8080. Nos pedirá el username y la contraseña por defecto, entonces se abre la página de configuración.

En este caso dejaremos la contraseña y username por defecto, y nos aseguramos que la IP address sea una IP estática:

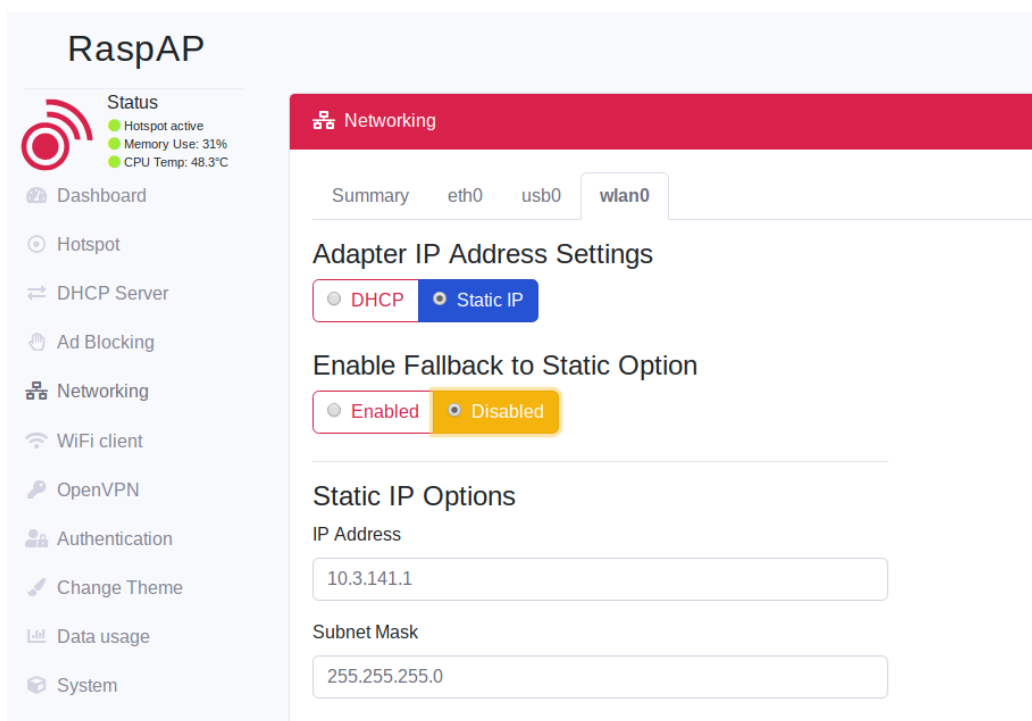


Figura 16 – Interfaz de configuración de RasAP

Verificamos que la red raspbi-webgui está disponible y que es posible conectarse con el teléfono móvil.



Figura 17 – señal wifi disponible desde Raspberry

### 3.3 Servidor APACHE y PHP

Podemos acceder a las páginas del servidor desde un navegador con la IP estática que configuramos en RaspAp o con localhost. El servidor apache funciona por defecto con el puerto 80 por lo que no entra en conflicto con la página de configuración de RaspAP ya que funciona en el puerto 8080.

Para este proyecto no utilizaremos HTML únicamente PHP ya que únicamente recibirá datos de una aplicación Android que se utilizará como interfaz del usuario.

Comenzamos con index.php, esta página php se encargará de recibir los datos de formulario y longitud y latitud, además escribir los datos recibidos en un archivo txt.

El código utilizado fue el siguiente:

```
<?php
$datos=$_GET;
$lugar=$datos["lugar"];
$uso=$datos["uso"];
$numero=$datos["numero"];
$colapso=$datos["colapso"];
$ventana=$datos["ventana"];
$falla=$datos["falla"];
$latitud=$datos["latitud"];
$longitud=$datos["longitud"];
#$comentario=$datos["comentario"];

$form="$lugar;$uso;$numero;$colapso;$ventana;$falla;$latitud,$longitud";
$info= fopen("formulario.txt","w");
fwrite($info,$form);
fclose($info);
¿>
```

Este código obtiene las variables que se asignaron en la URL de la aplicación, utilizando el método GET al enviar las variables y los datos en la URL, éstas son recibidas en el servidor, almacenando únicamente las respuestas.

Para la recepción de imágenes se diseñó un código en una página diferente llamada foto.php, el código es el siguiente:

```
<?php
$datos=$_POST;
$contentido=$datos['contenido'];
$nombre=$datos['nombre'];

$plainText=base64_decode(str_replace(array('-','_',' ','\n'), array('+',
'/','+',' '), $contentido));
$ifp = fopen( "foto.jpg", "wb" );
fwrite( $ifp, $plainText );
fclose( $ifp );

exec("sudo ./tx.sh");
?>
```

En este caso se utiliza el método POST por lo que el código base64 no se incluye en la URL, además el dato recibido no se guarda en un archivo txt si no que aparece en formato jpg en el servidor después de ser recibido.

Con el comando *exec* se ejecuta el script bash tx.sh después de recibir la fotografía en el servidor. En este script se encuentra la secuencia para la codificación y la transmisión de los archivos recibidos en apache.

### 3.4 Aplicación Android

Se realizó una aplicación de prueba que obtiene los datos de latitud y longitud, el botón enviar esta deshabilitado mientras no detecte que se ha completado un formulario o tomado una fotografía.



Figura 18 – Pantalla inicial de aplicación de prueba.

A modo de ejemplo completamos un formulario:

---

**Formulario**

**Lugar:**

**Uso de estructura:**

<input type="checkbox"/> Residencial	<input type="checkbox"/> Bodegas
<input type="checkbox"/> Industrial	<input type="checkbox"/> Educación
<input checked="" type="checkbox"/> Comercial	<input type="checkbox"/> Salud
<input type="checkbox"/> Otros:	<input type="text" value="especificar"/>

**Número de plantas:**

<input type="checkbox"/> 1	<input type="checkbox"/> 4
<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 5
<input type="checkbox"/> otro:	<input type="text"/>

**Existe colapso**

No

Parcial

---

Total

**Ventanas rotas**

<input type="checkbox"/> No	<input type="checkbox"/> Muchas
<input type="checkbox"/> Pocas	<input type="checkbox"/> Todas
<input checked="" type="checkbox"/> Algunas	<input type="checkbox"/> Sin ventanas

**Fallas en muros:**

<input type="checkbox"/> Ninguna	<input type="checkbox"/> Muy grave
<input type="checkbox"/> Leve	<input type="checkbox"/> Peligro de derrumbe
<input checked="" type="checkbox"/> grave	

Figura 19 - Formulación de la aplicación

El botón de formulario aparece marcado y el botón de enviar habilitado, por lo que es posible solo enviar un formulario, solo una foto o ambas.



Figura 20 – botones habilitados en la aplicación al tener datos almacenados.

Luego de tomar la fotografía esta es etiquetada con las coordenadas, la fecha y la hora además de que se cambia el tamaño de la imagen a 320x240 que es la resolución con la que opera Robot36. Al presionar el botón –Enviar- primeramente, se envían los datos del formulario y posteriormente la fotografía llega al servidor. Verificamos en el servidor si estos llegaron correctamente.



Figura 21 – Fotografía recibida en el servidor de la Raspberry.



Los datos del formulario se almacenan únicamente las respuestas ya que estos son los datos que varían, así que se envían en un orden establecido para que al momento de transmitir y sean recibidas se le especifique la pregunta a la que pertenece el dato recibido.

**ejemplo;Comercial;2;Algunas ;grave;13. , -89.**

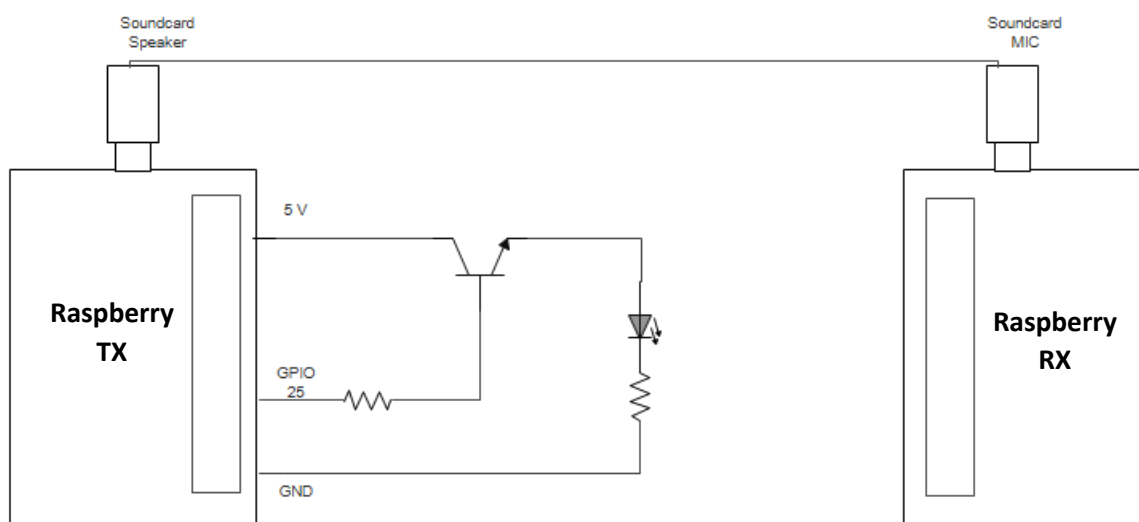
*Figura 22 – Datos recibidos del formulario.*

### 3.5 Transmisión de los datos obtenidos.

Debido a las limitantes de acceso al campus que surgieron por la pandemia, las pruebas finales se realizaron desde una Raspberry a otra, sin embargo se comprueba su funcionamiento, codificando, enviando y decodificando la imagen.

En la figura 23 se muestra el esquema del diseño para la prueba final realizada, la Raspberry RX hará de la base de monitoreo, y es la única Raspberry conectado a un monitor para visualizar la llegada de la imagen en el QSSTV.

En la Raspberry TX se encuentra el circuito donde el LED nos indicará la activación del PTT, en ese momento sabremos que la transmisión ha dado inicio.



*Figura 23 – Esquema prueba final*

La secuencia de transmisión inicia después de que el servidor haya recibido la fotografía primeramente convierte la imagen recibida a un archivo WAV en modo Robot36, una vez guardado se activa el pin25 con el que se controla el PTT, luego se reproduce el archivo WAV creado anteriormente y una vez finalizado el pin25 se apaga, esta secuencia se repite para los datos de texto obtenidos con la diferencia que la codificación es en AX.25. El código utilizado fue el siguiente:

```
#!/bin/bash
python -m pysstv -mode Robot36 foto.jpg foto.wav
python -c "import Rpi.GPIO as GPIO;GPIO.setmode(GPIO.BCM);
GPIO.setwarnings(False);GPIO.setup(25,GPIO.OUT);GPIO.output(25, True)"
sleep 0.25s
AUDIODEV=usb play /var/www/html/apachserver/foto.wav
sleep 0.25s
python -c "import Rpi.GPIO as GPIO; GPIO.setmode(GPIO.BCM); GPIO.setwarnings(False);
GPIO.setup(25,GPIO.OUT); GPIO.output(25, False)"
lugar=$(head -n 1 formulario.txt | cut -d ";" -f 1)
echo "UES>WORLD:ini."$lugar".p"| gen_packets -o x1.wav -B 300 -
python -c "import Rpi.GPIO as GPIO; GPIO.setmode(GPIO.BCM);
GPIO.setwarnings(False);GPIO.setup(25,GPIO.OUT);GPIO.output(25, True)"
sleep 0.25s
AUDIODEV=usb play /var/www/html/apachserver/x1.wav
sleep 0.25s
python -c "import Rpi.GPIO as GPIO; GPIO.setmode(GPIO.BCM);
GPIO.setwarnings(False);GPIO.setup(25,GPIO.OUT);GPIO.output(25, False)"
CADE=$(head -n 1 formulario.txt| cut -d ":" -f 1)
echo "UES>WORLD:f."$CADE".p" | gen_packets -o x2.wav -B 300 -
python -c "import Rpi.GPIO as GPIO; GPIO.setmode(GPIO.BCM); GPIO.setwarnings(False);
GPIO.setup(25,GPIO.OUT); GPIO.output(25, True)"
sleep 0.5s
AUDIODEV=usb play /var/www/html/apachserver/x2.wav
sleep 0.5s
python -c "import Rpi.GPIO as GPIO; GPIO.setmode(GPIO.BCM); GPIO.setwarnings(False);
GPIO.setup(25,GPIO.OUT); GPIO.output(25, False)"
```

### 3.6 Recepción de datos.

La recepción de datos ocurre en una base relativamente lejana, en ella estará activo un script de recepción rx.sh que detecta cuando Xastir recibe datos APRS y los guarda en un archivo de texto. El 41pa4liz a 41pa4lizer es el siguiente:

```
#!/bin/bash
nc localhost 2023 |grep -line-buffered "UES" | stdbuf -oL cut -d ":" -f 2 |grep
-line-buffered "p" |while read value;do
if [ $(echo $value | grep "ini") ];then
    41pa="ini"
else
    41pa="f"
fi

case $spaa in
    "ini" )
        lugar=$(echo $value | cut -d "." -f 2)
        name="$lugar"
        echo -n "Recibido de $lugar" >>
/var/www/html/apachserver/base/log.txt
        date >> /var/www/html/apachserver/base/log.txt
        ;;

    "f" )
        echo $value | cut -d "." -f 2 >/tmp/formulario.txt
        name=$(head -n 1 formulario.txt | cut -d ";" -f 1)
        touch /var/www/html/apachserver/base/$name.txt
        echo -e "Lugar: $name" >> name.txt
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 2)
        echo -e "uso de estructura: $form" >> /base/$name.txt
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 3)
        echo -e "numero de plantas: $form" >>/base/$name.txt
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 4)
        echo -e "existe colapso: $form" >> /base/$name.tx
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 5)
        echo -e "Ventanas rotas: $form" >> /base/$name.txt
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 6)
        echo -e "Fallas en muros: $form" >> /base/$name.txt
        form=$(head -n 1 /tmp/formulario.txt | cut -d ";" -f 7)
        echo -e "coordenadas: $form" >> /base/$name.txt

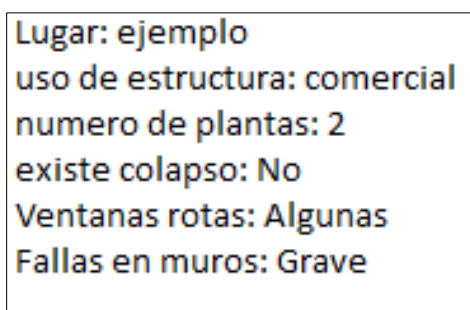
        mv /tmp/*.jpg /base/$name.jpg
        ;;

    * )
        echo $value>>/dev/null
        ;;
esac

done
```

Para que esto se lleve a cabo el transmisor y el receptor deben estar operando en la misma frecuencia. La recepción de datos de texto se realiza utilizando Xastir, para que únicamente los datos que queremos guardar recibidos en Xastir agregaremos un indicativo para que en caso de transmisiones diferentes al mismo tiempo no se mezclen, y solo las líneas pertenecientes al formulario se guarden como un único archivo, estos son *p*, *ini* y *f*.

Los indicativos *ini* y *f* determinan en que archivo deben guardarse, por ejemplo, *ini* se utiliza para el log tomando únicamente el lugar de donde proviene el mensaje, mientras que las líneas de texto que inicien con *f*. el script los guarda en un archivo temporal y luego va creando un archivo de texto con las preguntas y respuestas del formulario.



Lugar: ejemplo  
uso de estructura: comercial  
numero de plantas: 2  
existe colapso: No  
Ventanas rotas: Algunas  
Fallas en muros: Grave

Figura 24 – Formulario recibido.

Luego de guardar los datos del formulario el script busca el sitio donde ha sido guardada la fotografía que se recibió antes que el texto, y lo mueve a la misma carpeta donde se encuentra el formulario con el mismo nombre.

La imagen es recibida por QSSTV sin necesidad de un script adicional, QSSTV nos da la opción de autoguardado al recibir una imagen, esta se guarda en una carpeta seleccionada a nuestra preferencia, en nuestro caso es una carpeta llamada /tmp/ luego como ya se mencionó el script busca un archivo jpg para modificar el nombre del archivo y guardarlo donde se encuentra el formulario.



*Figura 25 – Fotografía recibida por QSSTV*

La calidad de la imagen dependerá en una parte la calidad de la imagen original, mientras mejor calidad tenga la imagen original la imagen recibida podrá tener una calidad mejor. Durante la transmisión siempre pueden perderse datos o haber interferencia que reducen la calidad de la imagen recibida comparado con la original.

## CONCLUSIONES

- La calidad de la imagen disminuye al llegar a su destino, este puede depender de la interferencia o ruido al momento de la transmisión/recepción, sin embargo, se observa que la calidad de la imagen recibida es aceptable ya que la imagen puede visualizar su contenido tanto como el lugar y las letras de las coordenadas.
- Aun cuando la comunicación por radio HF puede considerarse obsoleta, existe radioaficionados que mantienen activa el uso de este medio, así como también la innovación o mejora de software y equipos, haciendo de este medio muy importante sobre todo en casos de emergencia.

## GLOSARIO

**Baudios:** es una unidad de medida en la transmisión de señales que se expresa símbolos por segundos.

**Datos:** se refiere a una descripción o eventos documentados, el cual pueden ser analizados y estudiados para obtener información.

**Geolocalización:** es una tecnología capaz de determinar la ubicación geográfica de un objeto.

**Script:** en informática se trata de una secuencia de comandos que pueden ser ejecutados por un intérprete para realizar una determinada tarea.

**Servidor:** es una computadora que puede atender peticiones de uno o más cliente, según los diferentes servicios el cual el servidor este configurado.

**WAV:** (Waveform audio file format) es un formato de audio digital que comúnmente es utilizado sin compresión de datos, por lo que se considera una calidad de audio sin perdidas.

## BIBLIOGRAFÍA

- [1] Manual PHP  
<https://www.php.net/manual/es/index.php>
  
- [2] Sitio web: Rasp AP  
<https://raspap.com/>
  
- [3] Sitio web: Rasp AP  
[https://bioinf.comav.upv.es/courses/unix/scripts\\_bash.html](https://bioinf.comav.upv.es/courses/unix/scripts_bash.html)
  
- [4] Sitio web: Xastir APRS Program  
[https://xastir.org/index.php/Main\\_Page](https://xastir.org/index.php/Main_Page)
  
- [5] sstv-handbook.com  
<https://www.sstv-handbook.com/download/sstv-handbook.pdf>
  
- [6] Direwolf User guide  
<https://raw.githubusercontent.com/wb2osz/direwolf/master/doc/User-Guide.pdf>
  
- [7] Tutorial App Inventor  
<http://kio4.com/appinventor/>



## ANEXOS

### Instalación DireWolf en Linux

Antes de instalar DireWolf es necesario verificar si se encuentran instaladas las siguientes librerías. Y en caso de no tenerlas instaladas:

```
>sudo apt-get install libasound2-dev
>sudo apt-get install libudev-dev
```

Es posible instalar direwolf utilizando apt-get desde la terminal:

```
sudo apt-get install direwolf
```

Pero el método recomendado es siguiendo estos pasos:

```
>cd ~
>git clone https://www.github.com/wb2osz/direwolf
>cd direwolf
>git checkout dev
>mkdir build && cd build
>cmake ..
>make -j4
>sudo make install
>make install-conf
```

Para ejecutar el programa iniciar una terminal o CLI y escribir:

```
>direwolf
```

Más información: <https://github.com/wb2osz/direwolf>

### Instalación para QSSTV:

Antes de instalar QSSTV es necesario verificar si se encuentran instaladas las siguientes librerías. Y en caso de no tenerlas instaladas:

```
g++
libfftw3-dev
qt5-default
hamlib-dev
libasound2-dev
libpulse-dev
libopenjp2-7
```

```
libopenjp2-7-dev  
libv4l-dev
```

Para instalar las librerías puede escribir en la terminal:

```
>sudo apt-get install g++ libfftw3-dev qt5-default libpulse-dev  
>sudo apt-get install hamlib-dev libasound2-dev libv4l-dev  
>sudo apt-get install libopenjp2-7 libopenjp2-7-dev
```

A continuación, se procede a instalar el software siguiendo la siguiente secuencia de comandos:

```
cd ~/Downloads  
tar -xvzf qsstv_x.x.x.tar.gz  
cd qsstv_x.x.x  
qmake  
make  
sudo make install
```

Nota: reemplazar las X por la versión que se desea descargar.

Para ejecutar el programa iniciar una terminal o CLI y escribir:

```
>qsstv
```

También es posible buscar el icono de la aplicación en el menú de inicio del sistema operativo.

Más información: <http://users.telenet.be/on4qz/qsstv/manual/installation.html>