

Informe Mecatrónica

Reporte Semanal de Avances 12 de Abril

Marius Andrieux, Carole Huet, Juan Manuel Borquez, Matías Leonel Escobar, Juan Ignacio Luna Casabene, Mauricio Casarotto, Brandon Mamani, Ferran Martinez, Francisco Miranda, Matias Exequiel Molina, Juan Stella, Juan Sibecas, Enzo Tello, Valentín Panonto, Nicolas Piastrellini, Alan Vignolo

*División de Ingeniería Mecatrónica
Facultad de Ingeniería - Universidad Nacional de Cuyo
Ciudad de Mendoza, Mendoza, Argentina*

Abstract

Este informe aborda varios temas relacionados con la comunicación por radio y la programación en el ámbito de la radioafición. Se describen conceptos como Zello, una aplicación de comunicación por voz para Android, PC, TNC y Arduino. Se explican los fundamentos de APRS (Automatic Packet Reporting System) y el protocolo AX25 utilizado en la comunicación de paquetes por radio. Se aborda la decodificación de señales de PSK31 y FSK multiportadora, así como el experimento con el protocolo NMEA 0183 para la transmisión de datos de navegación. También se incluye la programación de un HT (Handy Talkie) y ejemplos de uso de API en lenguajes de programación R y Python. Este informe ofrece una visión general de estos conceptos y su aplicabilidad en el campo de la radioafición y la programación.

This report addresses various topics related to radio communication and programming in the field of amateur radio. Concepts such as Zello, a voice communication application for Android, PC, TNC, and Arduino are described. The fundamentals of APRS (Automatic Packet Reporting System) and the AX25 protocol used in packet radio communication are explained. Decoding of PSK31 and multi-carrier FSK signals, as well as experimentation with the NMEA 0183 protocol for navigation data transmission, are discussed. The programming of a Handy Talkie (HT) and examples of API usage in R and Python programming languages are also included. This report provides an overview of these concepts and their applicability in the field of amateur radio and programming.

Keywords: Comunicación, Decodificación, Investigación, Programación, Protocolos.

1. Introducción

El reporte presentado es una breve descripción del material investigado de las siguientes temáticas, con el propósito de dejar una base de contenido con el cual posteriormente se pueda trabajar en un proyecto integrador.

Las temáticas a tratar se enumeran a continuación:

- Uso API

- Programación HT
- Protocolo NMEA0182
- Decodificación PSK31 y FSK multi carrer
- APRS con Andriod, PC , TNC y Arduino. Definición de IGATE y protocolo AX25
- Zello

Email addresses: marius.andrieux22@gmail.com (Marius Andrieux), carolehuet99@gmail.com (Carole Huet), borquez.juan00@gmail.com (Juan Manuel Borquez), meescobarf87@gmail.com (Matías Leonel Escobar), juanicei99@gmail.com (Juan Ignacio Luna Casabene), mauriciocasarottocad@gmail.com (Mauricio Casarotto), brandonmamani1999@gmail.com (Brandon Mamani), oheydudeman@gmail.com (Ferran Martinez), francisconeuhuenmiranda@gmail.com (Francisco Miranda), matiming123@gmail.com (Matias Exequiel Molina), juanstella96@gmail.com (Juan Stella), juan.sibecas@gmail.com (Juan Sibecas), enzotello304@gmail.com (Enzo Tello), valentincarp66@gmail.com (Valentín Panonto), nicoslaspiastrellini00@gmail.com (Nicolas Piastrellini), alanvignolo1999@gmail.com (Alan Vignolo)

2. Resumen y abstract

El resumen se redacta al concluir el reporte de práctica puesto es que éste debe contener de manera breve los objetivos, el problema que se plantea, los resultados y las conclusiones que se obtuvieron al finalizar la práctica. El abstract es una traducción del resumen al Inglés.

Las *keywords* son las palabras clave que definen la práctica, deben estar en orden alfabético, se recomienda no usar las mismas palabras que en el título.

3. Objetivos

Introducir al grupo en la temática a desarrollar en el posterior proyecto integrador, formando una base informativa a partir de la investigación, de los temas ya mencionados en la introducción.

4. Desarrollo

Uso API

¿Que es una API y para que sirve?

Una API (Application Programming Interface) es un conjunto de reglas, protocolos y herramientas que permiten a diferentes aplicaciones interactuar entre sí. Es decir, es una interfaz de programación que expone ciertas funcionalidades de una aplicación o sistema, de tal forma que otros programas puedan acceder a ellas de manera controlada y segura.

Las APIs pueden utilizarse para una gran variedad de propósitos, pero algunos de los más comunes son:

- Integración de diferentes sistemas: Las APIs permiten que diferentes sistemas puedan comunicarse entre sí, intercambiando información y datos de manera eficiente. Por ejemplo, una API puede permitir que una aplicación de comercio electrónico se conecte con un sistema de pago para procesar transacciones.
- Desarrollo de aplicaciones: Las APIs también son utilizadas por desarrolladores para crear aplicaciones que se integran con otras plataformas y servicios. Por ejemplo, una aplicación móvil puede utilizar la API de Google Maps para mostrar mapas y direcciones.
- Automatización de procesos: Las APIs también pueden ser utilizadas para automatizar procesos y tareas. Por ejemplo, una API puede permitir que una aplicación de gestión de tareas se integre con un servicio de correo electrónico para crear automáticamente una tarea cada vez que se recibe un correo electrónico.

En nuestro caso particular, utilizamos la API para consumir datos de la misma mediante peticiones del tipo CRUD HTTP (Create, Read, Update, Delete), por medio de los comandos GET, POST, PUT y DELETE. En esta primera instancia de proyecto, generamos una codificación en python haciendo uso de la librería requests^a, la cual nos permite generar una "Query o Petición.^a la computadora que aloja la API y de esta manera obtener la información que buscamos. Se nos entregó como ejemplo para trabajar la API correspondiente a "Open Notify", la cual nos entrega información precisa y en tiempo real de la Estación Espacial Internacional (o por sus cifras en inglés, International Space Station, ISS).

Listing 1: Codigo Python para consumir datos de API ISS

```
1 #Lo primero que hacemos es importar la libreria  
request, que me permitira realizar el CRUD de la  
API
```

```
2 import requests  
3 #Adicionalmente importamos la libreria datetime que  
me permitira llevar el timestamp a la fecha y  
hora actual  
4 import datetime  
5 #Luego generamos el request de lectura de la api  
mediante el metodo get  
6 respuesta=requests.get("http://api.open-notify.org/  
iss-now.json")  
7 #La variable respuesta es un objeto de tipo response  
, que contiene la respuesta del servidor  
8 #Ahora con el metodo raise_for_status() podemos ver  
si la respuesta es correcta o no  
9 #El metodo escucha la respuesta y lanza la excepcion  
HTTPError si la respuesta es erronea  
10 respuesta.raise_for_status()  
11 #Esta excepcion establece un codigo de error del  
protocolo HTTP  
12 #Hay varios codigos de error, particularmente el que  
nos interesa es el 404 que indica que el  
recurso no existe  
13 #Si esto ocurre y el status code es igual error 204,  
entonces no generamos el objeto data, si no si  
lo generamos  
14 if respuesta.status_code != 204:  
15     data=respuesta.json()  
16  
17 #Una vez tenemos el objeto data, el cual es del tipo  
json, podemos acceder a su info como si se  
tratara de una biblioteca  
18 #Finalmente printeamos los datos  
19 print("Accediendo a informacion mediante API")  
20 print(f'Estado: {data.get("message")}')  
21 timestamp=data.get("timestamp")  
22 fecha_hora=datetime.datetime.fromtimestamp(timestamp  
)  
23 print(f'Fecha y hora: {fecha_hora}')  
24 print("La posicion actual de la ISS es: ")  
25 print(f'LATITUD {data.get("iss_position").get("latitude")}')  
26 print(f'LONGITUD {data.get("iss_position").get("longitude")}')
```

Finalmente el output en consola del codigo es el siguiente:

```
Accediendo a informacion mediante API  
Estado: success  
Fecha y hora: 2023-04-11 20:46:42  
La posicion actual de la ISS es:  
LATITUD 50.1998  
LONGITUD 171.8798
```

Figura 1: Captura output en consola

El output varia cada vez que se hace la request dado que la ISS va variando su posición y la API nos proporciona dicha información en tiempo real. Proximamente en el proyecto estaremos trabajando con API de APRS.

Programación HT

¿Qué es PMR446?

Los equipos PMR446 (Private Mobile Radio) son radios de uso libre en la banda de UHF (70 cm) y en el rango de frecuencias de 446.000 - 446.100 MHz, de gran aceptación por el público

en general, siendo usados tanto para actividades profesionales como de ocio. Su éxito radica, además de no necesitar licencia para portarlos y usarlos, en su fácil manejo, ya que carecen de cualidades excesivamente técnicas que impliquen un aprendizaje sobre radiotelefonía y su bajo coste. Estas emisoras (en la mayoría de los productos ofrecidos en el mercado como walky talkies de reducido tamaño) son muy sencillos de manejar incluso con una sola mano. Cuentan con la posibilidad de ser alimentados por pilas comunes o recargables y ofrecen en muchos casos una amplia garantía de resistencia a golpes e incluso impermeabilidad al agua o la nieve. Existen equipos con funcionalidad de subtonos y doble escucha. Los equipos de gama baja no suelen traer estas características. Usualmente cuentan con ocho canales de PMR446 separados por 12,5kHz de uno a otro. La regulación de esta banda permite una emisión máxima de 500mW de potencia.

¿Qué son los subtonos?

Los Subtonos CTCSS son unas señales que se emiten junto a la portadora y a nuestra voz, que permiten elegir qué señales vamos a oír y cuáles no. Cuando tenemos seleccionado un subtono, el Squelch, que es el silenciador del ruido del fondo, no permitirá que se oiga la señal que llega mientras el subtono del emisor no sea el que tenemos seleccionado. De esta manera podemos evitar molestias evitando conversaciones que nada tienen que ver con nosotros. Si elegimos el código o subtono 0 el sistema queda desactivado y podremos escuchar todas las conversaciones de todos los subtonos.

Usar subtonos garantiza varios factores:

- No ser interferido por actividades profesionales como vigilantes de seguridad, hoteles, taxistas.
- No ser interferido por niños jugando con estos equipos.
- No ser interferido por ruidos del espectro electromagnético.

Al seleccionar un canal y un subtono específico, pero que es conocido por todas las personas del mismo ámbito asegura las correctas comunicaciones.

Iniciativa Canal 7 Subtono 7 para Seguridad en Montaña

La Iniciativa Canal77PMR pretende aumentar la seguridad de todas aquellas personas que realizan actividades en la montaña portando una pequeña emisora/walkie sin licencia que pueden adquirir en cualquier gran superficie o por Internet.

Se ha establecido un canal único de comunicación (Canal 7 y subtono 7), ya que si utilizamos distintos canales, pierde todo el sentido.

La iniciativa consta de tres premisas:

1. Coordinación dentro del mismo grupo de montaña: En senderismo permite la comunicación entre el primero y el último evitando que los primeros se separen mucho de los más rezagados. En caso de necesidad se puede contactar con el primero para recibir ayuda. En escalada, permite la

comunicación entre el primero y el segundo de la cordada, evitando posibles confusiones y accidentes.

2. Prestar o recibir ayuda de otros grupos montañeros que estén en las cercanías: En caso de necesitar ayuda urgente, al estar todos en el mismo canal podemos prestar/recibir auxilio de grupos que estén realizando actividades en las cercanías.
3. Comunicación directa con los Grupos de Rescate: En el caso de que hayamos pedido auxilio a los grupos de rescate por los medios habituales, éstos podrán contactar con nosotros en su aproximación al lugar, pudiéndonos ampliar la información y/o guiar con más precisión.

Configuración manual para grabar el canal 7.7

Canal PMR 7 - frecuencia 446,08125

Subtono 7 - frecuencia 85.4

Lo guardaré en el canal de memoria 7, así sabré que estoy en el 7-7. Pulsamos VFO/MR y nos ponemos en la opción VFO (que no aparezcan canales de memoria a la derecha de la frecuencia)

Tecleamos la frecuencia del canal 7 PMR (446,00625) Solo teclearemos 446,081; el 25 lo pone sólo el walkie en un tamaño inferior. Como veis, no aparece el icono ÇT.^{es} es porque no tiene asignado ningún subtono. Ahora mismo este sería el canal puro 7.0 por lo que escucharemos cualquier comunicación emitida por los canales 7.0 al 7.38

Vamos a ponerle el SUBTONO 7 tanto en recepción (para poder escuchar) como en transmisión (para poder emitir).

- Recepción: MENU / 11 / R-CTCS / MENU / 854 / MENU / EXIT
- Transmisión: MENU / 13 / T-CTCS / MENU / 854 / MENU / EXIT

Como se puede ver, ahora ya aparece el icono ÇT.^{en} la parte izquierda, con lo que ya sabemos que ese canal tiene asignado un subtono.

Bien, vamos a memorizar este canal PMR (7) y el subtono asignado (7) en nuestro canal de memoria nº13 (por ejemplo) Para poder usar un canal de la memoria éste debe de estar vacío, sin uso. En el ejemplo que nos concierne, si se tiene algún canal guardado en la memoria 13 no nos dejará guardarlo, así que lo primero es borrar el canal de memoria a usar, en nuestro caso, el 13.

- Vaciar canal de memoria nº13: MENU / 28 / DEL-CH / MENU / 13 / MENU / EXIT
- Guardar en canal de memoria nº13: MENU / 27 / MEM-CH / 13 / MENU / EXIT Pulsa el botón VFO/MR. Y ahora teclea 013

Para configurar el Baofeng UV-82 en PMR 446 canal 7 subtono 7 mediante una computadora primero debe conseguir el cable auxiliar USB para conectar el Baofeng a la PC.

Lo siguiente es descargar el software de Baofeng de su página oficial específico para el modelo y serie correspondiente.

Una vez instalado el software, cuando se inicializa generalmente esta configurado en japonés, por lo tanto se debe seleccionar el botón en la ubicación marcada en rojo en la imagen a continuación:

Se abrirá entonces un desplegable que permite seleccionar la opción inglés.

Luego se procede a configurar el canal y subtono en la ventana abierta correspondiente a Channel Information (ventana con forma de planilla en la imagen anterior). Esto se consigue seleccionando en la fila correspondiente al canal 7 y columna Band la opción VHF/UFH:

Luego en la misma fila pero en la columna RX Frequency se debe ingresar el valor 446.08125, que corresponde a PMR 446 y canal 7. Al hacerlo y presionar enter los demás valores de la fila se rellenarán automáticamente:

Luego, en las columnas CTCSS/DCS Dec y CTCSS/DCS Enc debe seleccionar el valor 85.4, es el subtono 7:

Por último, seleccione Close en la ventana de Channel information, haga click en la pestaña Program en la barra de herramientas superior, seleccione la opción Write to radio, y seleccione Write en la ventana que se abrió.

Protocolo NMEA0182

Sobre el NMEA 0182 En la actualidad se utiliza el protocolo NMEA 0183. Este protocolo es una combinación de especificaciones eléctricas y de datos para la comunicación entre equipos electrónicos marítimos, tales como: profundímetro, anemómetro, corredera, girocompás, piloto automático, GPS y muchos otros tipos de instrumentos. Ha sido definido, y está controlado, por la organización estadounidense National Marine Electronics Association (organización de comercio electrónico estadounidense que establece estándares de comunicación entre electrónica marina).

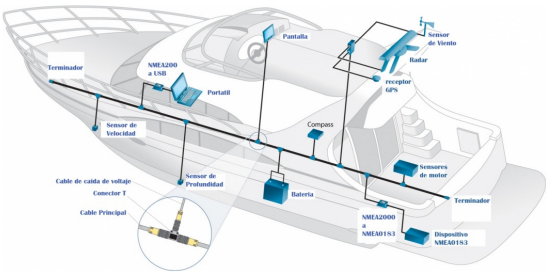


Figura 2: nmea

4.800 baudios (baudrate o bits por segundo) es la velocidad utilizada para la comunicación con plotters, antenas GPS o los diferentes instrumentos de navegación. 38.400 baudios es la alta velocidad empleada para que se pueda utilizar con aquellos equipos que utilizan la información AIS tal y como los

transpondedores o los receptores AIS. Esta diferencia de velocidades en ocasiones puede suponer un problema. Haciendo la suposición de que se contiene un moderno transpondedor AIS que emite datos NMEA0183 de alta velocidad a 38.400 baudios y se desea conectar, por ejemplo, con datos GPS o algunos plotters o PCs que solo tienen una entrada NMEA0183 y trabajan a una velocidad de 4.800 baudios, esta comunicación se imposibilita debido a estas diferentes velocidades. Ante este problema, se inventaron los denominados multiplexores NMEA como el modelo ofrecido por Digital Yacht MUX100 (<http://digitalyacht.es/producto/mux100/>) que tiene como función multiplexar u “ordenar” estos datos con diferentes velocidades para convertirlos en una salida “out” que puede conectarse a las entradas “in” de otros instrumentos. Existe una variante de la norma llamada NMEA-0183HS que especifica una velocidad de transmisión de 38.000 baudios. A pesar de que parece que NMEA0183 sigue estando muy presente como protocolo de comunicación y que parece que el cambio está costando más de lo esperado, está claro que el NMEA2000 lo sustituirá en los años venideros. Este nuevo estándar es muy fácil de utilizar y el conectar los equipos a la red del barco resulta más sencillo gracias a su un conector en “T” que permite que algunos instrumentos de navegación puedan alimentarse directamente de esta red NMEA2000. El objetivo de este nuevo estándar es así mejorar la velocidad de transmisión de datos y sobre todo facilitar la instalación y la conexión de los diferentes instrumentos de navegación.

Tabla 1: Velocidad de datos

Baud Rate típico	4800
Bits de datos	8
Bits de parada	1

Estructura de las sentencias

Todos los datos transmitidos son caracteres imprimibles ASCII entre 0x20 (espacio) a 0x7e ().

Los caracteres de datos son todos los anteriores además de los caracteres reservados. Los caracteres reservados se utilizan por NMEA0183 para los siguientes usos:

Las sentencias tienen una longitud máxima de 82 caracteres, incluyendo el \$ o ! del principio y el ;LF final.

El carácter de inicio para cada sentencia puede ser un \$ (para sentencias con campos convencionales o para sentencias que tienen encapsulación especial en ellos).

Los siguientes cinco caracteres identifican el emisor (dos caracteres) y el tipo de sentencia (tres caracteres).

Todos los siguientes campos de datos están separados por comas.

Cuando el dato no esta disponible, el campo correspondiente permanece en blanco (no contiene ningún carácter antes del siguiente delimitador (coma)).

Tabla 2: Estructura de sentencias

ASCII	Hex	Dec	Uso
<CR>	0x0d	13	Retorno de carro
<LF>	0x0un	10	Salto de línea, final de sentencia
!	0x21	33	Inicio de delimitador de encapsulación de sentencia
\$	0x24	36	Delimitador de inicio
*	0x2un	42	Delimitador de Suma de verificación
,	0x2c	44	Delimitador de campo
\	0x5c	92	Delimitador de bloque de ETIQUETA
^	0x5e	94	Delimitador de código para representación HEX de caracteres ISO/IEC 8859-1 (ASCII)
~	0x7e	126	Reservado

El primer carácter que inmediatamente sigue al último campo de datos es un asterisco seguido de una suma de verificación.

El asterisco es inmediatamente seguido por una suma de verificación representada como un número hexadecimal de dos dígitos. La suma de verificación es el OR exclusivo bit a bit de todos los caracteres ASCII entre el \$ y *, no inclusivos. Según la especificación oficial, la suma de verificación es opcional para la mayoría de las sentencias, pero es obligatorio para los tipos de sentencia RMA, RMB y RMC (entre otras).

Una suma de verificación, (también llamada suma de chequeo o checksum), en telecomunicación e informática, es una función de redundancia que tiene como propósito principal detectar cambios accidentales en una secuencia de datos para proteger la integridad de estos, verificando que no haya discrepancias entre los valores obtenidos al hacer una comprobación inicial y otra final tras la transmisión. La idea es que se transmita el dato junto con su valor suma, de esta forma el receptor puede calcular dicho valor y compararlo así con el valor suma recibido. Si hay una discrepancia se pueden rechazar los datos o pedir una retransmisión.

<CR><LF> finaliza la sentencia.

Como ejemplo, una alarma de waypoint de llegada tiene la forma:
\$GPAAM,Un,Un,0.10,N,WPTNME32

Decodificación PSK31 y FSK multi carrer

Decodificación de PSK31

El PSK31 es un modo de transmisión de radioaficionados que utiliza la modulación por desplazamiento de fase (PSK) para transmitir datos en una señal de radio. Es un modo popular utilizado en la radioafición para la transmisión de texto en tiempo real. Para lograr la decodificación del audio en formato psk31 se

utilizó la popular y reconocida aplicación FLDIGI ampliamente utilizada por radioaficionados. Una vez instalada la aplicación, se procedió a su configuración. Primero se configuró el modo de decodificación en PSK31. Esto se logró abriendo el menú de modos que se encuentra en la parte superior de la ventana del programa. Luego, se procedió a cargar el archivo para su decodificación. Esto se pudo realizar a través de la función de "playback" de FLDIGI. Esta función permite cargar un archivo de audio que contiene la señal de PSK31. Se accedió tras seleccionar la opción "Playback" en el menú "Archivo" en la parte superior de la ventana de Fldigi y posteriormente se seleccionó el archivo de audio. Cabe destacar que para que funcione correctamente el archivo se debe subir en formato ".wav". La decodificación se inició una vez que se presionó el botón "start" ubicado en la barra de herramientas de FLDIGI y el programa comenzó a mostrar el texto decodificado en la ventana de visualización.

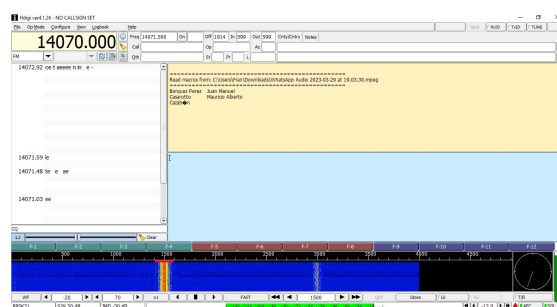


Figura 3: Foto del programa Fldigi

El texto obtenido fue el siguiente:

Borquez Perez Juan Manuel
Casarotto Mauricio Alberto
Cazabon Marton Gabriel
Escobar Matias Leonel
Huet Carole Audrey Mathilde
Luna Casabene Juan Ignacio
Mamani Ventura Brandon Nicolas
Martinez Arenas Ferran
Miranda Francisco Nehuen
Molina Matias Exequiel
Panonto Valenton Carlos
Piastrellini Nicolas
Sibecas Juan Pablo
Stella Juan
Tello Enzo Gabriel
Vignolo Alan Lihuen

Figura 4: Decodificación PSK

Se observó que la información se obtenía de forma relativamente lenta ya que la información se transmite a 31.25 bits de información por segundo. La zona fina en la visualización del espectro representa los cambios de fase en la señal y permite decodi-

ficar los datos transmitidos en forma de texto. destacamos que en protocolo psk31 se observa un espectro relativamente fino.

Decodificación de FSK multi Carrier - QPSK31 MC

El FSK es un modo de transmisión de radioaficionados que utiliza la modulación por desplazamiento de frecuencia. La información se codifica al variar la frecuencia de la señal portadora, generalmente entre dos frecuencias distintas, para representar los valores binarios 0 y 1. Es reconocida por su resistencia al ruido y la interferencia en entornos de transmisión difíciles. Para su decodificación se procedió igual que en el caso del audio PSK31, con la diferencia que el modo de decodificación seleccionado fue Op Mode-¿PSKR-¿Multicarrier-¿32xPSK63R. Finalmente se utilizó la herramienta "Playback", se cargó el archivo y se llevó a cabo su decodificación. Se obtuvo nuevamente la lista de alumnos de Mecatronica

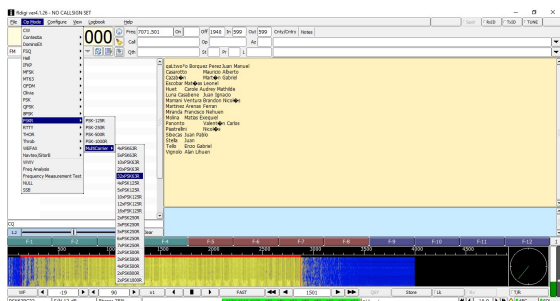


Figura 5: Decodificación FSK multicarrier

Se observó que la velocidad de transferencia de datos es altamente superior a la del audio en formato PSK32. Esto se debe a que la velocidad de transferencia del formato FSK multicarrier PSK63RC32 es de 1024 bits por segundo (bps).

APRS con Android, PC, TNC y Arduino. Definición de IGate y protocolo AX25

¿Que es APRS?

APRS (Automatic Packet Reporting System) es un protocolo de comunicación digital de radioaficionados que utiliza datos de posición y otros datos para transmitir información a través de ondas de radio. Fue desarrollado por Bob Bruninga en 1984 con el objetivo de poder enviar información en tiempo real de cualquier objetivo o persona a APRS-IS (Internet system) utilizando una red de radio frecuencia local. APRS funciona a partir de repetidores de paquetes conocidos como digipeaters, que recopilan y retransmiten datos a otros digipeaters de manera similar a cómo los repetidores retransmiten las ondas de radio. Cada vez que pasa por un digipeater, deja información para que otros la accedan y la retransmitan. A través de este método, los datos pueden viajar grandes distancias y eventualmente llegarán a un I-Gate donde todos en Internet tienen acceso a los datos que han estado viajando por los digipeaters.

APRS Con Android

Los teléfonos con sistema operativo Android son extremadamente populares y disponibles en la actualidad. Android es un sistema operativo en constante expansión y sus posibilidades son ilimitadas. Por ejemplo, ya se utiliza en televisores, netbooks, tablets, teléfonos, impresoras e incluso lavadoras. ¿Cómo se aprovecha el protocolo APRS en el sistema operativo Android? Actualmente, los transmisores APRS pueden enviar información de ubicación basada en los datos recibidos de una unidad receptora de GPS para compartir información de un usuario. Los smartphones Android cuentan con un receptor GPS incorporado en el teléfono Android, por lo que son capaces de compartir la ubicación del usuario convirtiendo los datos en un formato NMEA y enviándolos a través de Bluetooth a un servidor en un ordenador portátil. El servidor acepta la conexión Bluetooth y envía los datos NMEA recibidos del teléfono Android a un programa en C que los envía a un puerto serial en la computadora donde se conecta un TNC/transmisor APRS el cual enviará su ubicación a una I-Gate, y una vez que la ubicación esté disponible en la I-Gate, la aplicación APRS obtendrá la ubicación del servidor y la mostrará en un mapa incorporado en la aplicación. Una vez que la ubicación está en el transmisor, puede ser enviada a un APRS-IS I-Gate utilizando Radio Frecuencia y los datos de ubicación serán accesibles para todos. El principal objetivo de implementar APRS en Android, es el de proporcionar una buena medida de seguridad en áreas donde no hay buena conectividad de redes convencionales.

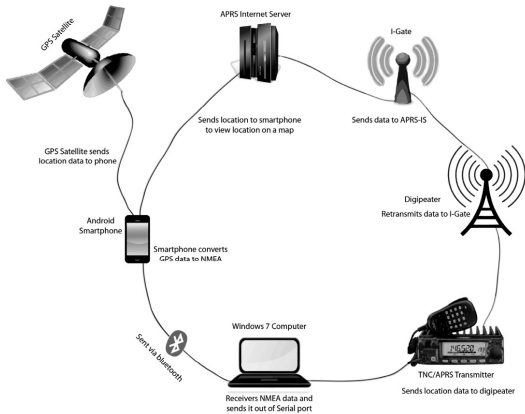
APRS - PC

El pc puede cumplir con varios roles a la hora de transmitir datos mediante el protocolo APRS donde principalmente se encarga de recibir, decodificar y mostrar los datos en un formato comprensible para el usuario. Por lo tanto, entre sus funciones más importantes se destacan: - Funciona como interfaz de usuario: Permite configurar y controlar un dispositivo de transmisión APRS, como un TNC (Terminal Node Controller) o un transceptor de radio. El usuario puede establecer la configuración de APRS, como la frecuencia de transmisión, el tipo de mensaje, etc. - Servidor: Un PC también puede alojar un servidor APRS, que es un nodo de la red APRS que almacena y comparte información APRS en tiempo real. Los servidores APRS recopilan, procesan y distribuyen datos APRS, como posiciones de estaciones, mensajes y otros datos, para que estén disponibles para otros usuarios de APRS en la red. - I-Gate (puerta): Un PC con un TNC y conexión a Internet puede actuar como un APRS Internet Gateway o I-Gate. El I-Gate es una estación que recibe paquetes APRS a través de la radio y los retransmite a la red APRS en Internet. Esto permite que las estaciones APRS que no tienen acceso directo a Internet, como estaciones móviles o portátiles, puedan enviar su información de posición y mensajes a través del I-Gate para que sean visualizados en la web o en otras estaciones APRS en todo el mundo.

APRS – TNC

TNC (Terminal Node Controller) es un dispositivo que se utiliza para conectar una estación de radio a una red de paquetes de datos. Entre sus funciones más importantes, destacan: Modulación y demodulación de datos: El TNC convierte datos digitales

en señales de radio y viceversa. Codificación y decodificación de paquetes: El TNC codifica y decodifica los datos en paquetes, que son unidades que se transmiten a través de la red. Control de flujo: El TNC tiene la importante función de controlar el flujo de datos entre la estación de radio y la red de paquetes de datos para evitar la congestión de la red y asegurar una transmisión de datos eficiente y confiable. Interfaz con el equipo de radio: El TNC proporciona una interfaz entre la estación de radio y el equipo de cómputo, como una computadora, para permitir la comunicación bidireccional de datos. Algunos TNCs también pueden incluir funciones de autenticación y gestión de red.



APRS con Arduino

Utilizando una placa Arduino, algo de software dedicado y hardware específico es posible transmitir y recibir paquetes APRS. En internet se puede encontrar gran cantidad información acerca de como llevar esto a cabo y las soluciones varían en el grado de conocimiento previo que uno debe tener o la dificultad de las mismas. A continuación se presenta información para la transmisión de paquetes con información de ubicación, temperatura, humedad, y en general cualquier información relevada por sensores. La información se obtuvo de blogs de personas que reproducen proyectos completos.

Trasmitir paquetes APRS

La guía del proyecto se obtiene del video [HX1 Arduino APRS Tracker](#) del canal de youtube de Christopher Lum. En el mismo se sigue el tutorial [Sparkfun HX1 APRS Transmitter Hookup Guide](#). Este tutorial está enfocado principalmente a la transmisión de paquetes con globos meteorológicos, sin embargo es posible hacer modificaciones para adaptarlo a las necesidades específicas. Otra cosa que se debe tener en cuenta es que debido a que el transmisor opera en la frecuencia de radio aficionado es necesario disponer de una licencia para operar en esta frecuencia. La más básica que se puede obtener es la de técnico y se adquiere rindiendo una evaluación. A continuación se indican los elementos del proyecto:

Materiales

- Un transmisor HX1: Un transmisor en la banda de radio-aficionados. [Transmisor HX1](#)
- Una placa Arduino como por ejemplo un Arduino UNO.
- Una fuente con una capacidad de corriente de al menos 150 mA. Un cargador de 9V o una fuente de PC con salida de 9V funciona bien.
- Los sensores que sean requeridos.
- El código del proyecto para manejar tanto el transmisor como los sensores y el GPS
- Una antena que pueda transmitir en la frecuencia de APRS que en Argentina es de 144.93 MHz
- Una licencia de radio aficionado

Revisión del Hardware

Modulo de RF HX1

Es el módulo utilizado para la comunicación a la frecuencia específica. Indicado en la figura 6

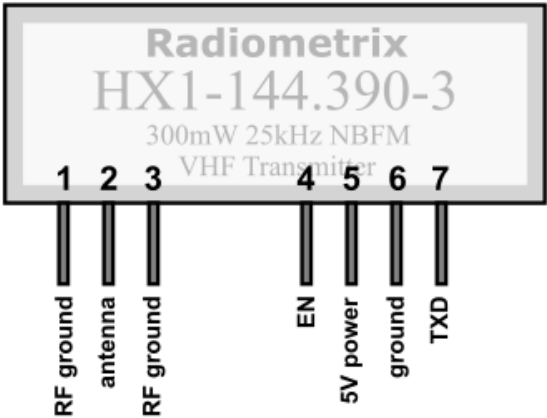


Figura 6: Modulo de radio HX1

Tabla 3: Descripción de pines del transmisor

Pin	Función	Descripción
1	RF Ground	Tierra de la Antena
2	Antenna OUT	Salida a la Antena
3	RF Ground	Tierra de la Antena
4	EN (Enable) Input	Habilitación de entrada (a 5V)
5	5V Power Input (VCC)	Fuente de 5V regulada
6	Ground	Tierra de la fuente
7	TXD (Transmit Data) Input	Señal modulada y formateada

Antena

En el tutorial indicado se muestra el diseño de una antena la cual será necesario construir para lograr la comunicación. Se presenta en las imágenes 7 y 8

Los elementos de la antena se pueden construir con cualquier material conductor. Para la conexión con el módulo de radio frecuencia se debe conectar el elemento vertical a la señal de salida del módulo y los elementos radiales al pin de tierra del módulo. La disposición de la antena debe ser sobre cualquier material no conductor de la electricidad.

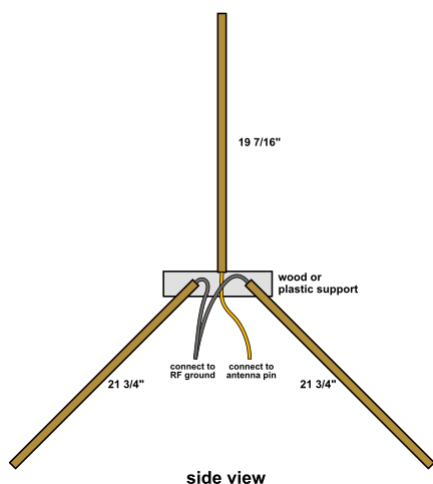


Figura 7: Antena vista frontal

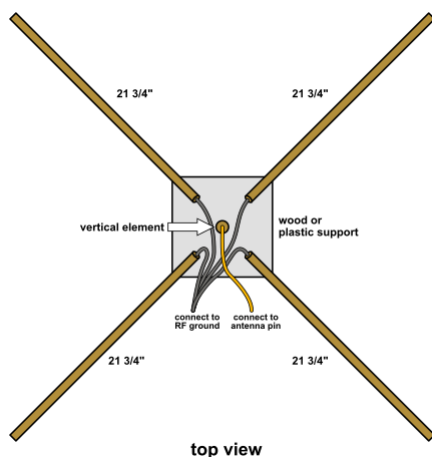


Figura 8: Antena vista lateral

Esquema de conexiones

Una conexión por default es la que se muestra en la figura 9. La placa utilizada puede ser cualquiera de las de Arduino, por lo tanto el esquema hay que cambiar la placa y los pines por los equivalentes según sea la que se esté utilizando. Así mismo los sensores y el buzzer junto con el GPS pueden cambiarse.

En la tabla 4 se puede observar las conexiones entre la placa el módulo RF, los sensores y el GPS

Software

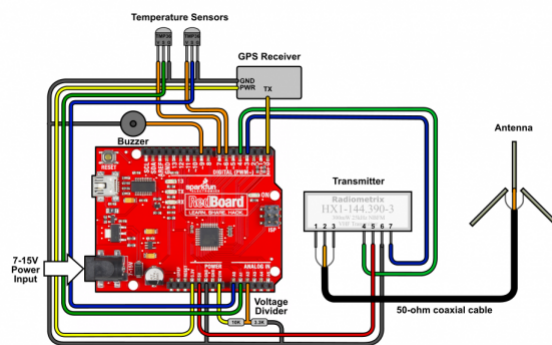


Figura 9: Esquema de conexiones

El software que se utiliza, [Trackuino](#), está pensado principalmente para el control de globos meteorológicos y para los sensores y conexiones que se indicaron en la figura 9 y en la tabla 4. Para que el proyecto funcione con sensores específicos hay que hacer cambios en el código fuente y en las configuraciones del software además de los cambios en las conexiones.

Los archivos que deben cambiarse del repositorio son: [aprs.cpp](#), en donde hay que modificar el código para los sensores específicos, [config.h](#), en el que hay que cambiar el CALLSIGN por el propio (también se puede cambiar la frecuencia de la transmisión) y [trackuino.ino](#) donde también se cambia según los sensores. Una guía mas específica se obtiene del vídeo mencionado anteriormente.

Recepción

Para la recepción de paquetes hay mucha información en internet en distintos repositorios. En concreto se puede seguir la guía del siguiente [blog](#) en el que el autor lleva a cabo la implementación en Arduino UNO de un receptor de paquetes

I-GATE

Un I-Gate de APRS es un dispositivo o sistema que permite conectar las señales de la red de APRS a Internet.

El I-Gate actúa como un puente entre las estaciones de radio APRS y los servidores de Internet, permitiendo que la información se transmita de forma remota. Esta información puede incluir ubicaciones, velocidades y otra información relevante, y puede ser utilizada por aplicaciones y servicios que se conectan a la red APRS a través de Internet.

Los I-Gates son comúnmente utilizados por los radioaficionados y otros entusiastas de la radio para monitorear y rastrear vehículos, barcos, aviones y otros objetos en movimiento. También se utilizan en aplicaciones de seguridad y de búsqueda y rescate para localizar y seguir a las personas en situaciones de emergencia.

Protocolo AX25

El protocolo AX.25 (Amateur X.25) es un protocolo de comunicación de paquetes utilizado en redes de radioaficionados. Fue desarrollado en la década de 1980 por la comunidad de

Tabla 4: Descripción de las conexiones

Desde	Hacia	Descripción
HX1 VCC(5)	Arduino 5V	Alimentación del HX1
HX1 GND(6)	Arduino GND	Tierra de la fuente de HX1
HX1 EN (4)	Arduino D4	Habilitación de Transmisión
HX1 TXD (7)	Arduino D3	Transmisión a HX1
HX1 Antena RF GND (1)	GND Antena	Tierra de la señal
HX1 Antena Output (2)	Antena	Señal a la Antena
HX1 Antena RF GND (1)	GND Antena	Tierra de la señal
GPS TX	Arduino D0 (RX)	Datos del GPS
Alimentación GPS	Arduino 3.3V	Alimentación
GND GPS	GND Arduino	Tierra del GPS
Alimentación TMP36	Arduino D6	Alimentación sensor de temperatura
Señal TMP36	Arduino A0	Señal de temperatura
GND TMP36	GND Arduino	Tierra del sensor
Buzzer power	Arduino D9	Alimentación del Buzzer
Buzzer GND	Arduino GND	Tierra del Buzzer
Arduino Vin (Battery Voltage)	Arduino A2 a través de un resistor de 10K	Sensor de voltaje de la batería
Arduino A2	Arduino GND a través de un resistor de 3.3K	Sensor de la batería

radioaficionados y se basa en el estándar X.25 de la Unión Internacional de Telecomunicaciones (UIT) para redes de conmutación de paquetes. Este protocolo define la estructura y el formato de los paquetes de datos transmitidos por radio y proporciona un conjunto de reglas para el intercambio de paquetes entre los nodos de la red. También establece un conjunto de direcciones de red para identificar los nodos y los servicios dentro de la red. AX.25 se utiliza principalmente en redes de radioaficionados para enviar datos de texto, correo electrónico, archivos, imágenes y otros tipos de información. También se ha utilizado en sistemas de seguimiento y telemetría en satélites de radioaficionados y en algunas redes de sensores de baja potencia y alcance. Aunque el protocolo AX.25 fue desarrollado originalmente para su uso en redes de radioaficionados, también ha sido utilizado en algunas redes de emergencia y comunicaciones de defensa civil debido a su capacidad para proporcionar comunicaciones de larga distancia y capacidad de respuesta en situaciones de emergencia.

El protocolo AX25 es la manera como las estaciones Packet se comunican entre sí. Packet radio utiliza como método de acceso múltiple por sensado de portadora (Carrier Sense Multiple Access CSMA), para permitir acceso a múltiples usuarios al sistema. La estación espera hasta que existe un canal desocupado antes de empezar el proceso de transmisión. Cuando una estación tiene un paquete por transmitir y el canal está desocupado se inicia un algoritmo de tiempo aleatorio; si el tiempo expira antes de que otra estación utilice el canal, el paquete es enviado. Si el canal está ocupado es posible que varios ciclos del algoritmo ocurran hasta que el paquete finalmente sea enviado. Desde el punto de vista del modelo de referencia OSI, dentro del protocolo AX.25 se toman las primeras 2 capas más bajas de este modelo, se subdividen en otras categorías encargadas del manejo general de la conexión y verificación de datos. Se puede realizar una conexión sencilla, o de forma múltiple.

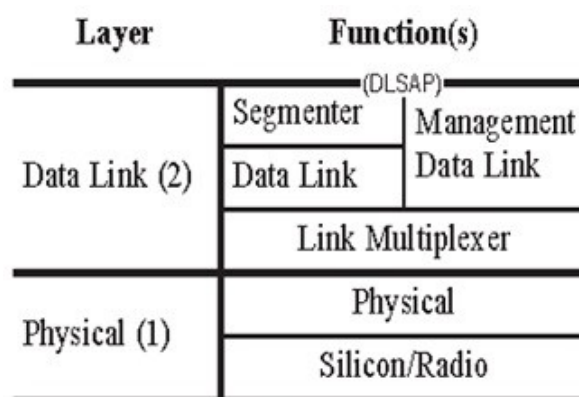


Figura 10: Conexión Sencilla

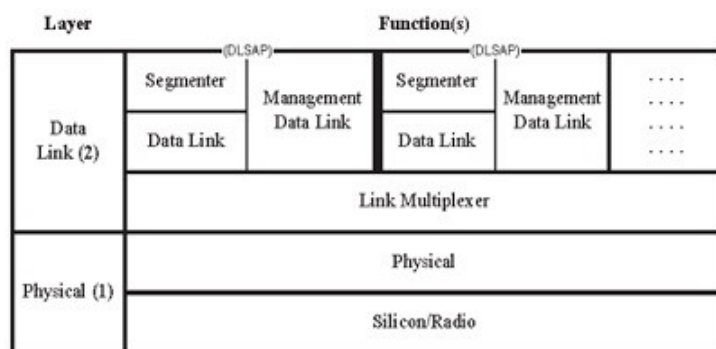


Figura 11: Conexión Múltiple

Donde el DLSAP (Data Link Service Access Point), es el punto de la capa de conexión de datos que provee el servicio o el paso a la capa 3. Sobre la capa 3 se observa un sistema de envío y respuesta, esta interacción se conoce como servicio primitivo, este es el intercambio entre la capa de conexión y las capas adyacentes a ella, este tipo de servicio primitivo funciona en el siguiente orden.

- Solicitud
- Indicación

- Respuesta
- Confirmación

Zello

¿Qué es Zello?

Zello es una aplicación móvil y web que permite la comunicación de voz en tiempo real entre usuarios. La aplicación utiliza la tecnología *push-to-talk* (PTT), que es similar a una radio de dos vías, pero en lugar de utilizar frecuencias de radio, la comunicación se realiza a través de la conexión a internet.

La aplicación está disponible en dispositivos móviles iOS y Android, así como en computadoras de escritorio y portátiles. Los usuarios pueden crear canales públicos o privados y unirse a ellos para comunicarse con otros usuarios que se encuentren en el mismo canal. También es posible enviar mensajes de texto y compartir imágenes a través de la aplicación.

¿En qué se usa?

Zello es popular entre los equipos de búsqueda y rescate, equipos de seguridad, voluntarios de emergencias y cualquier grupo que necesite una comunicación de voz rápida y confiable. Además, la aplicación cuenta con una función de historial de mensajes, que permite revisar las conversaciones anteriores.

En términos de seguridad, Zello ofrece cifrado de extremo a extremo en las comunicaciones de voz y mensajes de texto, lo que significa que la información está protegida y solo puede ser accedida por los usuarios que tienen la clave de cifrado.

Ventajas frente a otras apps

Algunas ventajas de Zello en comparación con otras aplicaciones similares como Discord son:

- Eficiencia en el uso de datos: Zello utiliza mucho menos datos que otras aplicaciones similares, lo que lo hace ideal para situaciones en las que la conectividad de internet es limitada o costosa.
- Simplicidad de uso: Zello es muy fácil de usar y no requiere mucha configuración. La aplicación se enfoca en la comunicación PTT y ofrece una interfaz sencilla y directa.
- Seguridad: Zello ofrece cifrado de extremo a extremo en las comunicaciones de voz y mensajes de texto, lo que significa que la información está protegida y solo puede ser accedida por los usuarios que tienen la clave de cifrado.
- Menor latencia: Zello ofrece una menor latencia que otras aplicaciones de comunicación, lo que significa que el tiempo entre cuando alguien habla y los demás lo escuchan es más corto. Esto hace que la comunicación sea más eficiente y efectiva en situaciones de emergencia.
- Bajo consumo de batería: Zello utiliza menos energía que otras aplicaciones similares, lo que significa que se puede utilizar durante períodos de tiempo más largos sin agotar la batería del dispositivo.

En resumen, Zello ofrece una comunicación PTT eficiente y simplificada, con menor latencia, menor consumo de datos y batería, y una mejor seguridad en comparación con otras aplicaciones similares como Discord. ¿Cómo podría aplicarse a nuestro proyecto?

Es posible comunicar un celular con Zello con una radio a través de un dispositivo intermediario. Este dispositivo se conoce como "Gateway" actúa como un puente entre la señal de radio y la señal de internet.

Para establecer la conexión, es necesario un dispositivo de radio que tenga un puerto de datos y un gateway compatible con Zello. El gateway actúa como una especie de servidor y se conecta a internet para establecer la comunicación con Zello. A su vez, el gateway se conecta al puerto de datos de la radio para transmitir la señal de radio al canal de Zello.

Existen diferentes tipos de gateways que funcionan con Zello, algunos de los más comunes son los gateways basados en software que se ejecutan en un ordenador y otros que son dispositivos físicos independientes que se conectan directamente a la red de internet.

Es importante destacar que el uso de gateways requiere de cierto nivel de conocimiento técnico y puede requerir de cierta configuración previa para funcionar correctamente.

Integración de diferentes tecnologías: El uso de un gateway permite integrar diferentes tecnologías de comunicación, como la comunicación por radio y la comunicación por internet, lo que puede ser beneficioso en situaciones en las que se necesite una comunicación flexible y adaptable a diferentes escenarios.

5. Conclusiones

[?] En conclusión, en este informe se abordaron varios temas relacionados con la comunicación por radiofrecuencia y la programación en el ámbito de la radioafición. El ecosistema de los Sistema Automatico de Informacion de Posicion tiene mucho potencial de investigacion, desarrollo y aplicacion por lo que podemos concluir que al analizar los distintos aspectos del proyecto encontramos viable su implementacion.