

iGate APRS

Informe de Entrega Preliminar: iGate Firmware

Kendy Raquel Arias Ortiz
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica
Cartago, Costa Rica
kendyarias@estudiantec.cr

Jean Carlos Chacón Perez
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica
Cartago, Costa Rica
jeanchacon@estudiantec.cr

Kendall Fallas Padilla
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica
Cartago, Costa Rica
kendallfallas@estudiantec.cr

Resumen—El presente proyecto implementa un iGate (Internet Gateway) para la red APRS LoRa, actuando como un puente de comunicación esencial entre dispositivos LoRa de largo alcance y la red global de APRS-IS. El sistema se materializa utilizando la placa LilyGO T3 LoRa32 V1.6.1, que integra un microcontrolador ESP32 y un transceptor SX1276 a 433 MHz. La metodología de desarrollo se centró en un diseño estructurado, que incluye la inicialización de hardware, el establecimiento de conectividad WiFi, y la gestión del ciclo continuo de operación. El firmware desarrollado opera bajo un bucle principal que realiza dos funciones clave: la recepción y reenvío automático de paquetes LoRa al servidor APRS-IS y la transmisión periódica de una baliza (beacon) para confirmar su estado. Las pruebas preliminares confirmaron el correcto reenvío de paquetes y la visualización del iGate en plataformas como APRS.fi. La aplicación práctica de este sistema se enfoca en el seguimiento de competiciones deportivas.

Index Terms—APRS-IS, ESP32, iGate, LilyGO T3, etc.

I. INTRODUCCIÓN

La comunicación inalámbrica y el intercambio de datos en tiempo real se han convertido en pilares fundamentales para el desarrollo de soluciones tecnológicas eficientes y seguras. Entre las múltiples herramientas disponibles, destacan tecnologías como el APRS (*Automatic Packet Reporting System*) y LoRa (*Long Range*), las cuales permiten la transmisión de información a través de radiofrecuencias con diferentes alcances, aplicaciones y requisitos técnicos. El presente trabajo tiene como objetivo analizar de forma detallada el funcionamiento, aplicaciones, frecuencias de operación y componentes esenciales de estas dos tecnologías, así como las normativas vigentes en Costa Rica que regulan su uso. Al realizar este trabajo se busca ofrecer información que permita comprender sus capacidades y limitaciones, facilitando su implementación en proyectos de comunicación, monitoreo y control.

II. MARCO TEÓRICO

II-A. APRS

El APRS (*Automatic Packet Reporting System*) o por su traducción Sistema Automático de Información de Posición mediante Radiopaquetes, es una tecnología cuyo propósito principal es el de transmitir información de manera rápida en tiempo real manteniendo en todo momento la seguridad de esa información. Inicialmente se creó para que se utilizara en estaciones fijas o móviles de radioaficionados. El APRS fue creado por Bob Bruninga como un sistema de comunicaciones

con el objetivo de integrar imágenes y datos en sistemas de computación y radio. **1**

Los APRS codifican su información utilizando el protocolo AX.25 para poder transmitir a través del espectro de radio. Estos paquetes son enviados por transceptores de radio o mediante repetidores digitales. La información enviada es recibida por otras estaciones con APRS que decodifican los paquetes de información recibida y luego esta información suele ser representada en una interfaz que muestra mapas que nos permite ver la ubicación de la información recibida. **2**

Los APRS tienen muchas funciones disponibles. Una de las principales aplicaciones es el envío de coordenadas que permiten rastrear el posicionamiento de barcos, vehículos y aviones mediante la tecnología GPS integrada en los APRS. Otra de sus aplicaciones es para la comunicación, ya que permite enviar mensajes que nos permiten indicar accidentes, emergencias, enviar señales de socorro. También se pueden utilizar para enviar información de eventos y reuniones. Hay varias formas de enviar mensajes a través de un APRS. Se pueden enviar mensajes directos que van dirigidos a estaciones específicas, boletines que se dirigen a varias estaciones y los mensajes grupales que son dirigidos a un grupo específico en vez de estaciones individuales. Además, los APRS son útiles para monitorear las condiciones climatológicas, ya que permiten transmitir en tiempo real la humedad, temperatura, presión barométrica y velocidad del viento. Permiten conectarse con estaciones meteorológicas y compartir sus datos a través de la red. Esta aplicación es utilizada para los observadores de tormentas y meteorológicos aficionados. Además de permitir las actualizaciones meteorológicas en tiempo real que permiten predecir posibles cambios climáticos. **3**

En cuanto a los protocolos que utilizan los APRS para la comunicación, podemos encontrar el AX.25, que es comunicación basada en paquetes. Utilizando el modo de tramas de información no numerada, permitiendo comunicación sin conexión. Este protocolo está basado en el protocolo X.25, que consiste en un protocolo para redes conmutadas por paquetes. **3**

Las frecuencias que se utilizan para los APRS son los 144.800 MHz para VHF (*Very High Frequency*) y los 432.500

MHz para UHF (*Ultra High Frequency*).

Los componentes que un APRS necesita para su funcionamiento son radios que puedan recibir y transmitir los paquetes de información. Trackers APRS que permitan enviar la información de ubicación y TNC (*Terminal Node Controllers*) que permiten transformar las señales digitales en paquetes compatibles con AX.25. También se necesita de software como el UI-View o el APRSISCE/32 que permiten gestionar los datos recolectados en las computadoras o aplicaciones para celulares que tengan la misma función 4. En cuanto a las redes existen los digipeaters que son repetidores digitales de paquetes que permiten recibir el paquete de información, almacenarla por un tiempo y luego la retransmiten con la misma frecuencia a la que fue recibido esto permitiendo que el alcance sea mayor ya que evita problemas por potencia, interferencia o obstrucciones. Un paquete puede pasar por varios digipeaters para tener un mayor alcance 5. Por último, están los igate que son estaciones que están conectadas a internet lo que permite que reciba las señales de radio y las transmita a la red APRS-IS que es una red de internet lo que permite tener un mayor alcance. Los mismos igate se encargan de realizar el proceso de pasar la información de internet a radio. 6

II-B. LoRa

El LoRa (*Long Range*) es una tecnología de comunicación inalámbrica patentada que se caracteriza por un alto alcance con un consumo bajo de energía útiles para enviar pequeñas cantidades de datos a distancias largas. Tiene un alcance de entre 13 km a 15 km. LoRa utiliza una modulación de espectro ampliado y su principal ventaja es su comunicación a larga distancia que se mantiene estable ante interferencias. Por sus características la vuelve ideal para redes IoT. Es una tecnología inalámbrica y utiliza modulación en radiofrecuencia Chirp Spread Spectrum (CSS) 7

La tecnología LoRa utiliza la frecuencia de 868 MHz en Europa, 433 MHz en Asia y en América utiliza la frecuencia de 915 MHz. Estas frecuencias están en un espectro de frecuencias que se pueden utilizar sin licencia 8

Las principales aplicaciones que se pueden encontrar para un LoRa es para proyectos de internet de las cosas (*IoT*) que se alimentan por medio de baterías. Además son útiles para conectar dispositivos a larga distancia. Se pueden encontrar aplicaciones en agricultura y riego inteligente, para casas y edificios inteligentes. En cuanto a ciudades se pueden utilizar para estacionamiento inteligente, monitoreo de residuos, alumbrado público, para seguimiento del transporte público. También se puede utilizar para sistemas de seguridad y monitorización de paneles fotovoltaicos. Puede servir para supervisar la infraestructura. Además, se pueden encargar de medir el consumo de energía, así como medir y detectar fugas de agua o gas. 9

Existen módulos basados ESP32 que utilizan chips LoRa en el mercado como lo son los casos de: TTGO t-beam

que utiliza un chip sx1276 operando en frecuencias de 433 MHz, 868 MHz y 915 MHz; TTGO LoRa 32 que opera en frecuencias 868 MHz y 915 MHz; Heltec WiFi LoRa que opera en frecuencias 868 MHz y 915 MHz, LilyGo T-Impulse que utiliza las frecuencias 868 MHz y 915 MHz ;y Ebyte E32 que utiliza un chip sx1278 y opera en frecuencia de 433 MHz.

II-C. Componentes del Sistema iGate LoRa/APRS

En esta sección se describen los conceptos y componentes principales necesarios para el funcionamiento del iGate LoRa/APRS:

- **iGate:** Puente entre la red LoRa e Internet, que recibe y retransmite los datos. Requiere un receptor de radio LoRa (generalmente en la frecuencia de 433 MHz) y un computador o mini-computador para conectarse a la web.
- **APRS (Automatic Packet Reporting System):** Sistema de radioaficionados que transmite paquetes de datos con información como posición, mensajes y datos ambientales.
- **LoRa (Long Range):** Tecnología de modulación que permite comunicación inalámbrica de largo alcance, mayor penetración de señal y bajo consumo energético.
- **Tracker:** Dispositivo LoRa, usualmente con módulo GPS, que envía automáticamente su posición y otros datos.
- **Frecuencia:** APRS-LoRa opera en frecuencias específicas; la banda de 433 MHz es común para estos sistemas.
- **Firmware:** Software específico instalado en las placas de desarrollo para que los dispositivos funcionen como iGate o tracker.
- **Placa de desarrollo:** Módulo electrónico con un módem LoRa, configurable como iGate o tracker mediante firmware.

II-D. Especificaciones del Hardware

En la figura 1 podemos observar las características que el T3 LoRa32 V1.6.1 – LILYGO® presenta.

Característica	Especificación
Microcontrolador	ESP32 (Wi-Fi 802.11 b/g/n y Bluetooth 4.2 BR/EDR & BLE)
Flash	4 MB (SPI)
SRAM	520 KB
Pantalla	OLED 0.96" (128x64) integrada
Módulo LoRa	SX1276 LoRa transceiver
Frecuencias	433 MHz (en este caso) / 868 MHz / 915 MHz (según modelo)
Interfaz	USB Type-C (programación y alimentación)
GPIOs	Compatible con ESP32 estándar (UART, SPI, I2C, ADC, DAC, PWM)
Alimentación	5V vía USB-C o batería Li-Po 3.7V (conector JST)
Carga de batería	Circuito de carga integrado para Li-Po
Antena	Conector SMA/IPEX para antena externa LoRa
Dimensiones	25.6 x 51.2 mm aprox.

Figura 1. Especificaciones del hardware T3LoRa32V1,6,1–LILYGO

II-D1. Conexiones de Hardware: En la figura 2 se pueden ver las conexiones del hardware LILYGO T3 LoRa32 V1.6.1

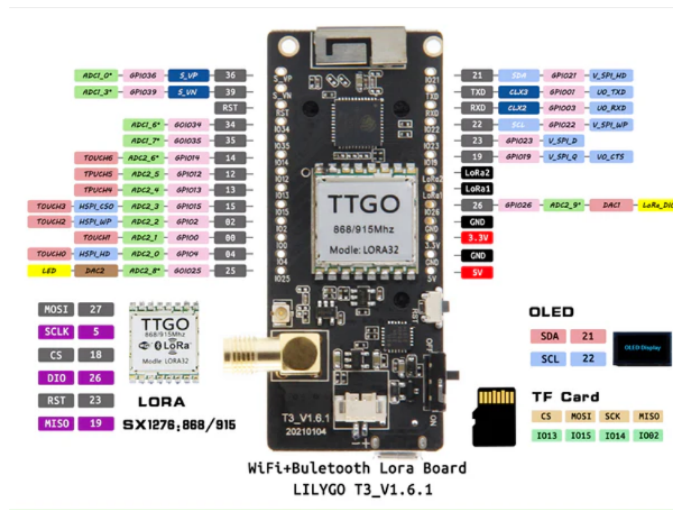


Figura 2. Diagrama de referencia de los buses del LilyGO T3 LoRa32. Muestra las conexiones principales de la placa con los periféricos y fuentes de alimentación.

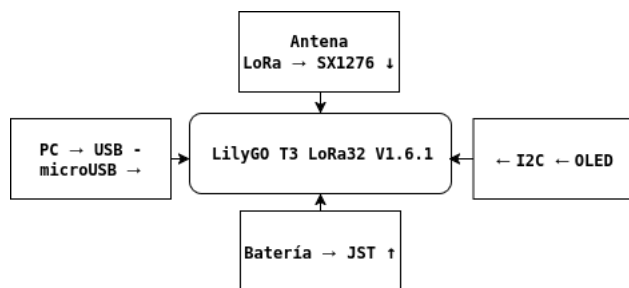


Figura 3. Diagrama de conexiones simplificado

Con el diagrama de la Figura 3, se busca detallar los elementos clave del hardware:

- **USB - Micro USB → PC:** Permite programar la placa y proporcionar alimentación mientras está conectada al computador.
- **Antena LoRa → SX1276:** El módulo SX1276 es el transceptor LoRa integrado en la placa. Se encarga de transmitir y recibir paquetes de datos en la frecuencia LoRa (433 MHz en nuestro caso). La antena se conecta a este módulo para mejorar la cobertura y la calidad de la señal.
- **OLED → I2C:** La pantalla OLED de 0.96" se comunica mediante I2C, mostrando información del iGate, como el número de paquetes recibidos o el estado de la conexión.
- **Batería → JST:** Conector para una batería Li-Po de 3.7 V, que permite que el iGate funcione sin necesidad de estar conectado al USB.

III. DISEÑO Y APLICACIÓN

Como aplicación para este sistema se pretende utilizar el LoRa iGate para el seguimiento en competiciones deportivas

como por ejemplo carreras o ciclismo. Cada corredor, ciclista o atleta lleva un pequeño tracker LoRa APRS. En diferentes puntos estratégicos de la competencia se colocan iGates. Cada iGate recibe la señal LoRa y la reenvía automáticamente a la red APRS-IS a través de Internet. Los datos recopilados se pueden visualizar en plataformas como aprs.fi, o integrarse en un mapa personalizado del evento.

El iGate se configurará mediante programación directa en la placa LilyGO T3 LoRa32, siguiendo un flujo planificado que garantice la recepción y transmisión correcta de los datos. La implementación se realizará en varias etapas:

1. Inicialización de Hardware y Firmware:

- Configuración del microcontrolador ESP32 y del módulo LoRa SX1276.
- Inicialización de la pantalla OLED para mostrar información de estado.
- Verificación de la conexión a la fuente de alimentación (USB o batería Li-Po).

2. Conexión a Internet:

- Establecimiento de la conexión WiFi para enviar los paquetes recibidos hacia APRS-IS.
- Implementación de control de errores para reconectar automáticamente en caso de caída de la red.

3. Recepción de Paquetes LoRa:

- Escucha continua de la frecuencia de 433 MHz para recibir paquetes de los trackers.
- Validación de que los paquetes cumplan con el formato correcto de APRS-LoRa.

4. Procesamiento y Registro de Datos:

- Almacenamiento de los datos en un log local para seguimiento y depuración.
- Visualización de información resumida en la pantalla OLED (por ejemplo, número de paquetes recibidos y estado de la conexión WiFi).

5. Transmisión a APRS-IS:

- Reenvío automático de los paquetes validados a la red APRS-IS.
- Implementación de reintentos en caso de fallos de envío.

6. Ciclo Continuo de Operación:

- Repetición indefinida de los pasos anteriores, garantizando la disponibilidad del iGate como puente entre los trackers LoRa y la red de Internet.

IV. DISEÑO PLANTEADO

IV-A. Diagrama de bloques

El diagrama de bloques de flujo representa de manera simplificada la arquitectura funcional del iGate LoRa/APRS. Permite visualizar cómo interactúan los distintos componentes del sistema para recibir, procesar y reenviar datos APRS. En la figura 4 podemos observar el diagrama de bloques planteado.

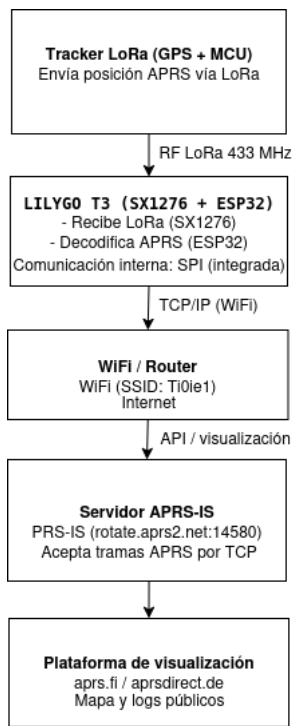


Figura 4. Diagrama de bloques del flujo de trabajo del iGate LoRa/APRS

IV-B. Pseudocódigo del iGate LoRa/APRS

El funcionamiento del iGate LoRa/APRS se puede resumir en el siguiente pseudocódigo:

1. Inicio del programa

2. Configuración:

- Definir la red WiFi (nombre y contraseña).
- Definir las credenciales de APRS-IS (indicativo y passcode).
- Definir el servidor APRS-IS (dirección y puerto).

3. Inicialización (Setup):

- Iniciar la comunicación serial para depuración.
- Conectar a la red WiFi.
 - Si la conexión WiFi falla, detener el programa.
- Configurar e iniciar el módulo LoRa.
 - Si el módulo LoRa falla, detener el programa.
- Conectar al servidor APRS-IS.
 - Si la conexión es exitosa, enviar la línea de login con indicativo, passcode y versión.
 - Si la conexión falla, mostrar un mensaje de error.

4. Bucle principal (Loop):

- Revisar si hay un paquete LoRa disponible.
 - Si hay un paquete:
 - Leer el mensaje completo del paquete LoRa.
 - Mostrar el mensaje recibido en el monitor serial.
 - Si la conexión a APRS-IS está activa:
 - ◊ Construir un mensaje APRS para reenvío (indicativo, >APRS, mensaje recibido).
 - ◊ Enviar el mensaje a APRS-IS.

- ◊ Mostrar el mensaje enviado en el monitor serial.
- Enviar beacon cada 30 segundos.
 - Comprobar si han pasado 30 segundos desde el último envío de beacon y si la conexión a APRS-IS está activa.
 - Si es así:
 - Construir el mensaje de beacon APRS (indicativo, >APRS, ubicación fija y mensaje de prueba).
 - Enviar el mensaje a APRS-IS.
 - Mostrar el mensaje de beacon enviado en el monitor serial.
 - Reiniciar el temporizador del beacon.

5. Fin del programa

IV-C. Máquina de Estados

El diagrama de estados muestra la máquina de estados interna del firmware del iGate, describiendo cómo cambia el comportamiento del sistema según las condiciones de operación. En la figura 5 se puede apreciar este diagrama.

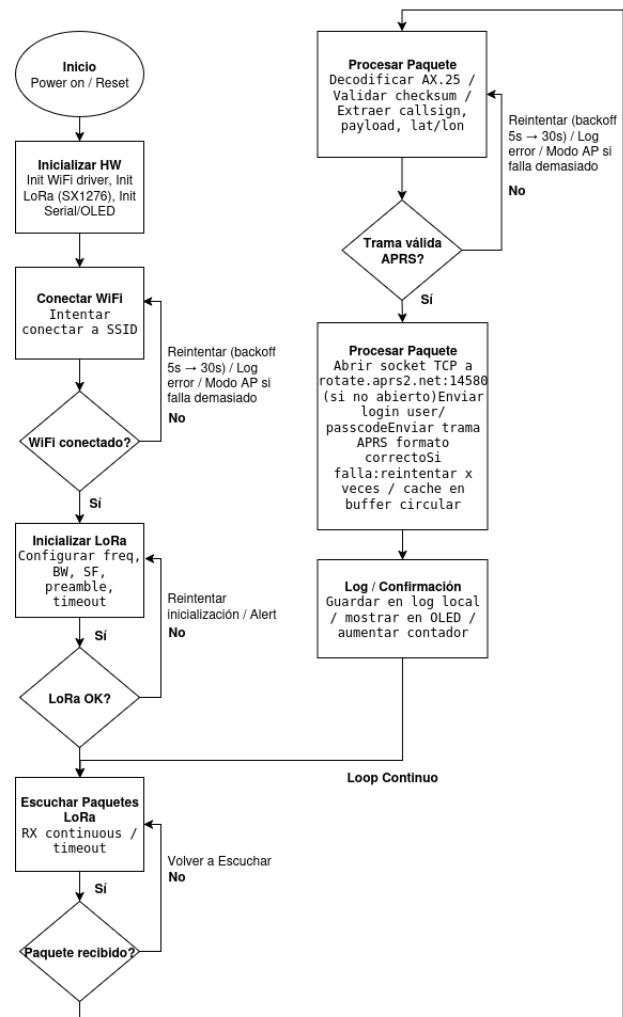


Figura 5. Máquina de Estados del firmware del iGate LoRa/APRS

V. CÓDIGO

V-A. Preparación del Entorno y Programación

Para el desarrollo del iGate LoRa/APRS se utilizó Arduino IDE en Linux (Ubuntu). La preparación del entorno y la configuración de la placa se realizó siguiendo los pasos descritos a continuación:

1. **Instalación del IDE:** Se descargó la última versión de Arduino IDE y se otorgaron los permisos de ejecución correspondientes. Además, se instalaron las dependencias necesarias para su funcionamiento en Ubuntu. Posteriormente, el IDE se ejecutó en modo sin sandbox para evitar problemas de compatibilidad.
2. **Configuración para ESP32:** En Arduino IDE se agregó la URL de los paquetes para ESP32 en *Archivo* → *Preferencias* y se procedió a instalar el soporte de la placa ESP32 mediante el Gestor de Placas. Se instalaron las librerías necesarias para la programación del iGate, incluyendo LoRa para el módulo SX1276, WiFi para la conectividad y TinyGPS++ de manera opcional para simulación de GPS.
3. **Conexión de la LilyGO T3 LoRa32:** La placa se conectó al computador mediante un cable USB-Micro USB. En Arduino IDE se seleccionó el puerto correspondiente y se configuró la placa LilyGO T3 LoRa32 para iniciar la programación y las pruebas iniciales.

V-B. Código Preliminar iGate LoRa APRS

Este código inicializa un iGate LoRa APRS en un ESP32. Se conecta a WiFi, recibe paquetes LoRa y los reenvía a la red APRS-IS. Además, envía un beacon de prueba cada 30 segundos. La comunicación LoRa se hace mediante el módulo SX1276 conectado por SPI, y APRS-IS mediante la librería APRSIS.

1. Preparar los datos:

- **Datos de Configuración importantes:**

```
1 // WiFi
2 const char* ssid = "TU_WIFI";
3 const char* password = "TU_PASSWORD";
4
5 // APRS-IS
6 const char* callsign = "TU_CALLSIGN";
7 const char* aprsPasscode = "TU_PASSCODE";
8 const char* aprsServer = "euro.aprs2.net";
9 const int aprsPort = 14580;
```

2. Conectar el hardware:

- Conecta la LilyGO T3 LoRa32 al PC usando el USB-Micro.
- Asegúrate de tener antena LoRa conectada, aunque sea solo para pruebas de recepción.
- Puedes usar la batería Li-Po para pruebas sin USB, pero no es obligatorio.

3. Seleccionar placa y puerto en Arduino IDE:

- Abrir Arduino IDE.
- Herramientas → Placa → ESP32 Arduino → LilyGO T3 LoRa32 (o ESP32 genérico si no aparece).
- Herramientas → Puerto y seleccionar el puerto USB donde está conectada la LilyGO.

4. Subir el código:

- Hacer clic en “Subir” (Upload) en Arduino IDE.
- Esperar a que compile y cargue.
- Abrir Monitor Serial a 115200 baudios para ver la salida en tiempo real.

5. Verificar funcionamiento:

- En el Monitor Serial deberías ver algo como:

```
Iniciando iGate LoRa APRS...
Conectando a WiFi...
WiFi conectado!
LoRa inicializado en 433.775 MHz
Conectado a APRS-IS!
Paquete recibido: T10IE1>APRS,TCPIP*:...
Paquete reenviado a APRS-IS
Beacon enviado a APRS-IS
```

- Cada paquete LoRa recibido se imprimirá y enviará a APRS-IS.
- Cada 30 segundos se envía un beacon de prueba automáticamente.

Código (C++): El código preliminar es el siguiente:

```
1 #include <WiFi.h>
2 #include <SPI.h>
3 #include <LoRa.h>
4
5 // --- Configuración de la red y APRS-IS ---
6 const char* ssid = "TU_WIFI";
7 const char* password = "TU_PASSWORD";
8
9 const char* callsign = "TU_CALLSIGN";
10 const char* passcode = "TU_PASSCODE";
11 const char* aprsServer = "euro.aprs2.net";
12 const int aprsPort = 14580;
13
14 WiFiClient client;
15
16 void setup() {
17   Serial.begin(115200);
18
19   --- Conexión WiFi ---
20   Serial.println("Conectando a WiFi...");
21   WiFi.begin(ssid, password);
22   while (WiFi.status() != WL_CONNECTED) {
23     delay(500);
24     Serial.print(".");
25   }
26   Serial.println("\nWiFi conectado");
27
28   --- Inicialización LoRa ---
29   LoRa.setPins(18, 14, 26);
30   if (!LoRa.begin(433775E3)) {
31     Serial.println("Error LoRa");
32     while(1);
33   }
34
35   --- Conexión a APRS-IS ---
36   Serial.println("Conectando a APRS-IS...");
37   if (client.connect(aprsServer, aprsPort)) {
38     Serial.println("Conectado a APRS-IS");
39     client.print("user_"); client.print(callsign);
40     client.print("_pass_"); client.print(passcode);
41     client.print("_vers_ESP32_iGate_1.0\r\n");
42   } else {
43     Serial.println("Error de conexión a APRS-IS");
44   }
45
46 void loop() {
47   int packetSize = LoRa.parsePacket();
48   if (packetSize) {
49     String loraMsg = "";
50     while (LoRa.available()) loraMsg += (char)LoRa.read();
51     Serial.println("LoRa recibido: " + loraMsg);
52
53     if (client.connected()) {
54       String aprsMsg = String(callsign) + ">APRS,TCPIP*: " +
55         loraMsg;
```

```

    client.print(aprsMsg + "\r\n");
    Serial.println("Enviado_a_APRS-IS:" + aprsMsg);
}

static unsigned long lastBeacon = 0;
if (millis() - lastBeacon > 30000 && client.connected())
{
    String beacon = String(callsign) + ">APRS,TCPIP
*:!0903.50N/07902.45W-Test_iGate";
    client.print(beacon + "\r\n");
    Serial.println("Beacon_enviado:" + beacon);
    lastBeacon = millis();
}
}

```

VI. PRUEBAS PRELIMINARES

Durante las primeras pruebas con el iGate LoRa/APRS, se realizaron las siguientes observaciones:

VI-A. Compilación y carga del código

El código inicial se compiló y cargó al LilyGO sin errores en el IDE.

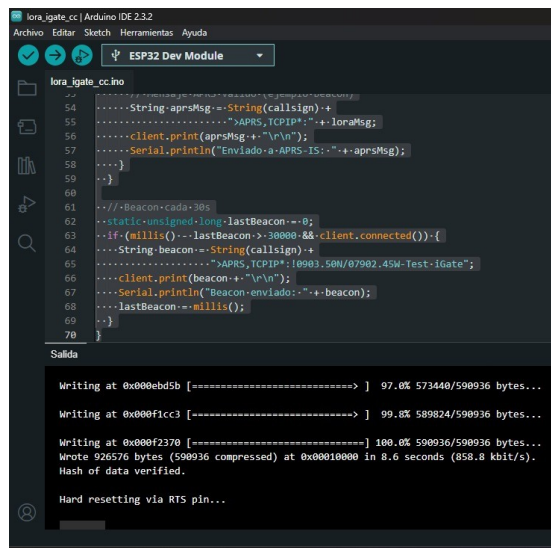


Figura 6. Monitor serial mostrando que el código se cargó correctamente y no se presentan errores.

VI-B. Modo Listening del LilyGO

Al encenderse, el dispositivo mostraba en la pantalla el mensaje **Listening**, indicando que estaba listo para recibir paquetes LoRa.

VI-C. Visualización en APRS.fi

Buscando el callsign del iGate en aprs.fi, el dispositivo aparecía en la ubicación programada, confirmando la conexión correcta con la red APRS-IS.

VI-D. Recepción de Raw Packets

Se observaron *raw packets* en la página de APRS.fi.

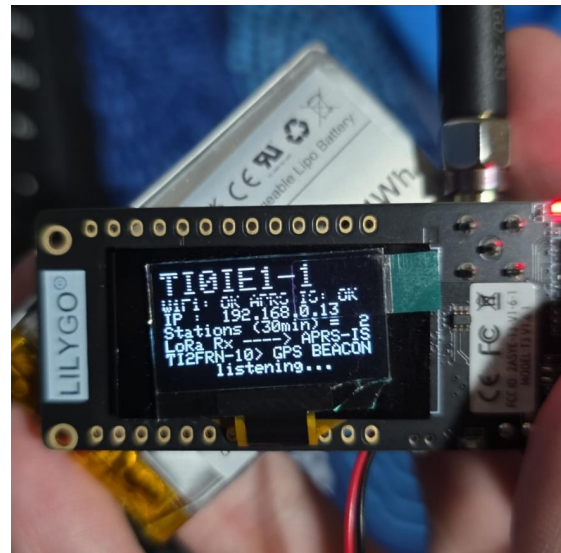


Figura 7. Pantalla del LilyGO mostrando *Listening*, indicando que el iGate está activo y a la espera de paquetes.

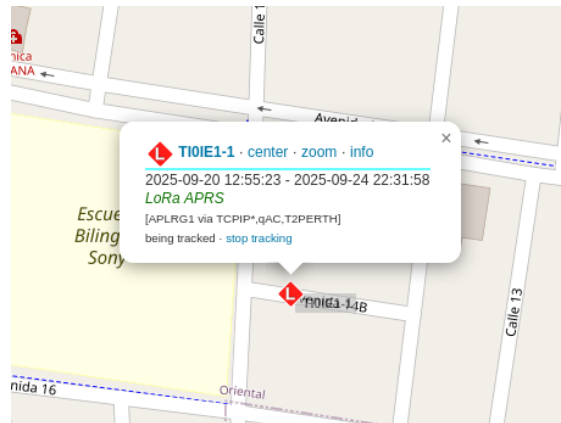


Figura 8. Mapa de APRS.fi mostrando la ubicación del iGate.

VI-E. Prueba de Beacons

Al enviar beacons personalizados, el iGate transmitió correctamente las coordenadas fijas configuradas en el código, ignorando la posición obtenida por el GPS integrado. De esta forma, los beacons se publicaron en APRS-IS mostrando las coordenadas de prueba programadas, las cuales corresponden a una ubicación en Panamá.

Las pruebas preliminares confirman que el iGate recibe y retransmite paquetes LoRa a APRS-IS, visualizables en el mapa y en *raw packets*. Además, el envío de beacons personalizados funcionó como se esperaba, evidenciando la ubicación de prueba en Panamá según los datos definidos en el firmware.

Este resultado valida el correcto control de las tramas de beacon y permitirá continuar con la integración de nuevas funciones, como la alternancia entre posiciones fijas de prueba y datos provenientes del GPS, según los requerimientos de la aplicación.

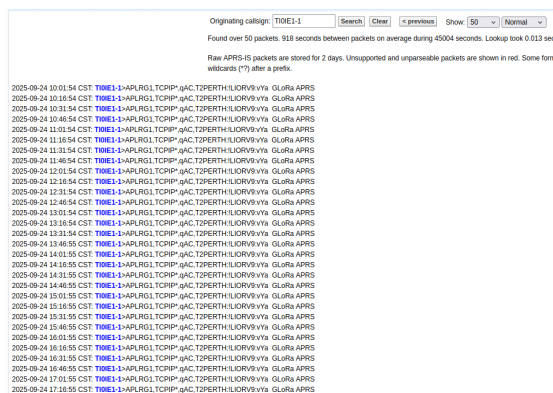


Figura 9. Sección de *raw packets* en APRS.fi. Esto confirma que el iGate recibe y retransmite correctamente los paquetes LoRa a APRS-IS.

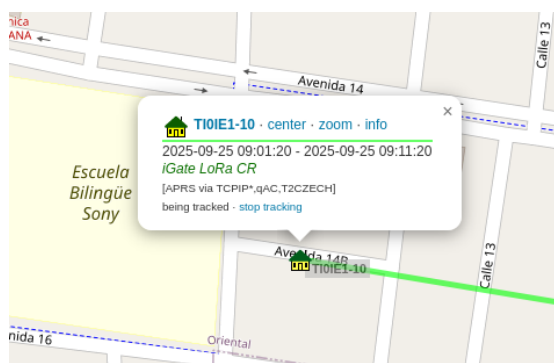


Figura 10. Sección de *raw packets* en APRS.fi. Esto confirma que el iGate recibe y retransmite correctamente los paquetes LoRa a APRS-IS.

VII. CONCLUSIONES

Las pruebas preliminares realizadas con el iGate LoRa/APRS confirman que el sistema cumple su función principal de recibir paquetes LoRa de los trackers y retransmitirlos a la red APRS-IS, permitiendo su visualización en plataformas como APRS.fi y la obtención de *raw packets*. El código inicial se cargó correctamente en la LilyGO T3, y el dispositivo opera en modo *Listening*, mostrando su disponibilidad para recibir información.

Adicionalmente, el envío de beacons personalizados funcionó de manera satisfactoria, transmitiendo las coordenadas fijas configuradas en el firmware y evidenciando en APRS-IS la ubicación de prueba en Panamá. Este comportamiento confirma que el sistema puede alternar entre datos manuales y automáticos, ofreciendo flexibilidad para futuras implementaciones.

En general, los resultados preliminares permiten validar la conectividad, la correcta integración hardware–software y la fiabilidad de la transmisión de datos. Además, brindan una base sólida para la depuración y refinamiento del sistema, asegurando que el iGate esté listo para su aplicación en el seguimiento de competiciones deportivas u otras

implementaciones de APRS-LoRa.

VIII. CRONOGRAMA

El cronograma actualizado para la entrega preliminar se puede visualizar a continuación:

Tabla I
CRONOGRAMA PRELIMINAR DE ACTIVIDADES DEL PROYECTO

Semana	Actividad / Objetivo	Avance Estimado
5	Configuración de entorno de programación y pruebas iniciales con la LilyGO T3.	Completado
6	Desarrollo inicial del código: - Recepción de paquetes LoRa - Conexión a WiFi - Estructura básica para envío a APRS-IS	Completado
7	Integración hardware-software y pruebas de funcionamiento: - Conectar antena LoRa - Verificar envío de datos a APRS-IS - Depurar errores iniciales	Completado
8	Entrega parcial (Informe y Presentación Parcial): - Diagramas de bloques y máquina de estados - Listado de hardware y justificación técnica - Planteamiento estructurado del diseño - Implementación en hardware y bus de conexión - Tipo de comunicación de cada periférico - Pseudo-código para control del sistema - Definición de tramas de datos - Cronograma y presupuesto para las semanas restantes - Código inicial en GitHub documentado	Completado
9-15	Implementación y Evaluación Continua: - Pruebas de recepción y transmisión de trackers - Depuración y refinamiento del código - Commits regulares y documentados en GitHub - Visualización de datos en APRS.fi y aprsdirect.de	En progreso
16	Defensa del Proyecto Final: - Entrega de informe completo - Presentación final - Código final documentado y funcional - Confirmación de publicación de datos en APRS	Planeado

IX. BIBLIOGRAFÍA

REFERENCIAS

- [1] Mario. (s. f.). Que es y cómo funciona el sistema APRS – Club de Radioexperimentadores de Cd. Juarez. Recuperado de <https://crecj.org/que-es-y-como-funciona-el-sistema-aprs/>
- [2] Qué es y cómo funciona el APRS. (s. f.). Radio Club Capitol. Recuperado de <https://www.radioclubcapitol.es/formacion/aprs>
- [3] Guía definitiva de APRS para radioaficionados. (s. f.). Radio Club Capitol. Recuperado de <https://www.radioclubcapitol.es/formacion/aprs/guia-aprs>
- [4] APRS (Automatic Packet Reporting System). (s. f.). Recuperado de <https://radioaficionados.mx/aprs-automatic-packet-reporting-system/>
- [5] Digipeaters – VK APRS Network. (s. f.). Recuperado de <https://aprs.net.au/info/digipeaters/>
- [6] Admin. (2022, 5 mayo). IGATE y DIGIPEATER APRS – blog de elihuzero. Recuperado de <https://blog.elihuzero.com.mx/uncategorized/igate-y-digipeater-aprs/>
- [7] LORA: ¿Qué es y para qué sirve? - AlaiSecure - España. (2022, 19 abril). AlaiSecure - España. Recuperado de <https://alaisecure.es/glosario/lora-que-es-y-para-que-sirve/>
- [8] Tecnología LORA y LoRAWAN - CatSensors. (s. f.). Recuperado de <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>

- [9] Alfaiot-Webmaster. (2024, 20 marzo). LoRa, una tecnología LPWAN ideal para el Internet de las Cosas. AlfaIoT. Recuperado de <https://alfaiot.com/iot/lora-una-tecnologia-lpwan-ideal-para-el-internet-de-las-cosas/>