

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE CIENCIAS APLICADAS
CARRERA DE TELECOMUNICACIONES



RADIO DEFINIDA POR SOFTWARE
PRUEBA 1B

Docente:

Ing. Edgar Maya.

Estudiante:

Anthony Miguel Panamá Chicaiza¹.
Diego Patricio Montezuma Caragulla².
Israel Sebastián Narváez Guevara³.

¹ampanamac@utn.edu.ec

²damontezumar@utn.edu.ec

³isnarvaezg@utn.edu.ec

TEMA

CAPTURA Y DEMODULACIÓN DE SEÑAL LORA MEDIANTE SDR ANGEL

OBJETIVOS

- **Objetivo general.**

Analizar y desarrollar un método eficiente para la captura y demodulación de señales LoRa utilizando un RTL-SDR V3 mediante el software SDRangel.

- **Objetivos específicos.**

1. Configurar adecuadamente el entorno de trabajo en SDRangel para la captura de señales LoRa, incluyendo la selección de dispositivos de hardware compatibles y la configuración de parámetros de captura.
2. Diseñar e implementar un proceso de demodulación de señales LoRa utilizando SDRangel, abordando aspectos como la selección de algoritmos de demodulación y la optimización de parámetros de configuración.
3. Evaluar la eficacia y el rendimiento del sistema propuesto mediante pruebas experimentales de captura y demodulación de señales LoRa en entornos controlados, analizando métricas como la tasa de éxito de demodulación y la calidad de la señal recuperada.

INTRODUCCIÓN

En el ámbito de las comunicaciones inalámbricas, la tecnología LoRa (Long Range) ha ganado prominencia debido a su capacidad para proporcionar conectividad de largo alcance y bajo consumo de energía, lo que la hace ideal para una amplia gama de aplicaciones IoT (Internet de las cosas). En este contexto, el software definido por radio (SDR) se presenta como una herramienta versátil y poderosa para capturar y demodular señales LoRa. En este estudio, exploramos el uso de SDRangel, una plataforma de código abierto para SDR, para la captura y demodulación de señales LoRa. Analizaremos el proceso de configuración del sistema, la adquisición de señales LoRa y su posterior demodulación, así como también discutiremos las posibles aplicaciones y ventajas de esta tecnología en diversos escenarios de comunicación inalámbrica.

MARCO TEORICO

RTL-SDR V3.

El RTL-SDR es un dispositivo USB de bajo costo, aproximadamente \$30, que puede funcionar como un escáner de radio basado en computadora para recibir señales de radio en tiempo real en un área determinada, sin necesidad de conexión a Internet. Dependiendo del modelo específico, es capaz de captar frecuencias que oscilan entre los 500 kHz y 1,75 GHz. Gran parte del software asociado con el RTL-SDR es desarrollado por la comunidad y se ofrece de manera gratuita. Es importante tener en cuenta que este dispositivo no está diseñado para la transmisión de señales [1].

El concepto del RTL-SDR se origina a partir de los dongles sintonizadores de televisión DVB-T, los cuales se producían en masa y estaban basados en el chipset RTL2832U. Mediante la colaboración de Antti Palosaari, Eric Fry y Osmocom

(especialmente Steve Markgraf), se descubrió la posibilidad de acceder directamente a los datos I/Q sin procesar en el chipset RTL2832U. Esto permitió transformar el sintonizador de televisión DVB-T en un receptor de radio definido por software de banda ancha, utilizando un controlador de software personalizado desarrollado por Steve Markgraf. Para aquellos que han beneficiado del proyecto RTL-SDR, se sugiere considerar hacer una contribución a Osmocom a través de Open Collective, ya que fueron ellos quienes desarrollaron los controladores y dieron vida al RTL-SDR [1].

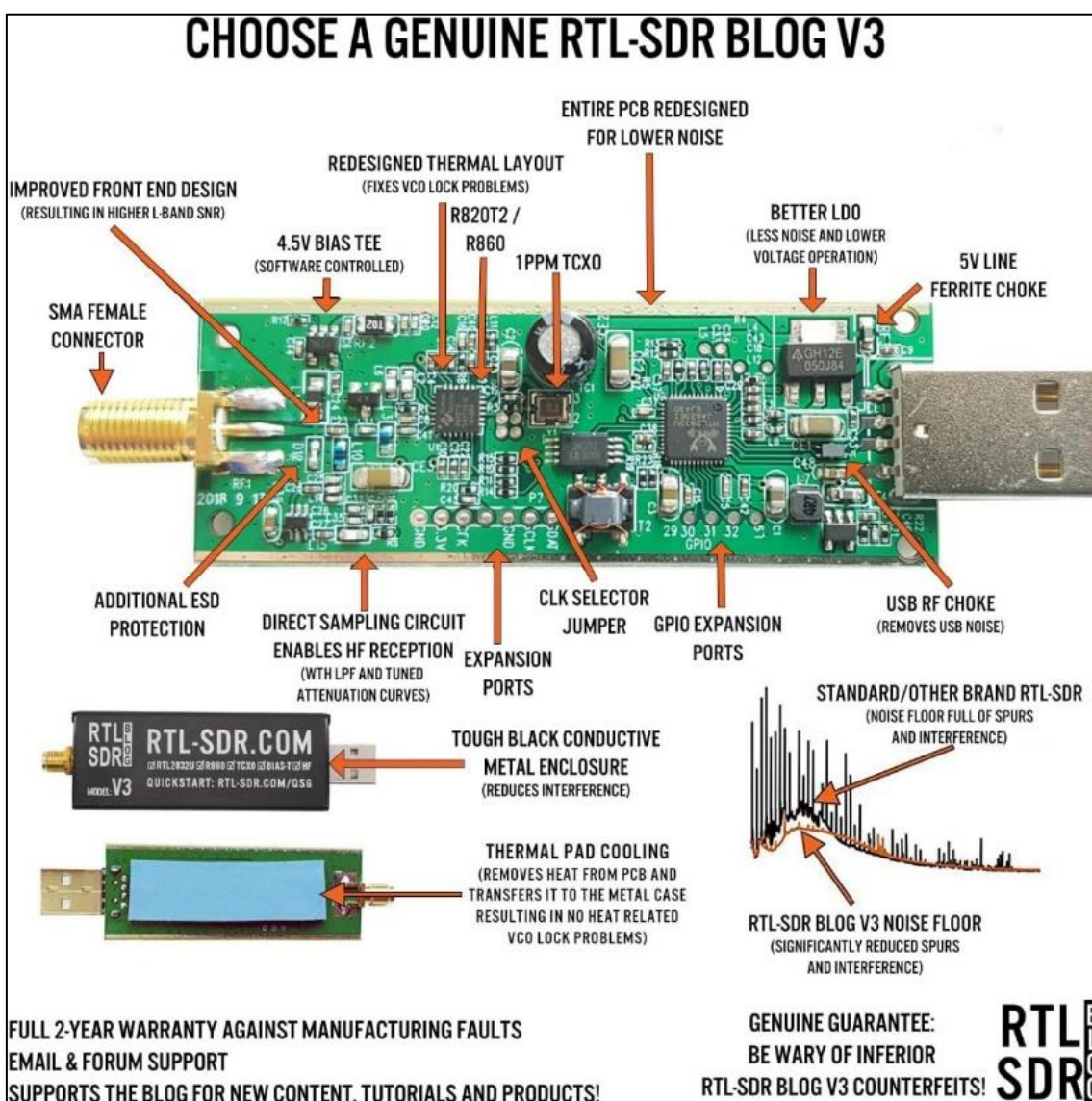


Figura 1. Esquema gráfico del SDR-RTL [1].

¿Cuáles son algunas aplicaciones de escáner de radio RTL-SDR?

Aplicaciones
Supervisión de frecuencias utilizadas por la policía.
Recepción de comunicaciones de servicios de emergencia como EMS, ambulancias y bomberos.
Escucha de conversaciones de control de tráfico aéreo.
Seguimiento de la ubicación de aeronaves mediante decodificación ADSB.
Decodificación de mensajes cortos ACARS de aviones.
Monitoreo de conversaciones en sistemas de radio troncal.
Decodificación de transmisiones de voz digitales no encriptadas como P25, DMR y D-STAR.
Seguimiento de la ubicación de embarcaciones marítimas mediante decodificación AIS.
Decodificación de tráfico de buscapersonas POCSAG/FLEX.
Detección de teléfonos inalámbricos y monitores para bebés.
Recepción y seguimiento de datos de globos meteorológicos lanzados por agencias meteorológicas.
Monitoreo de globos de gran altitud lanzados personalmente para recuperar cargas útiles.
Recepción de señales de sensores de temperatura y medidores de potencia inalámbricos.
Sintonización de radioaficionados en la banda VHF.
Decodificación de paquetes APRS de radioaficionados.
Visualización de transmisiones de televisión analógicas.
Monitoreo de señales GSM.
Utilización del RTL-SDR en dispositivos Android como un escáner de radio portátil.
Recepción y decodificación de señales GPS.
Uso del RTL-SDR como analizador de espectro.
Recepción de imágenes de satélite meteorológico NOAA.
Escucha de satélites y la Estación Espacial Internacional (ISS).
Realización de actividades de radioastronomía y mediciones de líneas de hidrógeno galáctico.
Seguimiento de la dispersión de meteoritos.
Sintonización de emisoras de radio FM y decodificación de información RDS.
Recepción de radiodifusión DAB.
Escucha y decodificación de HD-Radio (NRSC5).
Utilización del RTL-SDR como adaptador para radios de hardware tradicionales.
Decodificación de señales de terminales de datos móviles de taxis.
Uso del RTL-SDR como fuente de alta calidad de entropía para generación de números aleatorios.
Utilización del RTL-SDR como indicador de factor de ruido.
Ingeniería inversa de protocolos desconocidos.
Triangulación de la fuente de una señal.
Identificación de fuentes de ruido de RF.
Caracterización de filtros de RF y medición de ROE de antenas.
Decodificación de señales de satélites geosíncronicos Inmarsat STD-C EGC.
Escucha de la ISS (Estación Espacial Internacional).
Recepción de imágenes de satélites meteorológicos geoestacionarios.

Tabla 1. Aplicaciones en que este dispositivo se puede aplicar [1].

¿Cuál es el rango de frecuencia RTL-SDR y sus características?

SDR-RTL-.COM							
MODELO Y MARCA	ARQUITECTURA Y TECNOLOGIA	ANCHO DE BANDA	PUERTOS	ESPECIFICACIONES	PRECIO	OTROS	FRECUENCIA
RTL-SDR BLOG V3.	Tecnología LORA, Radio fusión AM, Radios móviles,	Hasta los 2.4 estables.	USB RF. DAT. CLK. GND. CRK. 3.3v. GND. GPIO PORTS.	Conector SMA. Chip Sintonizador R820T2 Polarización de 4.5V. Diplexor sintonizado a 25MHz. Preamplificador de 10 dB. Filtro de pasa bajos de 24MHz.	34 USD	Impedancia de entrada 50Ohmios (270-280 mA).	500 KHz -1766 MHz (500 KHz -24MHz modo de muestreo directo).

Tabla 2. Recopilación de información sobre SDR-RTL [1].

Con que software se puede utilizar el SDR-RTL

El RTL-SDR puede ser utilizado con una variedad de programas y software que permiten acceder y procesar las señales recibidas. Algunos de los programas más populares incluyen:

- **SDR# (SDRSharp):** Es un popular software de radio definido por software (SDR) que es fácil de usar y proporciona una interfaz gráfica intuitiva para la recepción y decodificación de señales.
- **HDSDR:** Otro software de SDR con una interfaz de usuario bastante flexible y capacidades avanzadas para el procesamiento de señales.
- **CubicSDR:** Un programa multiplataforma (disponible para Windows, Mac y Linux) que ofrece una interfaz simple pero poderosa para la recepción de señales.
- **GNU Radio:** Es una herramienta de procesamiento de señales de código abierto que proporciona una gran flexibilidad y capacidades avanzadas para el desarrollo de aplicaciones de radio definidas por software.
- **GQRX:** Similar a CubicSDR, GQRX es otro programa multiplataforma que proporciona una interfaz gráfica para la recepción de señales con dispositivos SDR como RTL-SDR.
- **RTL1090:** Específicamente diseñado para la recepción y decodificación de datos de aeronaves utilizando dispositivos RTL-SDR.

- **Dump1090:** Similar a RTL1090, Dump1090 es un programa específico para la decodificación de tráfico ADS-B de aviones.
- **PDW:** Este programa se utiliza para la decodificación de mensajes POCSAG/FLEX, comúnmente utilizados en sistemas de buscapersonas.

SDR Angel.

SDRangel emplea módulos de origen de muestra para adquirir datos de entrada I/Q desde un dispositivo hardware. Posteriormente, estos datos pueden ser procesados en la banda de paso de retorno, potencialmente con una reducción de la tasa de muestreo, mediante uno o varios módulos de canal de recepción para llevar a cabo operaciones como demodulación, decodificación o análisis de porciones específicas del espectro.

En contraste, SDRangel utiliza módulos receptores de muestra para enviar datos de salida I/Q a un dispositivo hardware. Estos datos pueden ser generados mediante uno o varios módulos de canal de transmisión para producir señales moduladas, que luego se mezclan en una banda de paso de transmisión, pudiendo ser interpoladas antes de su envío al dispositivo [2].

La interfaz de usuario se organiza en espacios de trabajo, dentro de los cuales se disponen distintos componentes como dispositivos, espectros principales, canales y funciones. Estos elementos de la interfaz pueden ser ajustados en tamaño y movidos libremente para permitir al usuario personalizar la disposición global según sus preferencias. Se tiene la opción de trabajar con múltiples espacios de trabajo y trasladar componentes entre ellos según sea necesario.

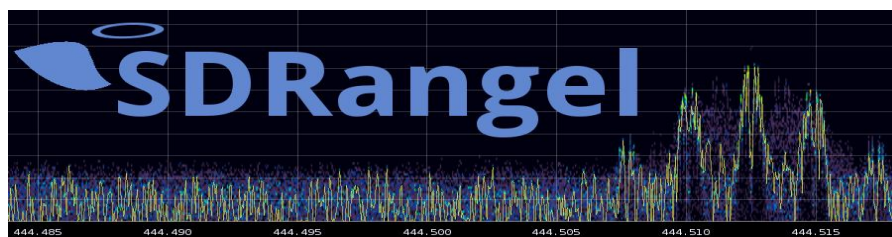


Figura 2. SDR Ángel [2] .

Obteniendo el programa

Compilar desde la fuente es una de las opciones recomendadas para instalar el software. Se pueden encontrar instrucciones detalladas para sistemas operativos como Ubuntu Linux, Windows y Mac.

Otra alternativa recomendada es construir una imagen de Docker, especialmente aplicable para sistemas Linux que cuentan con Docker instalado. Los archivos Docker y los scripts necesarios se encuentran en el proyecto auxiliar "sdrangel-docker", específicamente en la carpeta "sdrangel". Este método evita la necesidad de compilar manualmente desde el código fuente, ya que las imágenes se generan periódicamente con Dockerfile, asegurando así la utilización de la receta más actualizada.

Para aquellos que no deseen utilizar ninguna de las dos opciones anteriores, existe la posibilidad de instalar una distribución binaria. Aunque es factible crear un instalador para Windows desde el propio sistema, muchos usuarios prefieren utilizar el instalador distribuido. En la sección de versiones, para cada versión de la rama "master", el sistema CI/CD proporcionará dos archivos correspondientes a esa versión.

Para usuarios de Windows 10 y 11, se ofrece un instalador específico. Este binario ha sido creado mediante la canalización de Actions CI/CD utilizando MSVC 2022, y por lo tanto está diseñado para ser utilizado en estos sistemas operativos exclusivamente. El archivo contiene un instalador autoextraíble denominado "sdrangel-versión-win64.exe", que se puede ejecutar simplemente haciendo doble clic en él. A partir de ahí, el usuario puede seguir los pasos guiados por los diferentes menús para instalar el programa en la ubicación predeterminada o en otra de su elección.

Wifi LoRa V3 (Heltec).

Es una placa de desarrollo que integra el chip ESP32 de Espressif Systems con un módulo LoRa, lo que permite a los desarrolladores crear aplicaciones que combinan las capacidades de conectividad Wi-Fi y Bluetooth del ESP32 con la larga distancia y la baja potencia de comunicación del protocolo LoRa. El ESP32 es un microcontrolador de bajo consumo de energía con conectividad Wi-Fi y Bluetooth, mientras que el módulo LoRa proporciona una comunicación de largo alcance en bandas de frecuencia no licenciadas.

Esta combinación hace que el ESP32 LoRa V3 sea ideal para una variedad de aplicaciones de IoT, como monitoreo remoto, seguimiento de activos y sensores ambientales.[3]



Figura 3. ESP32 LoRa V3

MODELO Y MARCA	ARQUITECTURA	AB	PUERTOS	ESPECIFICACIONES	FRECUENCIA	POTENCIA TX	PRECIO
ESP32 LoRa V3	ESP32-S3FN8 (Xtensa@32-bit 1x7 dual core processor)	125KHz 250KHz 500KHz	Puertos GPIO. Comunicación (UART, SPI e I2C). Puerto USB Antena.	WiFi integrado, LoRa, Bluetooth tres conexiones de red, Wi-Fi integrado, antena de resorte metálico de 2,4 GHz dedicada a Bluetooth, interfaz IPEX (U.FL) reservada para uso de LoRa. Chip CP2102 USB a puerto serie integrado, conveniente para descargar programas e imprimir información de depuración.	470-510MHz 863-928MHz	21±1dBm	\$23.50

Tabla 3. Características de WiFi LoRa V3.

DESARROLLO

Instalación de librería en Arduino IDE.

Como primer paso, es necesario realizar la instalación de las librerías que comprenden el funcionamiento de la placa Wifi LoRa V3, a las cuales se tiene acceso mediante los instructivos proporcionados por el propio fabricante HELTEC. Para ello se ingresa en Arduino IDE y en la pestaña de archivos, se selecciona la opción de “Preferencias”, tal como se observa en la siguiente figura.

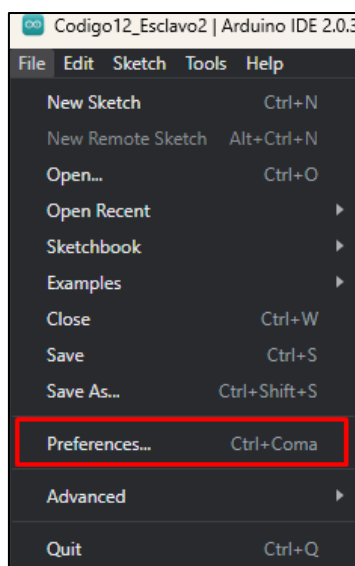


Figura 4. Inicio de instalación de librerías en Arduino IDE.

Una vez seleccionado el apartado anterior, se despliega una ventana como la que se ve a continuación, en la cual se debe seleccionar la opción marcada en rojo, con el objetivo de abrir un editor de URL.

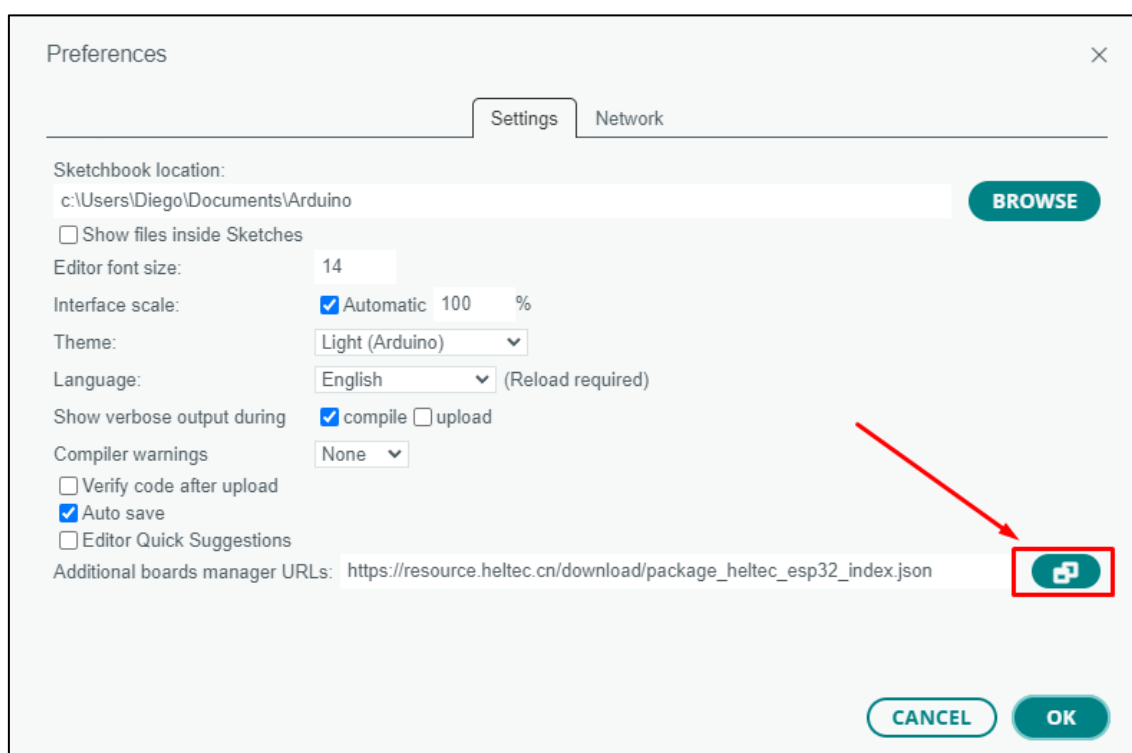


Figura 5. Ventana de asignación de preferencias.

En la siguiente figura se observa la ventana que se nos despliega, aquí vamos a ingresar el URL que se muestra en pantalla, este permite la descarga de recursos desde la

página de HELTEC, el cual descargara paquetes. json que permite interactuar con la placa WiFi LoRa V3.

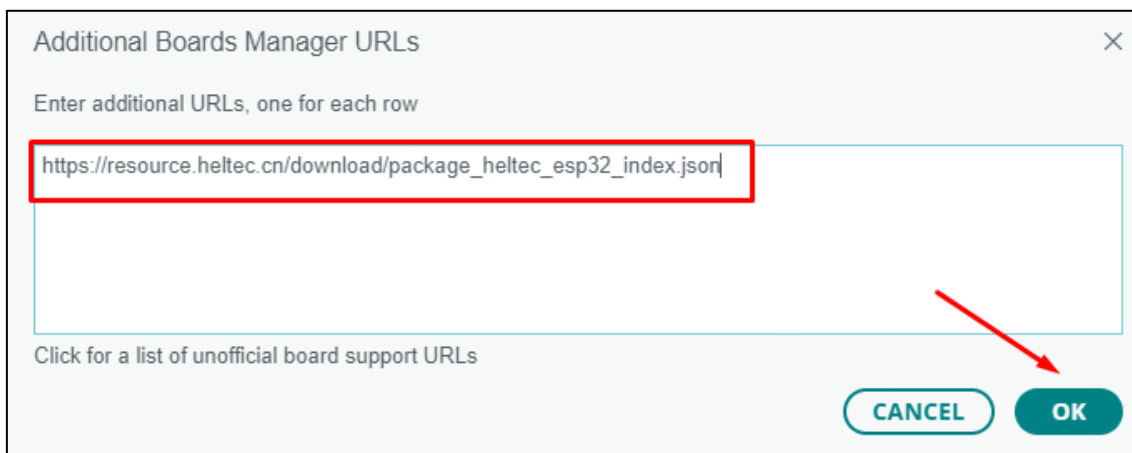
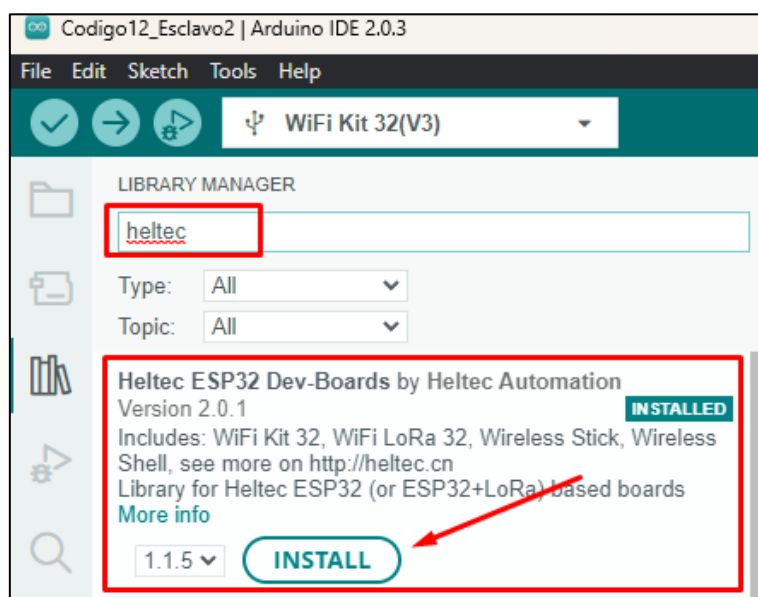


Figura 6. Ingreso de URL para descarga de recursos de HELTEC.

Una vez realizada esta acción, es necesario dirigirse al apartado de “Administración de librerías”, aquí vamos a ingresar en el buscador de librerías y mediante la palabra “heltec” se realiza el filtro de estas. Seleccionaremos la denominada “Heltec ESP32 Dev-Boards by Heltec Automation”, procedemos a instalar esta librería.



Script para el envío de señales con LoRa.

Cuando ya se tenga instaladas las librerías, se procede a la apertura del ejemplo que vamos a utilizar para la transmisión de datos, para este caso vamos a la sección “Archivos”, posteriormente seleccionaremos “Ejemplos”, bajamos hacia el apartado de “Heltec ESP32 Dev-Boards” y seleccionaremos la opción de “LoRa Basics”, por último,

el ejemplo que debemos seleccionar es el denominado “LoRa Sender”, este nos permite el envío de datos mediante una frecuencia específica.

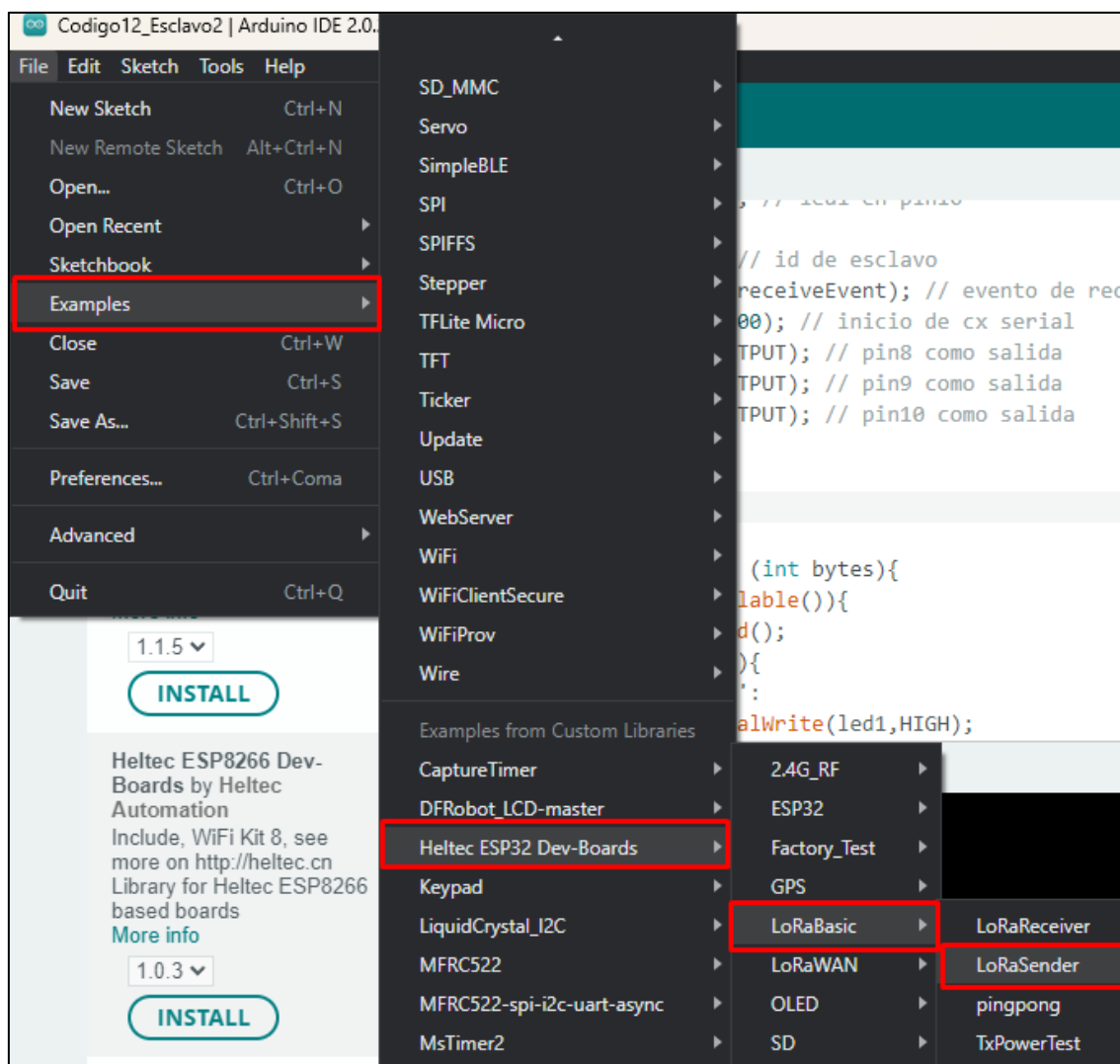
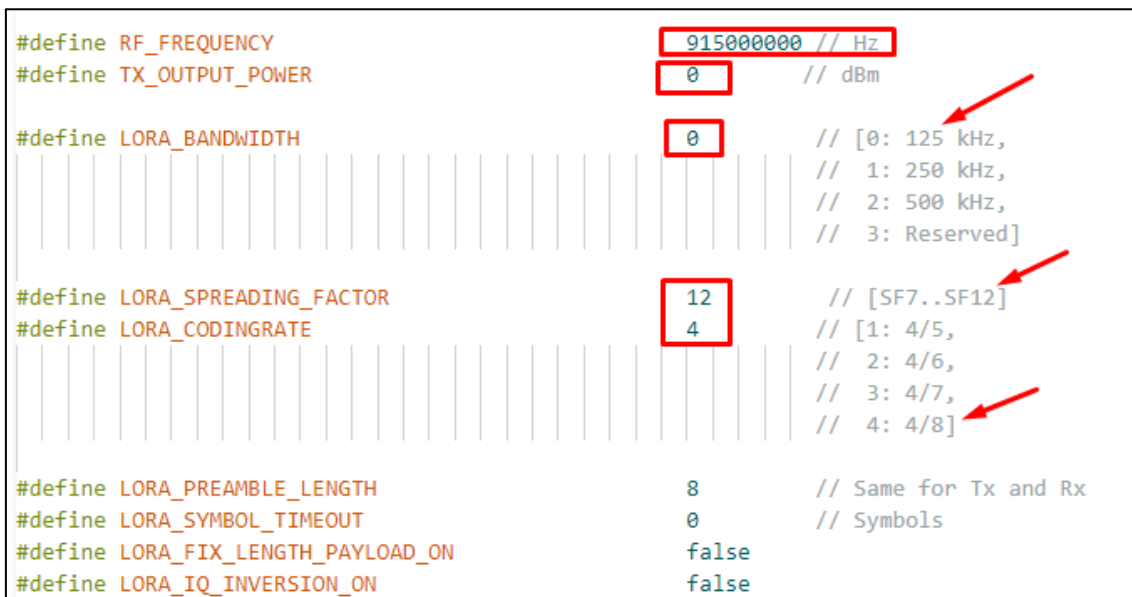


Figura 7. Apertura del ejemplo de envío de datos que se prevé utilizar en la comunicación.

Como se observa en la figura siguiente, el código que vamos a ejecutar se ha proporcionado correctamente, en este caso, vamos a modificar algunos parámetros esenciales para poder realizar el correcto envío de datos desde LoRa. Dichos datos son los siguientes:

- **Frecuencia:** En este caso vamos a utilizar una frecuencia de 915 MHz.
- **Potencia de transmisión:** Utilizaremos una potencia de transmisión de 0dBm, puesto que el alcance de la señal no será muy extenso.
- **Ancho de banda:** En el caso del ancho de banda, utilizaremos la primera opción que es de 125 kHz.

- **Factor de dispersión LoRa:** Este factor determina la tasa de dispersión espectral de la señal, esta afecta la velocidad de transmisión de datos y la distancia de cobertura de la red. Generalmente los factores de dispersión de LoRa van desde 7 a 12, entre más bajo el factor de dispersión, es decir 7, la velocidad de transmisiones será más alta, pero es más propensa al ruido, de igual forma utiliza distancias de propagación más bajas. Mientras que un factor de dispersión de 12 generara una velocidad retransmisión más baja, pero con grandes distancias de transmisión y menos propenso al ruido.
- **Tasa de codificación LoRa:** Este parámetro se relaciona con la forma en la que se codifican y decodifican en las comunicaciones LoRa. Esta tasa de codificación se puede expresar como 4/5, 4/6, 4/7 o 4/8. Esta relación nos indica la relación que hay entre los bits de información transmitidos y los bits de redundancia utilizados para realizar el proceso de corrección de errores. Es decir que si elegimos una configuración de 4/8, esto nos indica que por cada 4 bits de información se agregan 4 bit de redundancia para corregir la información. Una tasa de 4/8 es más robusta ante errores de transmisión, puesto que hay más bits redundantes, pero reduce la cantidad de información útil que se puede transmitir en un determinado ancho de banda (Menor eficiencia espectral), lo contrario e una tasa de 4/5, que proporciona más eficiencia espectral pero es más propensa a errores de transmisión.



```

#define RF_FREQUENCY          915000000 // Hz
#define TX_OUTPUT_POWER      0 // dBm

#define LORA_BANDWIDTH        0 // [0: 125 kHz,
                                // 1: 250 kHz,
                                // 2: 500 kHz,
                                // 3: Reserved]

#define LORA_SPREADING_FACTOR 12 // [SF7..SF12]
#define LORA_CODINGRATE       4 // [1: 4/5,
                                // 2: 4/6,
                                // 3: 4/7,
                                // 4: 4/8]

#define LORA_PREAMBLE_LENGTH 8 // Same for Tx and Rx
#define LORA_SYMBOL_TIMEOUT  0 // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false

```

Figura 8. Configuración de parámetros de transmisión en el código.

También podemos modificar los datos que queremos enviar, normalmente se enviaría un “Hello World”, pero en este caso enviaremos un texto acorde al trabajo realizado, como se puede observar en la siguiente figura.

```
void loop()
{
    if(lora_idle == true)
    {
        delay(1000);
        txNumber += 0.01;
        sprintf(txpacket, "Prueba 1B - SDR - Montezuma_Narvaez_Panama %0.2f", txNumber); //start a package
        Serial.printf("\r\nsending packet \"%s\" , length %d\r\n", txpacket, strlen(txpacket));
        Radio.Send( (uint8_t *)txpacket, strlen(txpacket) ); //send the package out
        lora_idle = false;
    }
    Radio.IrqProcess( );
}
```

Figura 9. Ingreso de texto a enviar por medio de la comunicación LoRa.

A continuación, se deja el código que se ha utilizado para el envío de datos desde la placa WiFi LoRa V3, el cual es uno de los ejemplos de las librerías de Heltec, dicho código lo podemos apreciar en su forma completa en las siguientes líneas.

Código

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#define RF_FREQUENCY          915000000 // Hz
#define TX_OUTPUT_POWER      0          // dBm
#define LORA_BANDWIDTH        0          // [0: 125 kHz,
                                         // 1: 250 kHz,
                                         // 2: 500 kHz,
                                         // 3: Reserved]
#define LORA_SPREADING_FACTOR 12         // [SF7..SF12]
#define LORA_CODINGRATE       4          // [1: 4/5,
                                         // 2: 4/6,
                                         // 3: 4/7,
                                         // 4: 4/8]
#define LORA_PREAMBLE_LENGTH  8          // Same for Tx and Rx
#define LORA_SYMBOL_TIMEOUT   0          // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON  false
#define RX_TIMEOUT_VALUE      1000
#define BUFFER_SIZE           30
```

```

char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];
double txNumber;
bool lora_idle=true;
static RadioEvents_t RadioEvents;
void OnTxDone( void );
void OnTxTimeout( void );

void setup() {
    Serial.begin(115200);
    Mcu.begin(HELTEC_BOARD,SLOW_CLK_TPYE);
    txNumber=0;
    RadioEvents.TxDone = OnTxDone;
    RadioEvents.TxTimeout = OnTxTimeout;
    Radio.Init( &RadioEvents );
    Radio.SetChannel( RF_FREQUENCY );
    Radio.SetTxConfig( MODEM_LORA, TX_OUTPUT_POWER, 0, LORA_BANDWIDTH,
                      LORA_SPREADING_FACTOR, LORA_CODINGRATE,
                      LORA_PREAMBLE_LENGTH, LORA_FIX_LENGTH_PAYLOAD_ON,
                      true, 0, 0, LORA_IQ_INVERSION_ON, 3000 );
}

void loop()
{
    if(lora_idle == true)
    {
        delay(1000);
        txNumber += 0.01;
        sprintf(txpacket,"Hello world number %0.2f",txNumber); //start a package
        Serial.printf("\r\nsending packet \"%s\" , length %d\r\n",txpacket,
strlen(txpacket));
        Radio.Send( (uint8_t *)txpacket, strlen(txpacket) ); //send the package out
        lora_idle = false;
    }
    Radio.IrqProcess( );
}

void OnTxDone( void )
{
    Serial.println("TX done.....");
    lora_idle = true;
}

void OnTxTimeout( void )
{
    Radio.Sleep( );
    Serial.println("TX Timeout.....");
    lora_idle = true;
}

```


Para realizar la carga del código a nuestra placa LoRa, es necesario realizar la búsqueda del dispositivo, como se puede observar en la siguiente figura, el dispositivo se denomina como “WiFi LoRa 32 (V3)”.

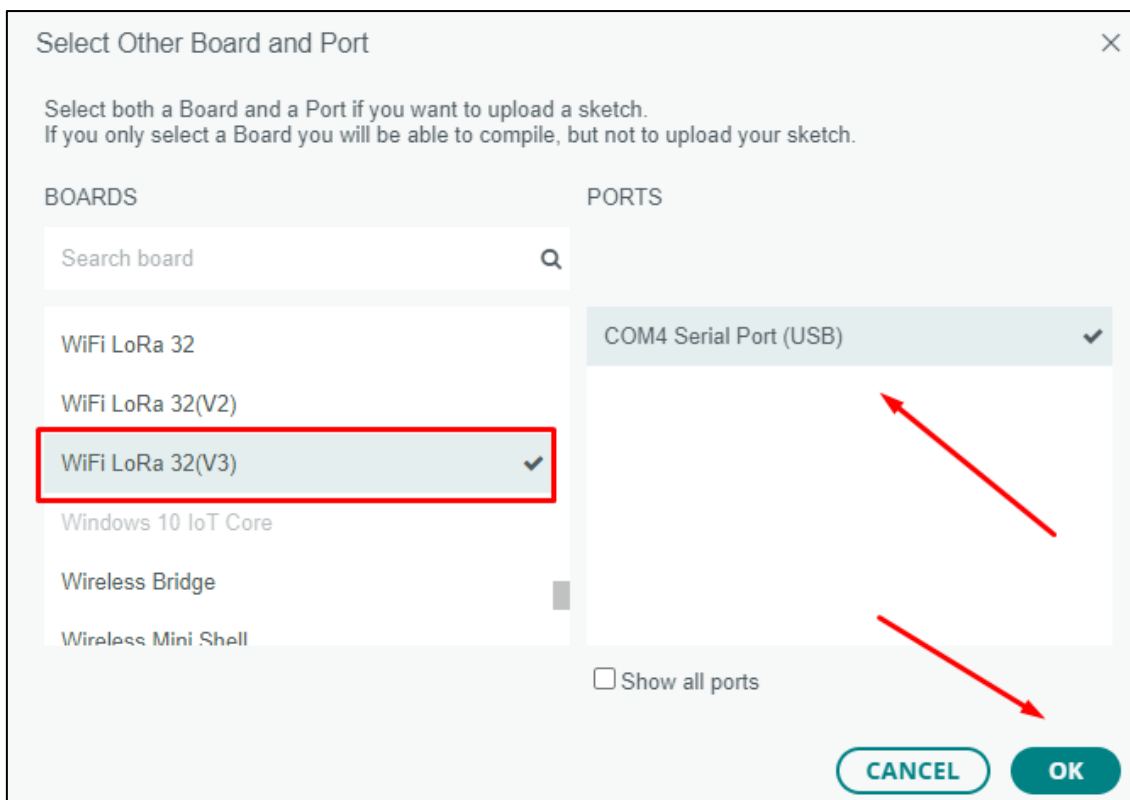


Figura 10. Selección de dispositivo para la carga del código.

Conexión de dispositivos.

Una Vez que se ha realizado todo el proceso anterior, se procede a la conexión del dispositivo a una fuente de alimentación con su respectiva antena de transmisión, como se aprecia en la siguiente figura. En este caso utilizamos una fuente por medio de su entrada tipo C conectada a la computadora personal.



Figura 11. Conexión de placa WiFi LoRa v3.

De igual forma se realiza la conexión del RTL-SDR V3, El cual nos ayudara a capturar las señales provenientes de LoRa en la frecuencia 915MHz, la conexión se realiza a la computadora personal mediante un puerto USB, mientras que en su otro extremos e realiza la conexión de una antena de tipo conejo, la cual captura las señales y las envía al RTL para su posterior administración en el software SDR Ángel.

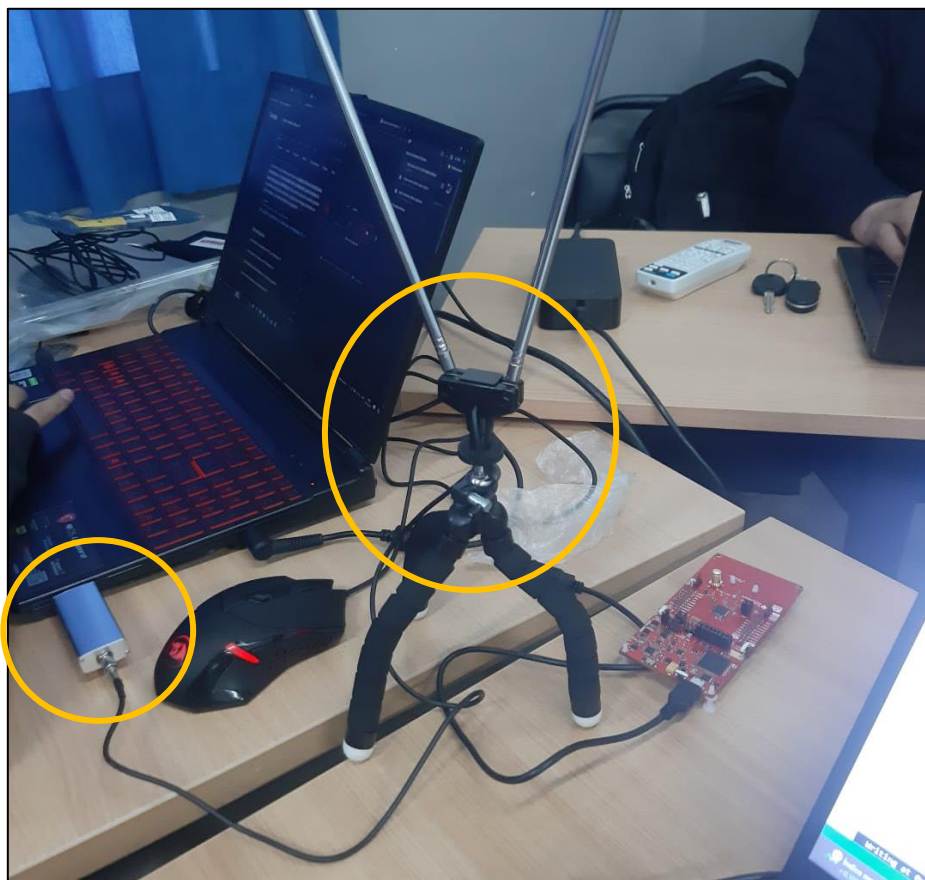


Figura 12. Conexión de RTL-SDR.

Configuración de RTL-SDR en SDR Ángel.

Una vez que se tenga toda la configuración planteada, es hora de abrir el software de SDR Ángel, como se observa en la siguiente figura, vamos a abrir una ventana de trabajo dando clic en el icono que se puede visualizar.

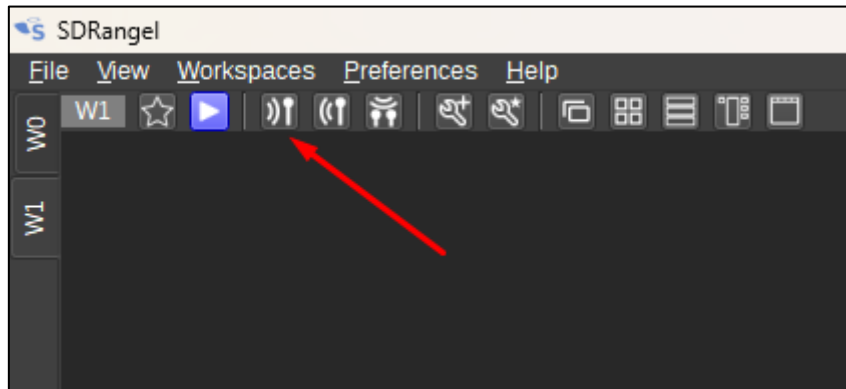


Figura 13. Apertura de ventana de trabajo en SDR Angel.

Se abrirá una ventana como la que se puede visualizar a continuación, en este caso debemos seleccionar el dispositivo RTL-SDR, el cual nos dará la interfaz para poder trabajar con este.

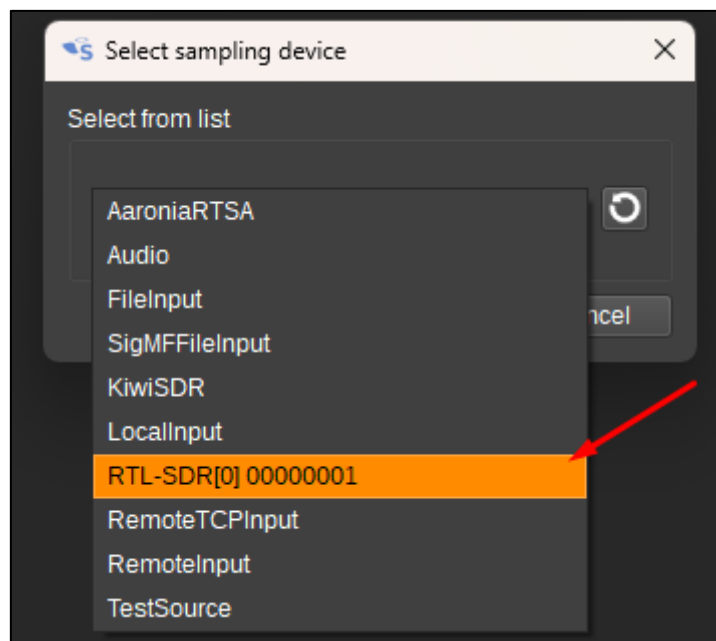


Figura 14. Selección de dispositivo RTL-SDR.

La interfaz que se genera se puede visualizar en la siguiente figura, como y aprecia este tiene en la parte izquierda el panel de configuraciones, mientras que en la parte derecha se tiene las gráficas de espectrogramas y espectros de frecuencia.

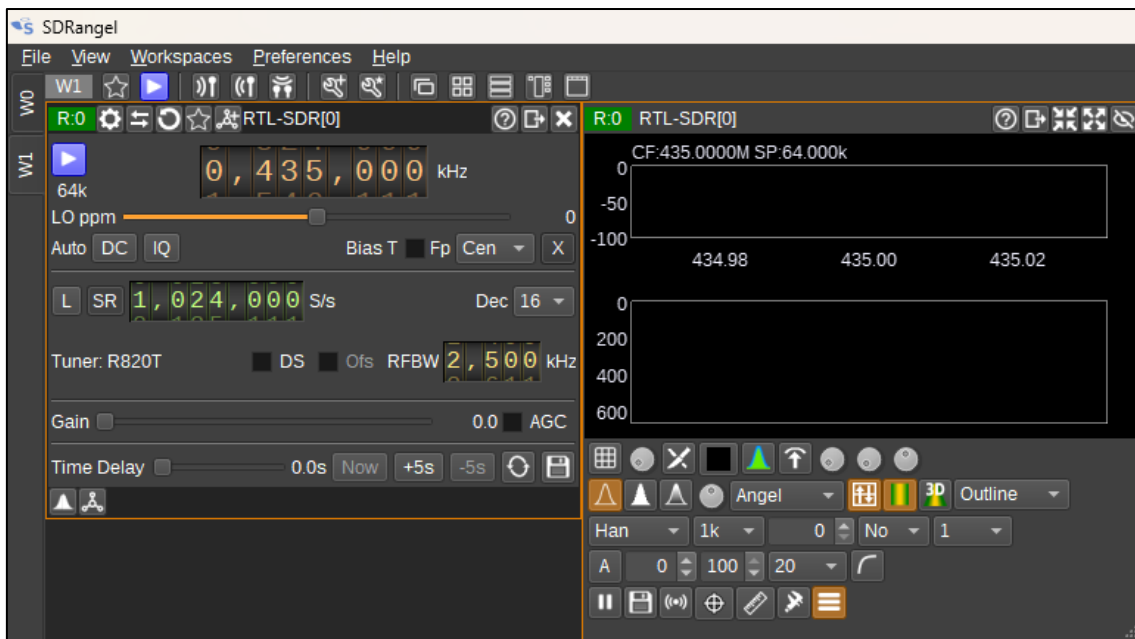


Figura 15. Interfaz de configuración para RTL-SDR.

Como primer paso, vamos a configurar la frecuencia de recepción de señal, en este caso debemos seleccionar la misma frecuencia a la que está transmitiendo el dispositivo LoRa, esta frecuencia es de 915 MHz, tal como se puede observar en la siguiente figura. De igual forma mediante la activación de DC, indicamos que se deshabilite la componente DC automática que viene por defecto.



Figura 16. Ingreso de frecuencia central 915 MHz.

Ahora se procede a configurar la tasa de muestreo de datos con la que se envía desde el dispositivo hacia el ordenador, para el caso podemos visualizar en la figura que esta tasa de muestreo se utiliza en 1000000 S/s. De igual forma también se establece el factor de decimación, el cual se utiliza para describir una reducción de la tasa de muestreo, en este caso se indica cuanto se reduce la tasa de muestreo en un factor de 4, por ende esta será de 250000 S/s.

Otro de los factores que se configuran en este apartado es el RFBW o ancho de banda de señal de radio, la cual se establece en 2500, puesto que sería el doble del ancho de banda original de 125 kHz.

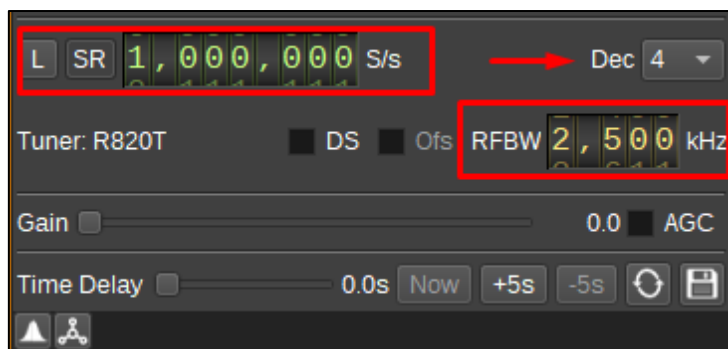


Figura 17. configuración de tasa de muestreo, ancho de banda y factor de decimación.

Una vez configurados estos factores, vamos a iniciar la captura de datos, para esto lo hacemos como se indica en la siguiente figura.

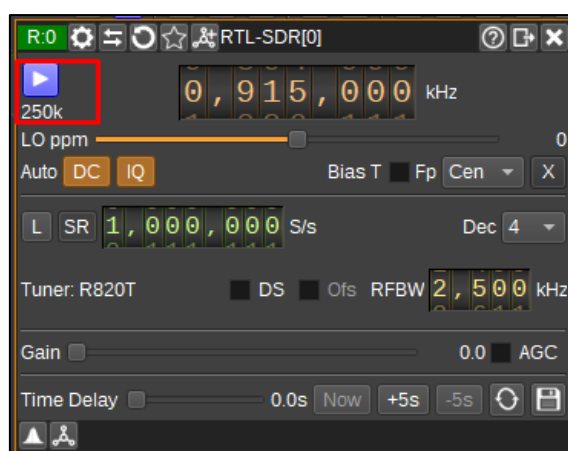


Figura 18. Inicio de captura de datos.

A continuación, en la siguiente figura se puede observar el espectro de frecuencia de la señal que se está capturando, así como el espectrograma respectivo. Aquí se puede observar que la frecuencia central que se está capturando es de 915 MHz.

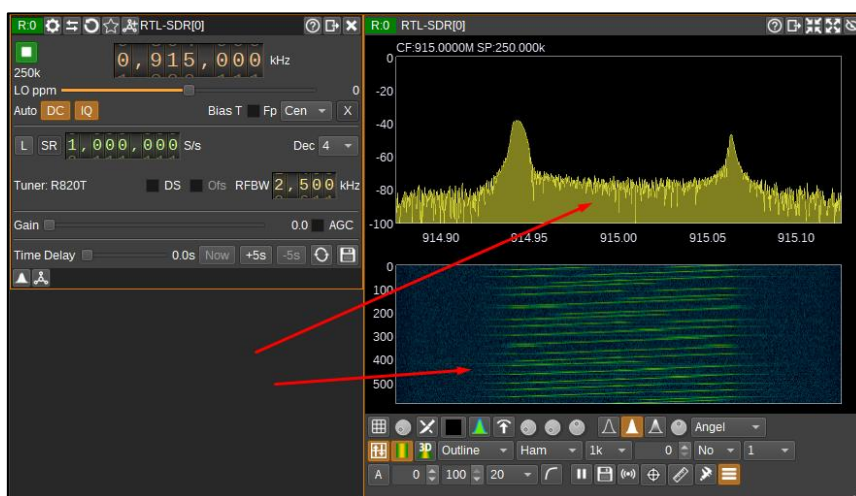


Figura 19. Visualización de espectrograma y grafica de espectro de frecuencia.

Ahora el siguiente paso es abrir el demodulador para poder capturar la información de los paquetes que está enviando el módulo LoRa, para ello vamos a dar clic tal como lo marca la siguiente figura.



Figura 20. Ingreso a la selección del demodulador.

Se abrirá un apartado como el siguiente, en este caso vamos a seleccionar la opción de “ChirpChat Demodulator”, este módulo nos ayudará a realizar la demodulación de señales LoRa. La modulación Chirp Spread Spectrum (CSS) es una técnica de dispersión espectral donde la señal se altera mediante una serie de pulsos denominados "chirps", los cuales experimentan cambios en su frecuencia a medida que transcurre el tiempo.

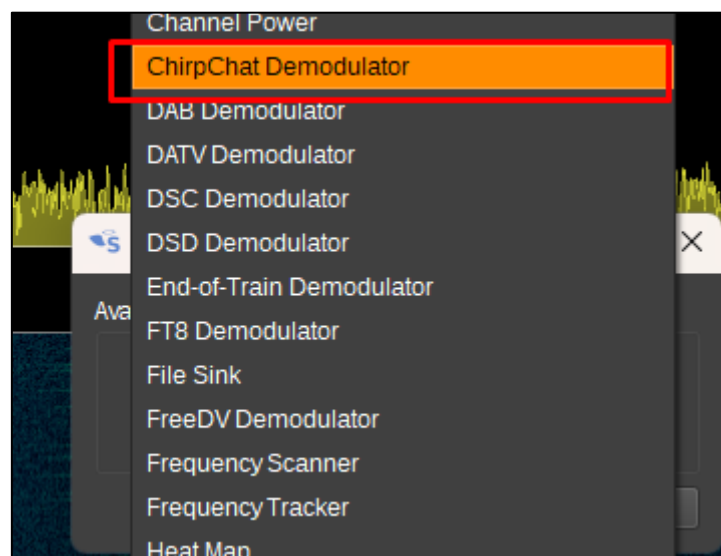


Figura 21. Elección del demodulador ChirpChat.

En este caso se nos abrirá el panel de configuraciones del demodulador, el cual se visualiza en la siguiente figura.

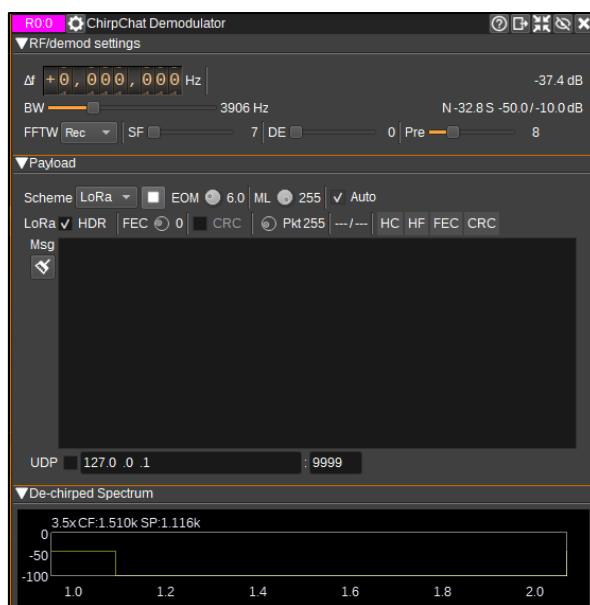


Figura 22. Panel de configuración del demodulador ChripChat.

Para realizar la configuración del demodulador, vamos a posicionar el ancho de banda en 125 kHz, esta es la modulación que está usando el dispositivo LoRa para realizar el envío de paquetes. También se va a realizar otras configuraciones como son:

Otras configuraciones que se establecen son el factor de dispersión, el cual se lo establece en 12, el cual fue establecido así en el script de LoRa. Otro factor es el DE, el cual se refiere a la cantidad de datos que se toman en cuenta para realizar la FFT y obtener los bins de frecuencia. Por último se establece el Pre en 8, este nos indica la cantidad de estos pulsos de preámbulo que se anticipa encontrar en una señal específica.

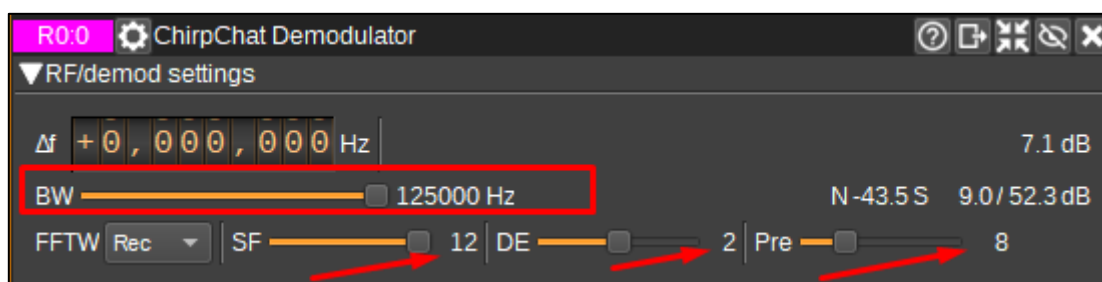


Figura 23. configuración de parámetros en el demodulador ChripChat.

RESULTADOS

Luego del proceso realizado anteriormente, se nos mostrara en la ventana de “Payload”, los datos que se están capturando a partir del demodulador de la señal LoRa, con los diferentes parámetros ya configurados. En este caso, como se puede observar en la siguiente figura, vemos como este es capaz de capturar el texto plano que está transmitiendo el dispositivo LoRa en la frecuencia de 915 MHz.

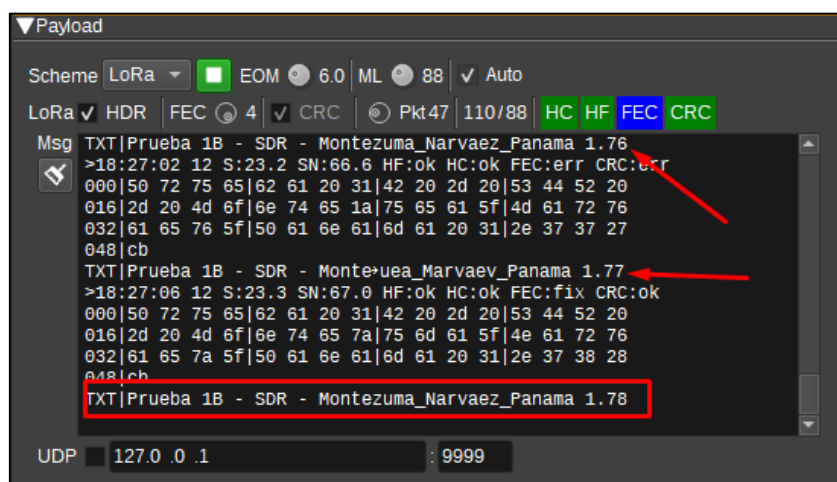


Figura 24. Captura y demodulación de paquetes para obtener la información transmitida por LoRa.

Mediante las ventanas de espectro de frecuencia, podemos visualizar ahora la frecuencia central de 915 MHz, en el cual el recuadro color violeta representa el ancho de banda de 125 kHz elegido en el demodulador para capturar la señal. En la parte baja también podemos visualizar el espectrograma de datos que se capturan en la señal transmitida.

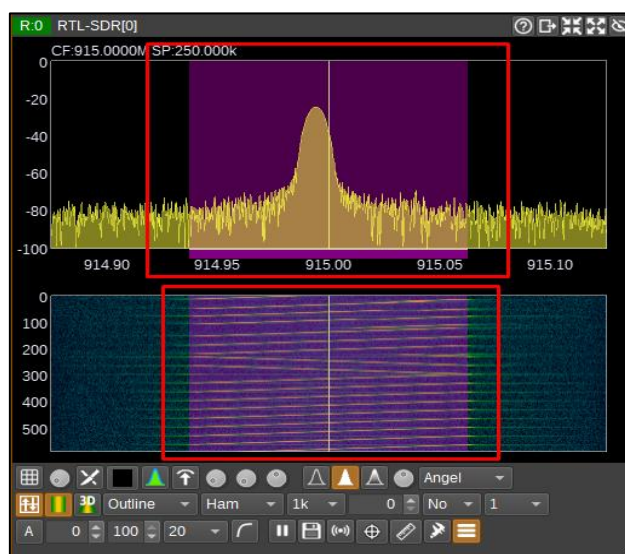


Figura 25. Espectrograma y espectro de frecuencia de la señal recibida.

De igual forma se puede apreciar este espectrograma en su forma 3D, en la cual se pueden distinguir los diferentes picos a lo largo del ancho de banda en el cual se está trabajando.

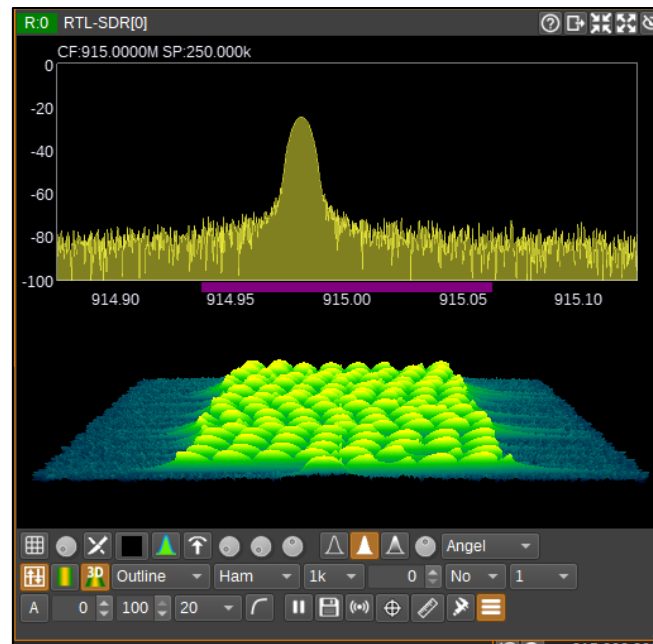


Figura 26. Espectrograma y espectro de frecuencia de la señal recibida en 3D.

En el espectrograma que se ha generado mediante el demodulador ChripChat, podemos visualizar los pulsos que se van generando a lo largo del ancho de banda de la señal capturada.

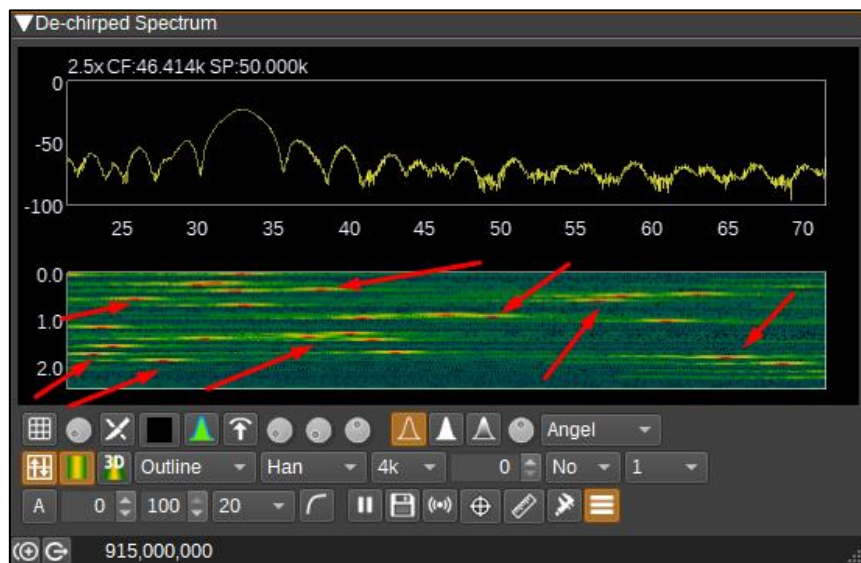


Figura 27. Pulsos generados en el espectrograma.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

En conclusión, una vez realizado todo el procedimiento de conexión, configuración tanto de frecuencias, factor de dispersión, tasa de codificación, ancho de banda, entre otros parámetros tanto en el dispositivo RTL-SDR V3 como en la placa WiFi LoRa V3, se ha podido realizar la correcta interceptación de datos provenientes desde esta comunicación, logrando visualizar el texto plano que está transmitiendo el dispositivo LoRa luego de realizar la demodulación con el uso del software SDR Angel.

Para lograr una comunicación, el RTL-SDR es capaz de sintonizar una gran gama de frecuencias, esto permite elegir diferentes canales de comunicación, con esto se evita posibles interferencias dentro del mismo canal utilizado y de igual manera lograr tener una señal más clara cambiando los valores de la tasa de muestreo, esto con el fin de producir una señal más clara sin mucho ruido.

Dentro de SDRAngel, es posible evidenciar los datos que se envían dentro de la frecuencia utilizada para la comunicación entre los dos hardware. Por otro lado, las diferentes gráficas anunciadas, por ejemplo, el chirped spectrum, donde indicaban los picos más altos por donde se envían los datos y muestrear la forma de onda en el dominio del tiempo a una frecuencia conectada.

En conclusión, el uso de SDRangel como herramienta para la captura y demodulación de señales LoRa presenta un potencial significativo en el ámbito de las comunicaciones inalámbricas. Durante el desarrollo de este estudio, se ha demostrado que SDRangel ofrece una plataforma versátil y flexible que permite configurar y adaptar fácilmente el sistema según las necesidades específicas de captura y demodulación de señales LoRa.

Recomendaciones:

Una de las recomendaciones que a las que se puede llegar luego de realizado este proceso, es el de la configuración, puesto que esta debe ser similar a las configuraciones que se realiza en el dispositivo LoRa, hablando de su tasa de codificación y factor de

dispersión, ya que tomo un tiempo llegar a relacionar los 2 parametros para que el SDR Angel pudiera realizar la demodulación.

Se recomienda tener conocimiento de las configuraciones realizadas en los dos dispositivos utilizados, esto con el fin de lograr la transmisión de los datos y recepción de la información de manera inalámbrica.

Se recomienda realizar pruebas con otras frecuencias y configuraciones dentro de ESP32 LoRa V3, esto con el fin de verificar si el mensaje llega incompleto, claro o con caracteres no identificados.

Antes de iniciar la captura y demodulación de señales LoRa, es crucial realizar una calibración adecuada del hardware SDR para garantizar resultados precisos y confiables. Esto incluye ajustar parámetros como ganancia, frecuencia de muestreo y ancho de banda para adaptarse a las características específicas de la señal LoRa que se desea analizar.

BIBLIOGRAFIA

- [1] RTL-SDR, «rtl-sdr.com,» 11 04 2013. [En línea]. Available: <https://www.rtl-sdr.com/about-rtl-sdr/>. [Último acceso: 28 04 2024].
- [2] E. Griffiths, «github.com,» 2022. [En línea]. Available: <https://github.com/f4exb/sdrangel/wiki/Quick-start>. [Último acceso: 28 04 2024].
- [3] “WiFi LoRa 32(V3) – Heltec Automation.” Accessed: Apr. 27, 2024. [Online]. Available: <https://heltec.org/project/wifi-lora-32-v3/>