# Descripción General

En un sistema operativo linux se ha instalado la libreria de LimeSuite la cual es un propuesta de código abierto para poder desarrollar sobre un software define Radio conocido como Lime7002M.

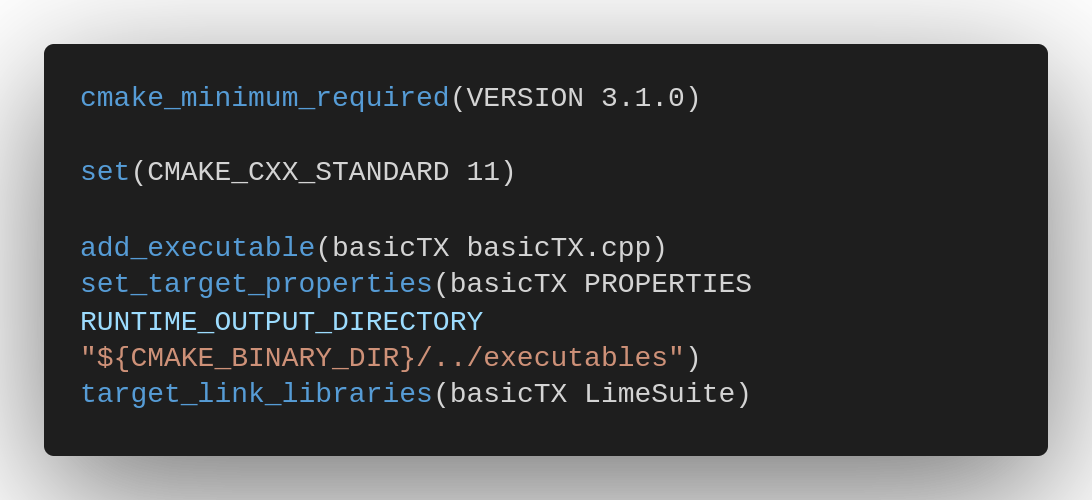
<https://github.com/myriadrf/LimeSuite>

El projecto cuenta con las librerias necesarias para trabajar en distintos modos; entre ellos esta : GUI, sistema embebido en C y C++.

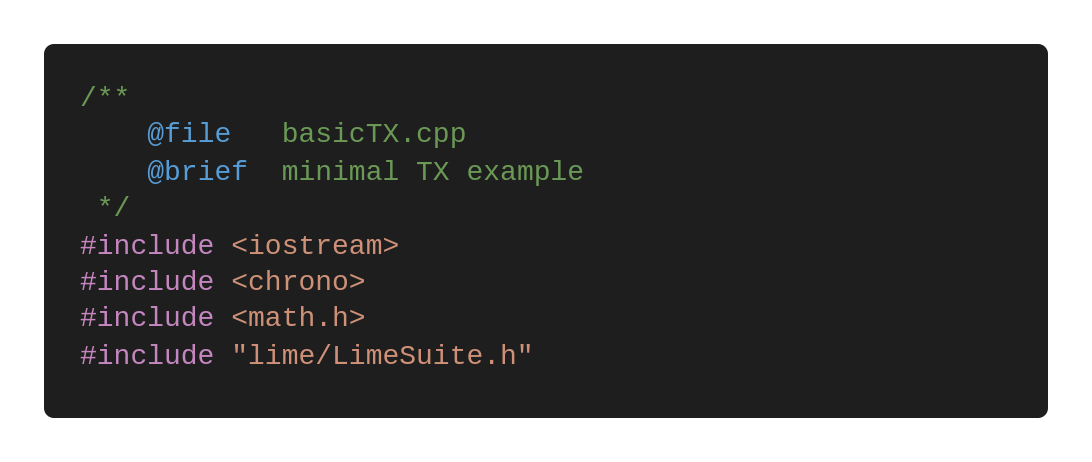
# Funcionamiento de codigo C++

Una vez instalada la libreria con Cmake install, se puede usar la misma libreria llamandola como libreria reconocida por el sistema operativo.

## Preparación libreria con CMAKE



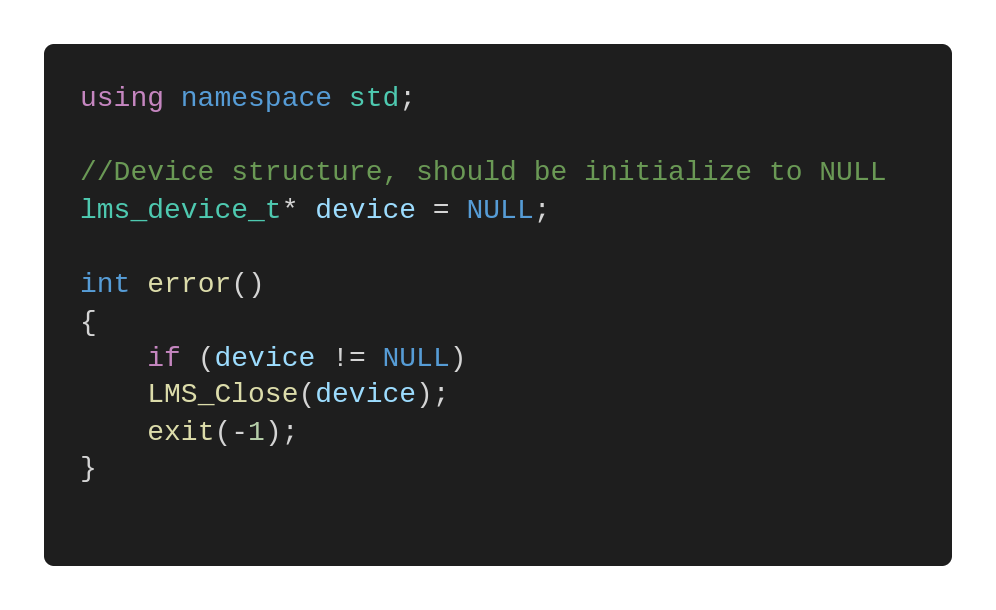
Posteriormente se mostrara el codigo basicTx.cpp donde se incluiran sus directorios de librearias



## Funciones de cabecera y variables globales

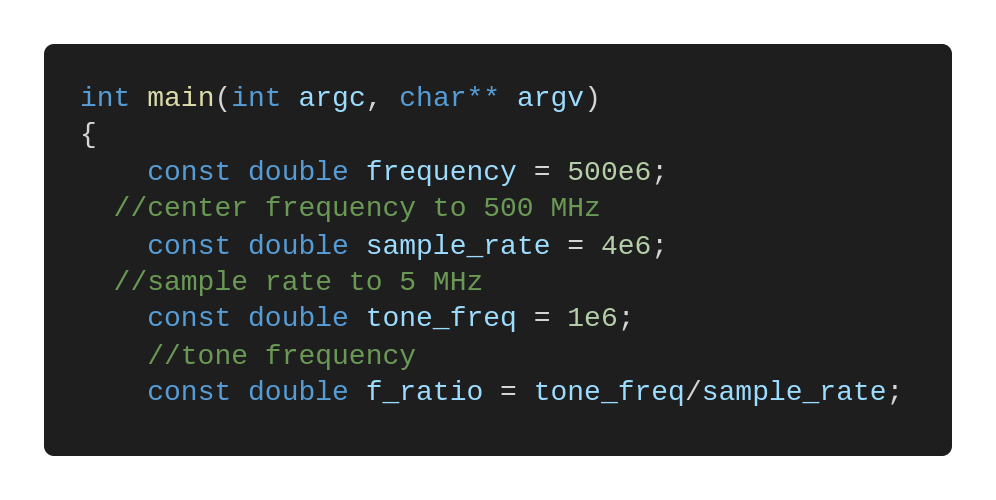
La estrucutra **lms\_device\_t** es la que nos dara acceso a todos los recursos del dispositivo, tanto a su conexión usb, como sus funciones para configurar propiedades de la antena.

Se agrega una funcion de error en caso de que haya problemas con el hilo de comunicación y no se quede perpetuamente abierto produciendo un death lock.



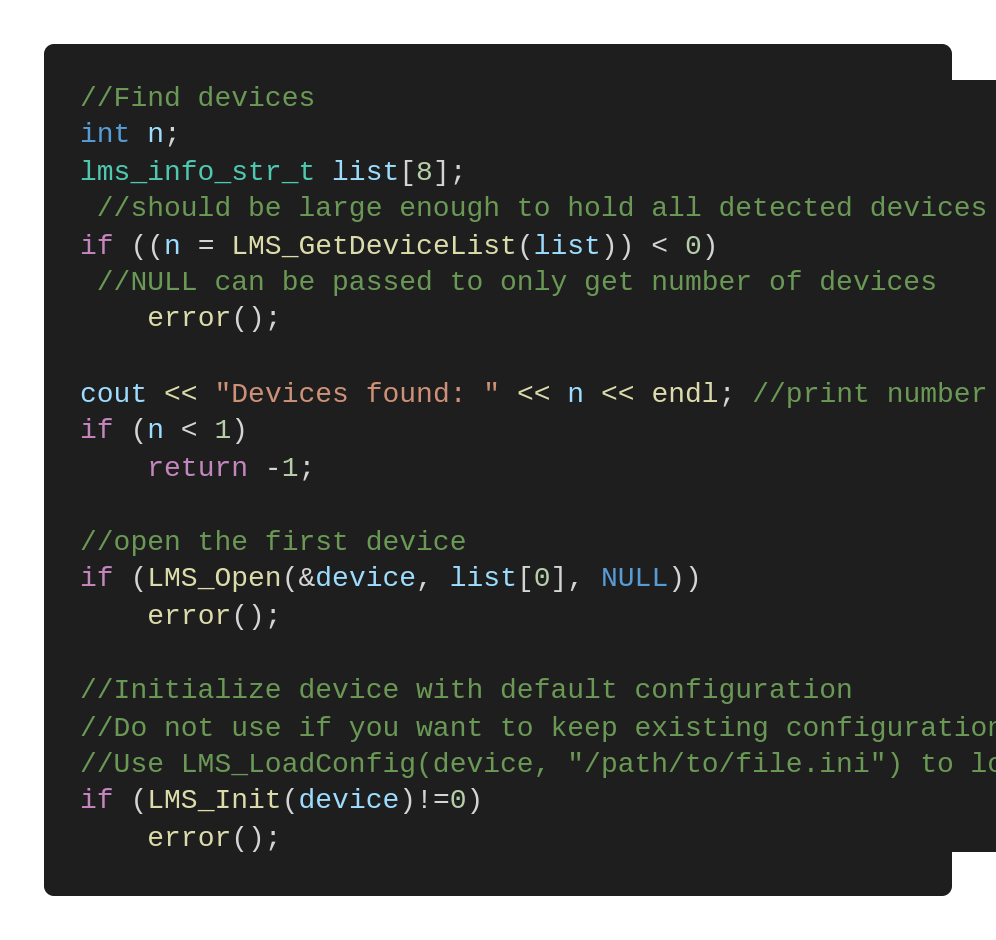
## Configuración de variables de radio

Una vez entrando al main se elige la configuración del radio con la que se va operar, en este codigo se dejan constantes los valores ya que se pretende dejarlos sin cambios, pero pueden ser perfectamente mutables sin ningun problema y configurables en tiempo de ejecución.



## Inicializar la conexión al radio

Se obtiene la lista de posibles radios, asi tambien se usa la función device list para encontrar el ID del radio que el kernel provee, una vez encontrado el ID adecuado de usb se abre la conexión y se busca inicializar el dispostivo, cabe recalcar que la inicialización no genera calibración, su unica función es alocar memoria necesaria para trabajar con los paquetes de USB e internamente de SPI.



## 

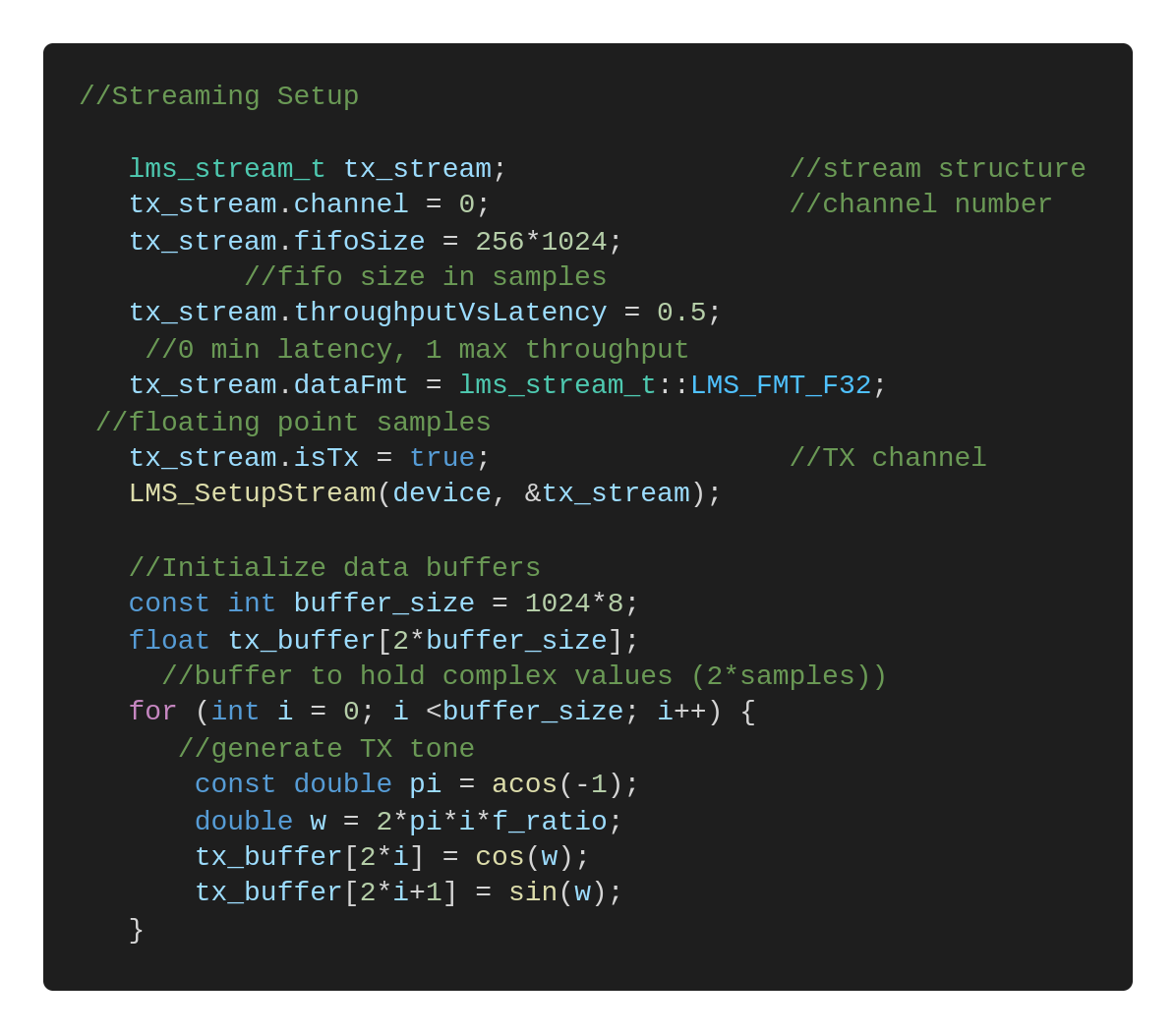
## Configurar radio con parametros

En esta sección se manda a llamar las funcion que internetamente tienen el arreglo de registros internos que se le llamaran al micro, estas API permiten directamente y sin tanto complicación , realizar configuración usualues del dispotivo. Las funciones tienen un diseño de este estilo.

**LMS\_<Configuración>(device, <TX/RX>, Channel, <valor numerico>)**

## 

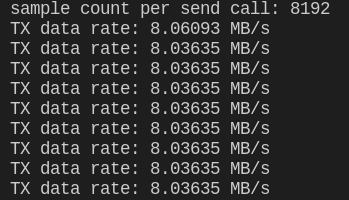
## Generación de señal por software

En la primera parte del codigo se hace una configuración para poner el tamaño del buffer que se va usar para la transmición, asi como el tipo de dato transmitido (float). Despues se genera la señal de tono , que sera transmitida, la frecuencia de la señal esta relaciónada con f\_ratio que se encuentra en la sección de “*Configuración de variables de radio”*

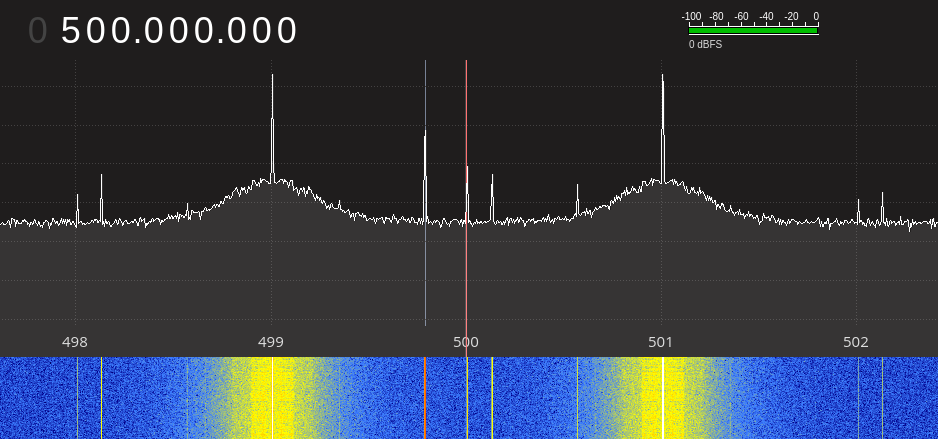
## Streaming

Se hace una carga de la señal al buffer, y el streaming se deja por 10 segundos

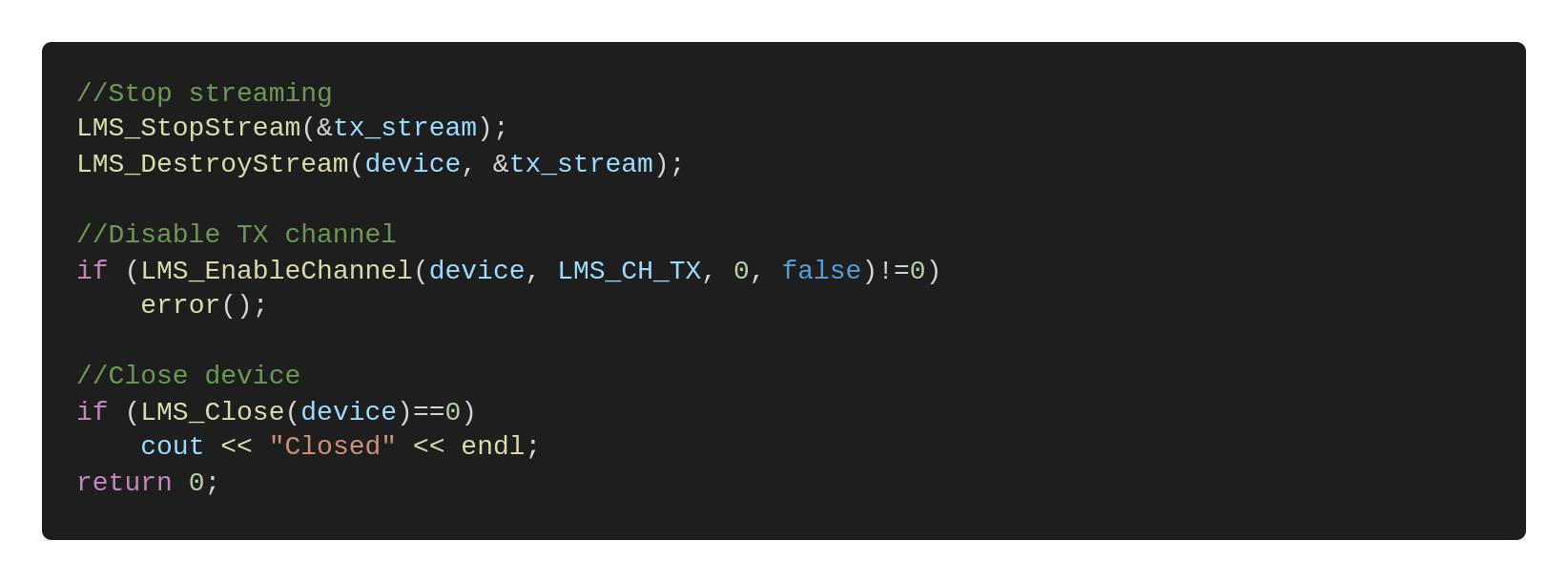
Una vez terminada la carga y el envio por paquete se dice el rate de transmición en MB/s



## Resultados

Para poder visualizar los resultados, se utilizo una HackRF con el software gqrx. En la imagen inferior se puede mostrar el resultado de la señal, la cual esta espaciada por un 1Mhz del centro de la frequencia principal, dado la frequencia de tono de la señal.

## Cierre

Una ves terminado el tiempo para streamear, se para el proceso y se libera el buffer reservado, despues se deshabilitan los canales de transmición y se cierra la comunicación por USB.