

COMUNICACIONES II – 27145

PROGRAMACIÓN EN RADIO DEFINIDA POR SOFTWARE (GNURADIO)

Fabián Yesid Moreno - 2191469

Fabián Leonardo Rincon Lizcano - 2174072

Florez Ramos Jose David - 2174241

Abstract

The objective of the internship is to provide the knowledge and skills to understand and program Software Defined Radios (SDR) using GNURadio. It focuses on theoretical concepts and their practical implementation. Including understanding the fundamentals of SDR, implementing blocks in GNURadio and evaluating the results. In practice we use software such as Ubuntu Linux, GNURadio and GRC (GNURadio Companion).

Resumen

El objetivo de la práctica es proporcionar los conocimientos y habilidades para entender y programar radios definidas por software (SDR) utilizando GNURadio. Se enfoca en conceptos teóricos y su implementación práctica. Incluyendo comprender los fundamentos de SDR, implementar bloques en GNURadio y evaluar los resultados. En la práctica utilizamos software como Ubuntu Linux, GNURadio y GRC (GNURadio Companion).

1. INTRODUCCIÓN

Se llevó a cabo una práctica centrada en la programación orientada en radio definida por software utilizando GNU Radio. Esta práctica se justifica en el contexto del desarrollo constante de las tecnologías y la necesidad de interactuar y conocer las distintas plataformas de programación de dispositivos. La práctica no sólo nos permitió a los estudiantes utilizar los bloques existentes, sino que también nos brindó la oportunidad de programar sus propios algoritmos, ampliando así su comprensión y habilidades en GNU Radio. El software utilizado para esta práctica incluyó Linux (Ubuntu), Terminal de Linux (git) y GNU Radio.

2. OBJETIVO PRINCIPAL

El objetivo principal de esta práctica es desarrollar competencias que nos permita interactuar eficazmente con la programación de funciones en GNU Radio, comprendiendo los aspectos fundamentales de los sistemas de tiempo real y la radio definida por software, y aplicando estos conocimientos para generar funciones y producir aplicaciones específicas para señales reales.

3. MARCO TEÓRICO

La radio definida por software (SDR, por sus siglas en inglés) es una tecnología que permite implementar por software partes de un sistema de comunicaciones que tradicionalmente se implementan en hardware. Esto proporciona una gran flexibilidad, ya que permite adaptar el sistema a diferentes

necesidades simplemente cambiando el software.

GNU Radio es una herramienta de desarrollo de software libre que proporciona bloques de procesamiento de señales para implementar sistemas de radio definidos por software. GNU Radio proporciona una interfaz gráfica, llamada GNU Radio Companion (GRC), donde los bloques se pueden arrastrar y soltar para construir un flujo de señales.

Los sistemas de tiempo real son aquellos sistemas en los que el tiempo en el que se producen las salidas es un factor crítico. En un sistema de tiempo real, no sólo importa que las salidas sean correctas, sino que deben producirse en el momento adecuado. En el contexto de la radio definida por software, los sistemas de tiempo real permiten procesar las señales de radio en tiempo real.

Los bloques de implementación de código en GNU Radio permiten a los usuarios implementar sus propios algoritmos. Estos bloques pueden interactuar con los bloques predefinidos de GNU Radio y pueden utilizarse para generar funciones y aplicaciones específicas para señales reales.

En resumen, la práctica realizada en la asignatura de Comunicaciones 2 se basó en estos conceptos para desarrollar competencias en la programación de funciones en GNU Radio, con el objetivo de entender y aplicar estos conceptos en un contexto práctico.

4. METODOLOGÍA EXPERIMENTAL

- Creación de la rama de desarrollo: Se creó una rama para el desarrollo de la práctica desde el repositorio en el navegador siguiendo el tutorial proporcionado. La rama se nombró “Practica_1”.
- Descarga de cambios: En la estación de cómputo, se descargaron los cambios de las ramas incluidas utilizando el comando git pull en la terminal.
- Creación de carpetas: En la misma terminal, se creó una carpeta llamada “Practica_1” y dos subcarpetas con los nombres “GNURadio” e “informe”.
- Uso de GNURadio y creación de informe: Se abrió la aplicación GNURadio y se guardaron los cambios del archivo creado en el directorio establecido para la práctica (Practica_1/GNURadio). Además, en la carpeta Practica_1/informe, se creó un archivo en Word para iniciar el informe.
- Implementación de bloques de Python: Se siguió el tutorial proporcionado para familiarizarse con el bloque de Python.
- Subida de cambios al repositorio: Se subieron los cambios al repositorio en la nube utilizando los

comandos git add ., git commit -m "", y git push.

- Compartir el repositorio: El usuario que creó el repositorio compartió con sus compañeros de grupo para la edición de los contenidos.
- Creación de nuevas ramas: Se crearon nuevas ramas a partir de "Practica 1" para que todos los integrantes del grupo tuvieran acceso a la estructura. Cada integrante creó su propia rama con un nombre único.
- Descarga de cambios en las ramas: En la terminal de cada integrante, se ejecutó el comando git pull.
- Ingreso a la rama personal: Cada integrante ingresó a la rama perteneciente a su nombre utilizando el comando git checkout P1_Nombre#.
- Implementación del algoritmo en el bloque de Python: Se utilizó el libro guía en la sección 1.2.0.1 e implementó en el bloque de Python el algoritmo sugerido.
- Implementación de bloques de acumulador y diferenciador: Se implementaron los bloques de acumulador y de diferenciador.
- Implementación de un bloque para mostrar estadísticas: Teniendo en cuenta lo aprendido, se implementó un bloque para mostrar la estadística vista en la clase teórica anterior y se sugirió una aplicación.
- Sugerencia de una aplicación y recreación de los resultados: Se sugirió una aplicación de los conceptos vistos en clase y se recreó una vista de los resultados.
- Verificación de cambios locales: Se utilizó el comando git status para verificar que hubo cambios de manera local en la PC.
- Aceptación de cambios: Si se encontraron cambios en la ventana de comandos, se mostraron en color rojo los archivos que cambiaron (se agregaron o quitaron). Si se estuvo de acuerdo con esos cambios, se utilizó el comando git add ..
- Confirmación de cambios: Si no se encontraron errores, se ejecutó el comando git commit -m "mensaje". El "mensaje" fue un comentario que hizo referencia a los archivos que se subieron (o eliminaron) o por qué se hicieron los cambios.
- Subida de cambios a la nube: Se ejecutó el comando git push y se verificó que los cambios hayan sido subidos a la nube (plataforma GitHub en el navegador).

Además, se sugirió una aplicación para los conceptos vistos en clase y se recreó una vista de los resultados. Esta aplicación demostró la utilidad de los conceptos aprendidos en clase y cómo pueden ser aplicados en situaciones reales.

```
import numpy as np
from gnuradio import gr

class blk(gr.sync_block):

    def __init__(self):
        gr.sync_block.__init__(
            self,
            name = "e_Diff",
            in_sig = [np.float32],
            out_sig = [np.float32]
        )
        self.acum_anterior = 0

    def work(self, input_items, output_items):
        x = input_items[0]
        y0 = output_items[0]

        N = len(x)
        diff = np.cumsum(x) - self.acum_anterior
        self.acum_anterior = diff[N-1]
        y0[:] = diff
        return len(y0)
```

Figura 1. código del bloque diferenciador

```
import numpy as np
from gnuradio import gr

class blk(gr.sync_block ):

    def __init__(self): # only default arguments here
        gr.sync_block.__init__(
            self,
            name = "e_Acum", # will show up in GRC
            in_sig = [np.float32],
            out_sig = [np.float32]
        )

    def work(self, input_items, output_items ):
        x = input_items[0] # Serial de entrada
        y0 = output_items[0] # Serial acumulada

        y0[:] = np.cumsum(x)

        return len(y)
```

Figura 2. Código del bloque Acumulador

5. ANÁLISIS DE RESULTADOS

Durante la práctica, se implementaron varios bloques en Python, incluyendo un acumulador y un diferenciador. Estos bloques fueron probados y se observó su comportamiento. Además, se implementó un bloque para mostrar estadísticas basadas en los conceptos vistos en clase.

Los resultados obtenidos a partir de estos bloques fueron analizados y se encontró que funcionaban como se esperaba. El bloque de acumulador acumulaba los valores de entrada y el bloque de diferenciador calculaba la diferencia entre los valores de entrada consecutivos.

El bloque de "promedio de tiempos" está diseñado para suavizar una señal a lo largo del tiempo. Consiste en un bloque acumulador que acumula la señal de entrada y un diferenciador que divide la suma acumulada por el número de muestras acumuladas. En esencia, realiza un promedio ponderado de la señal a lo largo de un intervalo de tiempo.

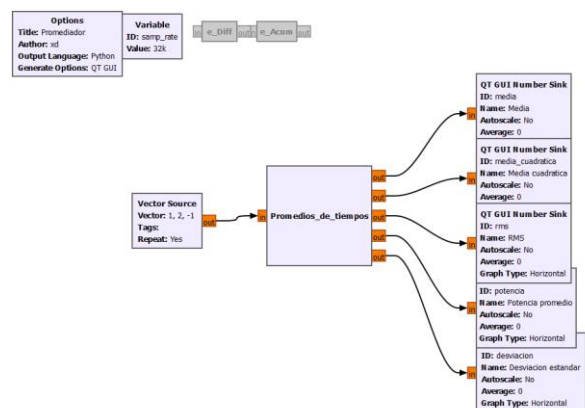


Figura 3. Bloque de Promedios

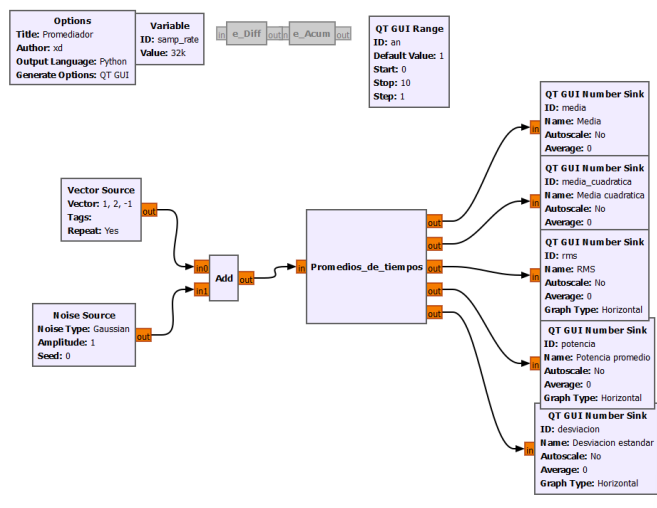


Figura 4. Bloque de ruido y promedio de tiempos

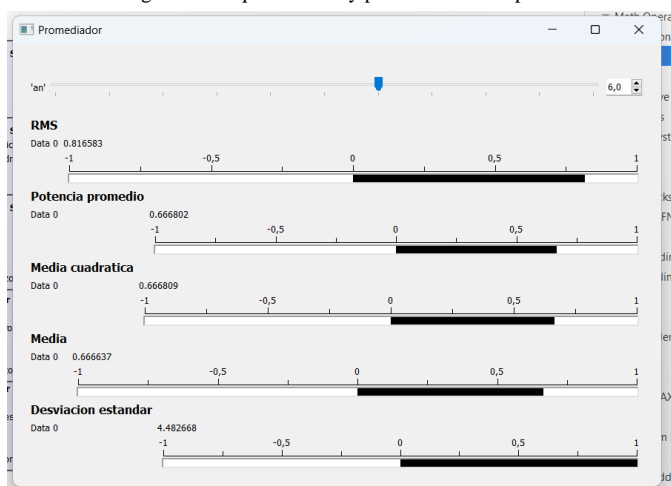


Figura 5. Visualización

Posibles aplicaciones

Filtrado de Ruido:

El bloque de "promedio de tiempos" puede ser utilizado para filtrar el ruido en una señal, ya que promedia varias muestras a lo largo del tiempo, suavizando las variaciones rápidas que pueden deberse al ruido. Se puede aplicar este bloque en situaciones donde la señal está afectada por ruido, como comunicaciones inalámbricas, y observar cómo mejora la calidad de la señal.

Monitoreo de Sensores:

En entornos donde los sensores pueden producir medidas ruidosas, el bloque de "promedio de tiempos" puede usarse para obtener lecturas más estables y representativas.

Se puede Implementar este bloque en sistemas de monitoreo ambiental, como sensores de temperatura o humedad, para mediciones más confiables.

Seguimiento de Movimiento:

Se puede utilizar el bloque en sistemas de seguimiento de vehículos o personas para mejorar la precisión de las posiciones estimadas.

Análisis de Señales Biológicas:

En señales biológicas como electrocardiogramas (ECG) o electroencefalogramas (EEG), el ruido puede afectar la interpretación de eventos importantes. El bloque ayuda a obtener una representación más clara.

6. CONCLUSIONES

De la práctica de laboratorio presentada previamente podemos concluir que:

- Aprendizaje de Git y GitHub: La práctica proporcionó una valiosa experiencia en el uso de Git y GitHub. Los comandos de Git como git pull, git add ., git commit -m "", y git push se utilizaron para gestionar el código y los cambios en el repositorio.
- Implementación de bloques en Python: La implementación de bloques en Python, incluyendo un acumulador y un diferenciador, proporcionó una comprensión práctica de estos conceptos.
- Aplicación de conceptos de clase: La sugerencia de una aplicación para los conceptos vistos en clase demostró cómo estos conceptos pueden ser aplicados en situaciones reales.
- Colaboración y trabajo en equipo: La práctica también destacó la importancia de la colaboración y el trabajo en equipo. Cada miembro del equipo trabajó en su propia rama y contribuyó al proyecto general.
- Verificación de cambios locales y subida a la nube: El proceso de verificación de cambios locales y subida de los mismos a la nube (plataforma GitHub en el navegador) reforzó la comprensión de cómo gestionar y compartir el código en un entorno de desarrollo colaborativo.

En resumen, la práctica fue una experiencia de aprendizaje valiosa que proporcionó una comprensión práctica de varios conceptos y herramientas importantes. También destacó la importancia de la colaboración y el trabajo en equipo en el desarrollo de software.

Referencias

- [1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," in IEEE Personal Communications, vol. 6, no. 4, pp. 13-18, Aug. 1999.
- [2] E. Blossom, "GNU radio: Tools for exploring the radio frequency spectrum," in Linux Journal, vol. 2004, no. 122, Jun. 2004.
- [3] P. A. Laplante and S. J. Ovaska, Real-Time Systems Design and Analysis: Tools for the Practitioner, 4th ed. Hoboken, NJ, USA: Wiley, 2011.
- [4] GNU Radio, "Creating Your First Block," GNU Radio Tutorials, [Online]. Available: https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block.