

Comunicaciones II

Diagrama de ojo

Plata Anghye; 2134736, Perez Jhon;2121636
Cordoba Augusto; 2145552
B1B.g2

20 de mayo de 2018

Resumen

A continuación se presentaran las instrucciones detalladas para utilizar los archivos anexados con el presente documento en el que se utilizan los programas de **GNU Radio Companion 3.7.9** y **Python 2.7.12**, todo esto fue realizado en **Ubuntu 16.04.4 LTS** para versiones anteriores a estas, pueden ser necesarias instalaciones adicionales o cambios en algunos comandos presentados en este documento

INTRODUCCION

El diagrama de ojo o patrón de ojo es un método utilizado para el análisis del comportamiento de los enlaces de transmisión. Permite analizar las formas de onda de los pulsos que se propagan en un enlace de comunicaciones, para lograr observar sus formas, desfases, niveles de ruido, potencias de las señales... y con ello apreciar la distorsión del canal (ISI), la severidad del ruido o interferencia y los errores de sincronismo en el receptor.

Nuestro objetivo con este proyecto consiste en mejorar la herramienta que ya tenemos en GNU Radio, que nos permite ver una versión aproximada de este diagrama, ya que solo gráfica 10 trazos al mismo tiempo, cuando es recomendado ver más de 100 trazos sobrepuestos para tener una versión más fiel y poder extraer los parámetros que necesitamos con mayor seguridad.

Para cumplir este objetivo se utiliza la programación en python el cual también es el código fuente en el que esta basado GNU Radio, el código principal que gráfica el diagrama de ojo no es de nuestra autoría y podrá ser encontrado en las referencias de este documento. Sin embargo, se presenta de manera muy general, nuestro enfoque principal es poder exportar señales de GNU Radio y poder visualizarlas con esta herramienta.

ANTES DE COMENZAR

Antes de empezar a seguir las instrucciones algunas cosas que podrían evitarnos algunos errores futuros.

VERIFICANDO VERSIONES

Con los siguientes comandos podremos hacer esto de una manera sencilla y desde la terminal de Ubuntu.

- **python -V** (Nos arrojará la versión de python que el sistema está usando por defecto)
- **gnuradio-companion --version** (Versión de GNU Radio)
- **lsb_release -a** (Con esto veremos la versión de Ubuntu)

Deberá obtener lo siguiente:



```
jhon@JHON-PC: ~  
jhon@JHON-PC:~$ python -V  
Python 2.7.12  
jhon@JHON-PC:~$ gnuradio-companion --version  
GNU Radio Companion 3.7.9  
  
This program is part of GNU Radio  
GRC comes with ABSOLUTELY NO WARRANTY.  
This is free software, and you are welcome to redistribute it.  
  
jhon@JHON-PC:~$ lsb_release -a  
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 16.04.4 LTS  
Release:        16.04  
Codename:       xenial
```

Figura 1: Verificando versiones

Si no tienen las versiones presentadas recomendamos descargarlas o actualizarlas.

Si la versión de python no corresponde podrá descargarla en el siguiente link:
<https://tecadmin.net/install-itempython-2-7-on-ubuntu-and-linuxmint/>.

Para actualizar Ubuntu solo necesitará los siguientes comandos:

- **sudo apt-get update**
- **sudo** - para obtener privilegios de ejecución de superusuario.
- **apt-get** - el gestor de paquetes de UBUNTU
- **update** - actualizará los repositorios
- **sudo apt-get upgrade**

ACTUALIZANDO O DESCARGANDO LAS LIBRERIAS NECESARIAS

En los computadores, en los que se ha instalado esta herramienta han sido necesarias 4 librerías pero se ha comprobado que no todos, necesitan todas. Sin embargo es mejor instalar cada una de estas y así, si ya está instalada las actualizará.

Las librerías se pueden instalar de la forma que el usuario prefiera; nosotros recomendamos hacerlo por medio de la herramienta **PIP**.

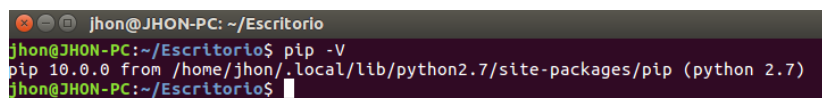
PIP es un sistema de gestión de paquetes utilizado para instalar y administrar software escritos en Python. Este puede ser instalado de manera sencilla con el siguiente comando:

- **sudo apt-get install python-pip**

Para verificar que ya se encuentra instalado utilizaremos el siguiente comando:

- **pip -V**

y deberemos obtener algo parecido a la figura 2 :



```
jhon@JHON-PC: ~/Escritorio
jhon@JHON-PC:~/Escritorio$ pip -V
pip 10.0.0 from /home/jhon/.local/lib/python2.7/site-packages/pip (python 2.7)
jhon@JHON-PC:~/Escritorio$
```

Figura 2:

Habiendo hecho esto procedemos a instalar algunas extensiones de python necesarias:

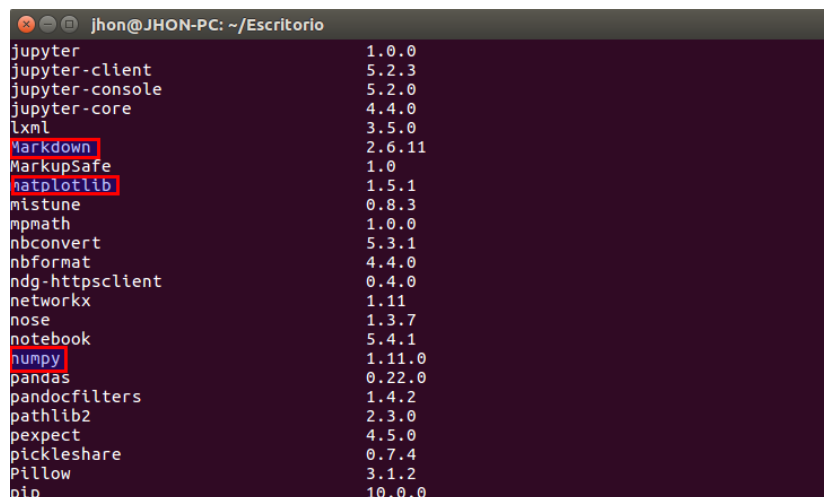
- **sudo apt-get install Cython:** Cython es una librería que permite generar extensiones para Python escritas en un lenguaje muy parecido a Python, pero con la posibilidad de utilizar funciones de C. En este caso nos permitirá aumentar el número de trazos para ver mejor el diagrama de ojo.
- **sudo -H pip install Markdown:** Markdown es un lenguaje de marcado ligero, lo que permite darle formatos a caracteres especiales posiblemente usado en los códigos; Markdown es requerida en algunos computadores para poder instalar las otras librerías.
- **sudo -H pip install scipy :** Scipy es una librería de herramientas numéricas para Python, el módulo scipy confiere capacidades de cálculo numérico de gran capacidad. En nuestro caso usaremos Scipy para poder importar los datos desde GNU Radio
- **sudo -H pip install numpy:** Numpy es una librería de Python, que permite hacer cálculos entre vectores y matrices. Es necesaria para hacer los cálculos presentes en el código.
- **sudo -H pip install matplotlib:** Matplotlib es una biblioteca de trazado 2D de Python que produce figuras de buena calidad, en una variedad de formatos impresos y entornos interactivos en todas las plataformas, necesaria en este caso para graficar nuestro diagrama de ojo.

Nota(La letra H le dice a sudo que mantenga el directorio de inicio del usuario actual)

IMPORTANTE: Para comprobar que las librerías están bien instaladas escribir el siguiente comando en la terminal de Ubuntu:

- **pip list**

A continuación aparecerá una lista con todas las librerías instaladas como la de la figura 3, entre las cuales debe aparecer las ya instaladas anteriormente.



```
jhon@JHON-PC: ~/Escritorio
jupyter 1.0.0
jupyter-client 5.2.3
jupyter-console 5.2.0
jupyter-core 4.4.0
lxml 3.5.0
Markdown 2.6.11
MarkupSafe 1.0
matplotlib 1.5.1
mistune 0.8.3
mpmath 1.0.0
nbconvert 5.3.1
nbformat 4.4.0
ndg-httpsclient 0.4.0
networkx 1.11
nose 1.3.7
notebook 5.4.1
numpy 1.11.0
pandas 0.22.0
pandocfilters 1.4.2
pathlib2 2.3.0
pexpect 4.5.0
pickleshare 0.7.4
pillow 3.1.2
pip 10.0.0
```

Figura 3: Lista de librerías instaladas con pip

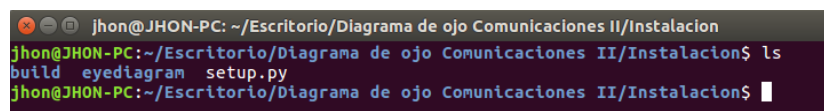
Si ya le aparecen todas las librerías mencionadas podrá avanzar a la siguiente etapa.

INSTALACION DEL DIAGRAMA DE OJO

Dentro de la carpeta llamada **"Diagrama-de-ojo-Comunicaciones-II"** encontrará una carpeta que dice instalación, entre y dentro de esta abra la terminal de Ubuntu y escriba el siguiente comando para comprobar sus elementos.

■ `ls`

obtendra algo igual a la figura 4.



```
jhon@JHON-PC: ~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion
jhon@JHON-PC:~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion$ ls
build_eyediagram  setup.py
jhon@JHON-PC:~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion$
```

Figura 4:

Ahora si el python que tiene configurado por defecto es **python 2.7** con el comando de la **Opción uno** podrá instalar el modulo **eyediagram** que instalara todo el código y facilitara el uso de este llamándolo con una sola linea como veremos mas adelante. Si por el contrario su python por defecto no es ese y tuvo que descargar de manera externa python 2.7 con el link suministrado anteriormente entonces utilice la **Opcion 2**

Opción 1

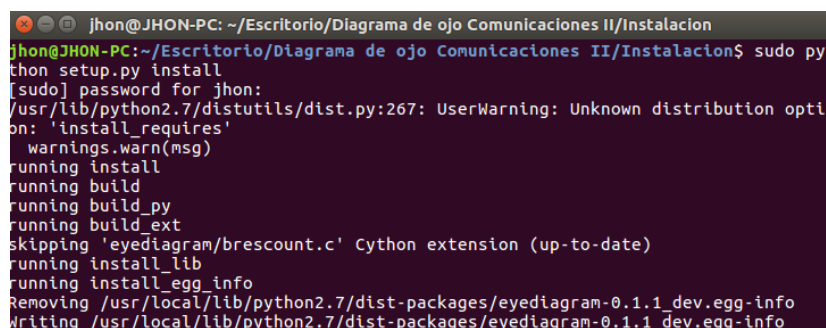
- **sudo python setup.py install** (Recuerde que debe estar dentro de la carpeta instalación)

Opción 2

- **sudo python2.7 setup.py install** (Recuerde que debe estar dentro de la carpeta instalación)

NOTA: Si no utilizo el link en este documento para instalar python 2.7 o si ya lo tenía instalado con otro nombre cambiar lo que va después del sudo por el nombre que su computador reconozca como python 2.7 ya que en algunos computadores se llamaba también python2

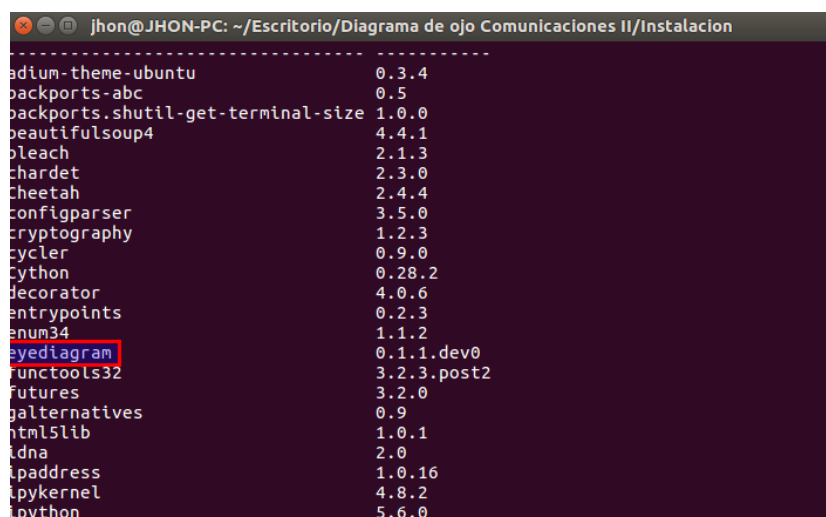
Cual sea la forma que utilice deberá obtener algo como la siguiente figura.



```
jhon@JHON-PC: ~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion
jhon@JHON-PC:~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion$ sudo py
thon setup.py install
[sudo] password for jhon:
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution opti
on: 'install_requires'
  warnings.warn(msg)
running install
running build
running build_py
running build_ext
skipping 'eyediagram/brescount.c' Cython extension (up-to-date)
running install_lib
running install_egg_info
Removing /usr/local/lib/python2.7/dist-packages/eyediagram-0.1.1_dev.egg-info
Writing /usr/local/lib/python2.7/dist-packages/eyediagram-0.1.1_dev.egg-info
```

Figura 5: Instalación del diagrama de ojo

Una forma de comprobar que el modulo se instalo correctamente es escribiendo **"pip list"** en la terminal y aparecera lo siguiente:



```
jhon@JHON-PC: ~/Escritorio/Diagrama de ojo Comunicaciones II/Instalacion
-----
adium-theme-ubuntu          0.3.4
backports-abc                0.5
backports.shutil-get-terminal-size 1.0.0
beautifulsoup4              4.4.1
bleach                       2.1.3
chardet                     2.3.0
cheetah                     2.4.4
configparser                 3.5.0
cryptography                 1.2.3
cyclr                        0.9.0
cython                      0.28.2
decorator                    4.0.6
entrypoints                  0.2.3
enum34                       1.1.2
eyediagram                   0.1.1.dev0
functools32                  3.2.3.post2
futures                      3.2.0
galternatives                0.9
html5lib                     1.0.1
idna                         2.0
ipaddress                    1.0.16
ipykernel                    4.8.2
python                       5.6.0
```

Figura 6: Comprobación de la instalación del modulo

En donde se puede observar que nuestro modulo ya esta instalado.

EJEMPLO

Entre a la carpeta ”**Diagrama-de-ojo-Comunicaciones-II**”, luego a la carpeta **Ejemplo**, encontrará una imagen y un código de python, utilizaremos editor de texto **gedit** que es el editor por defecto de Ubuntu , encontrara algo como esto:

```
# Copyright (c) 2015, Warren Weckesser. All rights reserved.
# This software is licensed according to the "BSD 2-clause" license.

from eyediagram.demo_data import demo_data
from eyediagram.mpl import eyediagram
import matplotlib.pyplot as plt

#Generamos señal de ejemplo
num_symbols = 100
samples_per_symbol = 24
y = demo_data(num_symbols, samples_per_symbol)

#llamamos el modulo que instalamos
eyediagram(y, 2*samples_per_symbol, offset=16, cmap=plt.cm.coolwarm)

#Graficamos
plt.plot(y)
plt.show()
```

Figura 7:

Puede ver claramente que el código se divide en 4 partes , en la primera importamos las librerías necesarias para el funcionamiento del código, luego de la librería importada generamos una señal de ejemplo que tendrá la siguiente forma :

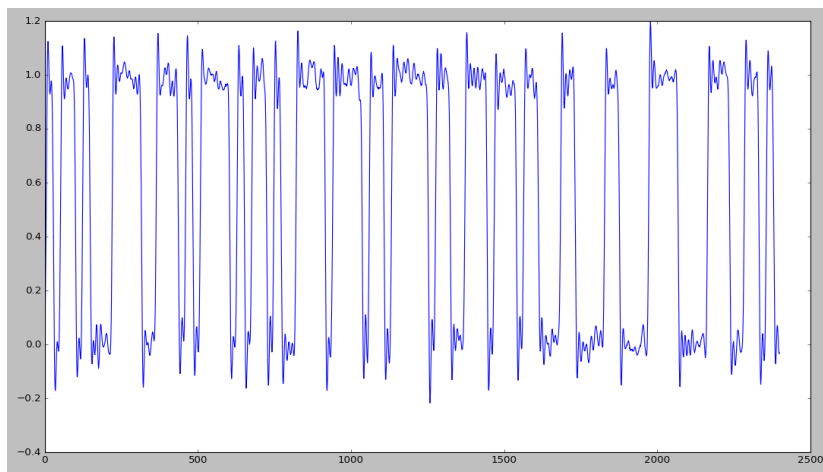


Figura 8: Señal en el tiempo

Como se puede apreciar en la figura 8 esta podría ser una señal luego de pasar por un canal con cierto ruido.El diagrama de ojo correspondiente seria el

de la figura 9 , adicional a eso podemos aumentar el numero de trazos con la variable **num_symbols** y como se puede ver en la figura 10 el diagrama mejora.

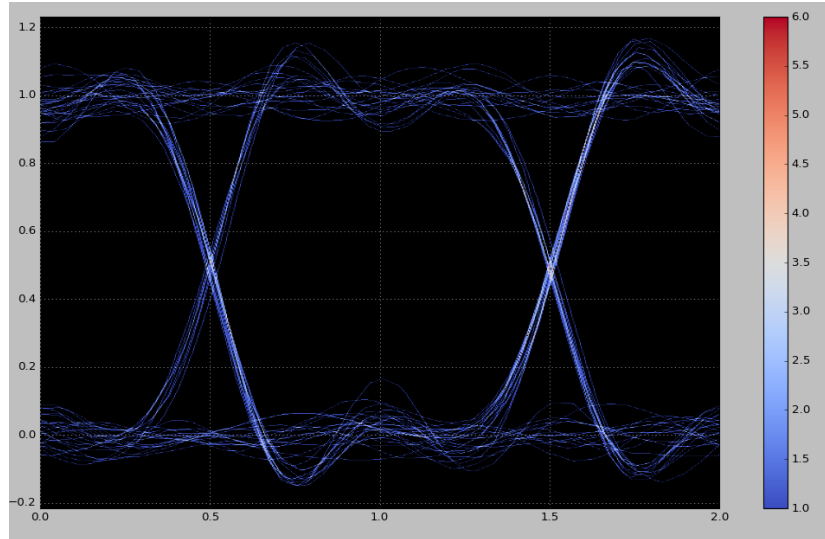


Figura 9: Diagrama de ojo para num_symbols=100

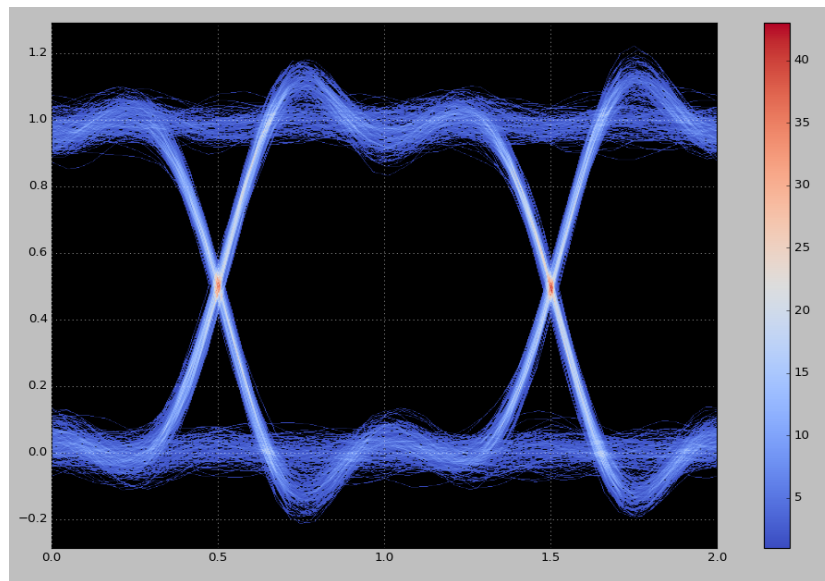


Figura 10: Diagrama de ojo para num_symbols=1000

USO CON GNU RADIO

Como se menciona antes nuestro principal objetivo es integrar GNU radio con nuestra herramienta , exportando señales generadas por cualquier flujo-

grama e importarla desde nuestro código para realizar el análisis que necesitemos hacer.No obstante hay que recordar que las aplicaciones de esta herramienta no solo se limitan a GNU Radio.

Siguiendo con nuestro análisis deberemos realizar los siguientes pasos:

- En nuestra carpeta llamada **Diagrama-de-ojo-Comunicaciones-II** debemos abrir una carpeta que se llama "**GNU Radio**"
- Dentro de esta carpeta crearemos un documento vacío y le pondremos el nombre que queramos en nuestro caso este se llama **Datos.Recibidos** y ya se encuentra presente en la carpeta, no necesariamente tienen que usar ese.
- Abriremos el archivo de GNU Radio llamado **Wave_Form.grc** dentro de este encontraremos un flujo-grama creado por nosotros en el cual utilizaremos distintos tipos de filtros que harán el papel de wave_formings por los cuales haremos pasar una señal aleatoria binaria.
- Antes de correr el flujo-grama debemos cambiar la dirección del bloque **File Sink** y asignarle la del documento vacío que creamos anteriormente.Esto hará que cuando corramos el flujo-grama se guarden los datos de mi señal de salida en el documento vacío, también podremos cambiar el tipo de filtro que esta generando mi señal. Si corremos el flujo-grama podremos observar los siguientes aspectos importantes:

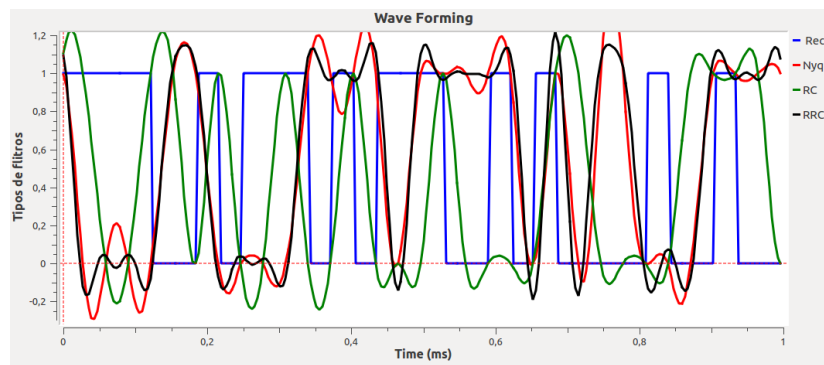


Figura 11: Wave_formings en el tiempo

Como se puede ver en la figura 11 veremos como es la respuesta de cada filtro en el tiempo, y estimar que tendrán distintos diagramas de ojo.

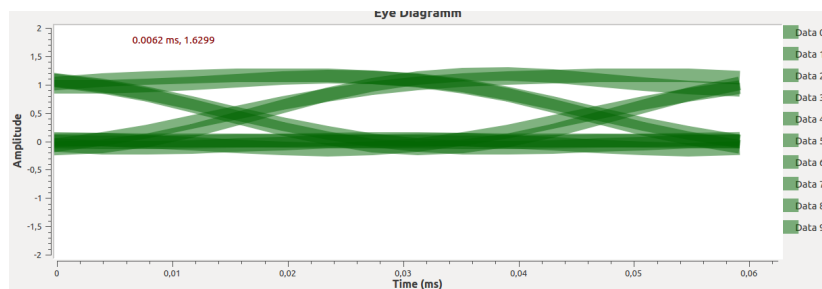


Figura 12: Wave_formings en el tiempo

En la figura 12 vemos el diagrama de ojo para el filtro **Root Raised Cosine Filter(RRC)** que puede ser cambiado dentro de el flujo-grama. En la figura anterior podemos corroborar que el diagrama de ojo solo tiene 10 trazos al mismo tiempo que nos permite hacer un análisis rápido pero no es suficiente como por ejemplo, para un informe profesional.

- El paso siguiente es abrir el archivo **prueba.py** que esta dentro de la carpeta **GNU Radio** , podemos observar la similitud que tiene con el que abrimos anteriormente de ejemplo. Hay que tener en cuenta los siguientes aspectos
 - Si creamos un documento vacío con otro nombre y no utilizamos el llamado **"Datos recibidos"** deberemos entonces poner el nombre de este nuevo documento en donde esta el anteriormente mencionado.
 - El **sps** utilizado debe ser el mismo que el usado en el flujo-grama que genero mi señal.
 - **fp** es el intervalo que vamos a graficar, que equivale a la cantidad de trazos, no podemos graficar todo el archivo entero por que GNU Radio genera muchas muestras en poco tiempo. Entre mas aumentemos este valor mejor se vera el diagrama de ojo pero requerirá mas computo
- Como paso final ejecutaremos este archivo.py desde la terminal con el siguiente comando
 - **python prueba.py**(Recordar que no necesariamente es python, todo depende de como el sistema reconozca a python 2.7)

Al final se nos abrirá una ventana con el siguiente diagrama de ojo:

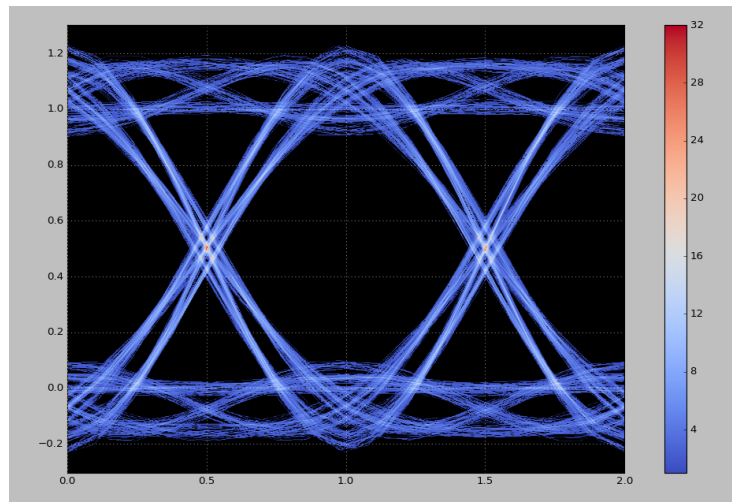
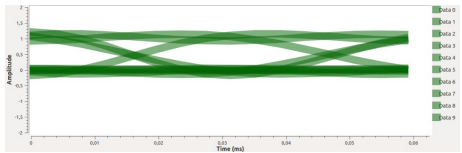
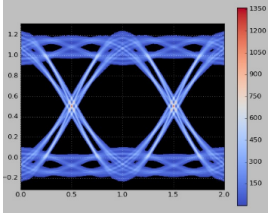
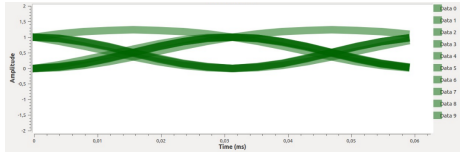
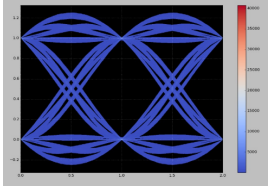
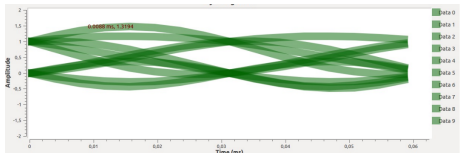
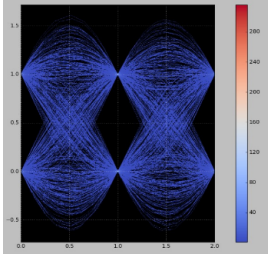


Figura 13: Diagrama de ojo de filtro RRC ; SPS=8 ; NUMERO DE TRAZOS=10000

NOTA: El flujo grama utilizado en GNU-Radio utiliza bloques pertenecientes al grupo de investigación Comdiguís.

En la siguiente tabla veremos algunos resultados cuando cambiamos el tipo de filtro.

Cuadro 1: Aplicaciones del diagrama de ojo

Tipo de Filtro	Diagrama de ojo con Gnu Radio	Diagrama de ojo con Python
RCC		
RC		
Nyquist		

OBSERVACIONES

- Cuando se corre el flujo-ograma en GNU-Radio, el por defecto genera unos archivos llamados **wform.py** , **wform.pyc** y **top_block.py** pueden eliminarse pero cada vez que corra el flujo-grama aparecerán.
- La herramienta gráfica **matplotlib** no es estrictamente necesaria , si tienen conceptos de python pueden realizar este diagrama casi en cualquier interfaz gráfica.ejemplos de estas son :
 - bokeh
 - pyqtgraph
- En [1] se encontró el código, se deja como referencia para aquellos que quieran estar al día con las actualizaciones.

CONCLUSIONES

- Este código ha sido probado con las especificaciones de hardware mencionadas en el documento, cualquier variación de éstas no has sido corroborada y no se asegura el funcionamiento del modulo.
- Lás gráficas que genera este módulo no son en tiempo real, son generadas por GNU, guardadas en un archivo para después graficarlas en python por medio de éste módulo.

- Se puede notar un mejoramiento en la calidad de las gráficas que se generan con este módulo en comparación con las generadas por medio de GNU.

LÍNEAS DE TRABAJO FUTURO

La principal línea de continuación para este trabajo, es realizar este proceso en tiempo real, de manera que la gráfica generada por GNU sea graficada de forma simultanea por la aplicación de python, esto con el fin de simular los aparatos profesionales con los que no se cuenta para hacer este proceso.

Referencias

[1] <https://github.com/WarrenWeckesser/eyediagram>

[2] <https://www.python.org/>