

Instrucciones para usar la aplicación

1 de abril de 2019

1. Trabajo Preparativo

Antes de comenzar la instalación debes verificar si cumples los requerimientos. En caso de no hacerlo, deberán realizar los pasos que se te ofrecen a continuación.

1.1. Requerimientos

No conocemos bien los requerimientos, pero esta información es una buena aproximación:

- descarga el código eyediagram así:
 - <https://github.com/hortegab/Diagrama-de-ojo-Comunicaciones-II>
Es la versión UIS, la cual se diferencia de la del autor original en que ha sido actualizada para las nuevas versiones de python, esto teniendo en cuenta que la version original es de hace más de 4 años
 - <https://github.com/WarrenWeckesser/eyediagram>
Es la versión del autor original que esperamos no sea necesario usar, porque es para versiones antiguas de python.
- Las pruebas exitosas que le hemos realizado a esta aplicación han sido en las siguientes condiciones:
 - versiones de Ubuntu en que ha sido probado:
 - **Ubuntu 16.04.4 LTS**. Aquí hemos probado la versión original, pero suponemos que la versión de Ubuntu es transparente para ambas versiones.
 - **Ubuntu 18.04.2 LTS**. Aquí hemos probado la versión UIS, pero suponemos que la versión de Ubuntu es transparente para ambas versiones.
 - Versiones de python en que ha sido probado
 - **Python 2.7.15rc1**. Aquí funciona bien la version UIS [2]
 - **Python 2.7.12**. Aquí funciona bien la versión del autor original [1].
 - Versiones de GNU Radio en que ha sido probado. Eso es transparente para Eyediagram, aplica más bien para los flujogramas:
 - **GNU Radio Companion 3.7.9**
 - **GNU Radio Companion 3.7.13.4**

1.2. Paso 1. Verifica que tienes versiones compatibles de software que debe estar pre instalado

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

- **python -V** (Nos arroja la version de python que el sistema esta usando por defecto). Con las versiones de python puede ocurrir lo siguiente:
 - que no tengas instalado python o que tu versión sea inferior a 2.7 o superior a 3.0. En este caso deberás instalarla, para lo cual puede descargarla en el siguiente link:
<https://tecadmin.net/install-itpython-2-7-on-ubuntu-and-linuxmint/>
 - Que tu version sea superior a 2.7 e inferior a 3.0. Entonces no deberá hacer nada. Al parecer basta con el comando de upgrade que se menciona en el siguiente paso.
 - Que tengas tanto una version 3.0 o superior, como otra de version 2.7. En este caso, cuando requiera llamar a python deberá llamar de manera explicita la version2, escribiendo el comando python2, en vez de python.
- **gnuradio-companion - -version** (Version de GNU Radio)
- **lsb_release -a** (Con esto veremos la versión de Ubuntu). Este comando no funciona en todos los PCs, pues puede no estar instalado. Puede instalarlo, pero otra alternativa es este comando `$cat/etc/issue/`

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.

1.3. Paso2. Actualiza tu sistema

Siempre que desees instalar algo nuevo en Ubuntu es conveniente realizar una actualización del sistema. 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**

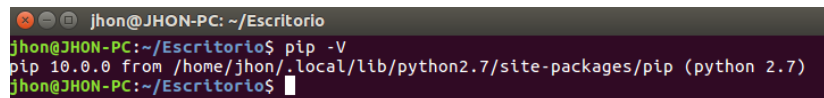
1.4. Paso 3. Instala o actualiza librerías que van a ser usadas y que no vienen por defecto con Ubuntu

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. Sigue estos pasos:

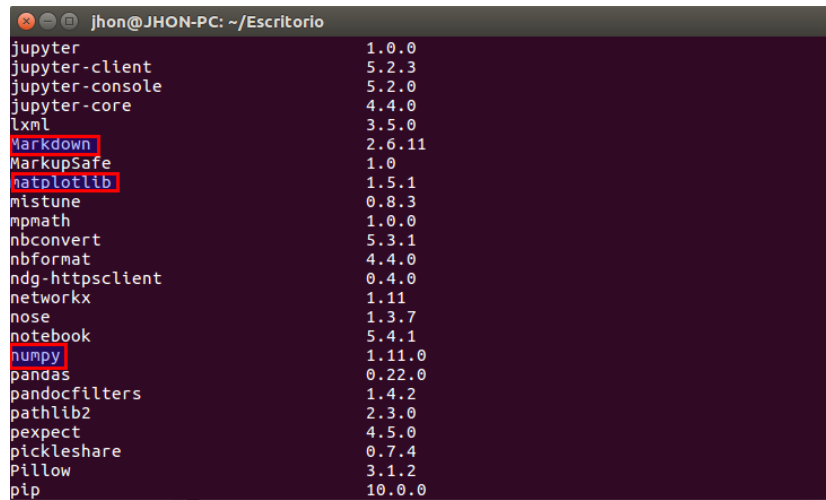
- **pip -V**: para comprobar si tienes instalado PIP. Deberías obtener algo parecido a la figura 2



```
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:

- **sudo apt-get install python-pip**. Se usa si en el paso anterior descubres que no tienes instalado PIP. Al parecer, para quienes tienen instalado al mismo tiempo python2.7 y python3, el anterior comando actúa para python2.7
- **pip list**: Sirve para que compruebes si ya tienes instaladas las librerías requeridas por eyedigraph. Si todo está instalado, debería aparecer una lista con todas las librerías instaladas como la de la figura 3.



```
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
```

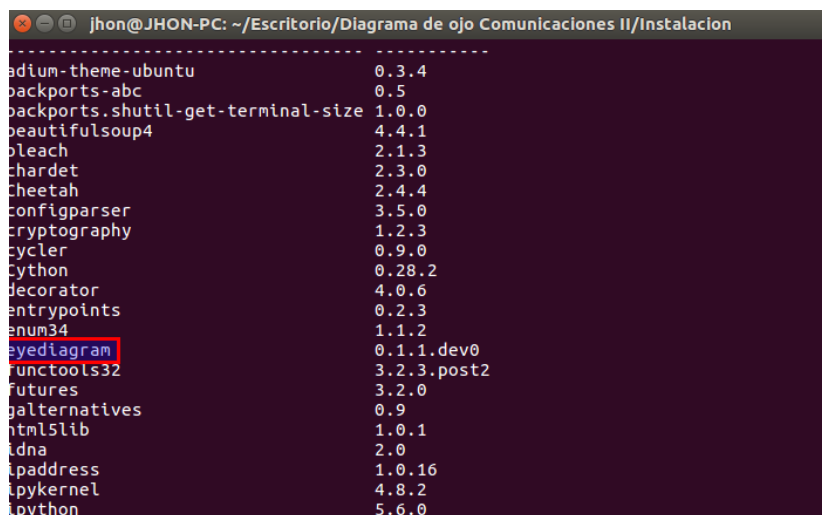
Figura 3: Lista de librerías instaladas con pip

- **sudo apt-get install cython**: Para instalar Cython que 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)
- **pip list**: Para volver a revisar si ahora sí están instaladas todas las librerías de python.

2. La instalación del Diagrama de Ojo

Entre a la carpeta que “Instalación”, y dentro de esta abra la terminal de Ubuntu y usa los siguientes comandos.

- **sudo python setup.py install.** Esto es para el caso en que solo tienes una versión de python instalada en tu computador
- **sudo python2.7 setup.py install.** Esto es para el caso en que tienes varias opciones de python instaladas, es necesario señalar a cual de ellas se le instala la aplicación del diagrama de ojo
- **sudo python2 setup.py install.** Esto es para el mismo caso anterior, solo que en unos computadores entiendo esto en vez de lo anterior.
- **pip list:** para verificar que ya está instalada la librería de eyediagram. Se deberá listar algo parecido a la Figura 4



```
-----
adum-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 4: Comprobación de la instalación del modulo

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

3. Pruebas de funcionamiento

Sigue estos pasos:

- ve a la raíz de las carpetas bajadas para la instalacion, la carpeta raíz se llama “Diagrama-de-ojo-Comunicaciones-II”
- Entra a la carpeta Ejemplo
- dentro de esa carpeta abre un terminal de Ubuntu.
- **python Ejemplo1.py**

- **python2 Ejemplo1.py** esto es para el caso en que tienes varias versiones python instaladas.
- Si todo está bien se obtiene una gráfica como la Figura 5

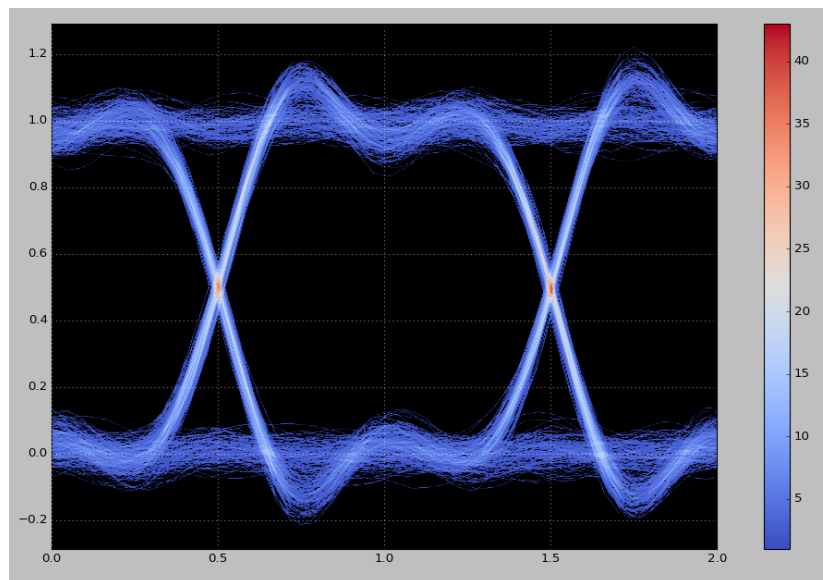


Figura 5: Diagrama de ojo para num.symbols=1000

Problemas que hemos encontrado en este punto y forma en que los hemos resuelto:

- al arrancar el Ejemplo1.py, con la versión original de eyediagram, se presentaba el error siguiente:

```
AttributeError: 'AxesSubplot' object has no attribute 'set_axis_bgcolor'
```

Lo hemos resuelto así:

- al parecer es un problema de últimas versiones de python 2.7.xx
- abrimos el archivo Diagrama-de-ojo-Comunicaciones-II/Instalacion/eyediagram/mpl.py
- cambiamos `ax.set_axis_bgcolor('k')` por `ax.set_facecolor('k')`
- volvimos a probar el ejemplo y todo okay

4. GNU Radio con EyeDiadram

Se requieren los siguientes componentes y preparativos:

- **Diagrama-de-ojo-Comunicaciones-II/GNU Radio/**: Es la carpeta donde tenemos un ejemplo de los archivos necesarios. El primer paso es entrar en ella.

- **senalx**: la verdad no importa como se llame, sino que allí esté almacenada la señal a la que deseamos buscar y graficar el Diagrama de Ojo. Si no ves estos archivos no importa pues son los que debe generar el flujograma.
- **Generar_senal.grc**: La verdad no importa como se llame pero es un flujograma que genera la señal, a la que nos interesa sacarle el Diagrama de Ojo, y la guarda en un archivo. En este flujograma es necesario generar la señal o señales a las que se les va a buscar el Diagrama de Ojo. La idea es que comiences pranicando con él, corriendolo y mirando lo que produce, pero necesitarán configurar algunas cosas previamente:
 - hay que revisar la dirección que tienen configurada los bloques “File Sink” para guardar los datos de la señal generada. Hay que recordar los nombres dados a los archivos de esas señales como por ejemplo `senal0`, `senal1`
 - hay poner especial atención al parámetro `Sps`, hay que recordarlo para usarlo más adelante
 - es importante asegurarse que los valores de amplitud, de la señal a la que se le quiere aplicar el diagrama de ojo, no se salgan de los límites entre -1 y 1, ya que si no se hace así, la gráfica del ojo saldrá deformada, pues el eye diagram considera que esos son los límites de amplitud.
 - no dejes correr el flujograma demasiado tiempo, pues en cuestion de segundos ocupará varios gigas de información. Basta con que cuentes hasta cinco. Si esto te incomoda agrega un bloque `Throttle` al flujograma para que limite un poco la velocidad.
 - cuando ya seas más experto pueden modificar este flujograma de acuerdo a tus necesidades o desarrollar uno similar.
- **Producir_Ojo.py**: La verdad no importa como se llame, pero es un programa en python que usa comandos de `eyediagram` y `python` para producir la gráfica del diagrama de ojo. Esto es lo que debes saber para cuando desees modificarlo:
 - Es un programa en python, por lo tanto la manera de correrlo es así:
\$ `python Producir_Ojo.py`. Tan pronto lo hagas verás los diagramas de ojo que el tiene programado producir
 - para editarlo, sigue los consejos que tiene en los comentarios. Observe que hay que señalarle los archivos de señal que debe abrir, pues es a ellos que les saca el diagrama de ojo.

5. Otros apuntes para sacarle mayor provecho a EyeDiagram

- 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.

- Las 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 este módulo.
- 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 original, se deja como referencia para aquellos que quieran hacerle seguimiento. Pero la verdad ya nosotros le hicimos una modificación como se explicó arriba. Nuestro código está en [2].

6. Detalles del archivo que produce el diagrama de ojo

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 senal 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 6:

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 :

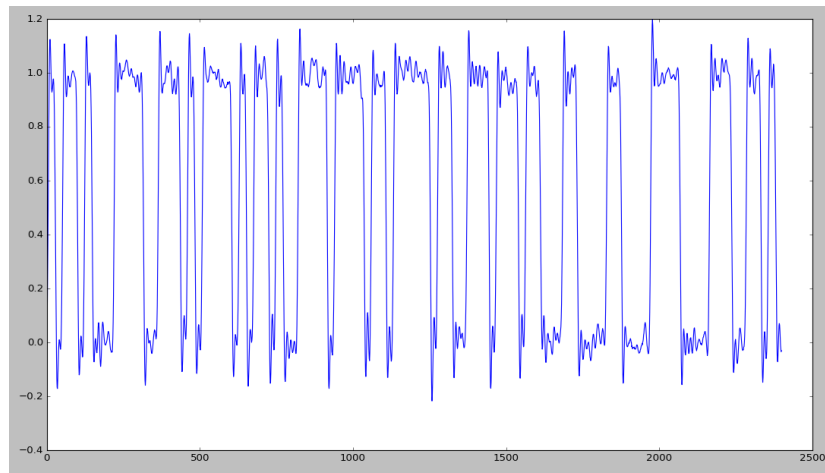


Figura 7: Señal en el tiempo

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

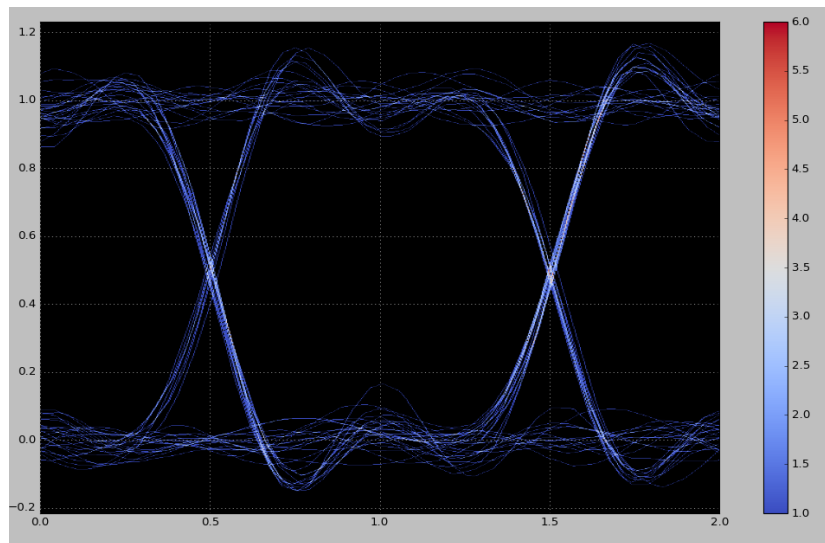


Figura 8: Diagrama de ojo para num_symbols=100

Recomendaciones para trabajos futuros

La principal línea de continuación para este trabajo es realizar poder crear un bloque en gnuradio que logre producir este diagrama de ojo. En este sentido pueden servir los siguientes considerandos:

- producir el diagrama de ojo como un bloque de gnuradio, cuando ya las librerías están instaladas puede ser tan sencillo como:

- organizar la señal en un vector `y0` de longitud `Ndat`
 - usar el comando para crear el diagrama de ojo: `eyediagram(y0, 2*Sps, offset=16, cmap=plt.cm.coolwarm)`
 - hacer que la gráfica aparezca, por ejemplo con el comando `plt.show()`
- Las dificultades pueden surgir en los siguientes sentidos:
- lo que hay dentro del comando `eyediagram()`, donde dice “`cmap=plt.cm.coolwarm`” es lo que hay que adaptar a `gnuradio`. hemos dicho que esa parte es flexible para ser usada no solo con `matplotlib`, habria que ver que tanto se adapta para ser usada con un visualizador tipo QT, como los que usa `gnuradio`.
 - para lograr lo anterior, es necesario dominar el tema de gráficas con PyQT a nivel de código de python, para probar si eso es compatible con `eyediagram()`.
 - otra opción es probar si `gnuradio` es compatible con `matplolib`

Referencias

- [1] <https://github.com/WarrenWeckesser/eyediagram>
- [2] <https://github.com/hortegab/Diagrama-de-ojo-Comunicaciones-II.git>
- [3] <https://www.python.org/>