

**LABORATORIO**

**EQUIPOS Y SISTEMAS DE**

**AUDIO Y VIDEO**

**(ESAV)**

*Práctica Final (Borrador)*

***Analizador de video Vectorscopio para interfaces SDI***

Curso 2018-2019

Grupo 2:

Alba Blanco, Miguel Ángel

Durán Castro, Patricia

Rodrigo Castillo, Adrián

## Introducción.

**En este informe se presenta la arquitectura general y realización de un Vectorscopio.**

**El sistema es capaz de realizar las transformaciones necesarias para representar en un diagrama polar la información de color extraída de vídeo digital YCbCr.**

A continuación, se muestra un diagrama de bloques con los módulos necesarios los cuales se explicarán posteriormente.

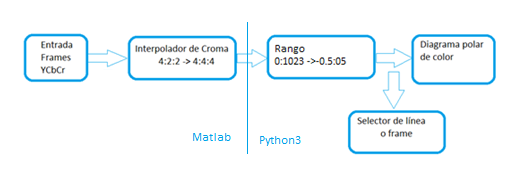


Ilustración : Diagrama de bloques.

* **Entrada de frames.** Este módulo es el encargado de recibir la información de vídeo de una trama SDI y determinar la resolución espacial y de color.
* **Interpolador de Croma.** Este módulo es el encargado de realizar una interpolación espacial de la croma en caso de que sea necesario.
* **Rango 0:1023🡪-0.5:0.5.** Este módulo aplican las transformaciones necesarias para poder representar los valores en el Vectorscopio.
* **Diagrama polar de color.** Representa gráficamente la información de color en un diagrama polar.
* **Selector de línea o frame.** Ofrece la posibilidad de poder representar en el vectorscopio un frame completo o una línea determinada.

|  |  |
| --- | --- |
| Ilustración 2: Imagen diagrama de barras laboratorio | Ilustración 3: Representación del diagrama de barras en Vectorscopio del laboratorio |

Todos los módulos presentados se desarrollan en lenguaje de programación Matlab excepto el módulo de visualización que es desarrollado en lenguaje de programación Python3.

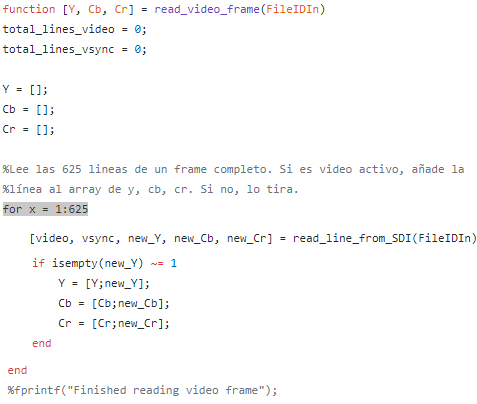
## Matlab

## Entrada de frames

C:\Users\gisam3\Desktop\lectura.PNGLa lectura de una trama SDI se realizará de la siguiente manera:

Dicha instrucción iterará en función del número de frames de la trama SDI.

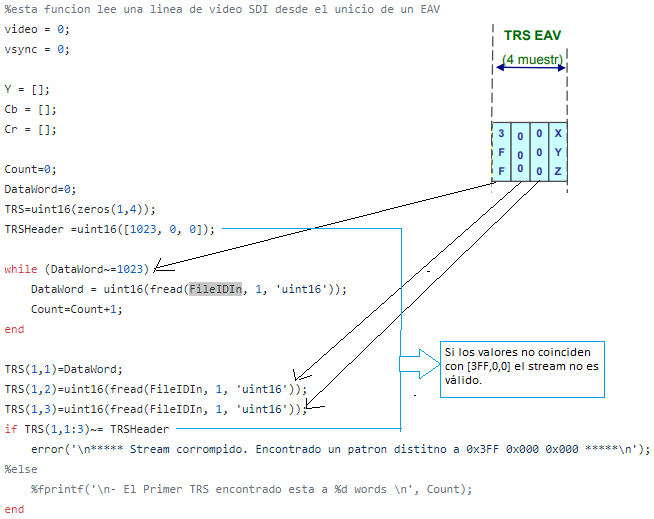
La función **read\_video\_frame()** es la siguiente:

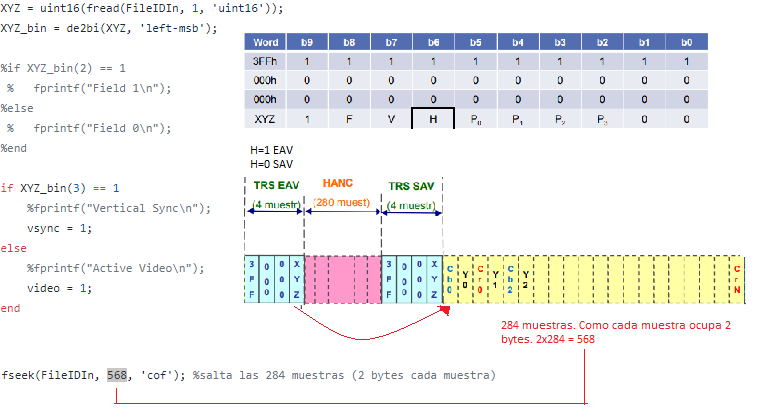


Se encarga de identificar vídeo activo en la trama SDI para su posterior lectura. En caso de ser video activo almacena los valores de Cb y Cr.

La función **read\_line\_from\_SDI()** se encarga de comprobar que la trama SDI comience por un EAV.

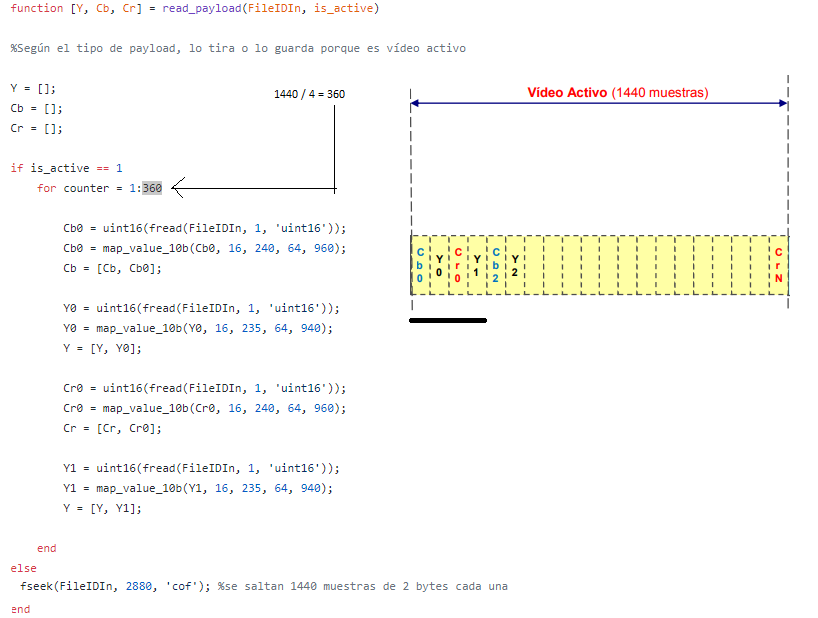
C:\Users\gisam3\Desktop\lee trama sdi.PNG



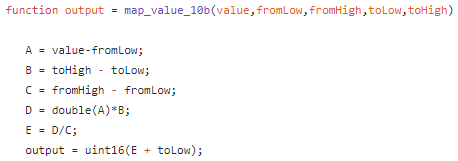


Tras esto se lee la información de luma y croma o se desprecia la información de sincronismo vertical. La función utilizada es **read\_playload().**





Cabe destacar que la función **map\_value\_10b()** es una conversión de profundidad de bit de 8 bit por pixel a 10. Esto es debido a que el vídeo que contiene la trama es de 8 bit por píxel.



## Interpolador de croma.

Es necesario que el submuestreo de la crominancia sea 4:4:4

## Python3

## El programa principal (main()) de Python es el siguiente:

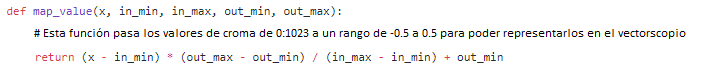
## 

Donde se llamaran las siguientes funciones:

* get\_video\_frame() 🡪 explicada en el apartado c).
* draw\_vectorscope() 🡪 explicada en el apartado b).
* draw\_video\_info() 🡪 explicada en el apartado b).
* button()🡪 explicada en el apartado c).

## Cambio de Rango.

La función **map\_value** está contenida dentrode **get\_video\_frame**(). Es la encargada de realizar la transformación de un rango de 0 a 1023 procedente de los 1024 valores correspondientes a los 10 bit de profundidad de croma al rango del vectorscopio el cual es entre -0.5 a 0.5.



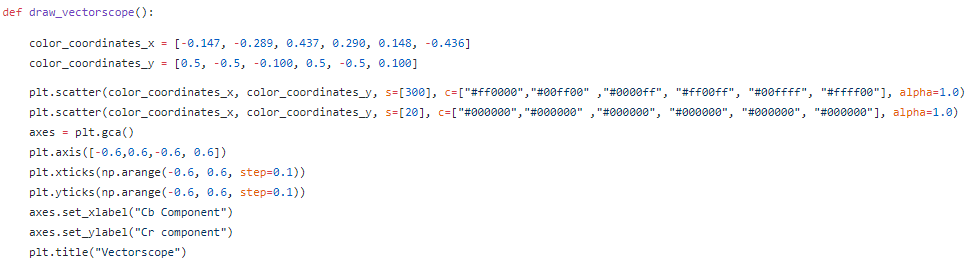
## Diagrama polar de color.

La función **draw\_vectorscope** representa en un plano el diagrama polar con los valores de Crominancia.

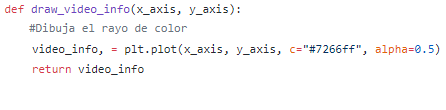
La función plt.scatter() se representa mediante círculos, los valores de los colores primarios y secundarios: R, G, B, M, Y, C.

s= [] representa el área del círculo para dichos colores. El valor de 300 sirve para poder percibir de manera apropiada el color y 20 el punto exacto en el que se centran.

El eje x representará los valores correspondientes a la componente Cb mientras que él eje y representará los valores correspondientes a Cr.



La función **draw\_video\_info** representa en el plano el rayo correspondiente al valor de crominancia del frame o línea seleccionado.



## Selector de línea o frame.

Get\_video\_frame()