

# **PROCESAMIENTO DIGITAL DE SEÑALES**

## **Laboratorio N°3: FFT con FRDM-K64**

### **DOCENTES:**

Parlanti, Gustavo.

Rossi, Roberto.

Molina, German.

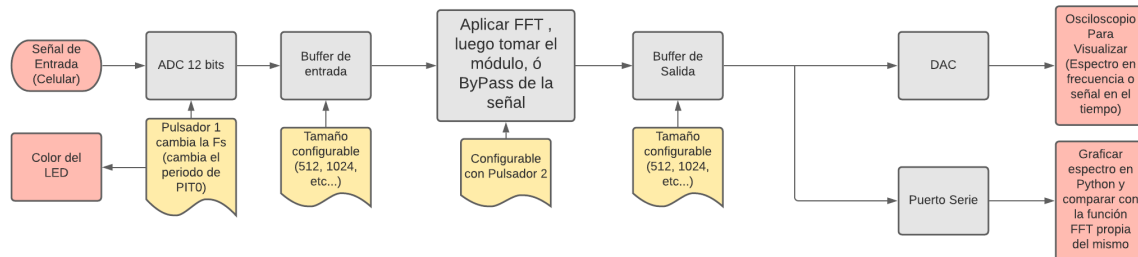
### **ALUMNOS:**

Gomez Augusto Rodrigo, matricula: 39807998

Ferraris, Domingo Jesús, matricula: 36656566

Alaniz Franco, matricula: 37234066

### Diagrama de bloques simplificado de funcionamiento:

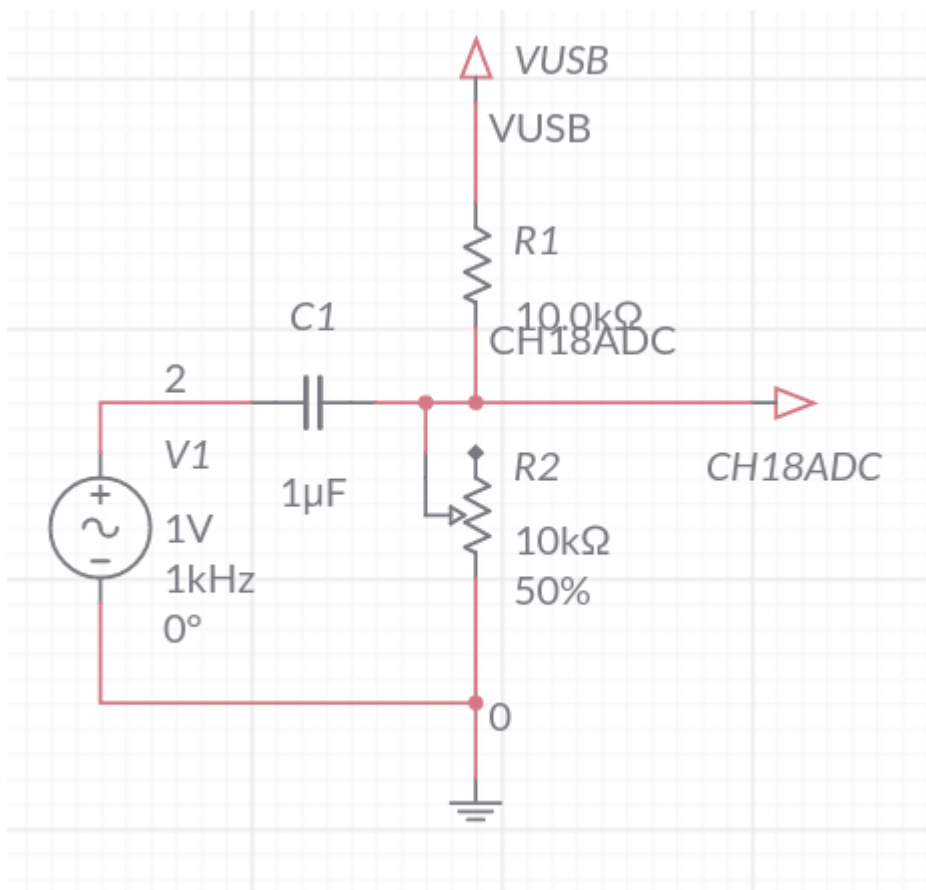


El sistema consta del ADC, trabajando en modo 12 bits SE, al que le se le ingresa la señal a muestrear, que en este caso se generó con la placa de sonido de la PC (o el celular) . Además consta con un pulsador selector de velocidades e indicador de las mismas por medio de los colores de un led RGB, de un pulsador con función de start/stop para la FFT y de otro que cambia la cantidad de puntos de la FFT de forma cíclica. Finalmente se envía por UART el resultado de la FFT y se gráfica con Python. Además por medio del DAC de 12 bits se puede reconstruir el espectro para observar en el osciloscopio.

### Fotos tomadas del trabajo

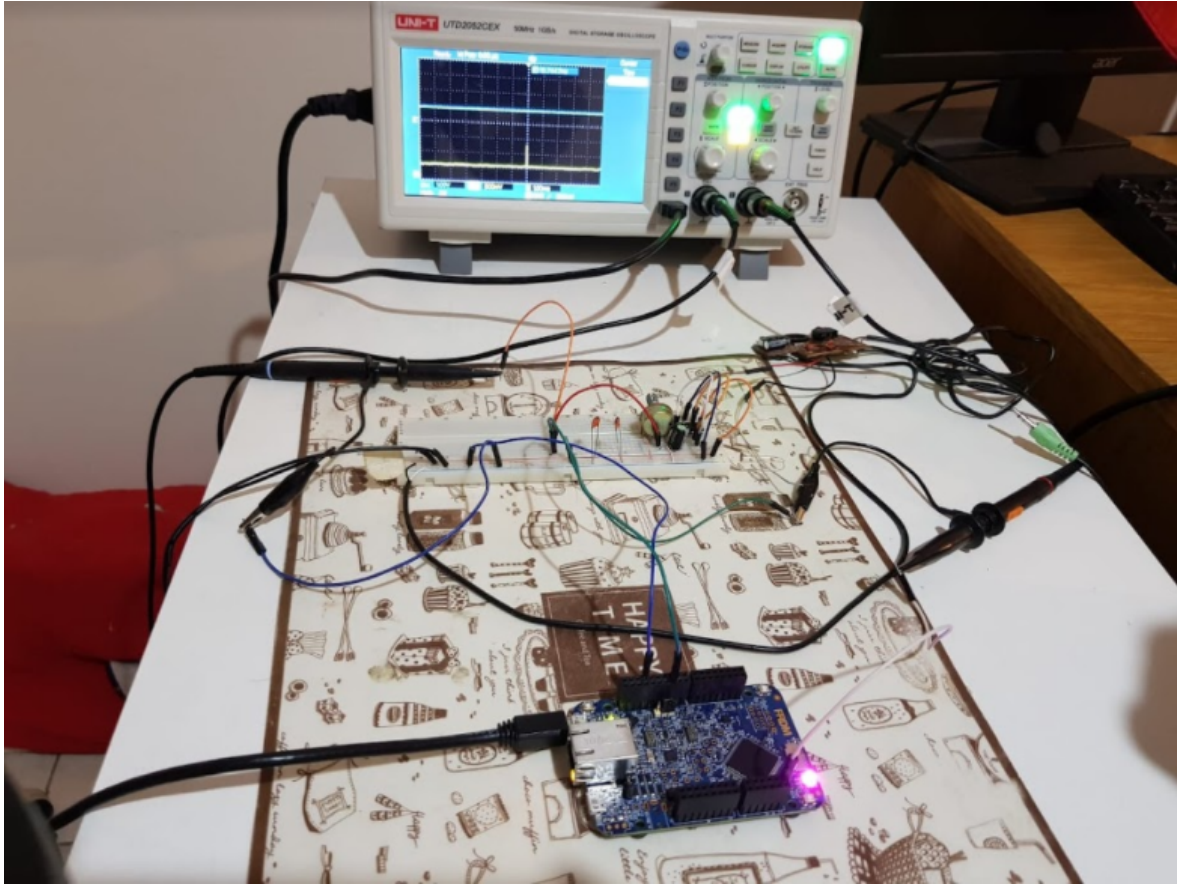
La señal de entrada, es la especificada en el párrafo anterior, es la misma que la de los trabajos anteriores, y el circuito de acondicionamiento también. El diagrama circuital mostrado a continuación lo indica:

**Facultad de Ciencia Exactas, Físicas y Naturales**  
**Procesamiento digital de señales**

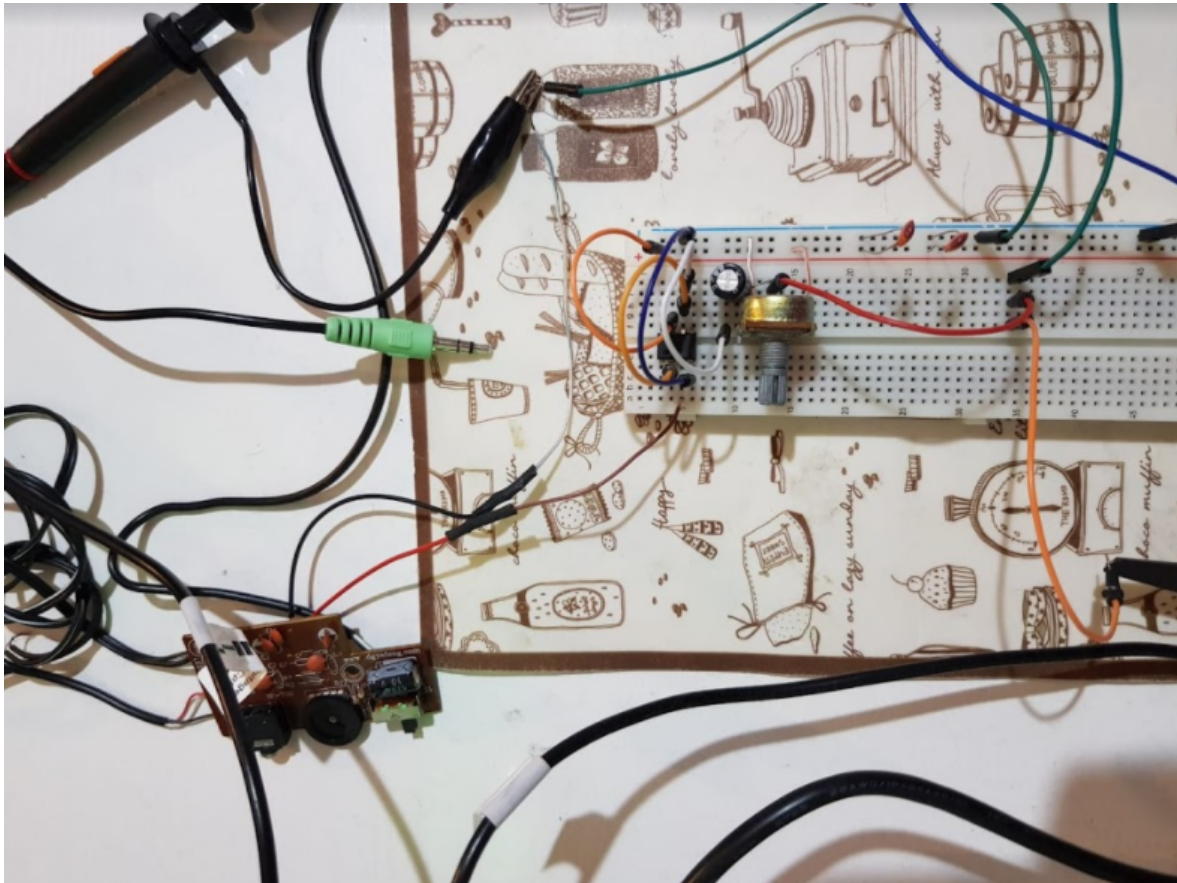


También se adjunta una imagen de todo el sistema implementado físicamente:

**Facultad de Ciencia Exactas, Físicas y Naturales**  
**Procesamiento digital de señales**



**Facultad de Ciencia Exactas, Físicas y Naturales**  
**Procesamiento digital de señales**

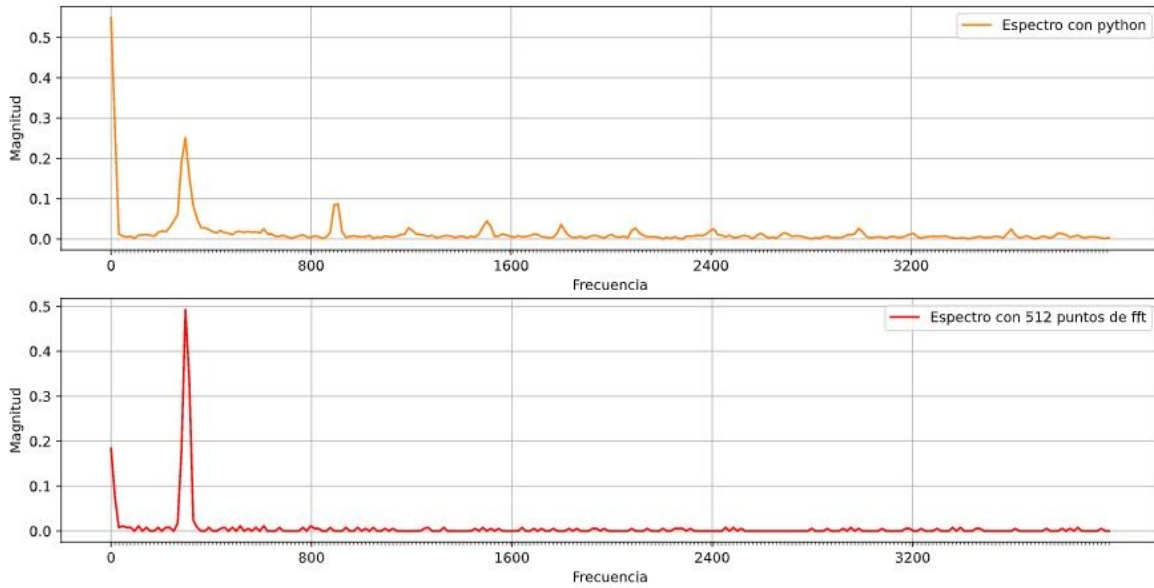


Para poder visualizar los resultados, los datos en el buffer de salida se transmiten por puerto serie a un Script de Python, para que allí se pueda graficar el espectro y además, computar la FFT en el propio entorno para poder compararlo con la que resulta del DSP.

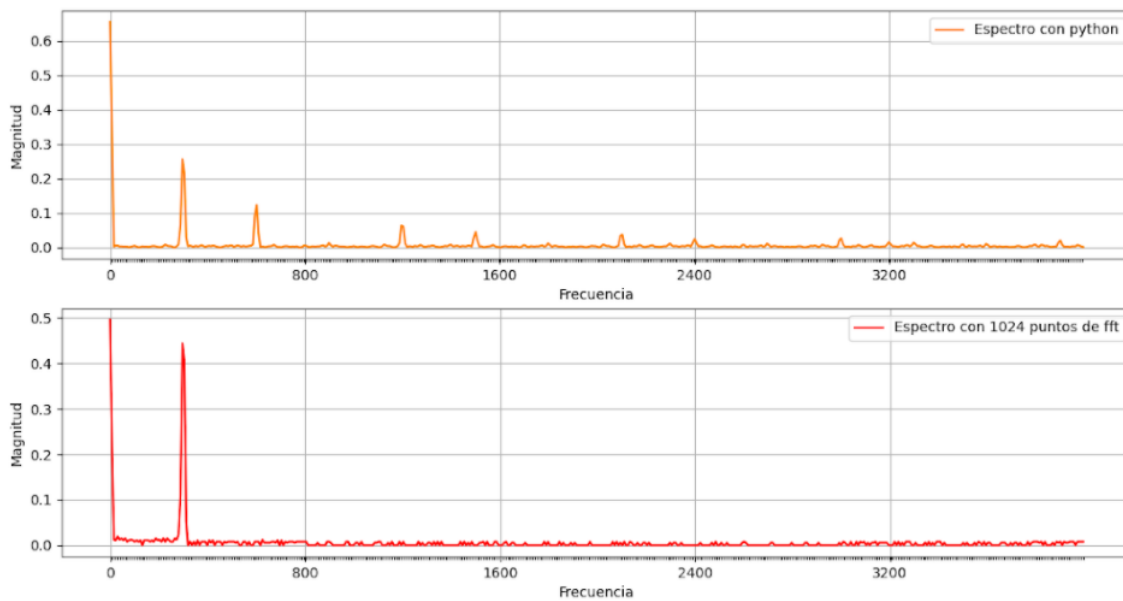
**Señal de entrada senoidal de  $f=300\text{Hz}$**

**Para  $F_s= 8\text{kHz}$ ,  $N= 512$**

## Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



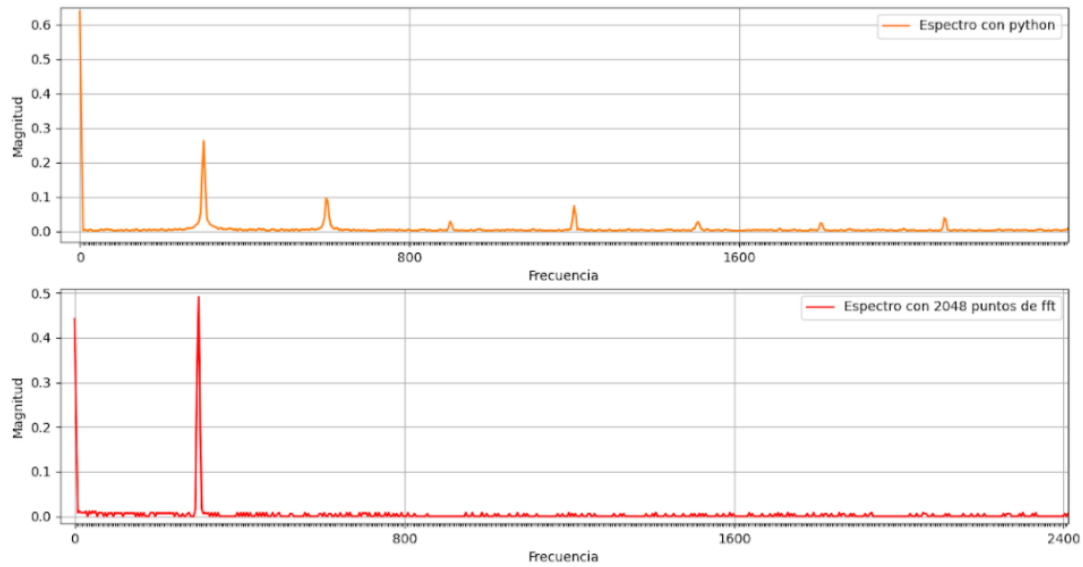
Para  $F_s = 8\text{kHz}$ ,  $N = 1024$



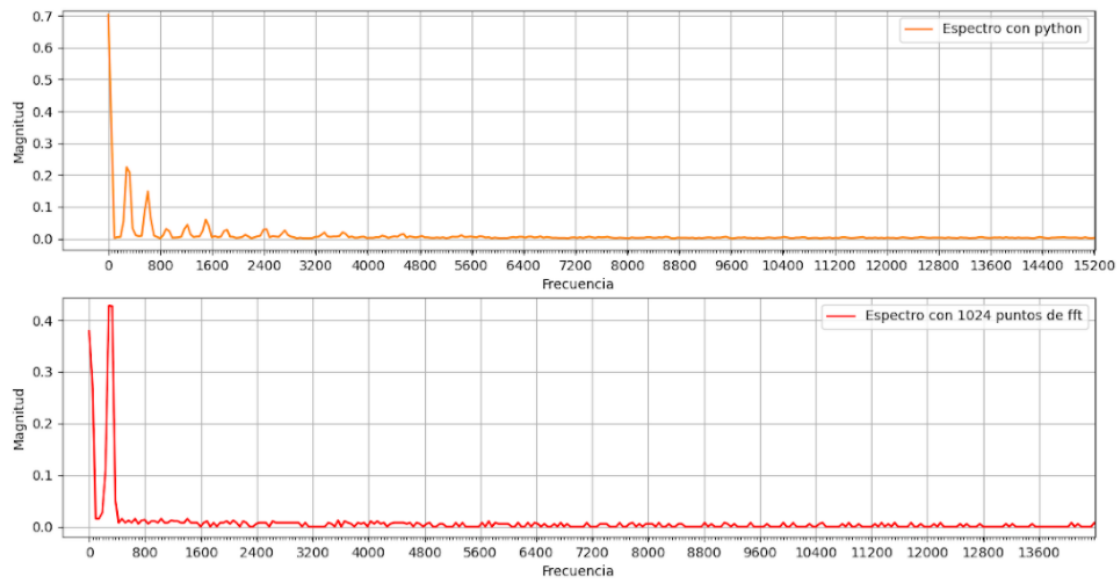
Para  $F_s = 8\text{kHz}$ ,  $N = 2048$

## Facultad de Ciencia Exactas, Físicas y Naturales

### Procesamiento digital de señales

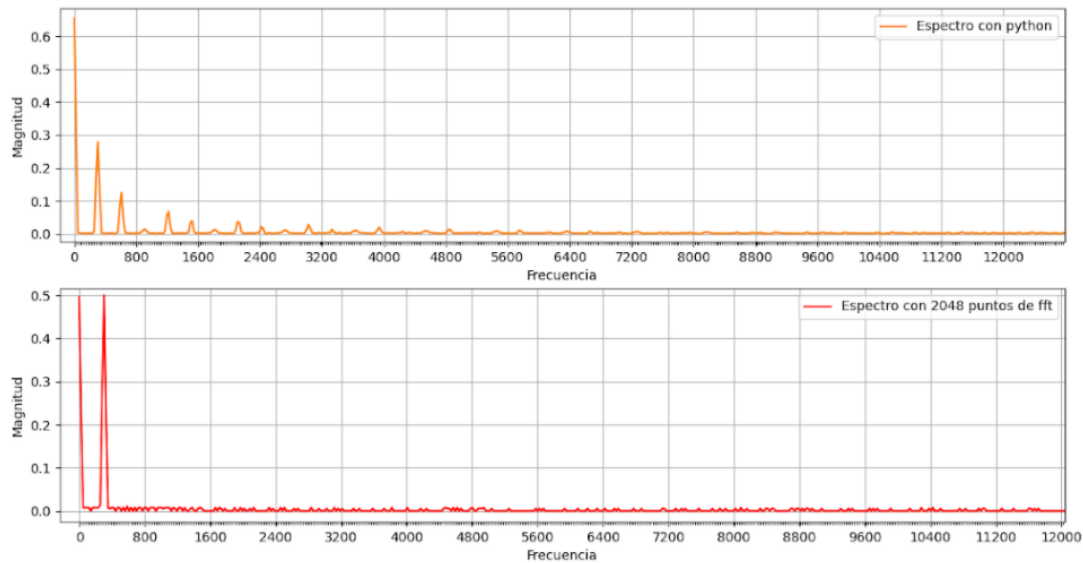


**Para  $F_s = 48\text{kHz}$ ,  $N = 1024$**



**Para  $F_s = 48\text{kHz}$ ,  $N = 2048$**

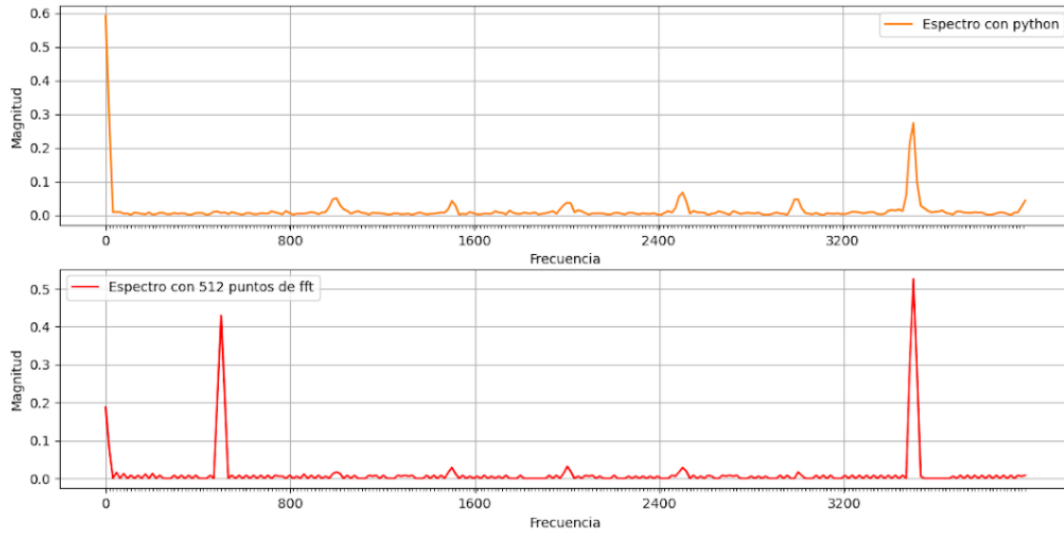
## Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



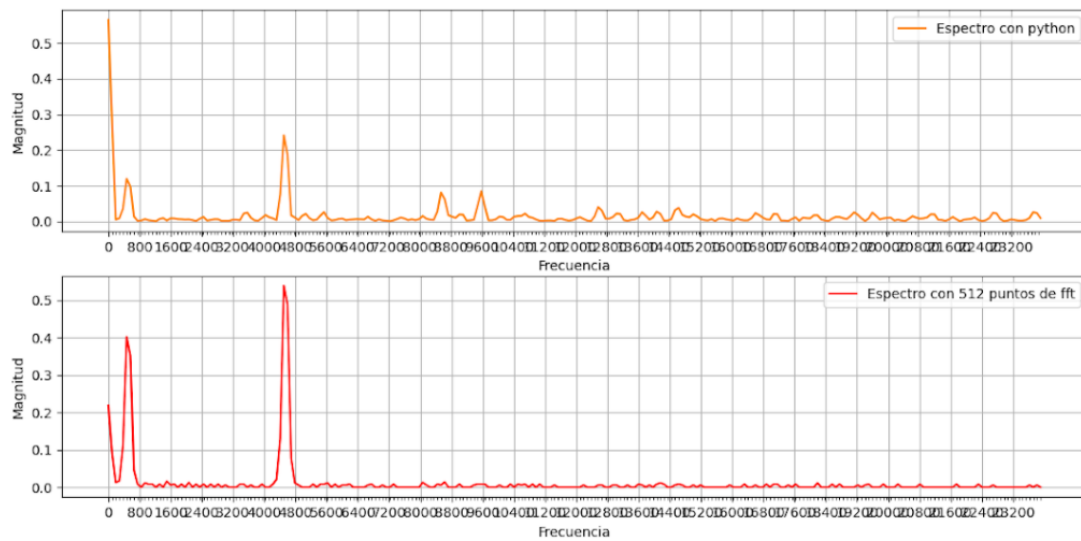
Es interesante también destacar que el fenómeno del aliasing también puede estar presente si no se escoge una  $F_s$  adecuada o se muestrea una señal sin satisfacer el teorema del muestreo, esto se pudo observar cuando excitamos con una señal que es la suma de dos sinusoides de frecuencias  $f_1 = 500$  Hz y  $f_2 = 4500$  Hz, y obteniendo el espectro para un mismo  $N$ , pero cambiando  $F_s = 8$  kHz y luego  $F_s = 48$  kHz. En el primer caso, las frecuencias de mayor módulo se dan en  $f_1 = 500$  Hz aproximadamente (lo cual es verdadero), pero el otro pico se da en  $f_2 = 3500$  Hz lo cual es incorrecto, este es un aliasing de la señal de entrada, que sale de hacer  $f_{alias} = F_s - f_2 = 8000 \text{ Hz} - 4500 \text{ Hz} = 3500 \text{ Hz}$ :



## Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



Este problema se soluciona cuando se cambia  $F_s = 48\text{kHz}$ , pudiéndose ver los picos en  $f_1 = 500\text{ Hz}$ , y  $f_2 = 4500\text{ Hz}$  como es debido:

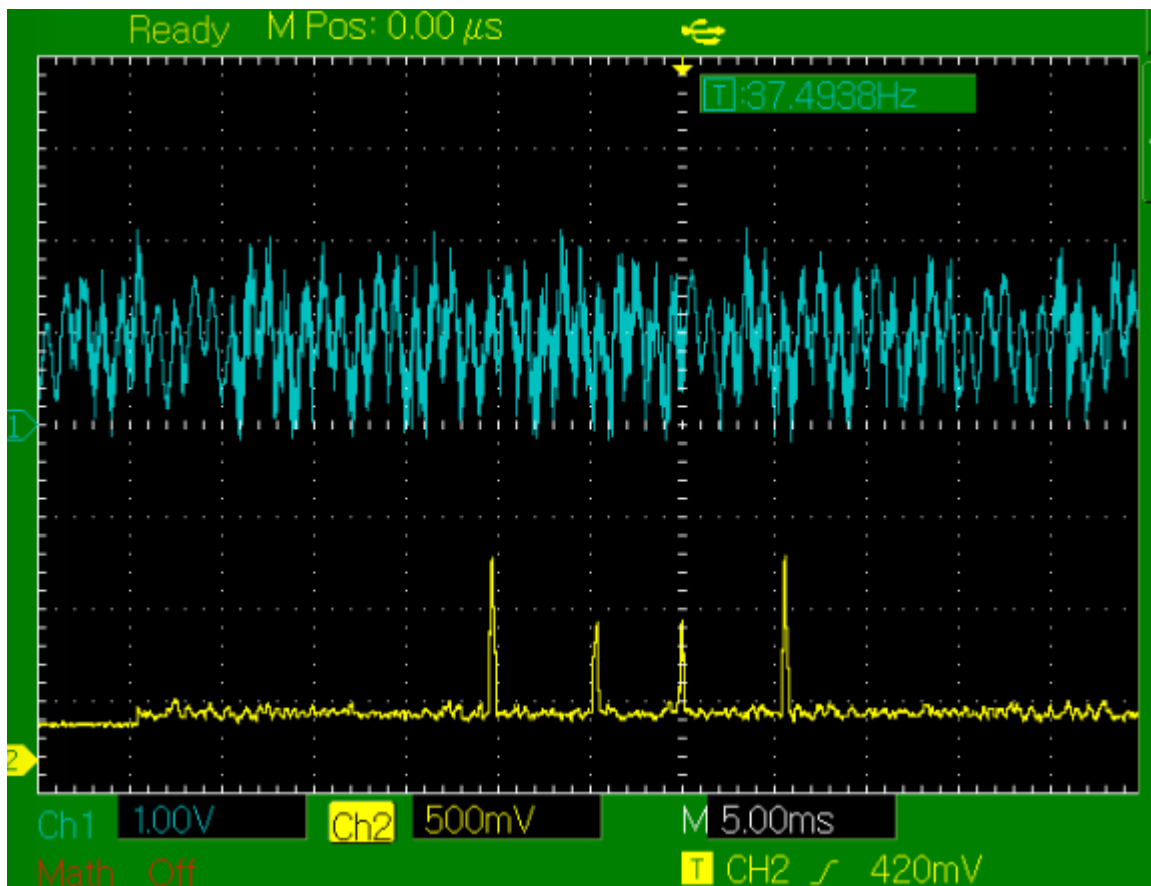


**Facultad de Ciencia Exactas, Físicas y Naturales**
  
**Procesamiento digital de señales**

Adicionalmente el buffer de salida es transmitido al DAC, pudiéndose visualizar el espectro de la señal de entrada, en el osciloscopio.

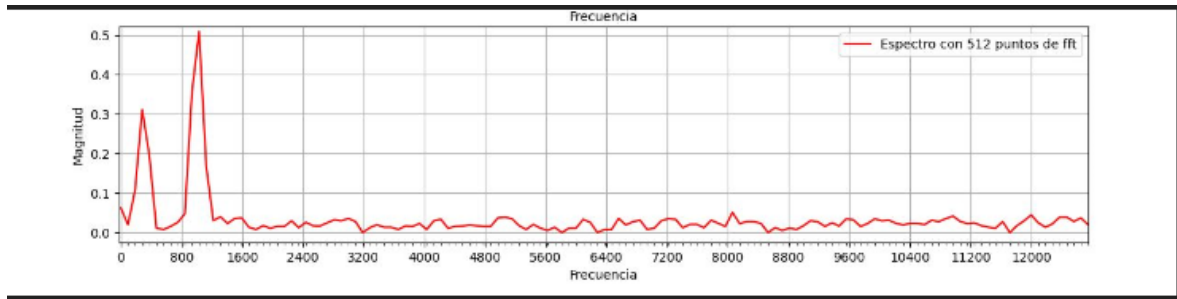
**Señal de entrada senoidal de  $f=300\text{ Hz}+1000\text{ Hz}+\text{noise}$**

**Para  $F_s=8\text{ kHz}$ ,  $N=512$**



**Para  $F_s=48\text{ kHz}$ ,  $N=512$**

## Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



### [Proyecto en formato .zip](#)

### Conclusión:

El saber cómo implementar una transformada de Fourier discreta es una herramienta vital y básica en toda aplicación de DSP, el proceso para llevarlo a cabo tuvo algunos inconvenientes principalmente por las inconsistencias encontradas en la API de CMSIS sobre la aplicación del algoritmo FFT, donde se contradice qué valor debe tener el flag correspondiente al orden en bit inverso para que los valores de la salida posean valores normales o expresados en bit inverso. Solucionada la cuestión el resto del trabajo fue relativamente sencillo.

#### Parameters

[in,out] <b>S</b>	points to an instance of the Q15 RFFT/RIFFT structure
[in] <b>fftLenReal</b>	length of the FFT
[in] <b>ifftFlagR</b>	flag that selects transform direction <ul style="list-style-type: none"> <li>• value = 0: forward transform</li> <li>• value = 1: inverse transform</li> </ul>
[in] <b>bitReverseFlag</b>	flag that enables / disables bit reversal of output <ul style="list-style-type: none"> <li>• value = 0: <u>disables bit reversal of output</u></li> <li>• value = 1: enables bit reversal of output</li> </ul>

#### Returns

execution status

- **ARM\_MATH\_SUCCESS** : Operation successful
- **ARM\_MATH\_ARGUMENT\_ERROR** : fftLenReal is not a supported length

#### Details

The parameter fftLenReal specifies length of RFFT/RIFFT Process. Supported FFT Lengths are 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192.

The parameter ifftFlagR controls whether a forward or inverse transform is computed. Set(=1) ifftFlagR to calculate RIFFT, otherwise RFFT is calculated.

The parameter bitReverseFlag controls whether output is in normal order or bit reversed order. Set(=1) bitReverseFlag for output to be in normal order otherwise output is in bit reversed order.

This function also initializes Twiddle factor table.

**Facultad de Ciencia Exactas, Físicas y Naturales**  
**Procesamiento digital de señales**

Un concepto importante aprendido a lo largo del trabajo, es que a mayores puntos tomados en la FFT ( $N$ ), la resolución en el dominio de la frecuencia aumenta (bin) esto también se puede lograr disminuyendo la frecuencia de muestreo, pero muchas veces no conviene hacer eso porque se corre riesgo de obtener aliasing de la señal de entrada (todo depende de la frecuencia máxima de la señal a analizar).