



**FACULTAD
DE INGENIERIA**
Universidad de Buenos Aires

**Procesamiento de Señales de Carrera
de la Maestría en Sistemas Embebidos**

10pts

TP1

Autor

Esp. Ing. Alejandro Permingeat

Tabla de contenido

Registros de cambios	3
1 . LTI	4
2. Ruido de cuantización	5
3. Filtro antialias	7
4. Generación y simulación	8
5. Adquisición y reconstrucción con la CIAA	15
5.1 Generación y muestreo de señal senoidal de 440 Hz	15
5.2. Realice el mismo experimento con una cuadrada y una triangular	30
5.2.a Señal Triangular.	30
5.2.b Señal Cuadrada.	44
6. Número Q	58

Registros de cambios

Revisión	Detalle de los cambios realizados	Fecha
1.0	Creación del documento	17/06/2020

1 . LTI

Demuestre si los siguientes sistemas son LTI:

- a) $y(t) = x(t) * \cos(t)$
- b) $y(t) = \cos(x(t))$
- c) $y(t) = e^{x(t)}$
- d) $y(t) = 1/2 x(t)$

Un sistema es lineal invariante en el tiempo cuando cumple dos condiciones: linealidad (escalonado, adición y superposición) e invariabilidad en el tiempo.

- a) $y(t) = x(t) * \cos(t)$

El sistema es variante en el tiempo, ya que el $\cos(t)$ es dependiente del tiempo. Basta con un ejemplo. Consideremos 2 tiempos distintos $t=1$ y $t=45$.

con $t=1$, $x(t) * \cos(t) \rightarrow x(1)*\cos(1) = x(1) * 0,999847695$

con $t=45$, $x(t) * \cos(t) \rightarrow x(45)*\cos(45) = x(45) * 0,707106781$

como se puede ver, sin importar $x(t)$, la parte $\cos(t)$ va a afectar de manera distinta según t .



Con lo cual sin analizar las demás propiedades se puede afirmar que este sistema no es LTI (Lineal invariante en el tiempo), porque no cumple con al menos ser invariante en el tiempo.

- b) $y(t) = \cos(x(t))$

El sistema no es lineal (al menos no cumple con el escalado). Por ejemplo, si $y = x(t)$ fuera $y = 6t$, y probamos escalar $x(t)$ por la constante 10 nos quedaría:

$y(t) = \cos(6t) \rightarrow$ escalado por 10 sería $\cos(10*(6t))$ lo cual es distinto a $10 \cos(6t)$.

Para demostrar esto solo basta asignarle a "t" un valor cualquiera, por ejemplo $t=1$

con $t=1$, $\cos(10*(6*1)) \rightarrow \cos(60) = 0,5$

con $t=1$, $10 * \cos(6*1) \rightarrow 10 * \cos(6) = 9,945218954$



No hace falta analizar las demás propiedades. Se puede afirmar que el sistema NO es LTI (lineal invariante en el tiempo) porque no es lineal (no cumple al menos con el escalado)

c) $y(t) = e^{x(t)}$

El sistema no es lineal (al menos no cumple con el escalado). Por ejemplo, si $y = x(t)$ fuera $y = 4t$, y probamos escalar $x(t)$ por la constante 8 nos quedaría:

$y(t) = e^{4t} \rightarrow$ escalado por 8 sería $e^{(8 \cdot 4 \cdot t)} = e^{(32 \cdot t)}$ que no es igual a $8 \cdot e^{4t}$. Para demostrar esto solo basta asignarle a "t" un valor cualquiera, por ejemplo $t=7$

con $t=7$, $e^{(32 \cdot t)} \rightarrow e^{(32 \cdot 7)} = 1,914097017 \times 10^9$

con $t=7$, $8 \cdot e^{(4 \cdot t)} \rightarrow 8 \cdot e^{(4 \cdot 7)} = 1,157005651 \times 10^{13}$

No hace falta analizar las demás propiedades. Se puede afirmar que el sistema NO es LTI (lineal invariante en el tiempo) porque no es lineal (no cumple al menos con el escalado)



d) $y(t) = 1/2 x(t)$

El sistema es lineal (cumple con escalado, adición y superposición)

El sistema es invariante en el tiempo



Este sistema es LTI (Lineal invariante en el tiempo)

2. Ruido de cuantización

a - Calcule la relación señal a ruido de cuantización teórica máxima de un sistema con un ADC de:

- 24 bits
- 16 bits
- 10 bits
- 8 bits



- 2 bits

Para calcular la relación señal a ruido se utiliza la siguiente fórmula:

$$SNR = 1,76 + 6,02 * N$$

Con lo cual, la relación señal a ruido de los ADC, según la cantidad de bits es:

- 24 bits -> 146.24 dB
- 16 bits -> 98.08 dB
- 10 bits -> 61,96 dB
- 8 bits -> 49,92 dB
- 2 bits -> 13.8 dB



b - Dado un sistema con un ADC de 10 bits, que técnica le permitiría aumentar la SNR? En qué consiste?

Dado un sistema con un ADC de 10 bits, la relación SNR por definición será de aproximadamente 62 dB. Para mejorar la relación SNR, una técnica que se puede utilizar es realizar el sobremuestreo. Si se sobremuestra al doble de la frecuencia se ganan 2 dB. Si se sobremuestra 4 veces de la frecuencia de muestreo original, se ganan 4 dB.

6 **3**

Esto se debe a que al incrementar la frecuencia de muestreo, la densidad espectral de potencia de error de cuantización se distribuye entre más muestras, con lo cual cada muestra tendrá menor error de cuantización.



3. Filtro antialias

Calcular el filtro antialias que utilizara para su práctica y/o trabajo final y justifique su decisión.

Para la práctica, la frecuencia máxima de muestreo quedará condicionada a la tasa de transferencia de los datos desde la Placa hacia la PC via USRT. La tasa máxima a la que puede enviar la CIAA via UART es 460800bps. Se envían 8 bits de carga útil cada 10 bits, y el ADC es de 16 bits, con lo cual se podrán transmitir 23040 muestras por segundo (23040 Hz). **10**

Según Nyquist para reconstruir una señal de manera adecuada hace falta muestrear al doble de su frecuencia. De lo anterior se desprende que con una frecuencia de muestreo de 23040 Hz, se podrá considerar como máximo una señal de frecuencia $23040 \text{ Hz} / 2 = 11520 \text{ Hz}$.

De todo lo anterior, se propone como filtro antialias, un filtro pasabajos con una frecuencia de corte fijada en 11520 Hz.

Para el trabajo final se utilizarán muestras de un acelerómetro funcionando a 200 Hz, por lo cual la frecuencia máxima a considerar en la señal del acelerómetro es de 100Hz, con lo cual el filtro antialias debería ser un filtro pasabajo con una frecuencia de corte de 100 Hz. La lectura de valores del acelerómetro las realiza un integrado MEMs con lo cual no se puede colocar un filtro analógico. Se analizará si el integrado tienen la posibilidad de configurar un filtro pasabajos en su lógica interna y en caso afirmativo se aplicará.





4. Generación y simulación

1. Genere un módulo o paquete con al menos las siguientes funciones

- senoidal (fs[Hz], f0[Hz], amp[0 a 1], muestras),fase [radianes]
- Cuadrada (fs[Hz], f0[Hz], amp[0 a 1], muestras)
- Triangular(fs[Hz], f0[Hz], amp[0 a 1], muestras)

-senoidal:

```
import numpy as np
```

```
def generateSinSignal(fs,f,amp,sample,fase):
```

```
    """ fs: frecuencia muestreo
        f: frecuencia de la señal
        amp: amplitud [0,1]
        sample: muestras
        fase: fase en radianes
    """
```

```
    sec = sample / fs
    t = np.arange ( 0,sec,1/fs )
    signal = amp*np.sin(2 * np.pi *f*t+fase)
    return signal
```

- Cuadrada

```
import numpy as np
import scipy.signal as sc
```

```
def generateSquareSignal(fs,f,amp,sample,fase):
```

```
    """ fs: frecuencia muestreo
        f: frecuencia de la señal
        amp: amplitud [0,1]
        sample: muestras
    """
```

```
    sec = sample / fs
    t = np.arange ( 0,sec,1/fs )
    signal = amp*sc.square(2 * np.pi *f*t)
    return signal
```




- Triangular

```
import numpy as np
import scipy.signal as sc
```

```
def generateTriangularSignal(fs,f,amp,sample,fase):
```

```
    """ fs: frecuencia muestreo
        f: frecuencia de la señal
        amp: amplitud [0,1]
        sample: muestras
    """
```



```
    sec = sample / fs
    t = np.arange ( 0,sec,1/fs )
    signal = amp*sc.sawtooth(2 * np.pi *f*t)
    return signal
```

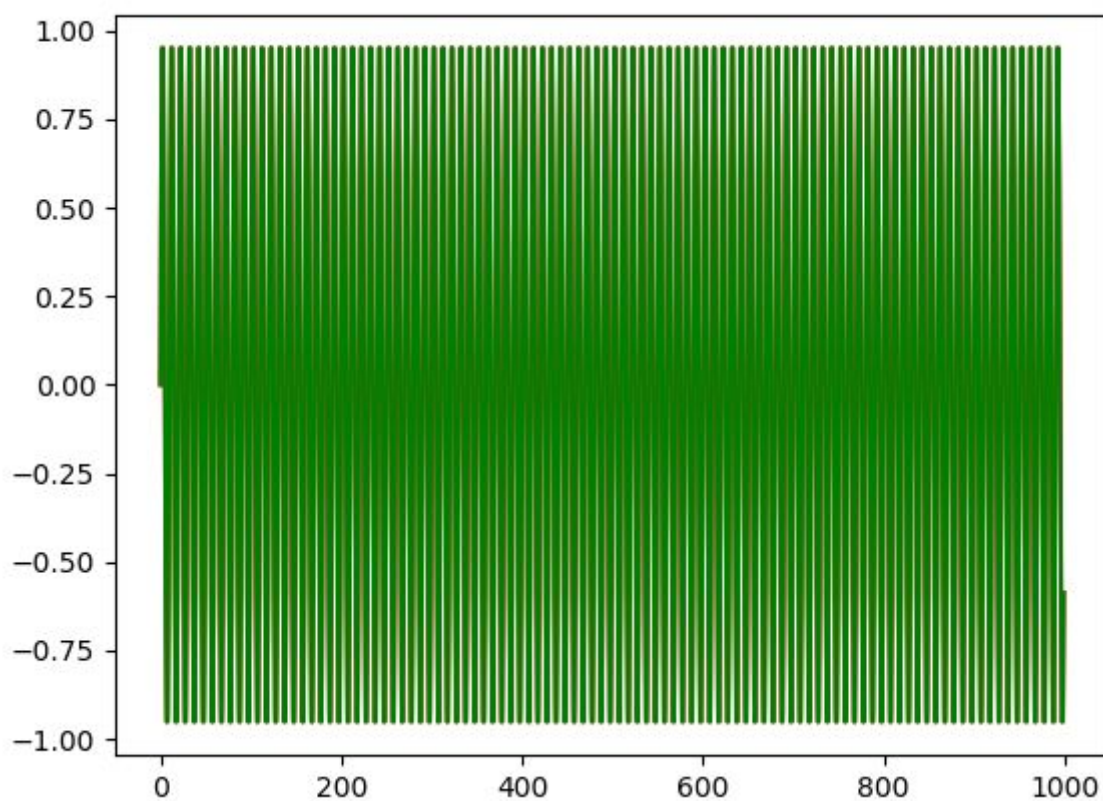
2. Realice los siguientes experimentos

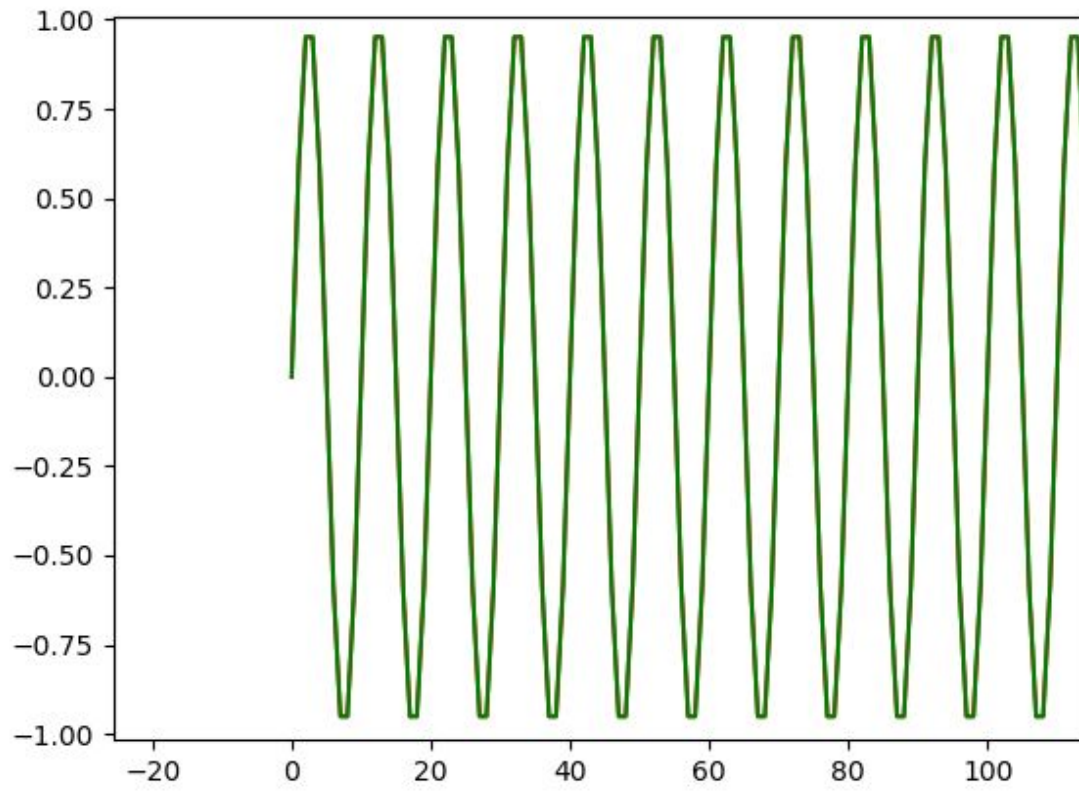
- $fs = 1000$
- $N = 1000$
- $fase = 0$
- $amp = 1$
-

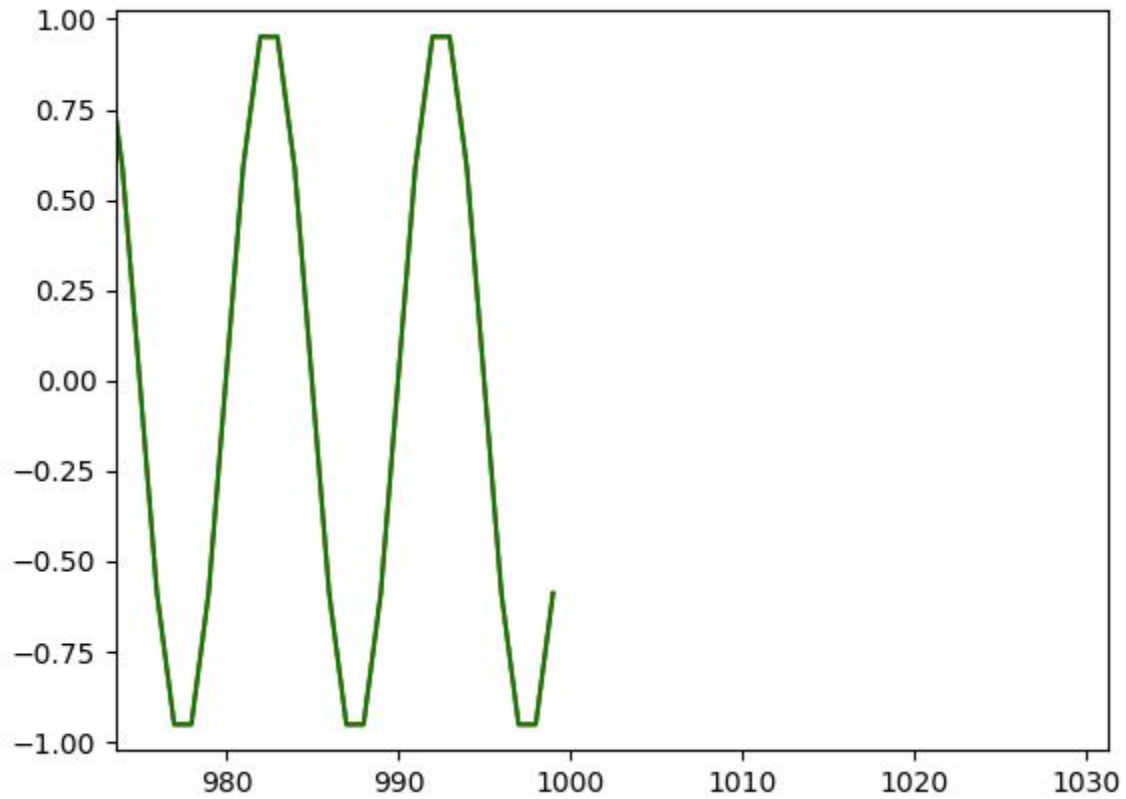


2.1 $f_0 = 0.1*fs$ y $1.1*fs$ Cómo podría diferenciar las senoidales?

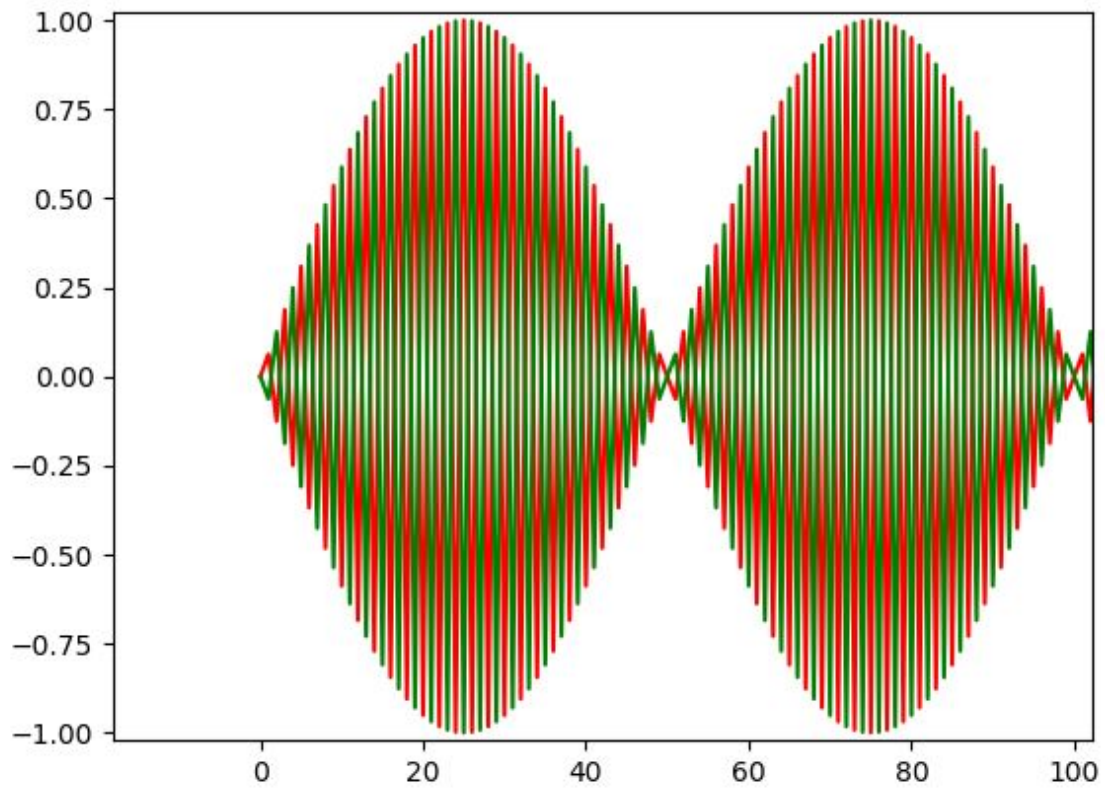
No se pueden diferenciar.

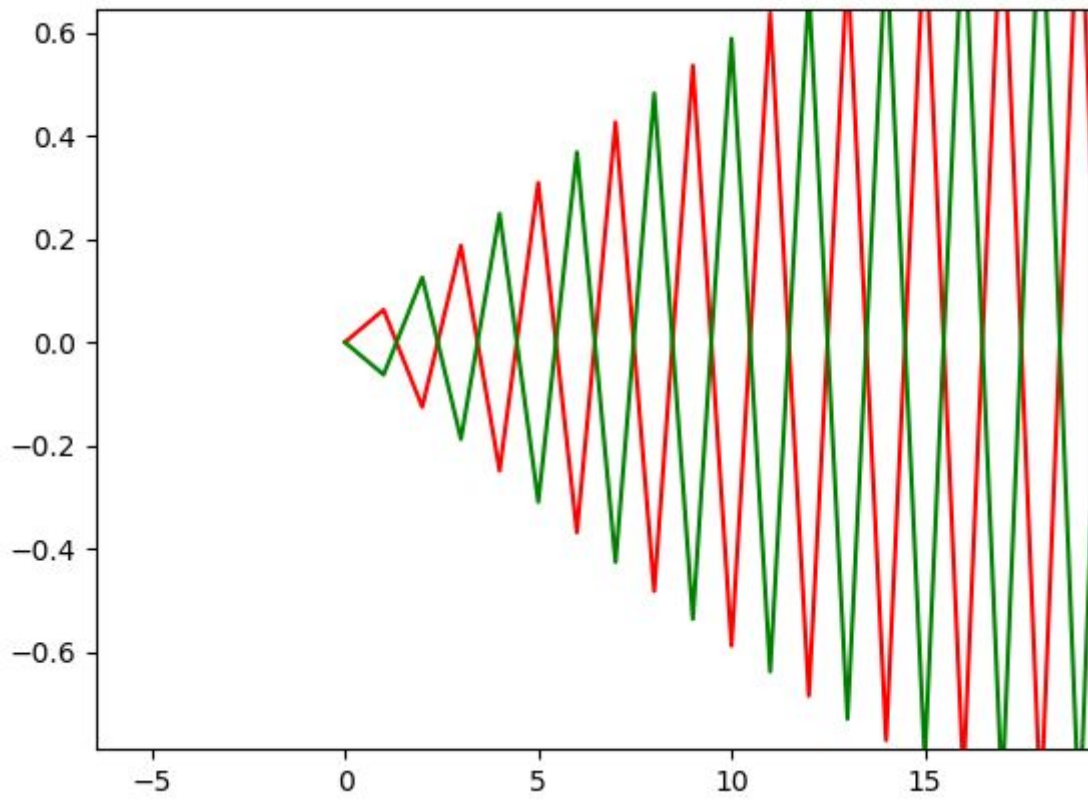






2.2 $f_0 = 0.49 \cdot f_s$ y $0.51 \cdot f_s$ Como es la frecuencia y la fase entre ambas?
La frecuencia es la misma y la fase es 180° una respecto de la otra







5. Adquisición y reconstrucción con la CIAA

5.1 Generación y muestreo de señal senoidal de 440 Hz

Genere con un tono de LA-440. Digitalice con 10, 8, 4 y 2 bits con el ADC, envíe los datos a la PC, grafique y comente los resultados

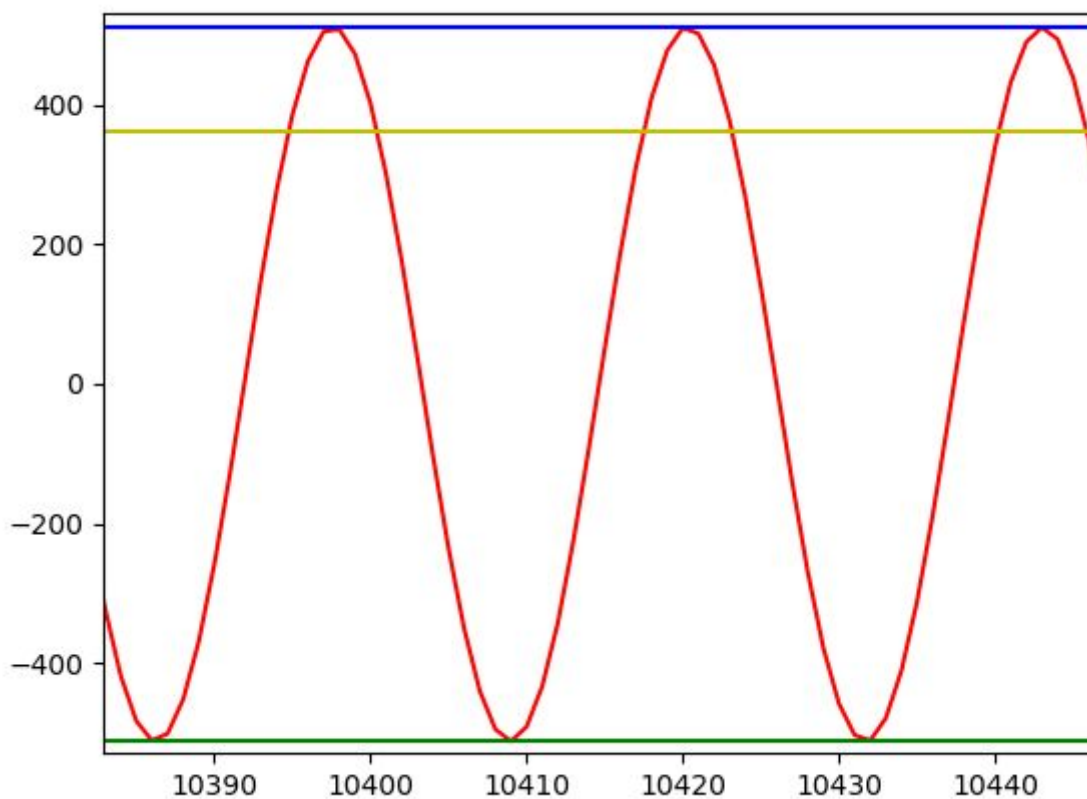
- Señal original con su máximo, mínimo y RMS
- Señal adquirida con su máximo, mínimo y RMS
- Señal error = Original-Adquirida
- Histograma del error

Señal original, generada por el DAC de la EDU-CIAA (se elige generarla de esta manera para facilitar el cálculo de la señal ruido)

valor máximo (línea azul): 511

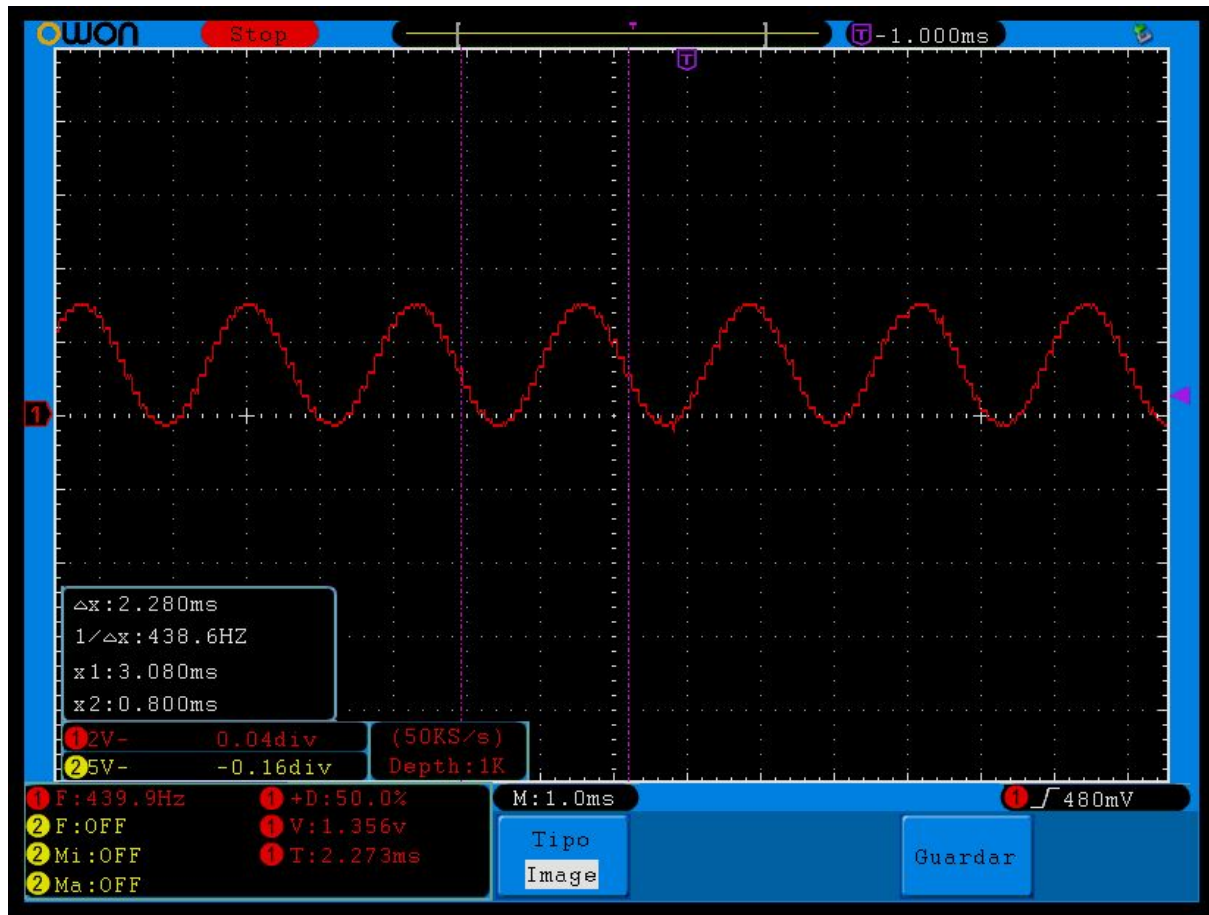
valor mínimo (línea verde): -512

valor RMS (línea amarilla): 361.9991140126944



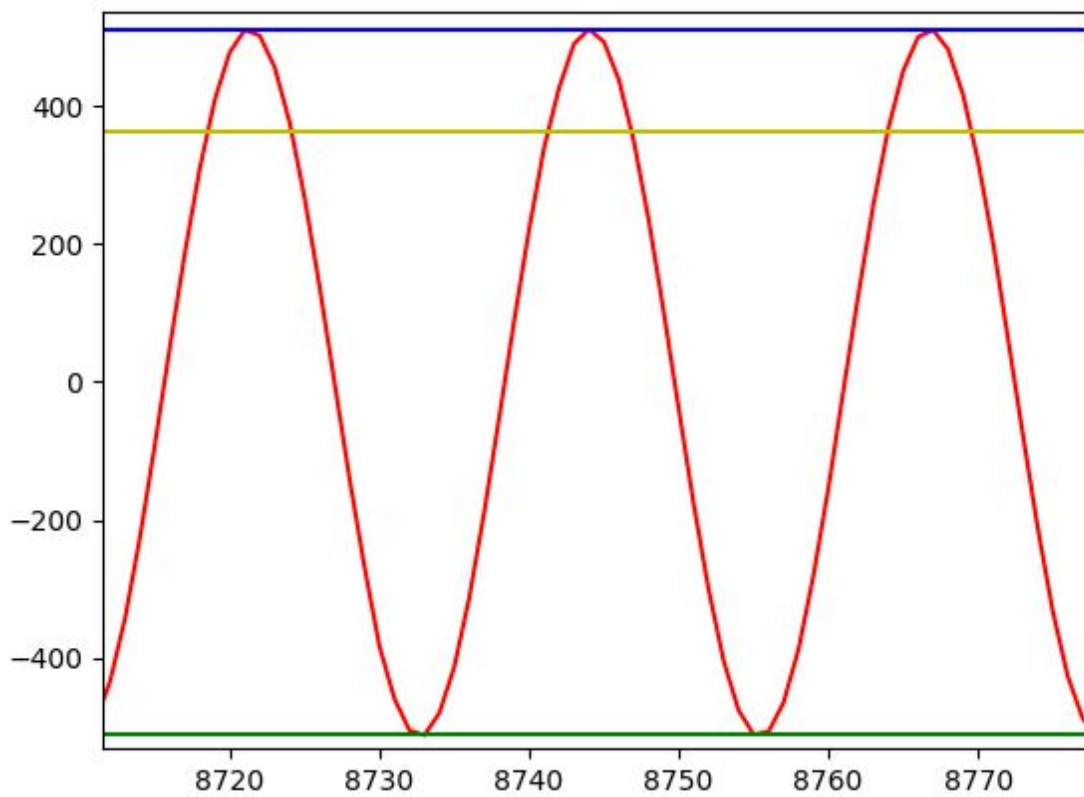


Señal observada en Osciloscopio

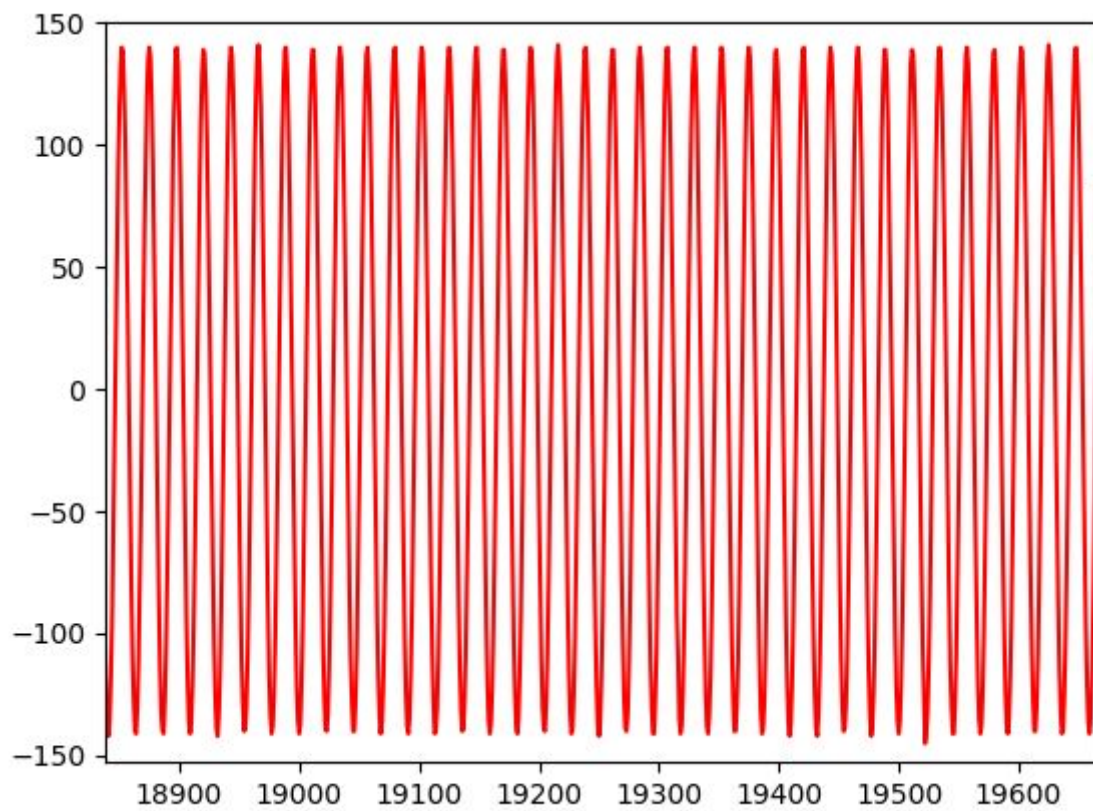


Señal adquirida (ADC 10 bits)

- valor máximo (línea azul): 511
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 362.21870893005837

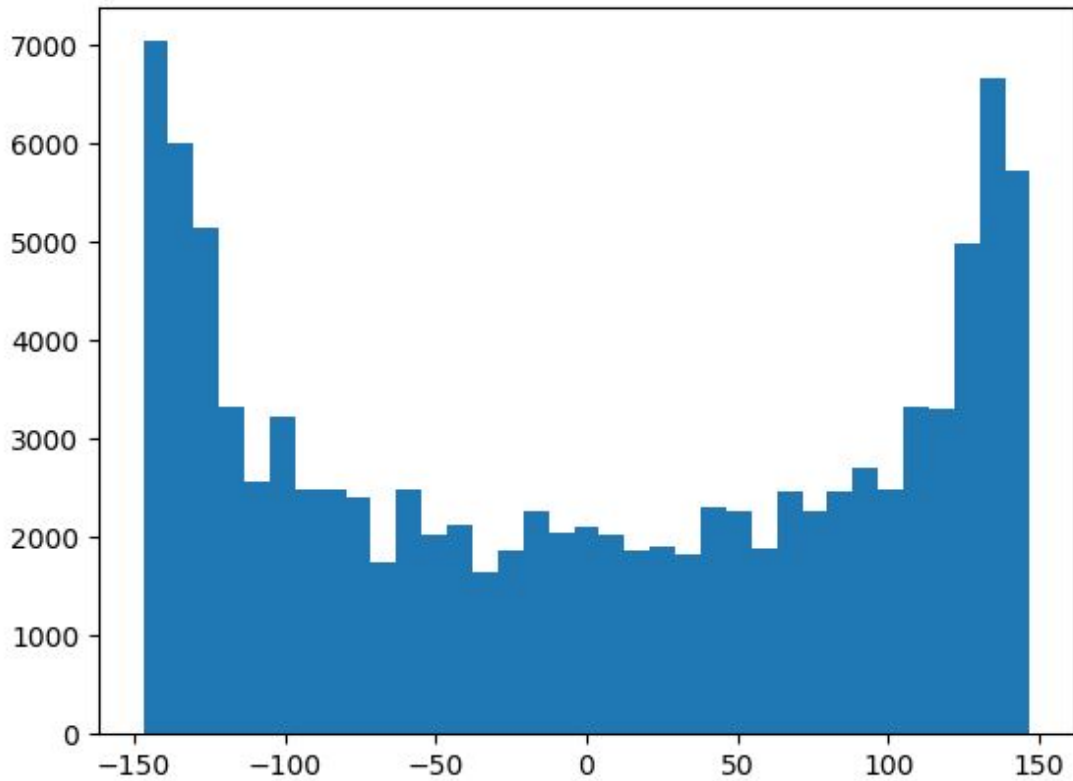


Señal error (10 bits)





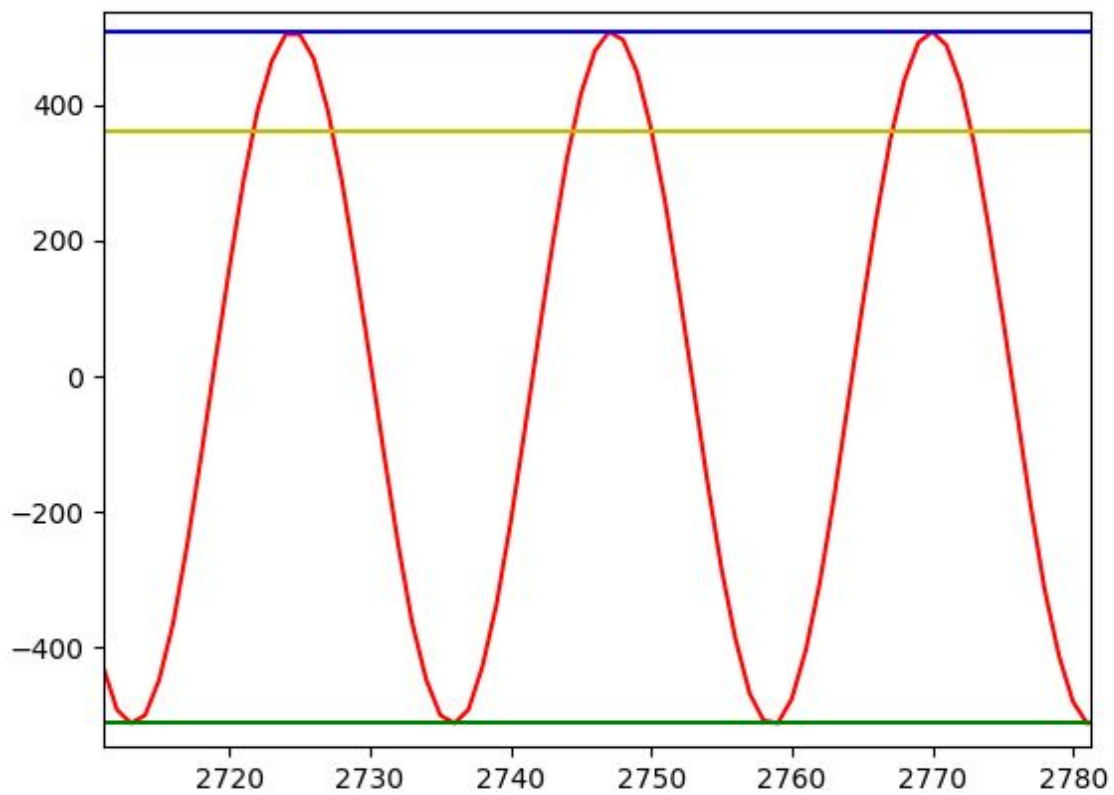
Histograma señal error (10 bits)



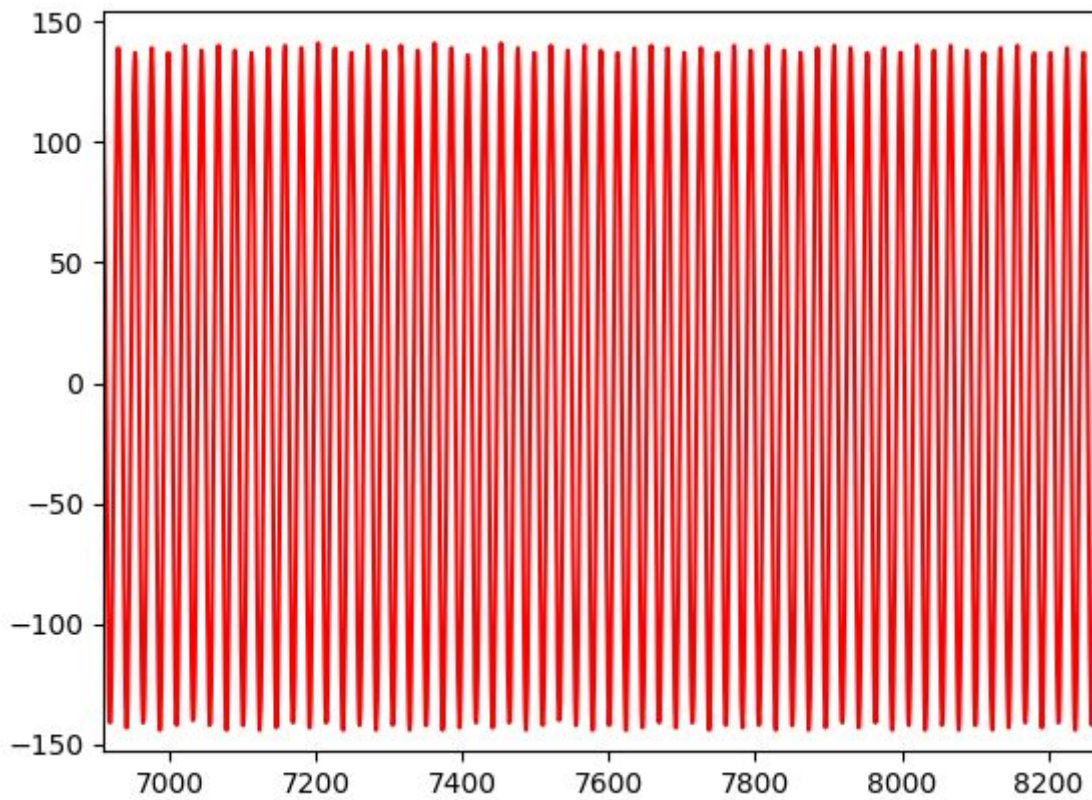


Señal adquirida (ADC 8 bits)

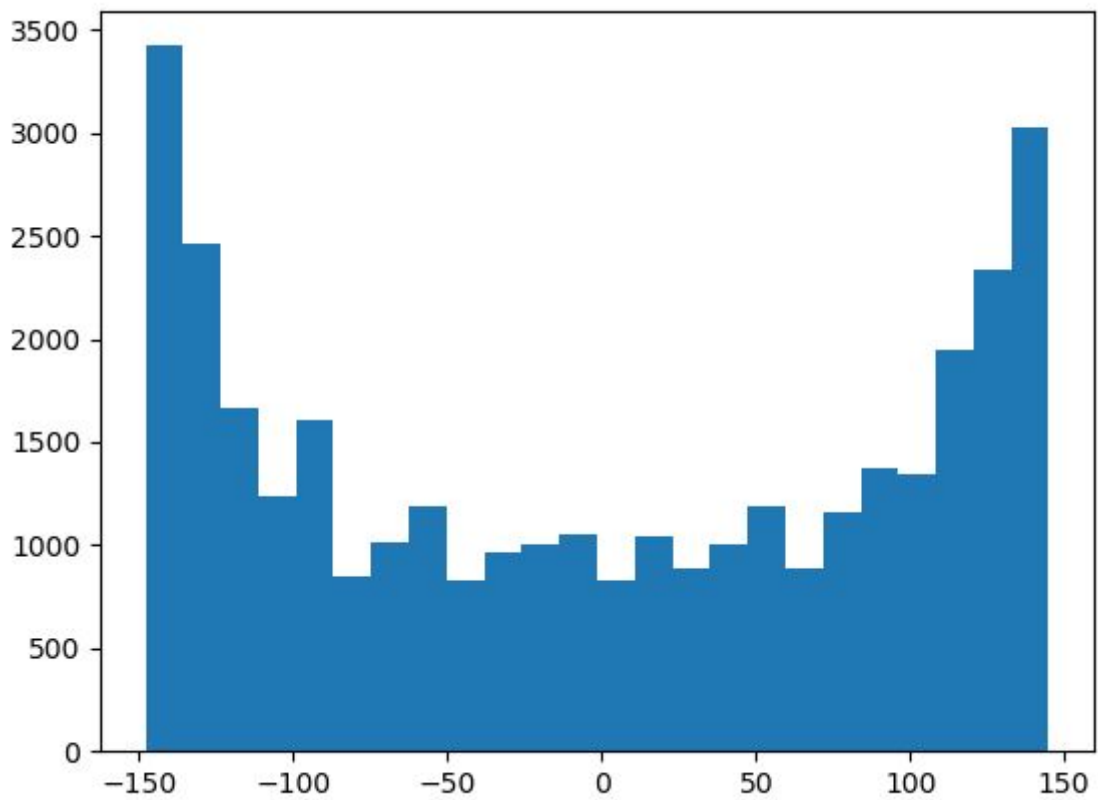
- valor máximo (línea azul): 508
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 362.1314784471127



Señal error (8 bits)

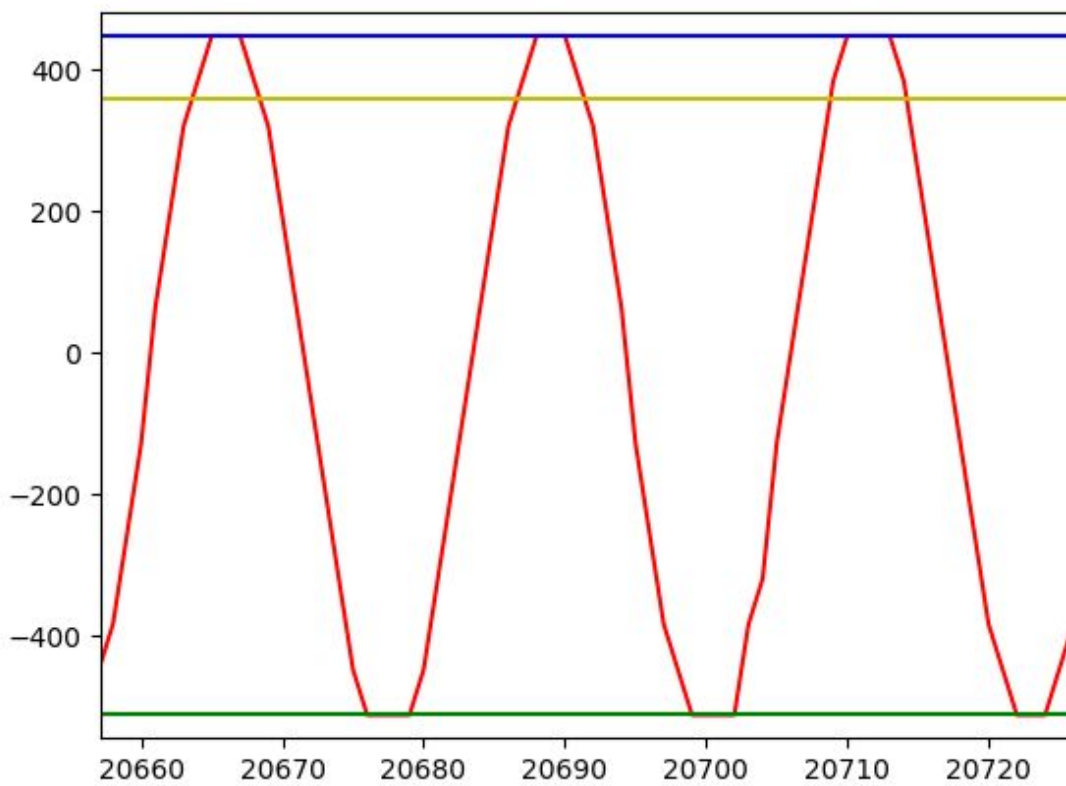


histograma (señal error 8 bits)

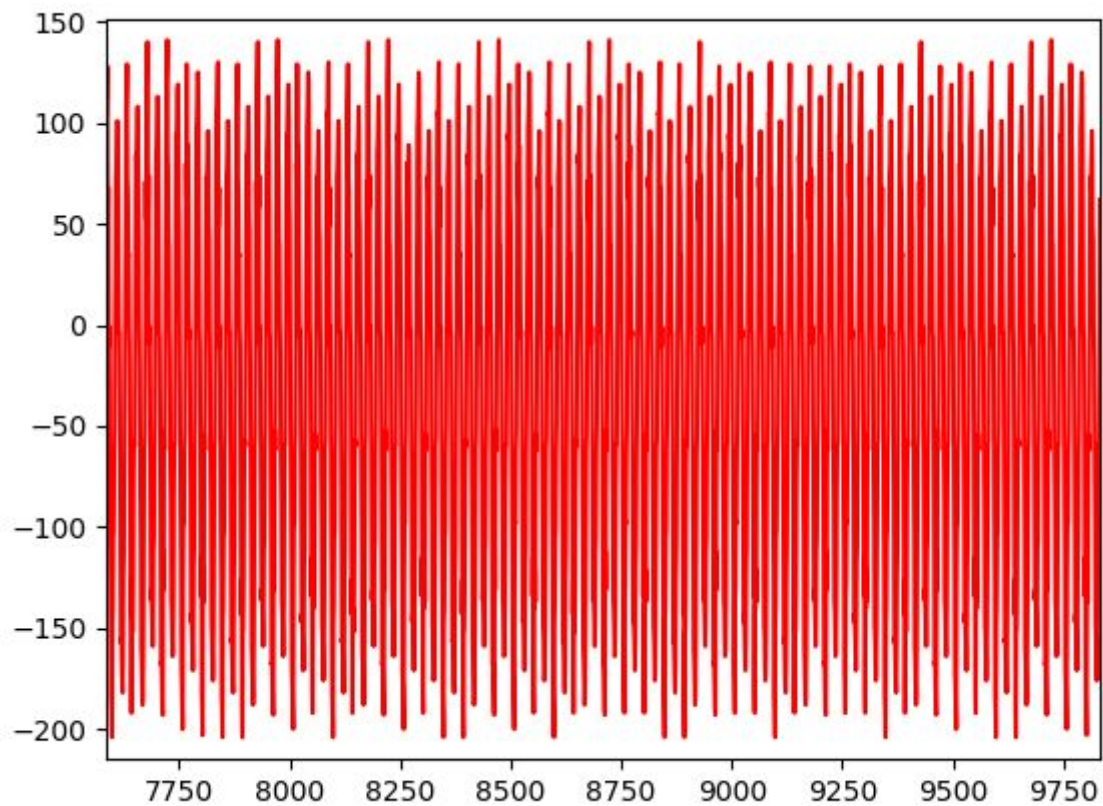


Señal adquirida (ADC 4 bits)

- valor máximo (línea azul): 448
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 358.5503069035713

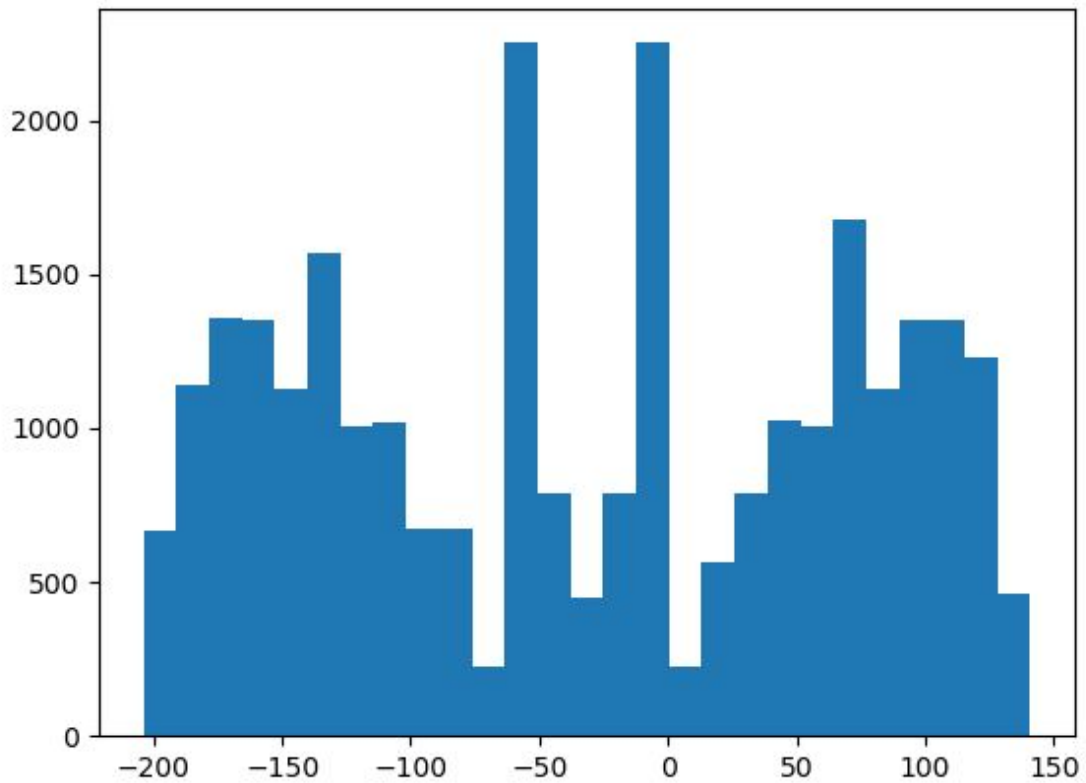


Señal error (4 bits)



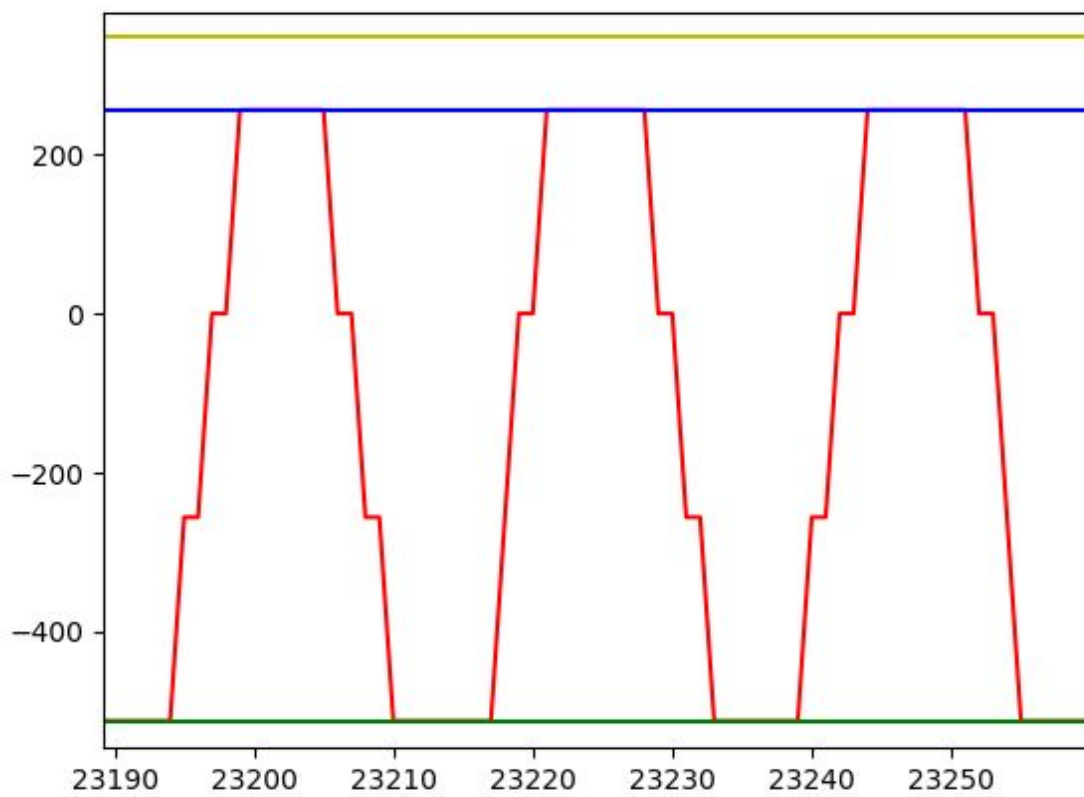


histograma (señal error 4 bits)

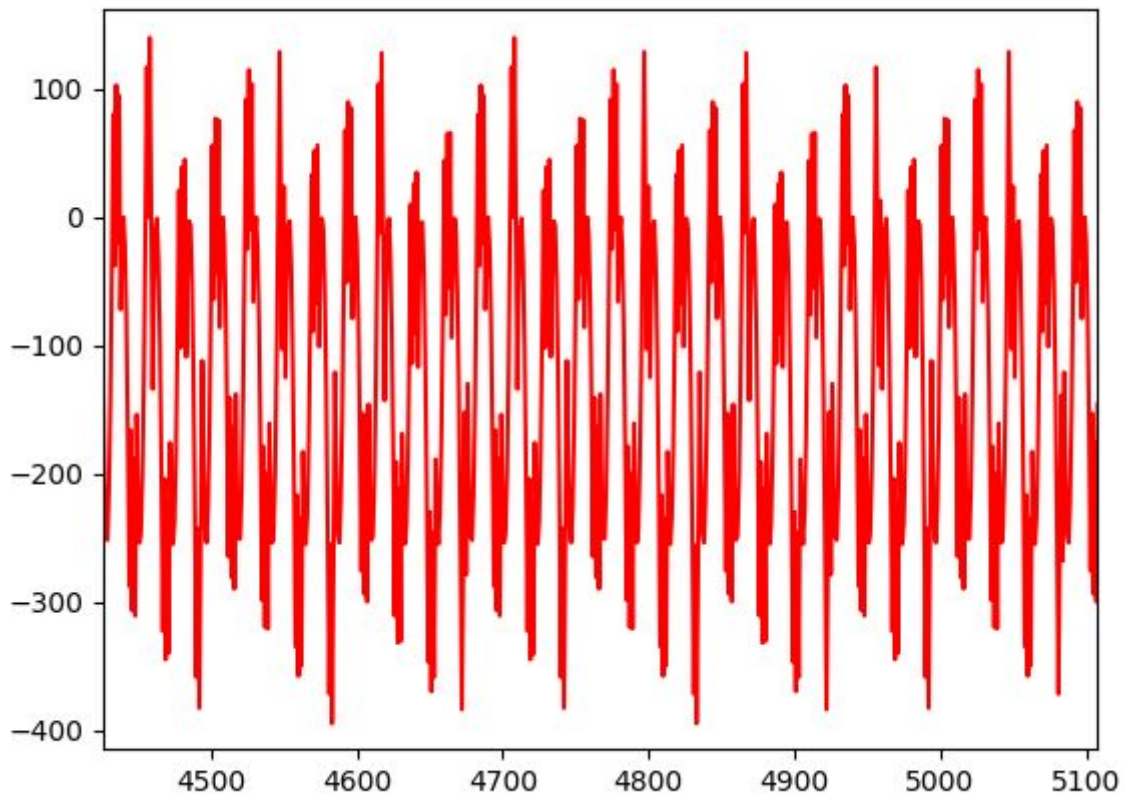


Señal adquirida (ADC 2 bits)

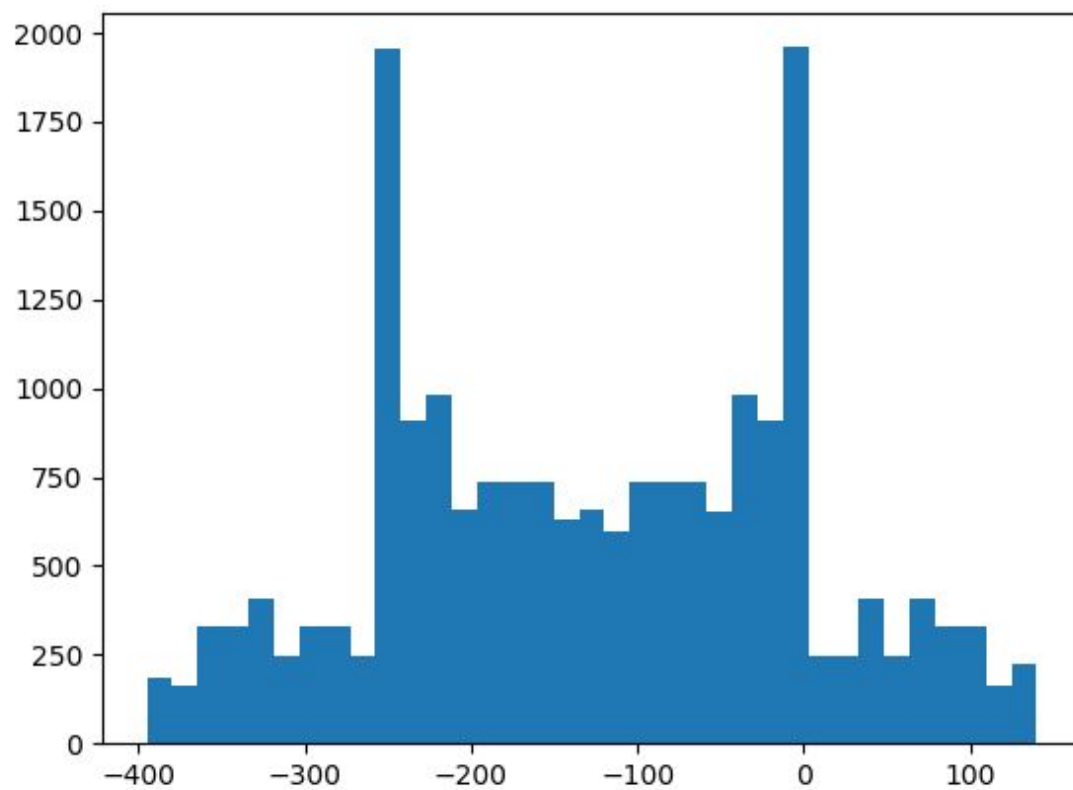
- valor máximo (línea azul): 256
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 347.4529110882924



Señal error (2 bits)



histograma (señal error 2 bits)

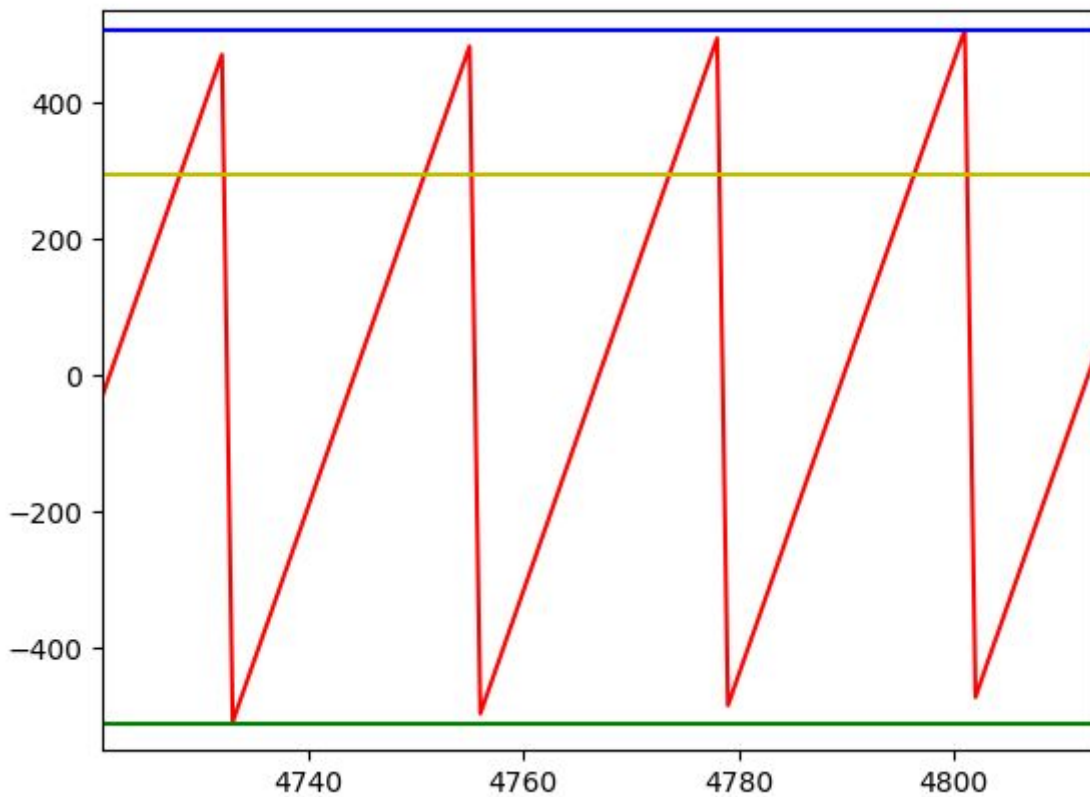


5.2. Realice el mismo experimento con una cuadrada y una triangular

5.2.a Señal Triangular.

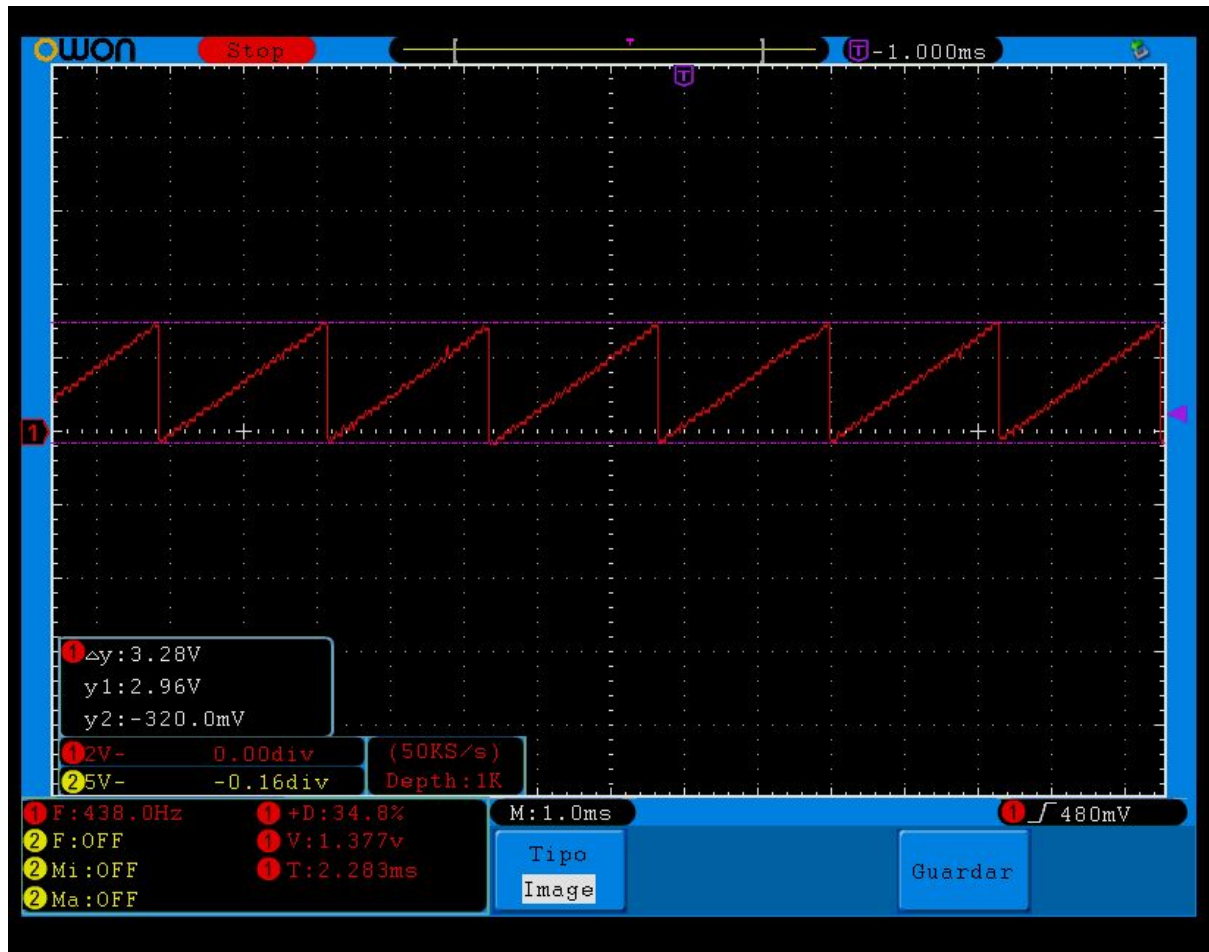
Señal triangular original, generada por el DAC de la EDU-CIAA (se elige generarla de esta manera para facilitar el cálculo de la señal ruido)

- valor máximo (línea azul): 507
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 295.60714853166326



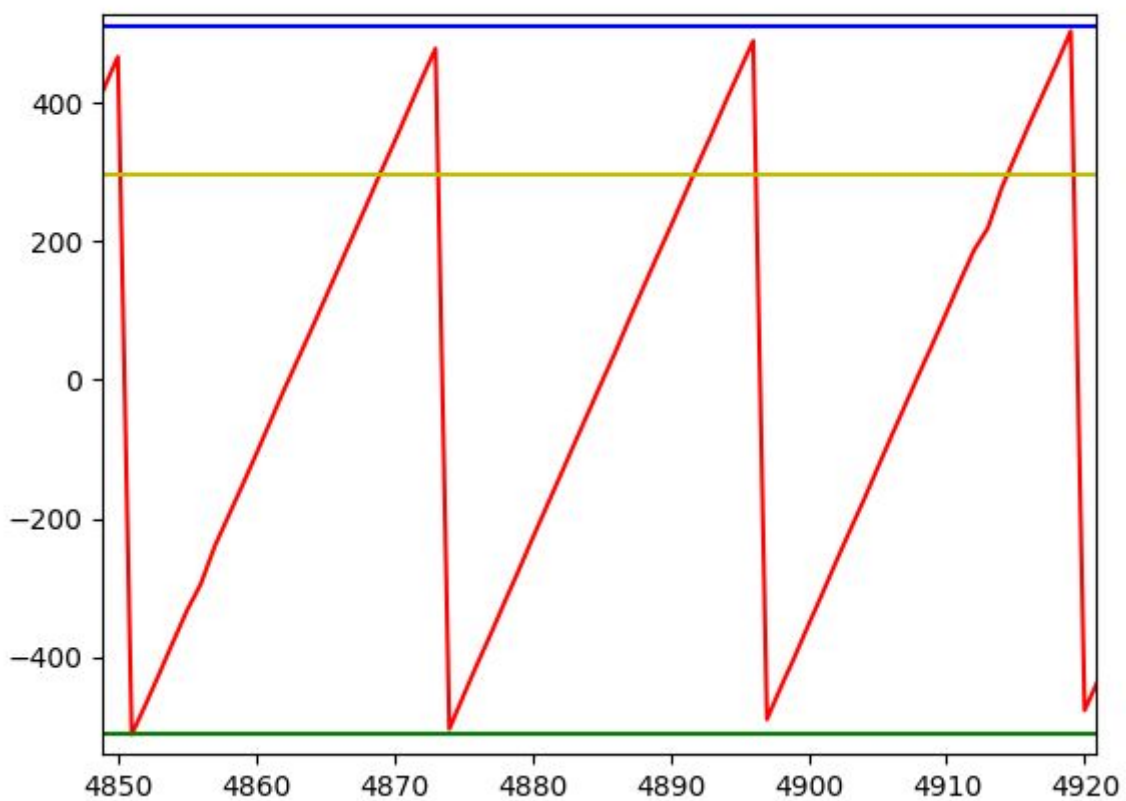


Señal observada en Osciloscopio



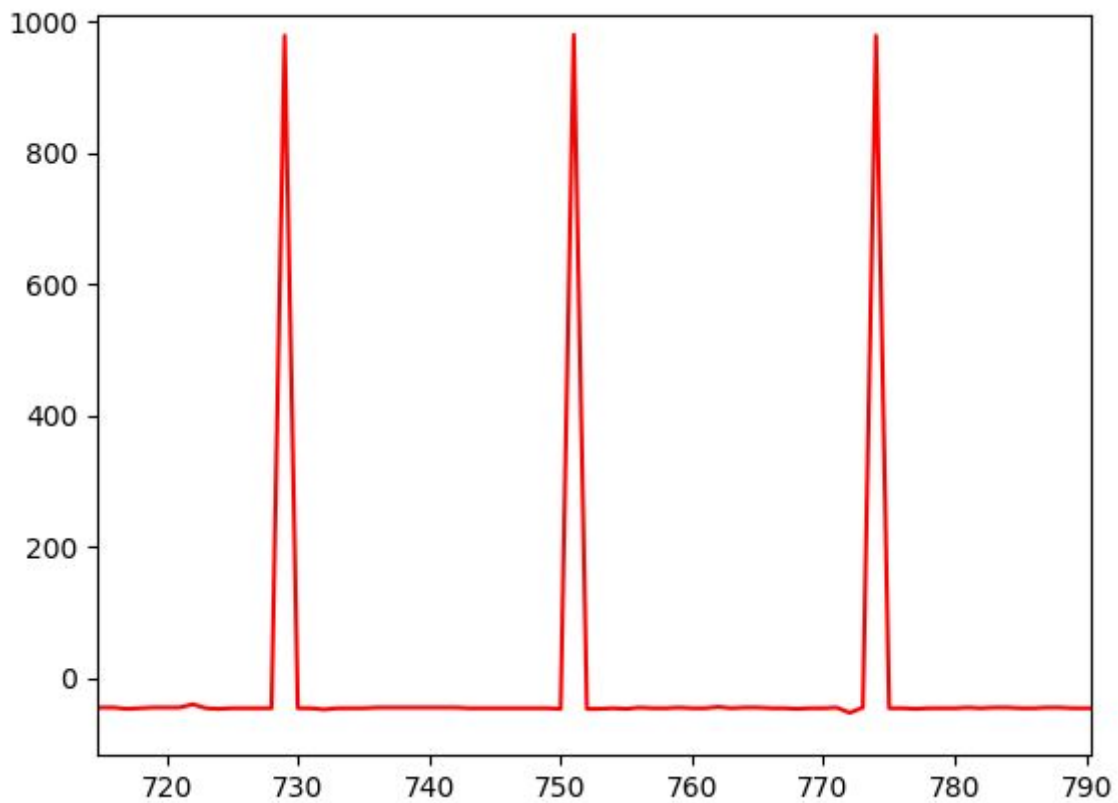
Señal adquirida (ADC 10 bits)

- valor máximo (línea azul): 511
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 295.7390347082903

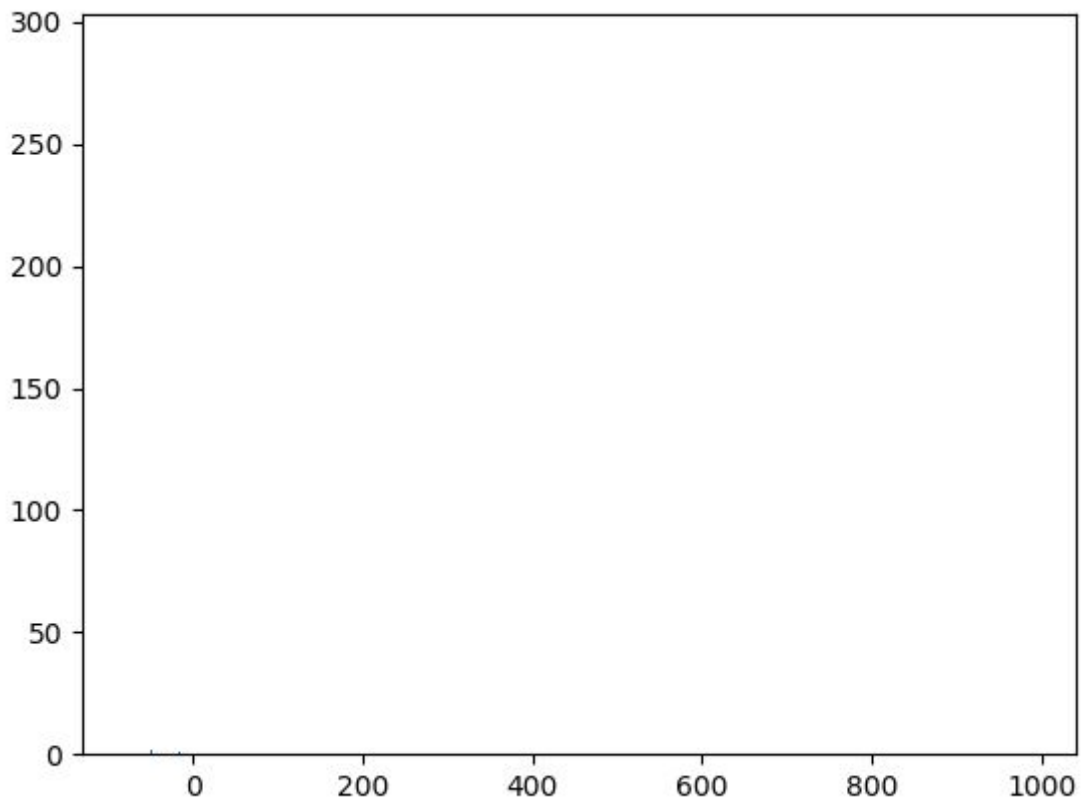




Señal error (10 bits)



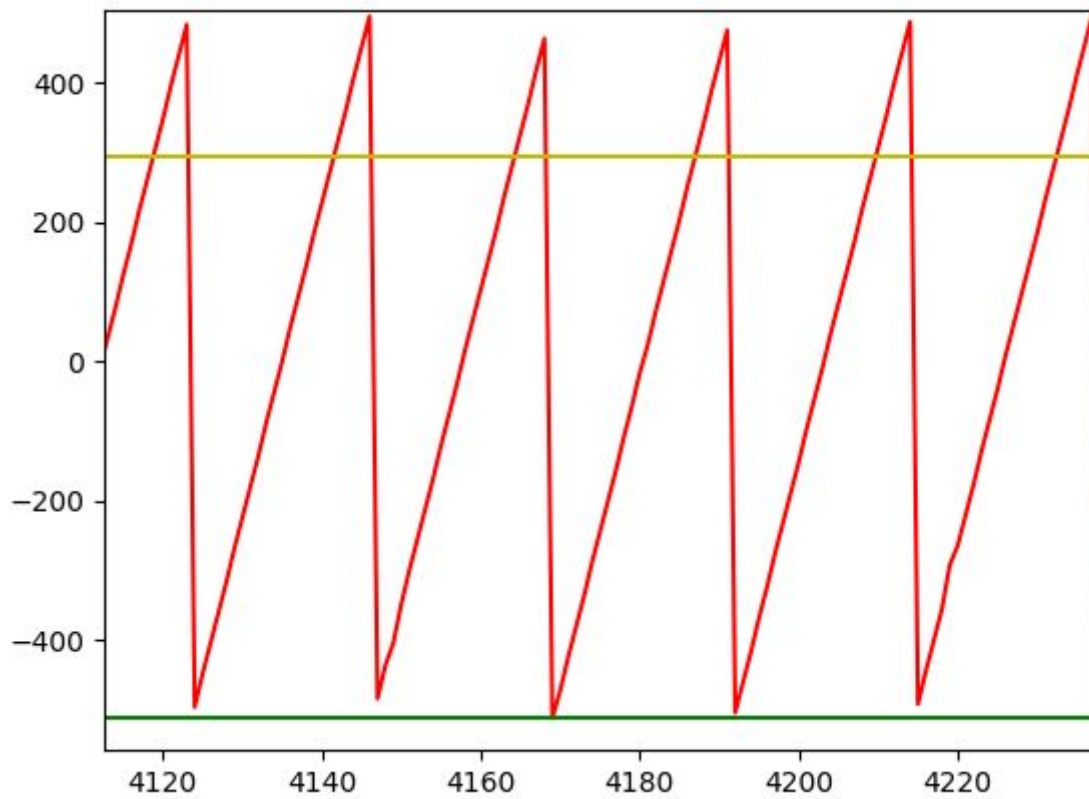
histograma (señal error 10 bits)





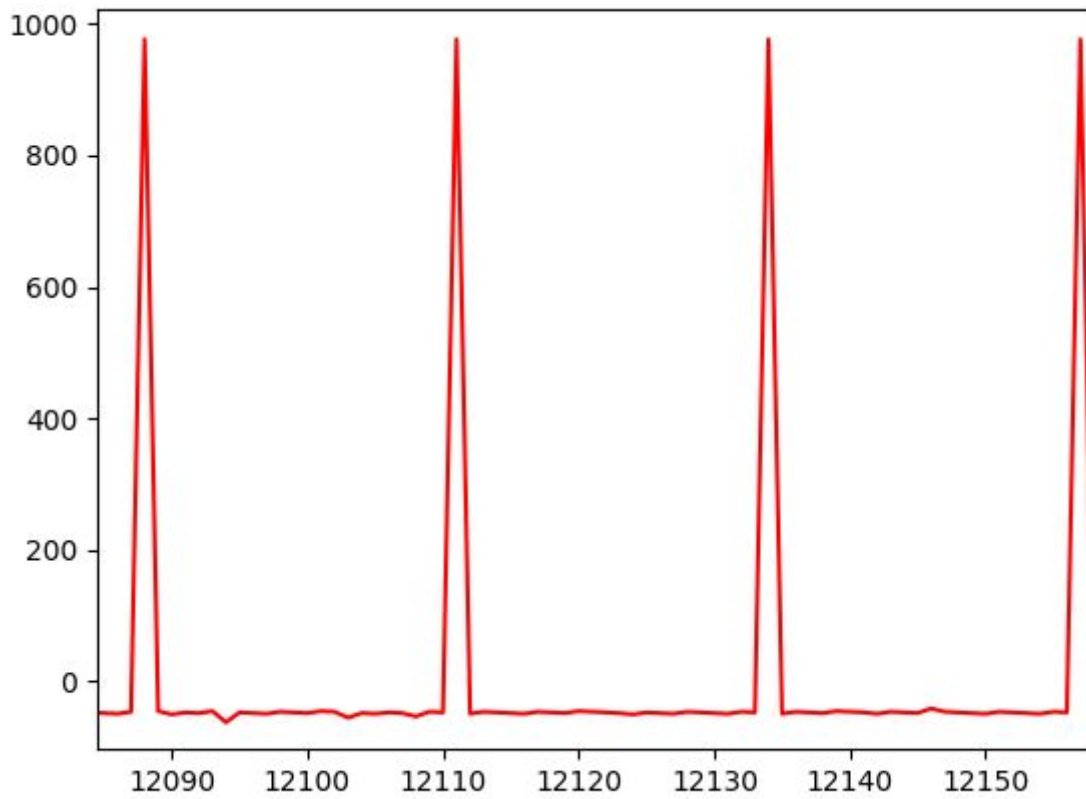
Señal adquirida (ADC 8 bits)

- valor máximo (línea azul): 508
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 295.77062687074283



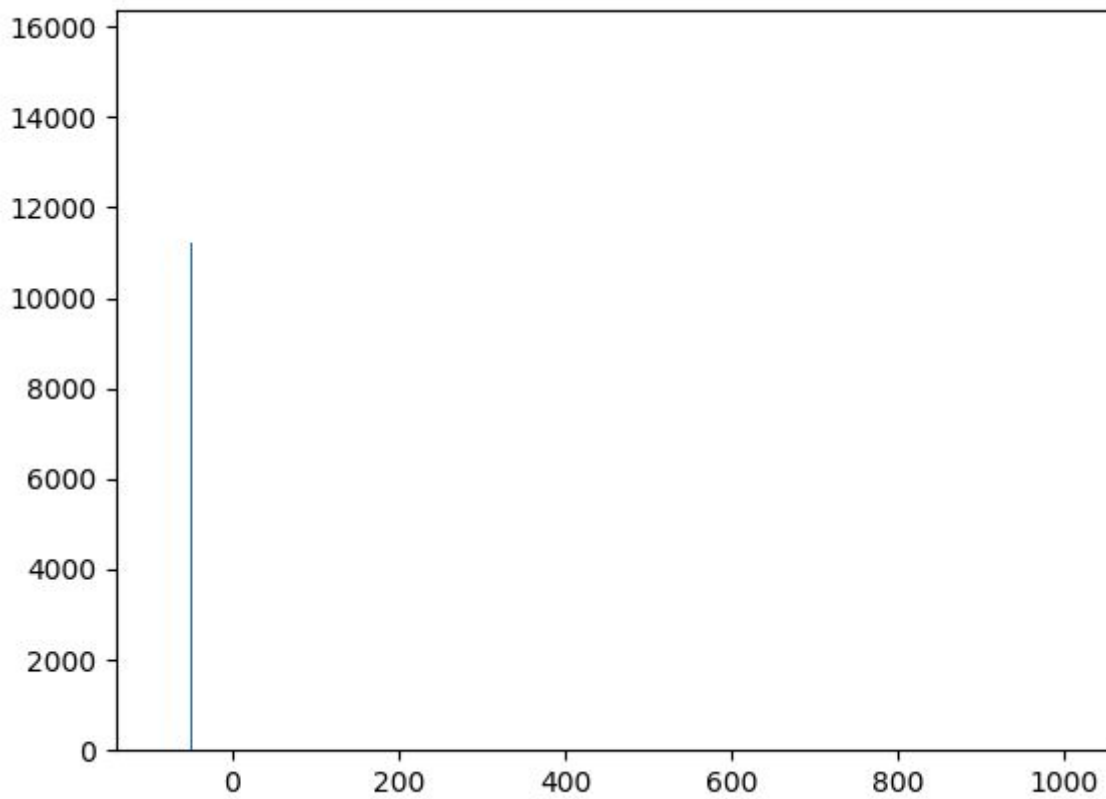


Señal error (8 bits)



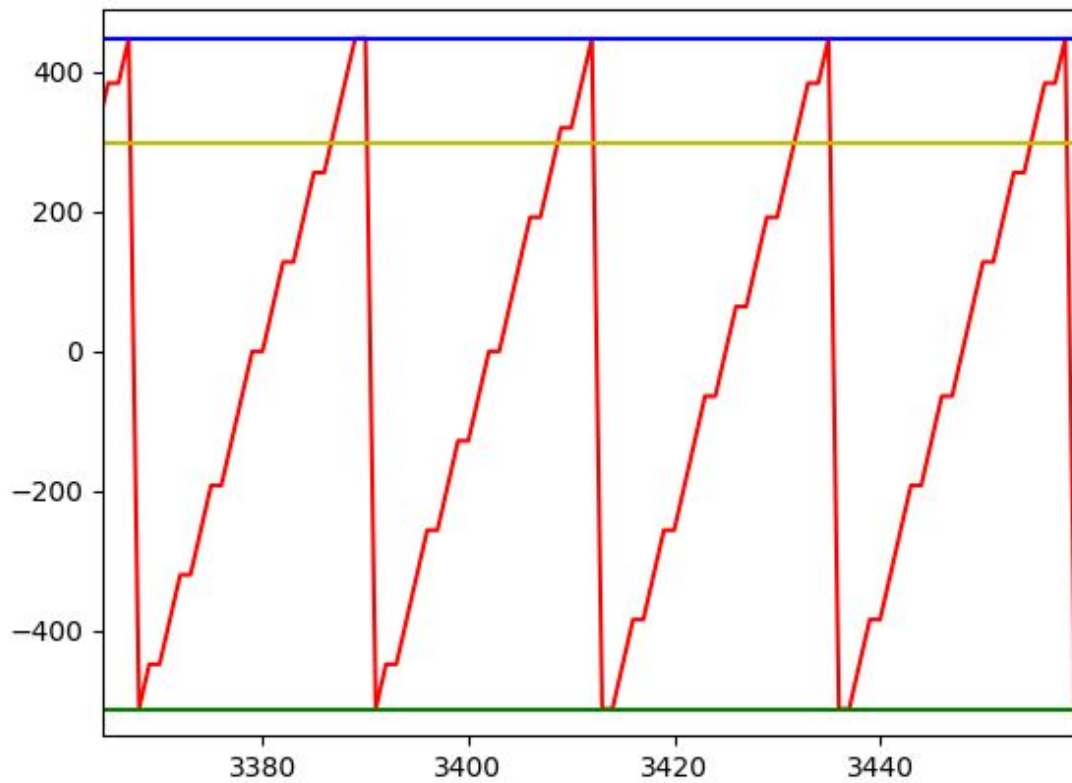


histograma (señal error 8 bits)

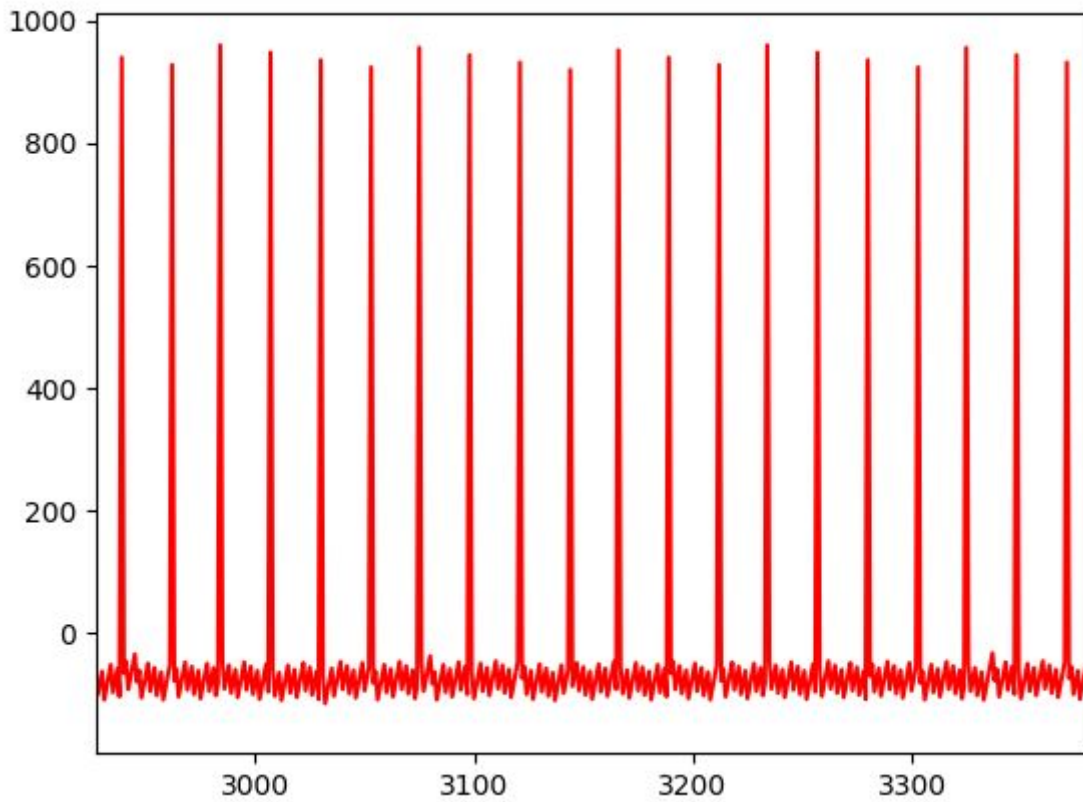


Señal adquirida (ADC 4 bits)

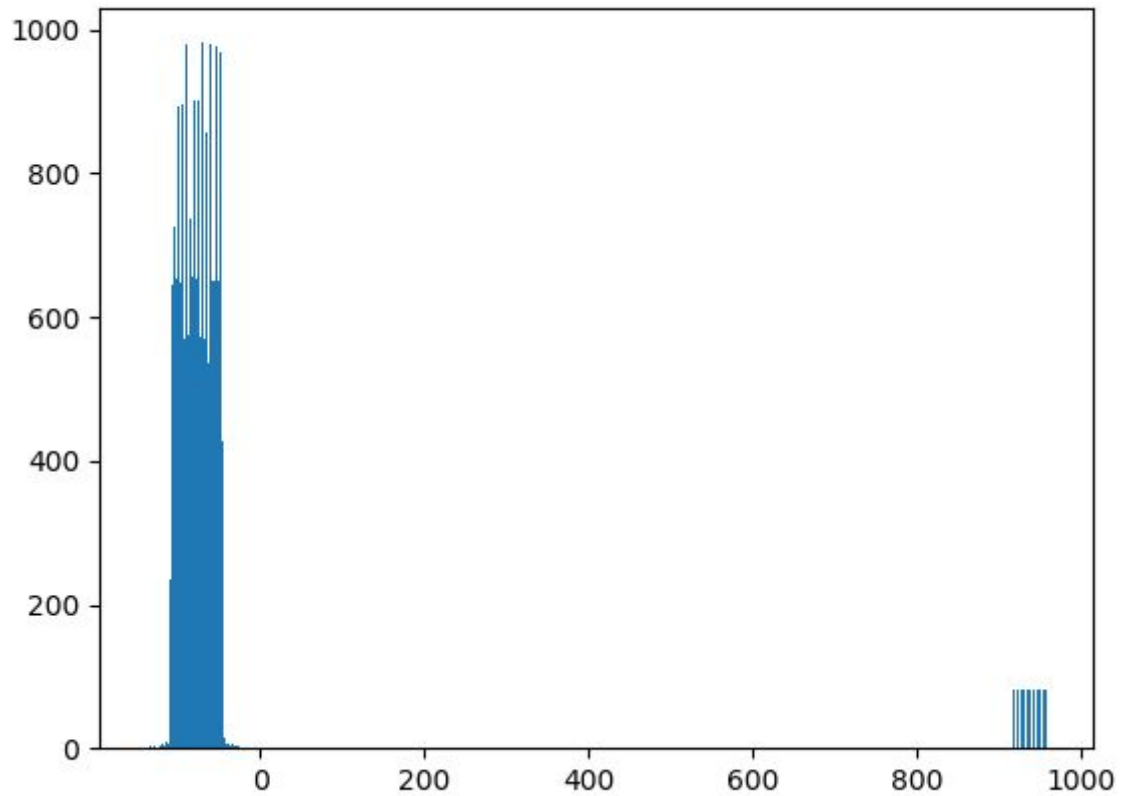
- valor máximo (línea azul): 448
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 297.36069783242135



Señal error (4 bits)



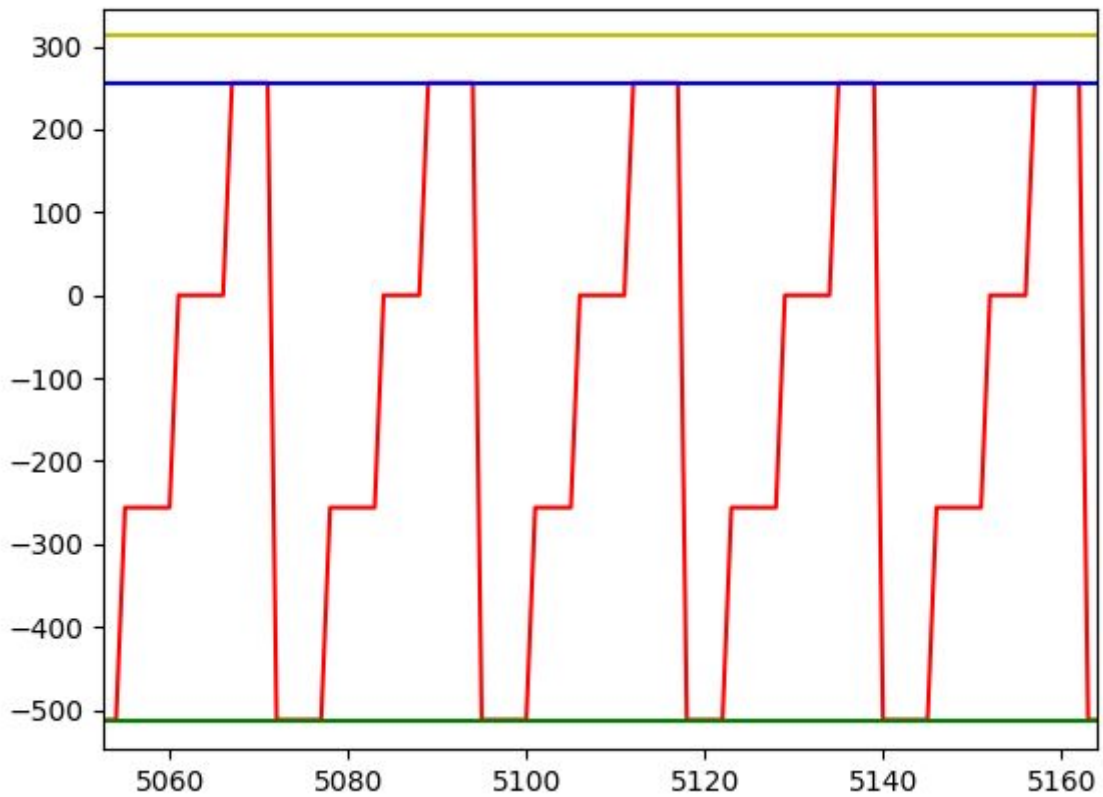
histograma (señal error 4 bits)





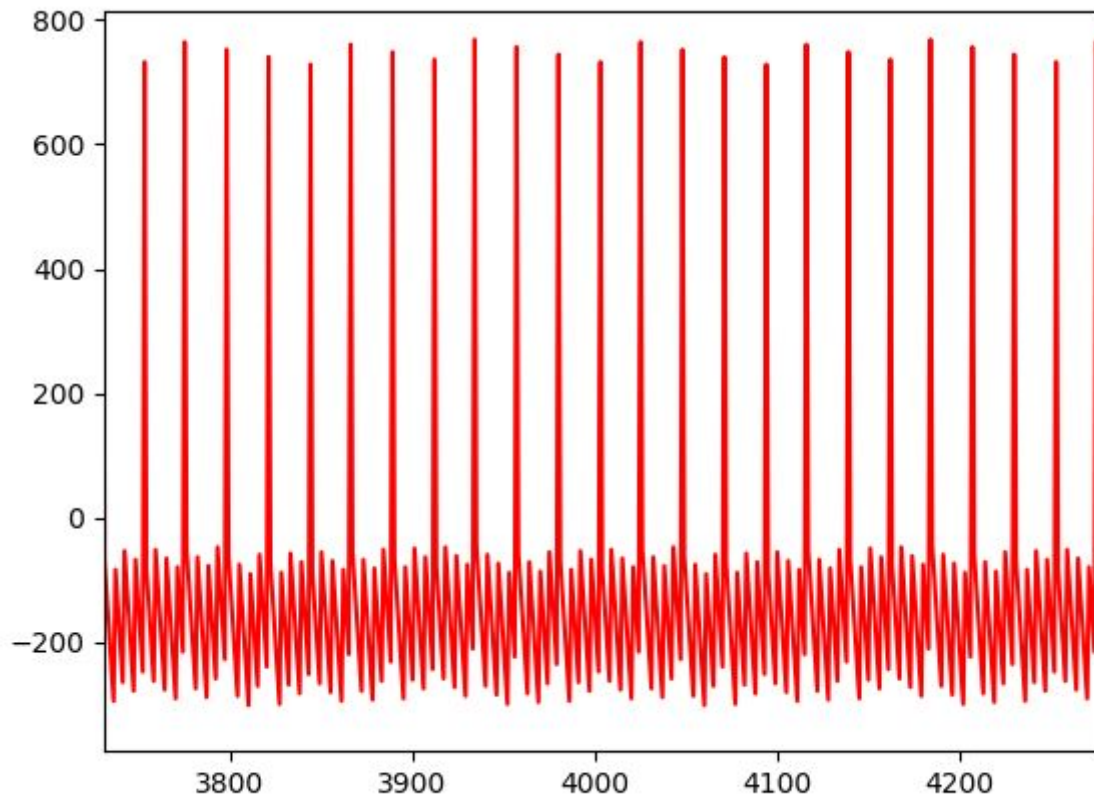
Señal adquirida (ADC 2 bits)

- valor máximo (línea azul): 256
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 314.2286573863164

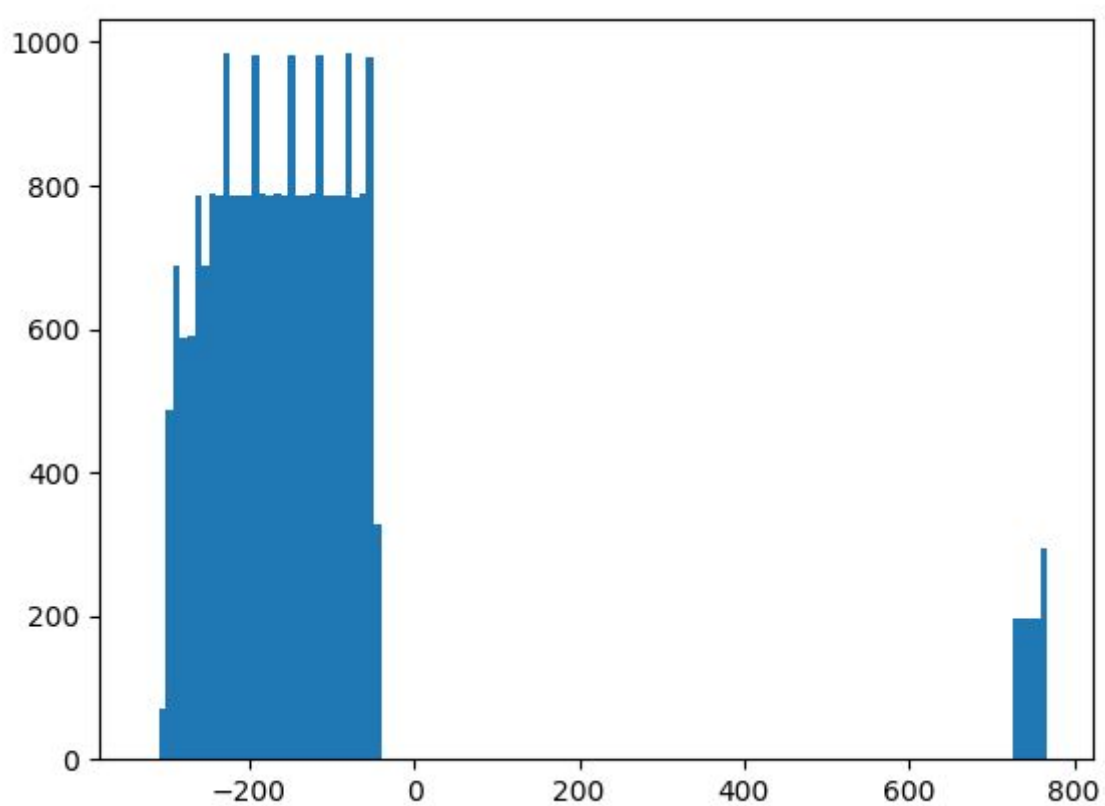




Señal error (2 bits)



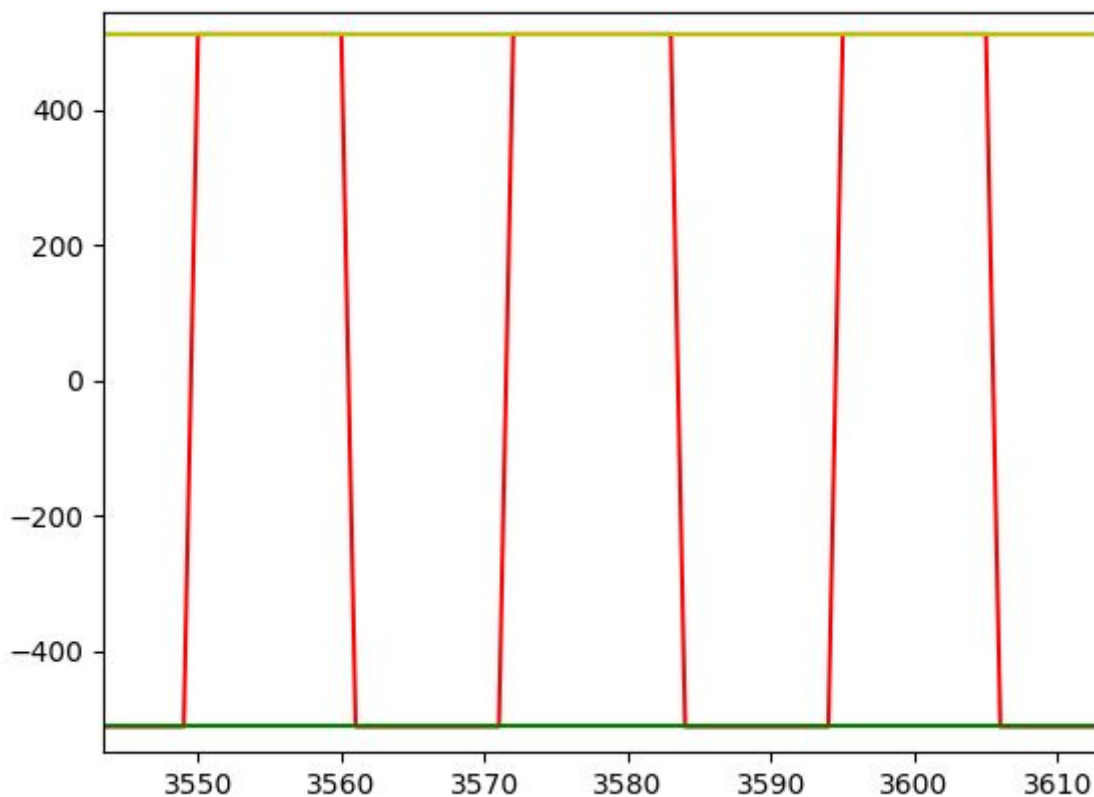
histograma (señal error 2 bits)



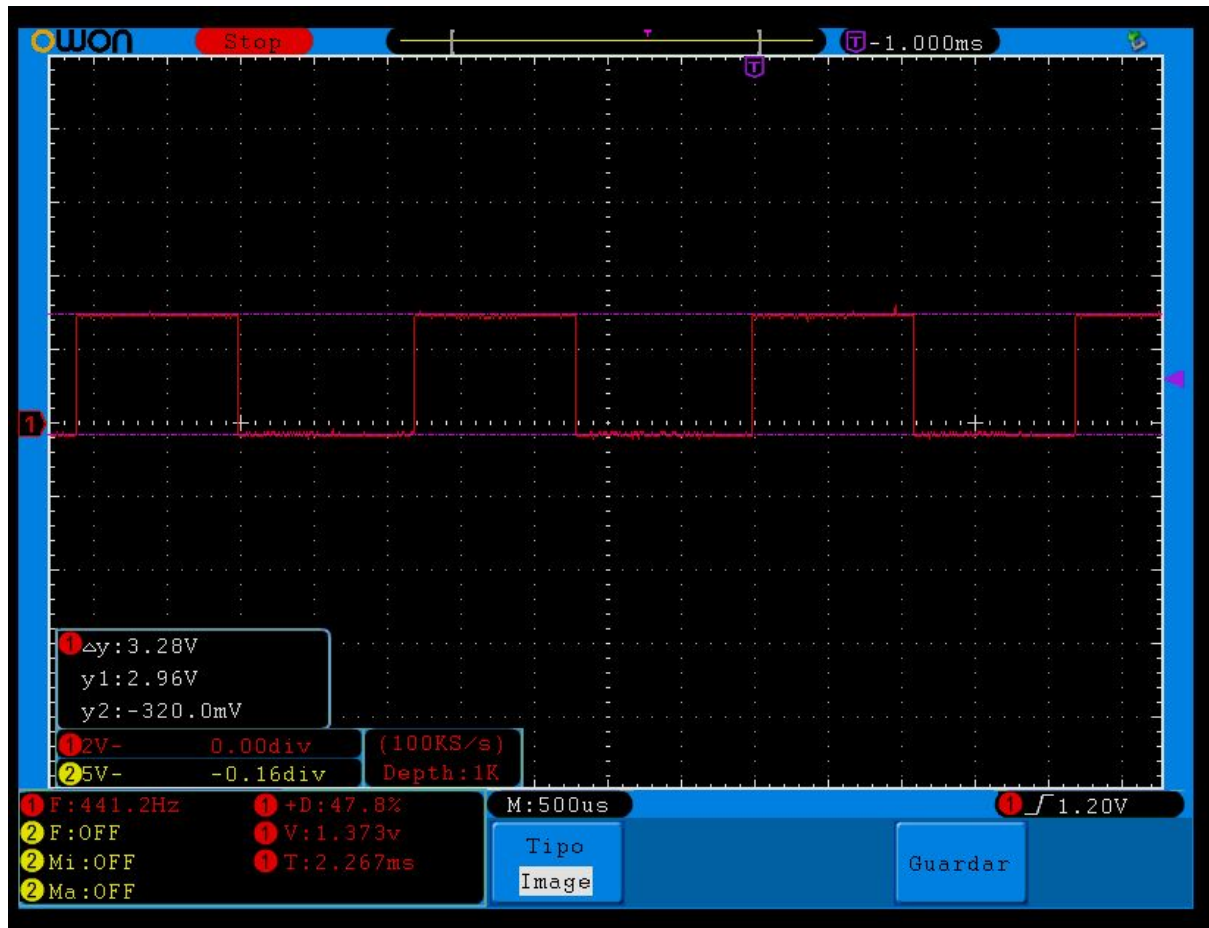
5.2.b Señal Cuadrada.

Señal cuadrada original, generada por el DAC de la EDU-CIAA (se elige generarla de esta manera para facilitar el cálculo de la señal ruido)

- valor máximo (línea azul): 512
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 512.0

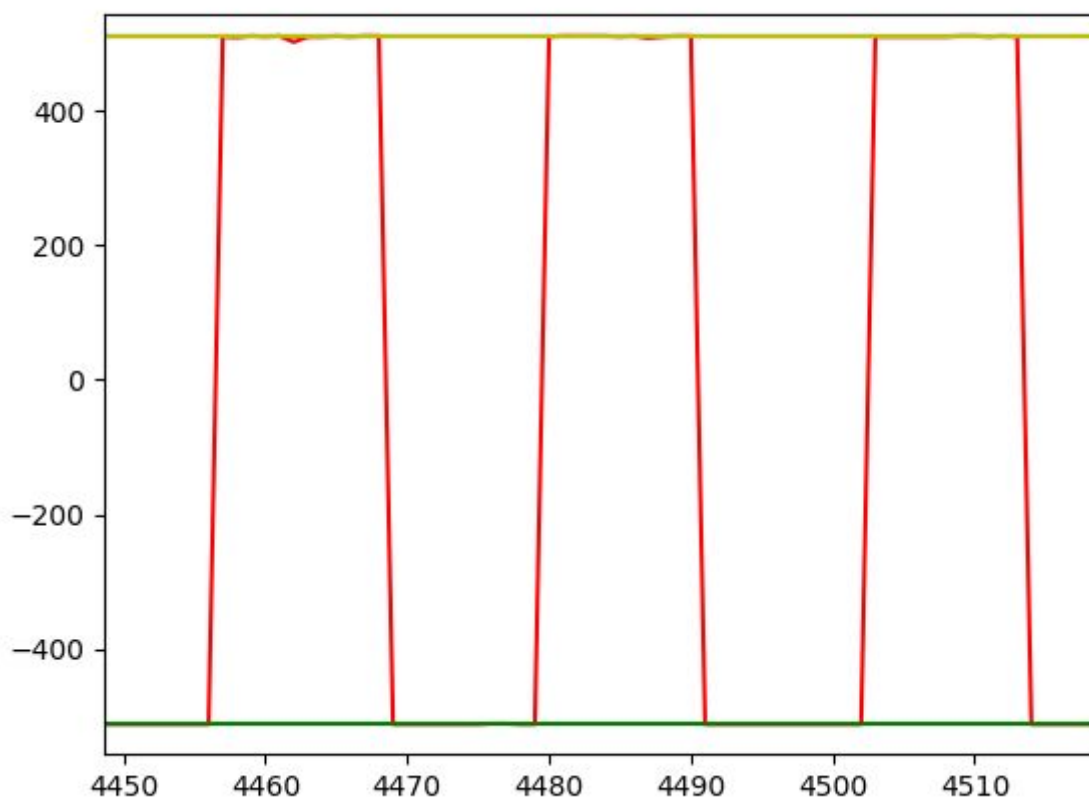


Señal observada en Osciloscopio



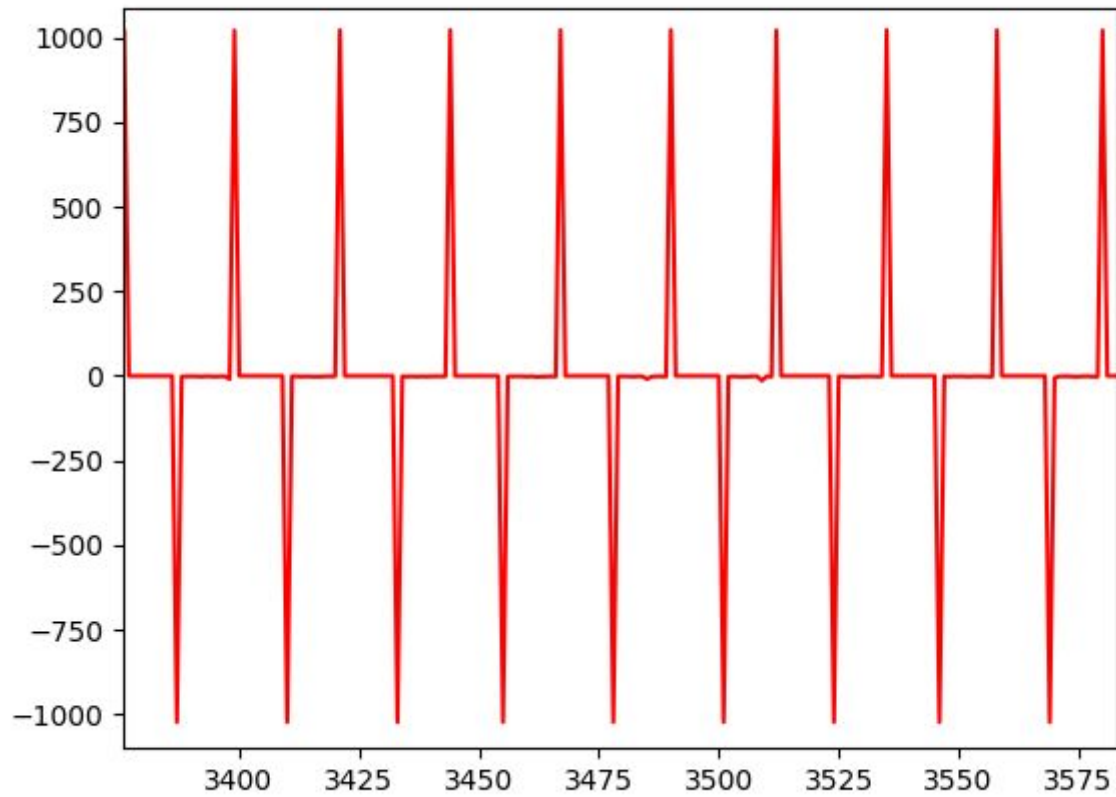
Señal adquirida (ADC 10 bits)

- valor máximo (línea azul): 511
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 511.1064695375368

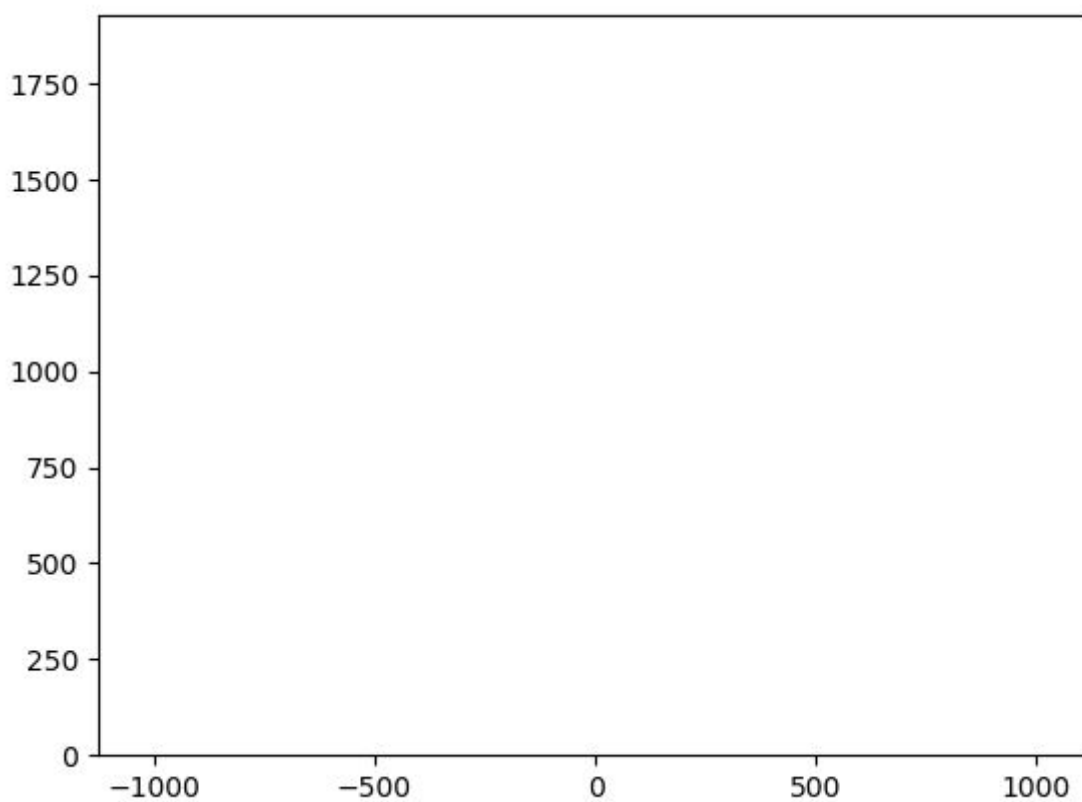




Señal error (10 bits)

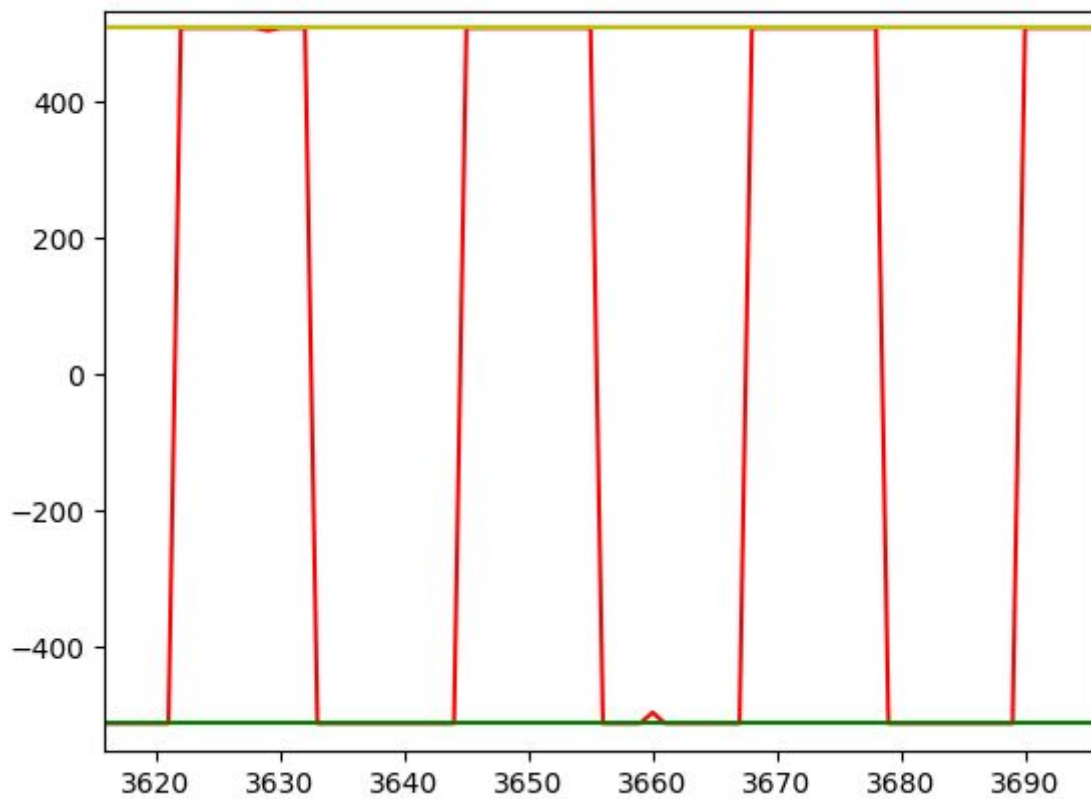


histograma (señal error 10 bits)

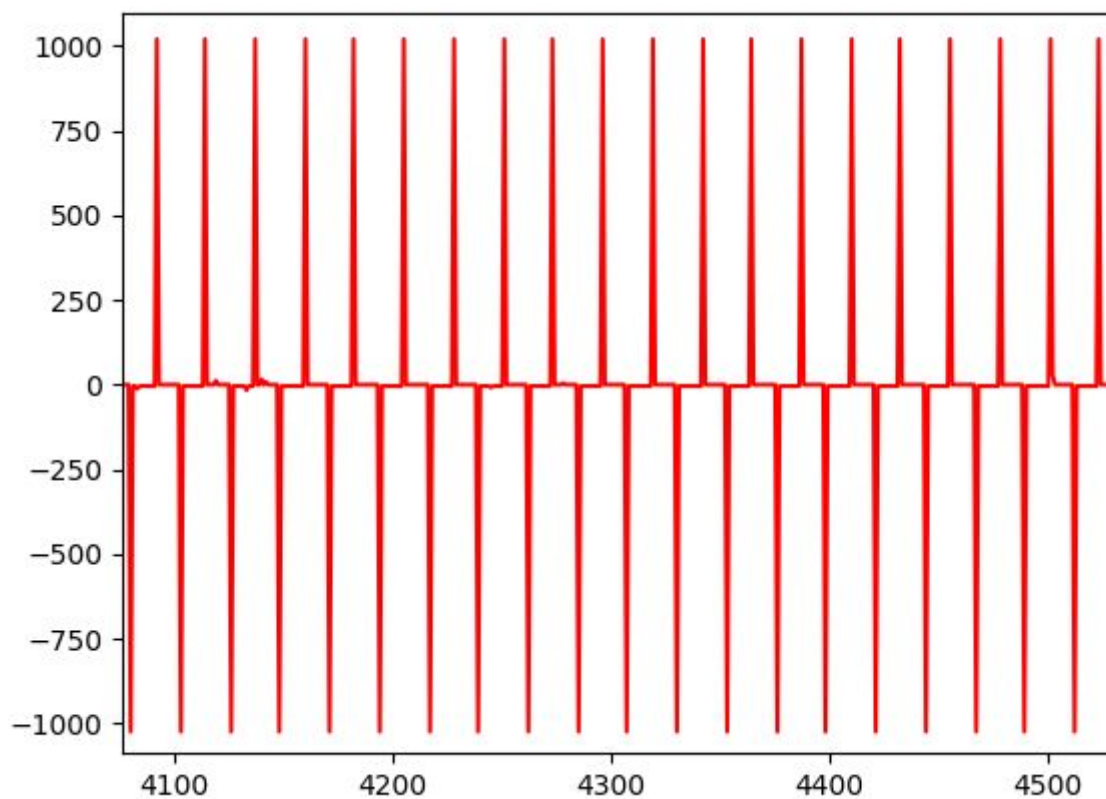


Señal adquirida (ADC 8 bits)

- valor máximo (línea azul): 508
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 509.87514580975227

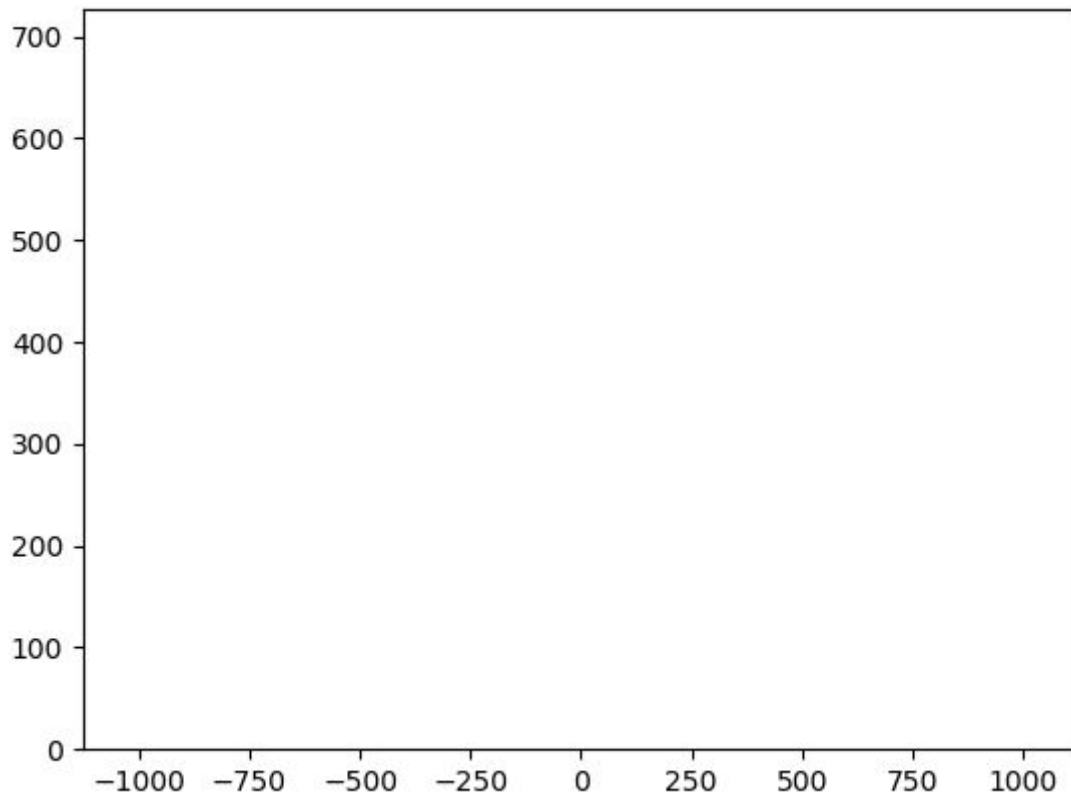


Señal error (8 bits)



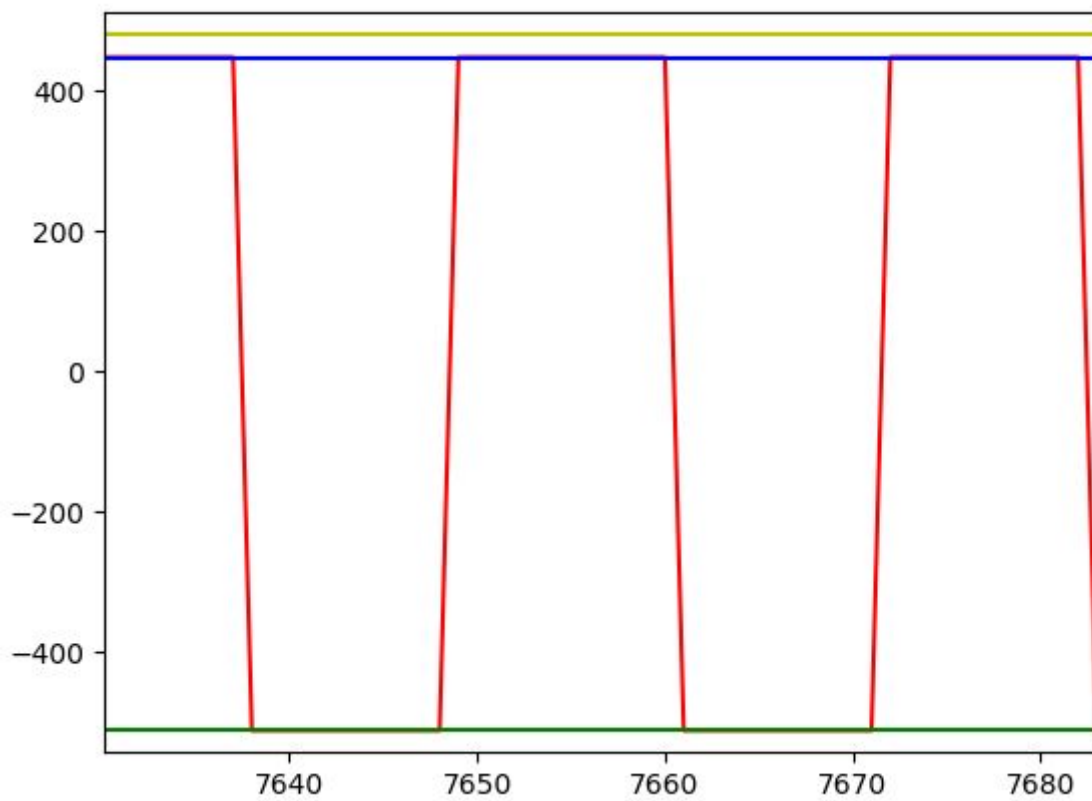


histograma (señal error 8 bits)

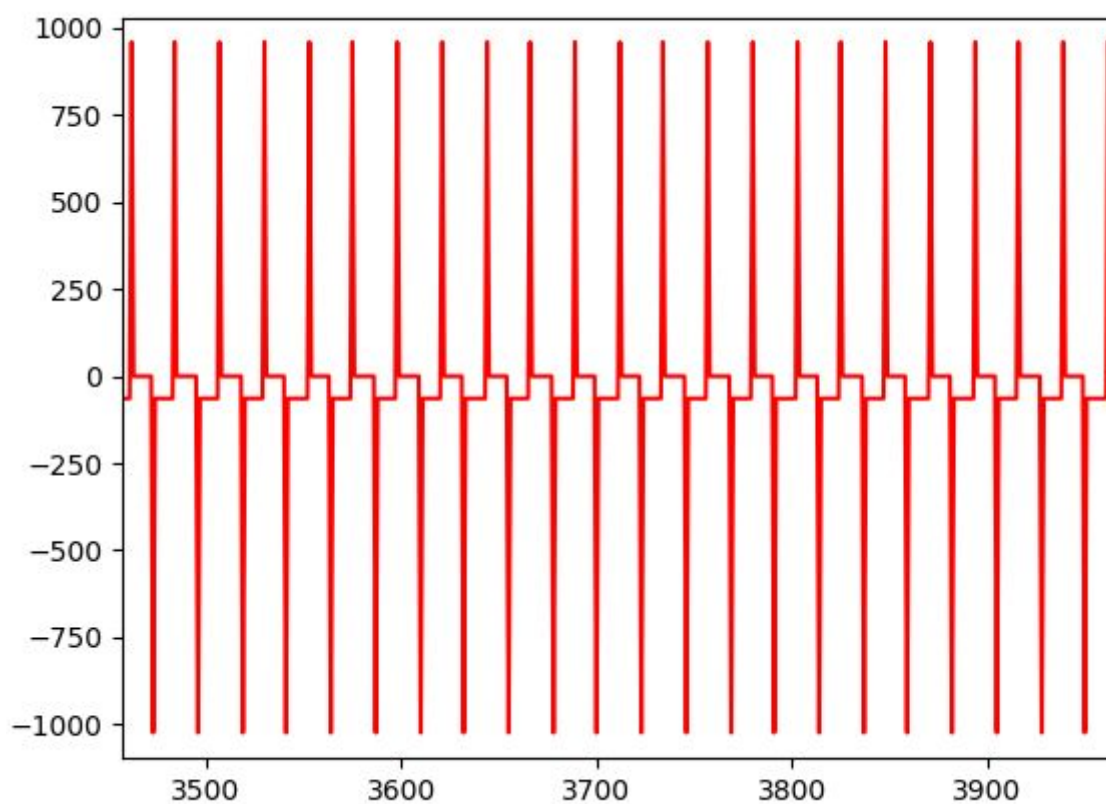


Señal adquirida (ADC 4 bits)

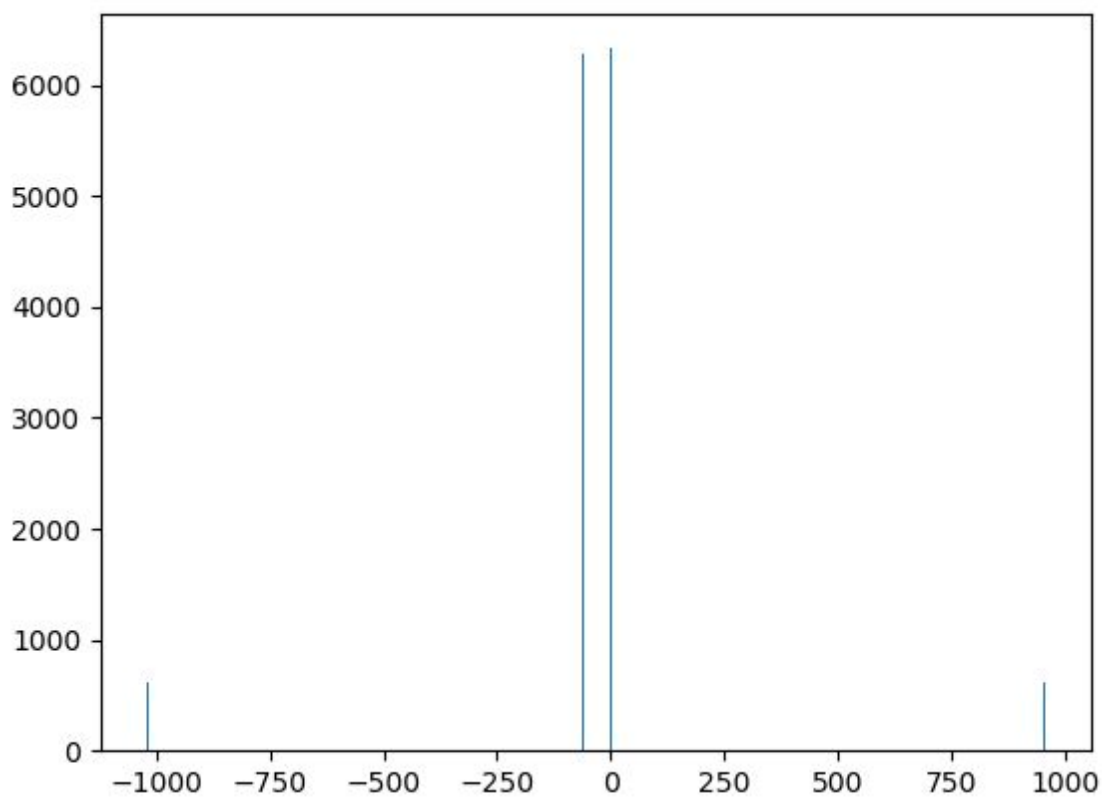
- valor máximo (línea azul): 448
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 481.21438422192824



Señal error (4 bits)

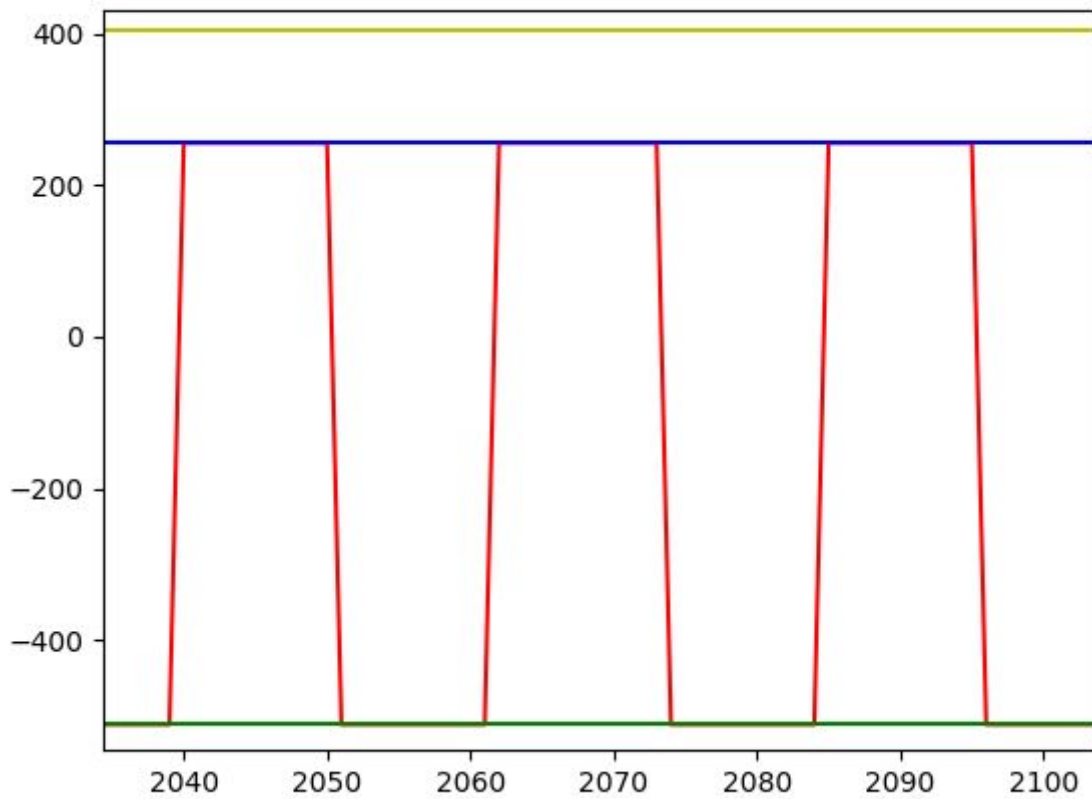


histograma (señal error 4 bits)

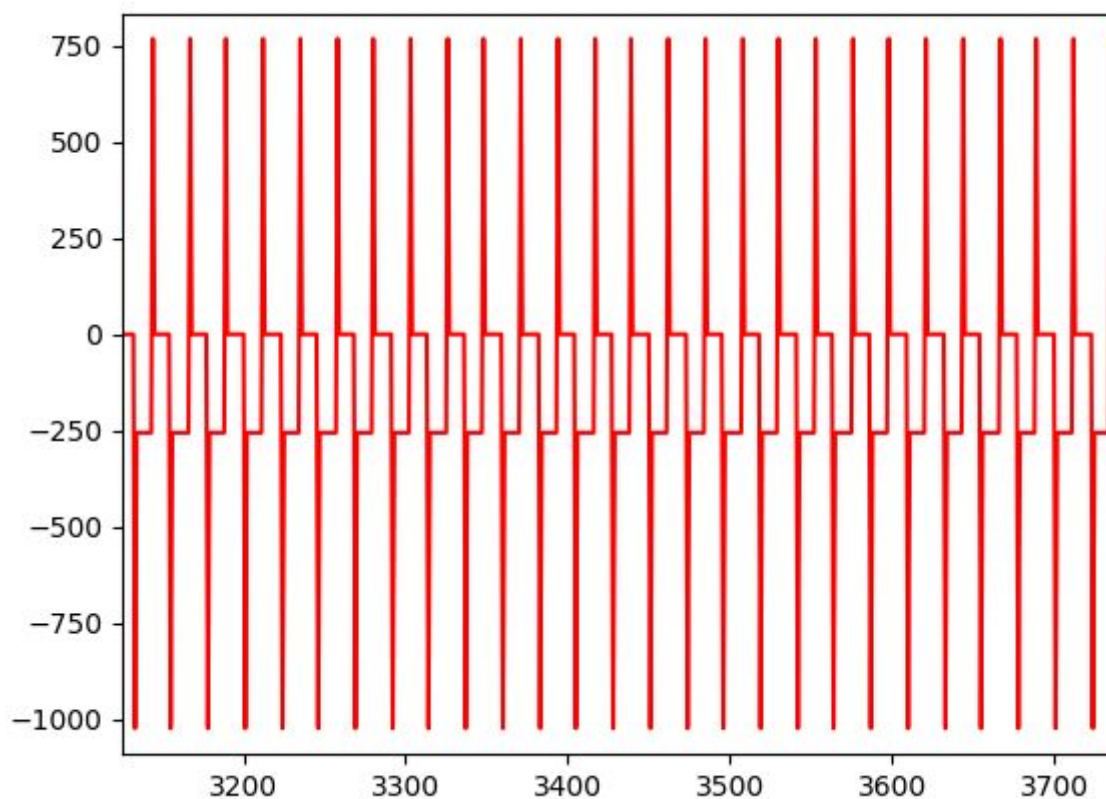


Señal adquirida (ADC 2 bits)

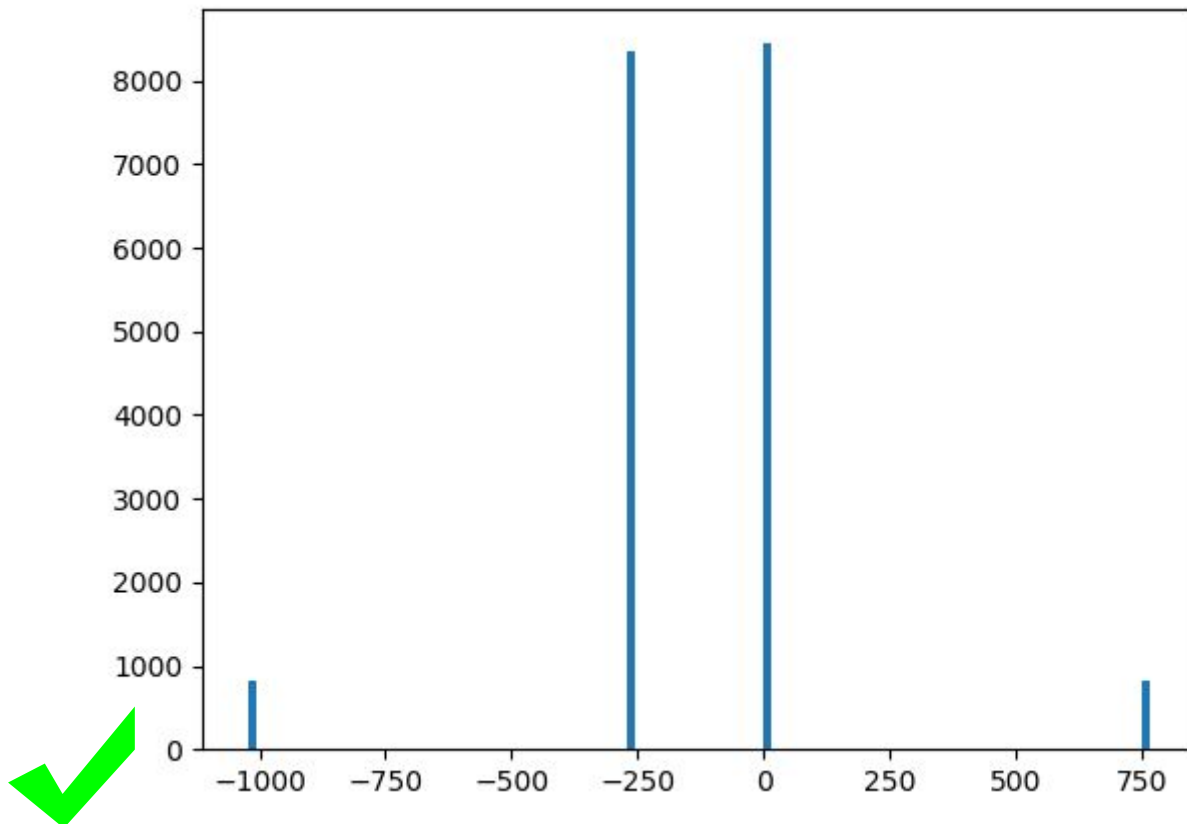
- valor máximo (línea azul): 256
- valor mínimo (línea verde): -512
- valor RMS (línea amarilla): 405.28069833689887



Señal error (2 bits)



histograma (señal error 2 bits)



Buen trabajo en este punto, se nota que le dedicastes tiempo. Lo unico a destacar es el tema del error, que en realidad cuando cuantizas por ej. con 4 bits y tenes un spam de 1024 valores del adc, el error entre la señal generada y la adquirida nunca podria ser mayor a $1024/2^4=64$

Es decir que en el histograma de error de 4bits, el eje x deberia ir de 0 a 64 o de -32 a 31 es decir de $-LSB/2$ a $+LSB/2$ o como esta en la clase podrias normalizarlo de -0.5 a 0.5 sabiendo que se trata de 0.5 LSB.

En cualquier caso el concepto es que no podrias tener en la señal error (signal-adc) algo que valga 100 o 250 como se ve en los histogramas

Puede ser de gran utilidad aca graficar la señal que generas con el DAC y la que adquiris con el ADC en el mismo grafico para verificar que todo este funcionando ok. Es decir vos generas con el DAC el numero digamos 70 y en 4 bits el ADC despues de shiftear y eliminar los bits menos significativos retorna 63, entonces mandas por el puerto serie el 70 y el 63 y los graficas juntos, entonces se ven las 2 curvas, una mas 'redondita' y otra mas cuadriculada y en ningun caso la cuadriculada podria separarse en mas de 64 puntos de la redondita. Sino pasa eso, podria haber algun error de sincronizacion o algo en el codigo a revisar. No hace falta que lo hagas de nuevo, solo que le pegues una mirada a la clase 1 a la parte del error y asimiles la idea

6. Número Q

1. Explique brevemente algunas de las diferencias entre la representación flotante de simple precisión (32b) y el sistema de punto fijo Qn.m

Diferencias:

a - rango

flotante simple precisión(32b): Rango dinámico $\pm 3,4e1038$, $\pm 1,2e10-38$

Qn.m: rango entre $[-2^{(m-1)}, 2^{(m-1)} - 2^{-n}]$ para Qn.m con signo y
rango entre $[0, 2^m - 2^{-n}]$ para Qn.m sin signo

b - resolución

flotante simple precisión(32b): variable

Qn.m: la resolución es fija en -2



2. Escriba los bits de los siguientes números decimales (o el más cercano) en float, Q1.15, Q2.14

- 0.5
- -0.5
- -1.25
- 0.001
- -2.001
- 204000000

número	float	S Q1.15	S Q2.14
0.5	00111111000000000000000000000000	0100000000000000	0010000000000000
-0.5	10111111000000000000000000000000	1100000000000000	1110000000000000
-1.25	10111111101000000000000000000000	1000000000000000	1011000000000000
0.001	00111010100000110001001001101111	0000000001000000	0000000000100000
-2.001	110000000000000000001000001100010	1000000000000000	1000000000000000
204000000	01001101010000101000110010110000	0111111111111111	0111111111111111

0x0021

0x0010

Entiendo que se te desplazo 1 bit ahí