

# Learning-ADC\_TO\_DAC

Ce document résume mes observations suite à la programmation de la carte de développement : « Nucleo-L476RG ».

L'objectif est d'enregistrer un signal en entrée de l'ADC et le rejouer sur le DAC.

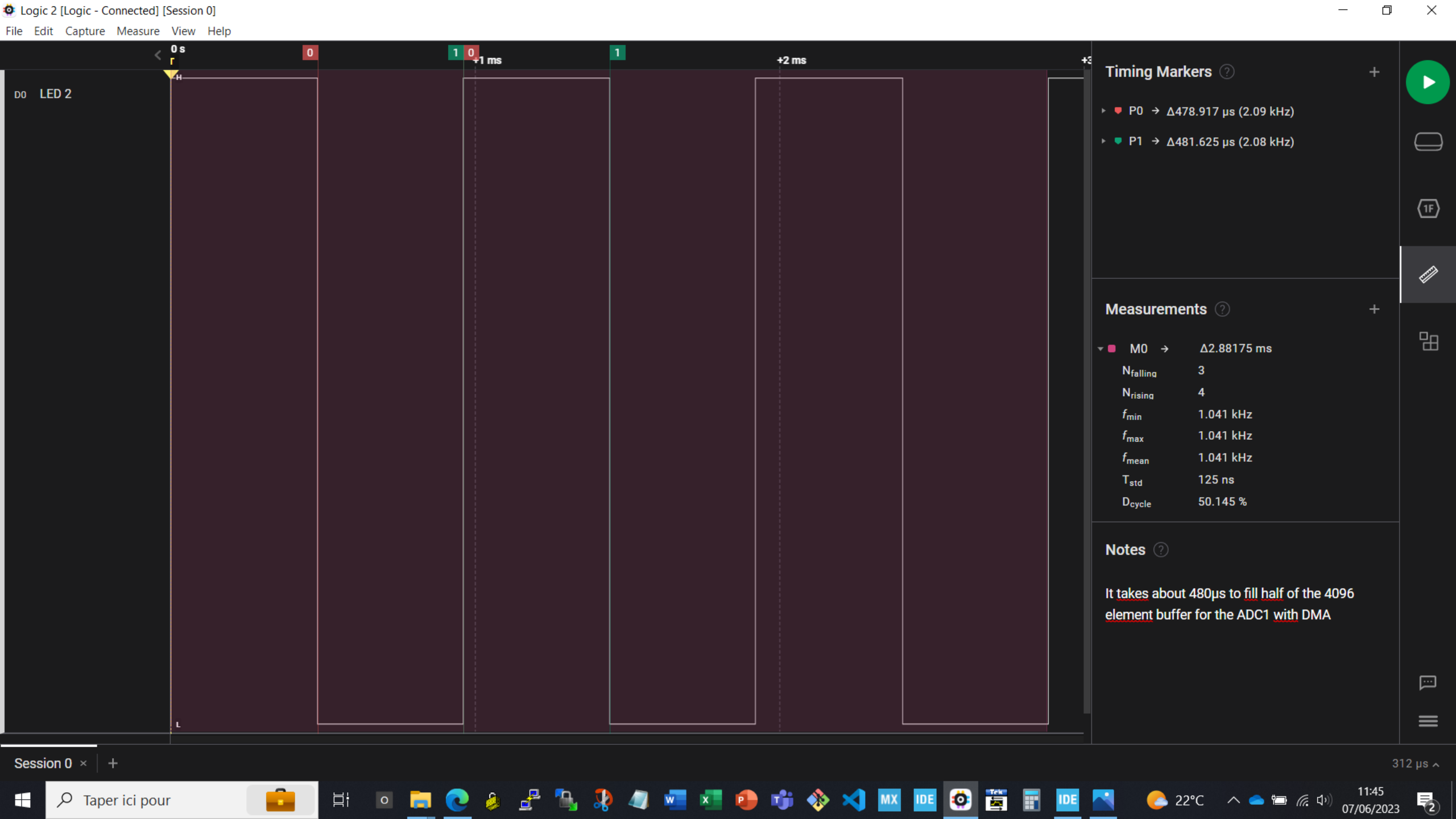
# Première étape l'ADC

Dans un premier temps, j'ai cherché à comprendre le fonctionnement de l'ADC pour connaître sa fréquence d'acquisition.

Ensuite, Il faut trouver le bon compromis entre la taille du buffer et la fréquence d'échantillonnage.

J'ai pu faire des enregistrements avec l'ADC de 4 façons :

- 1) En « Polling »
- 2) En lançant la DMA de l'ADC depuis ma boucle « While »
- 3) En utilisant le mode circulaire de la DMA
- 4) En synchronisant mes acquisitions grâce à un Timer toujours en DMA mode circulaire



Logic 2 [Logic - Connected] [Session 0]

File Edit Capture Measure View Help

<0 s

0

+1 ms

10

+2 ms

1

+3

D0 LED 2

Timing Markers ?

P0 → Δ936.5 μs (1.07 kHz)

P1 → Δ939.583 μs (1.06 kHz)

Measurements ?

M0 → Δ1.875958 ms

Nfalling1

Nrising2

fminN/A

fmax533.061 Hz

fmean533.061 Hz

TstdN/A

Dcycle50.079 %

Notes ?

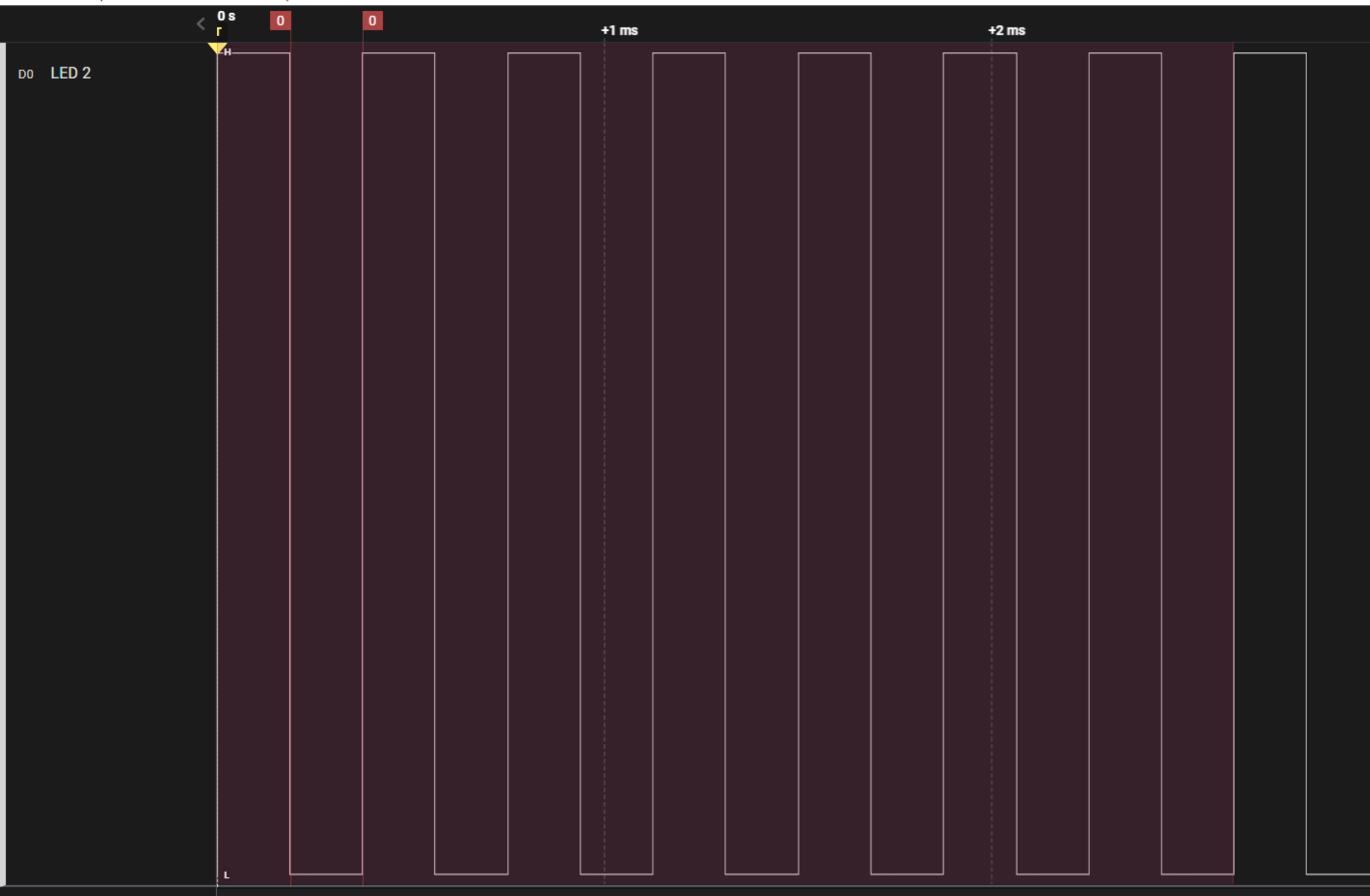
It takes about 940μs to fill half of the 8000 element buffer for the ADC1 with DMA

Session 0 × +

312 μs ^

Taper ici pour

26°C 13:50 07/06/2023



## Timing Markers ?

P0 →  $\Delta 187.583 \mu s$  (5.33 kHz)

## Measurements ?

M0	→	$\Delta 2.626 ms$
$N_{falling}$		7
$N_{rising}$		8
$f_{min}$		2.661 kHz
$f_{max}$		2.669 kHz
$f_{mean}$		2.666 kHz
$T_{std}$		377.964 ns
$D_{cycle}$		49.99 %

## Notes ?

It takes about 187,5 μs to fill of the 800 element buffer for the ADC1 with DMA2

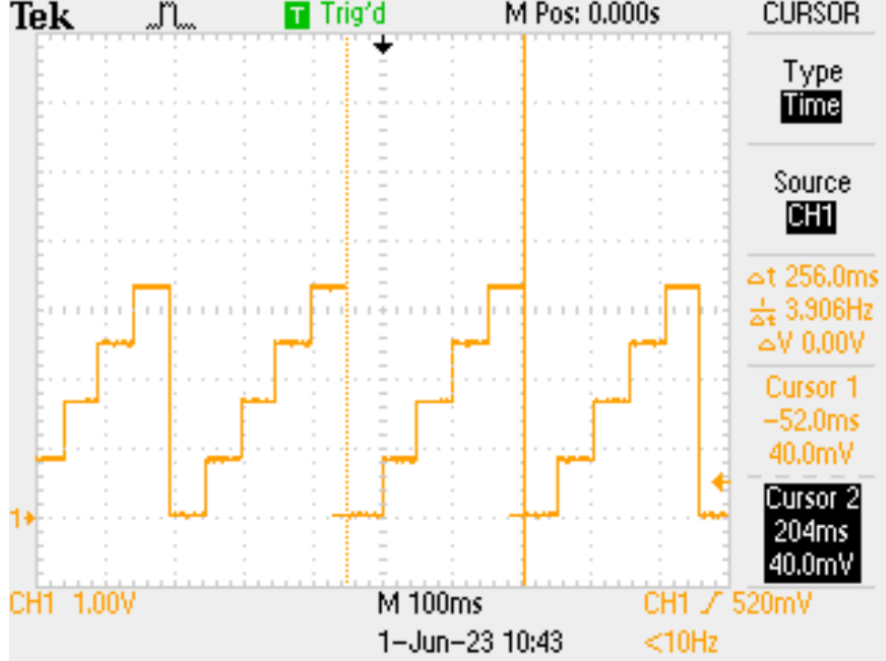
# Deuxième étape le DAC

Le DAC doit jouer le buffer enregistrer pour l'ADC.

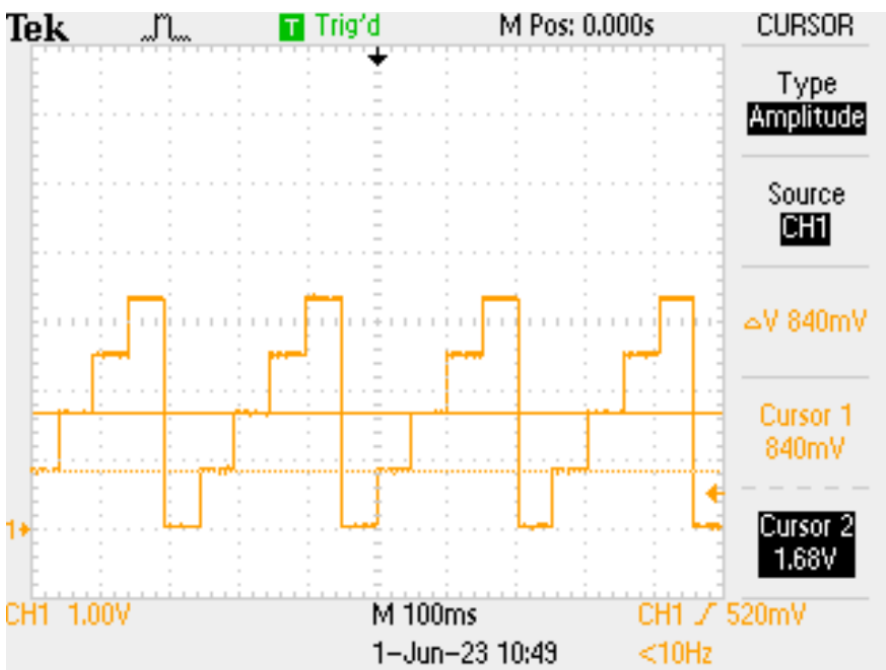
Pour cela, j'ai utilisé un Timer qui lance la conversion numérique du buffer vers la sortie du microcontrôleur.

J'ai fait différentes expérimentations :

- 1) Générer un escalier avec 5 niveaux en « polling »
- 2) Générer un sinus avec une LUT de différente taille de buffer en « polling »
- 3) Générer un sinus avec une DMA mode normal
- 4) Générer un sinus avec une DMA mode circulaire

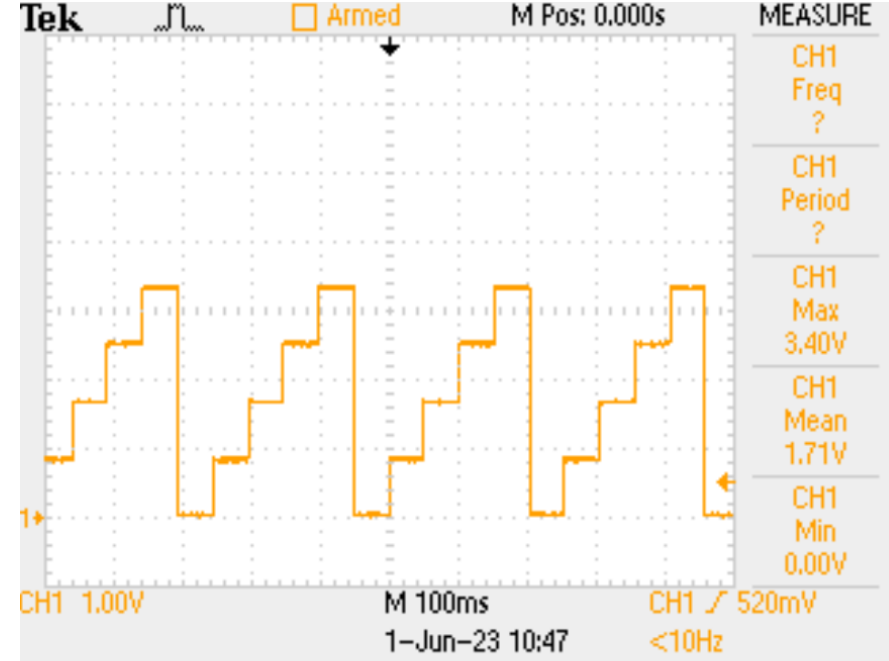


TDS 2024B - 10:35:46 01/06/2023

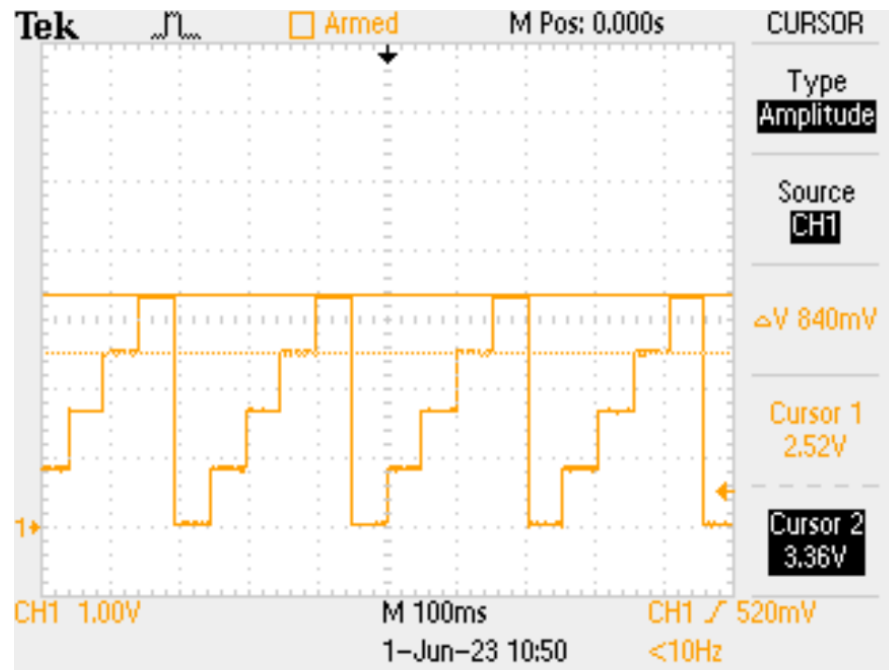


TDS 2024B - 10:41:44 01/06/2023

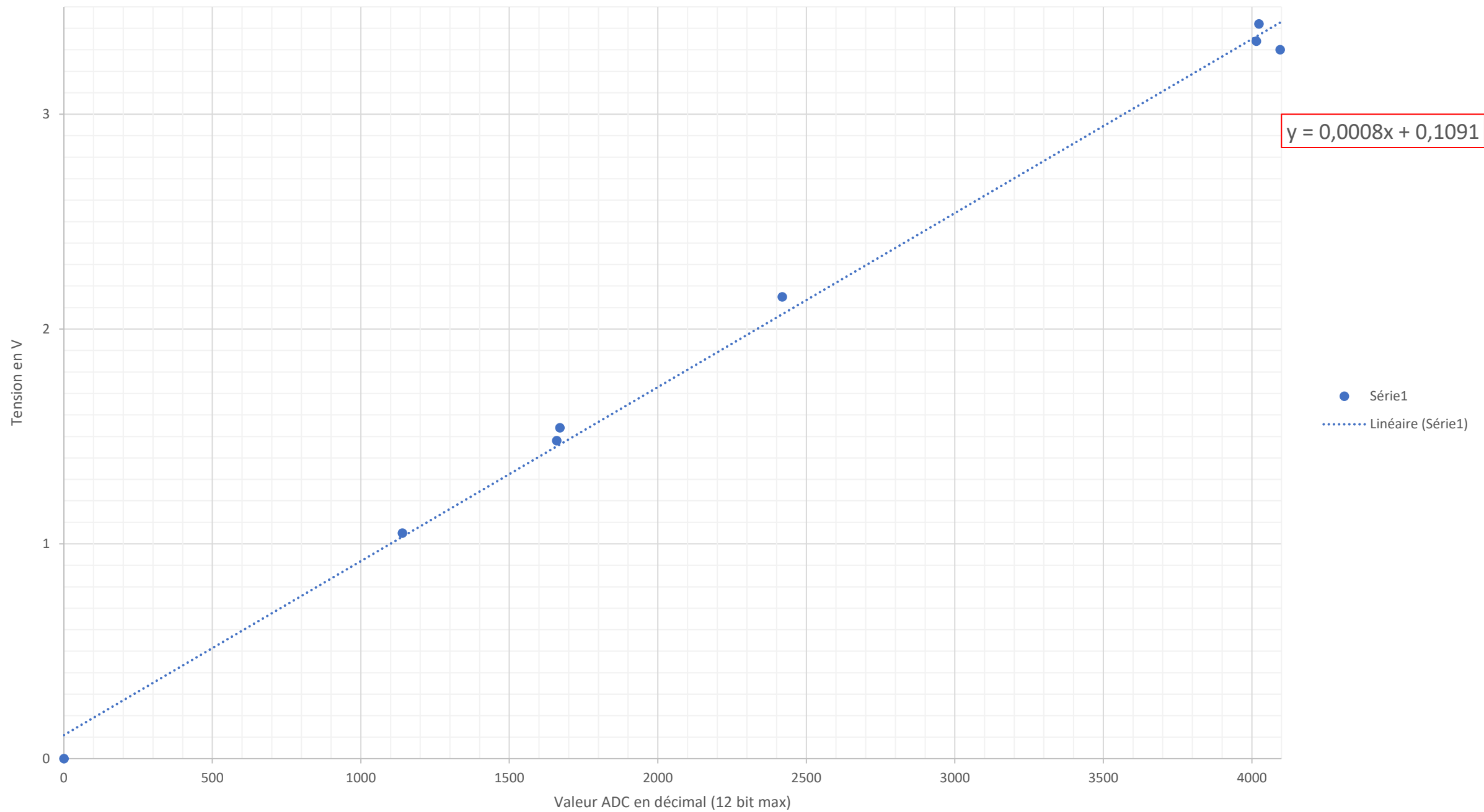
Génération d'un  
escalier à 5 niveaux



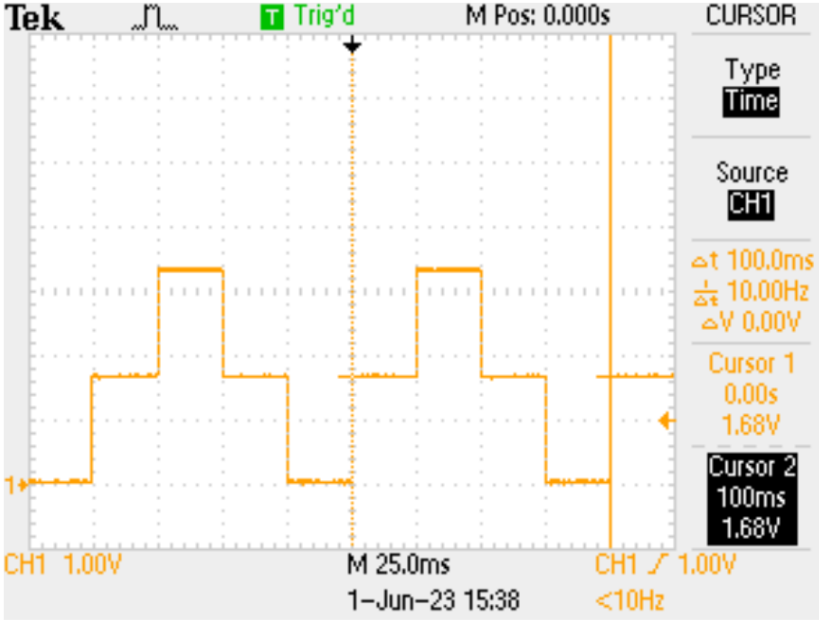
TDS 2024B - 10:39:05 01/06/2023



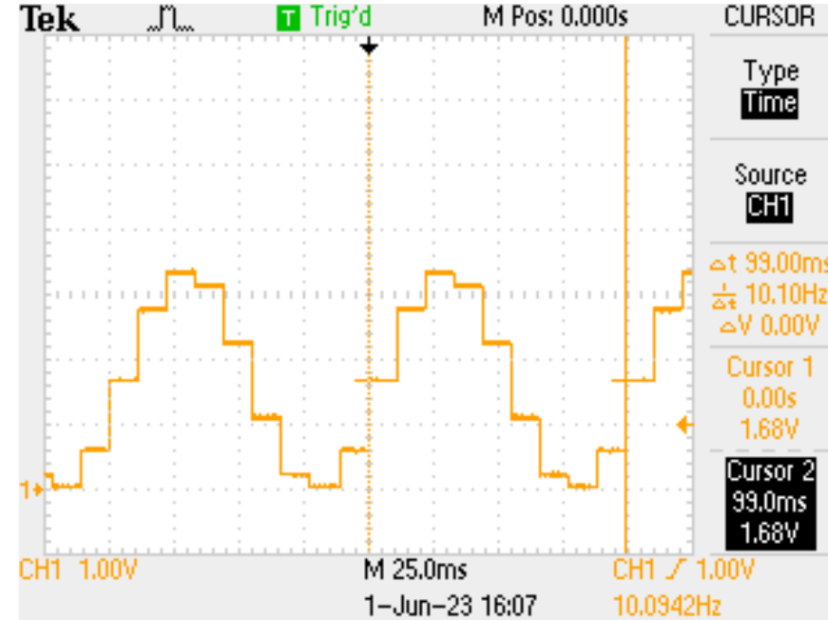
TDS 2024B - 10:42:41 01/06/2023



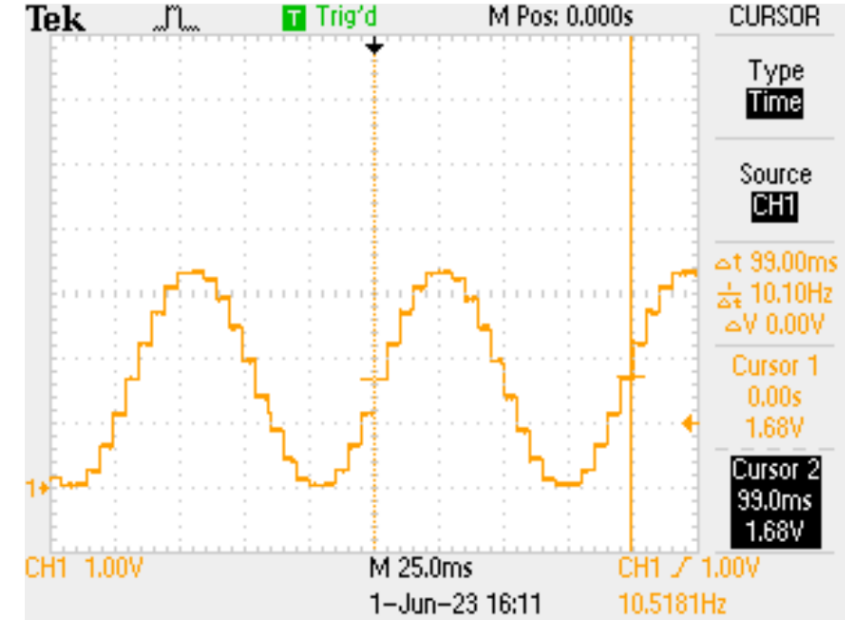




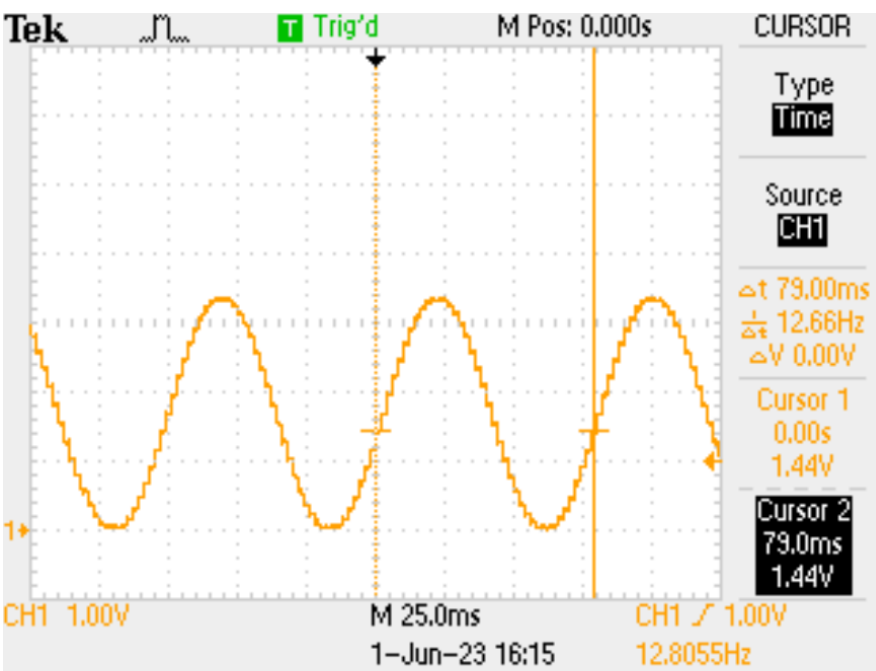
TDS 2024B - 15:30:41 01/06/2023



TDS 2024B - 15:59:21 01/06/2023

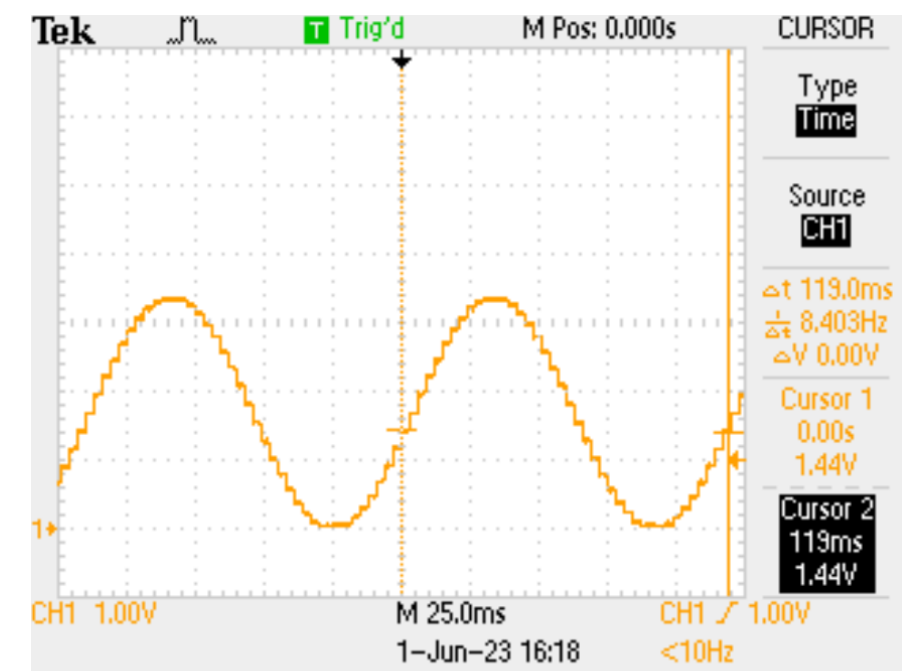


TDS 2024B - 16:03:50 01/06/2023

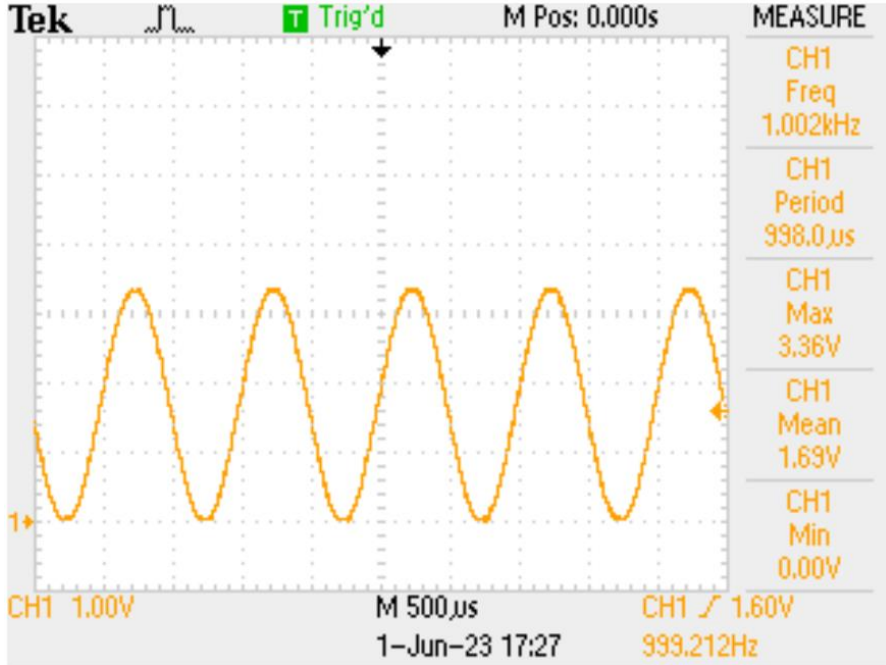


TDS 2024B - 16:07:55 01/06/2023

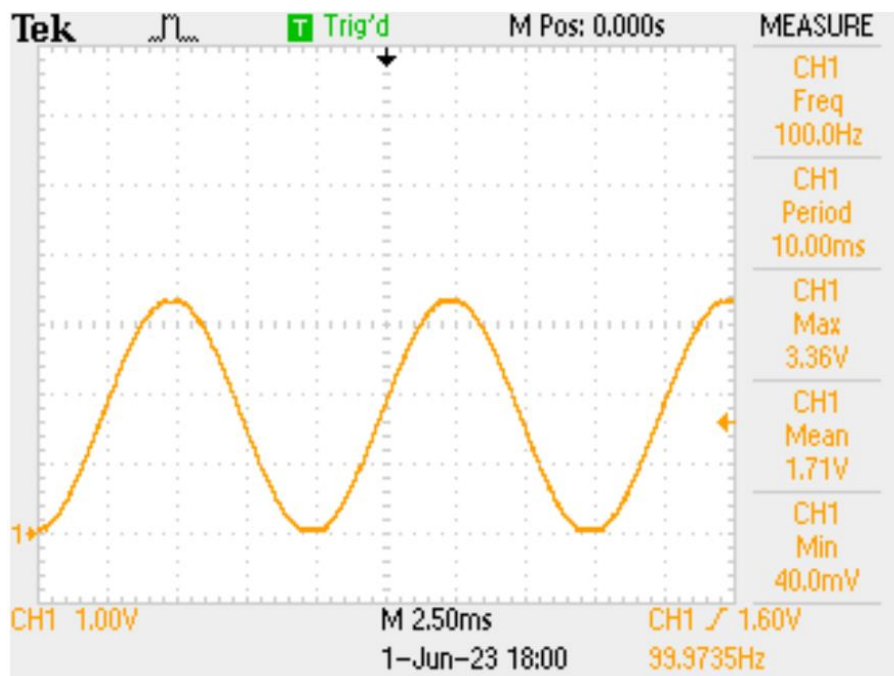
Génération d'un sinus à 10 Hz avec différent niveau d'échantillonnage



TDS 2024B - 16:09:59 01/06/2023

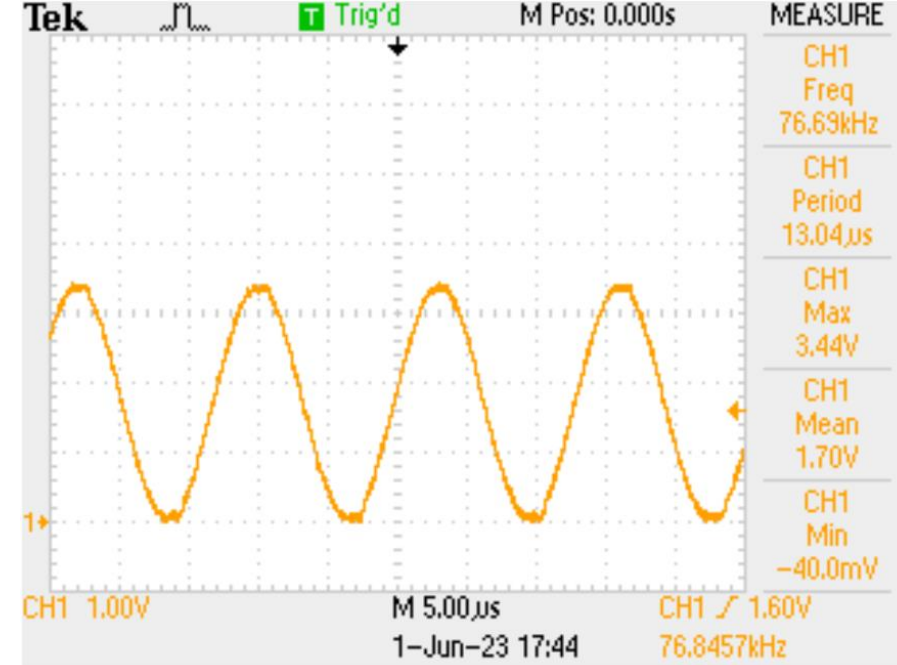


TDS 2024B - 17:19:14 01/06/2023

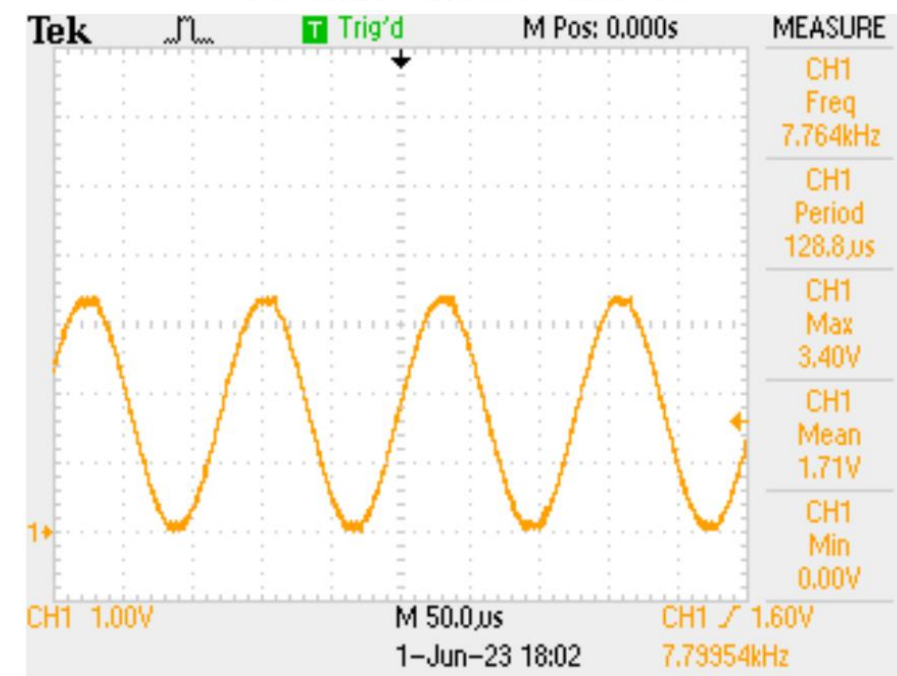


TDS 2024B - 17:52:36 01/06/2023

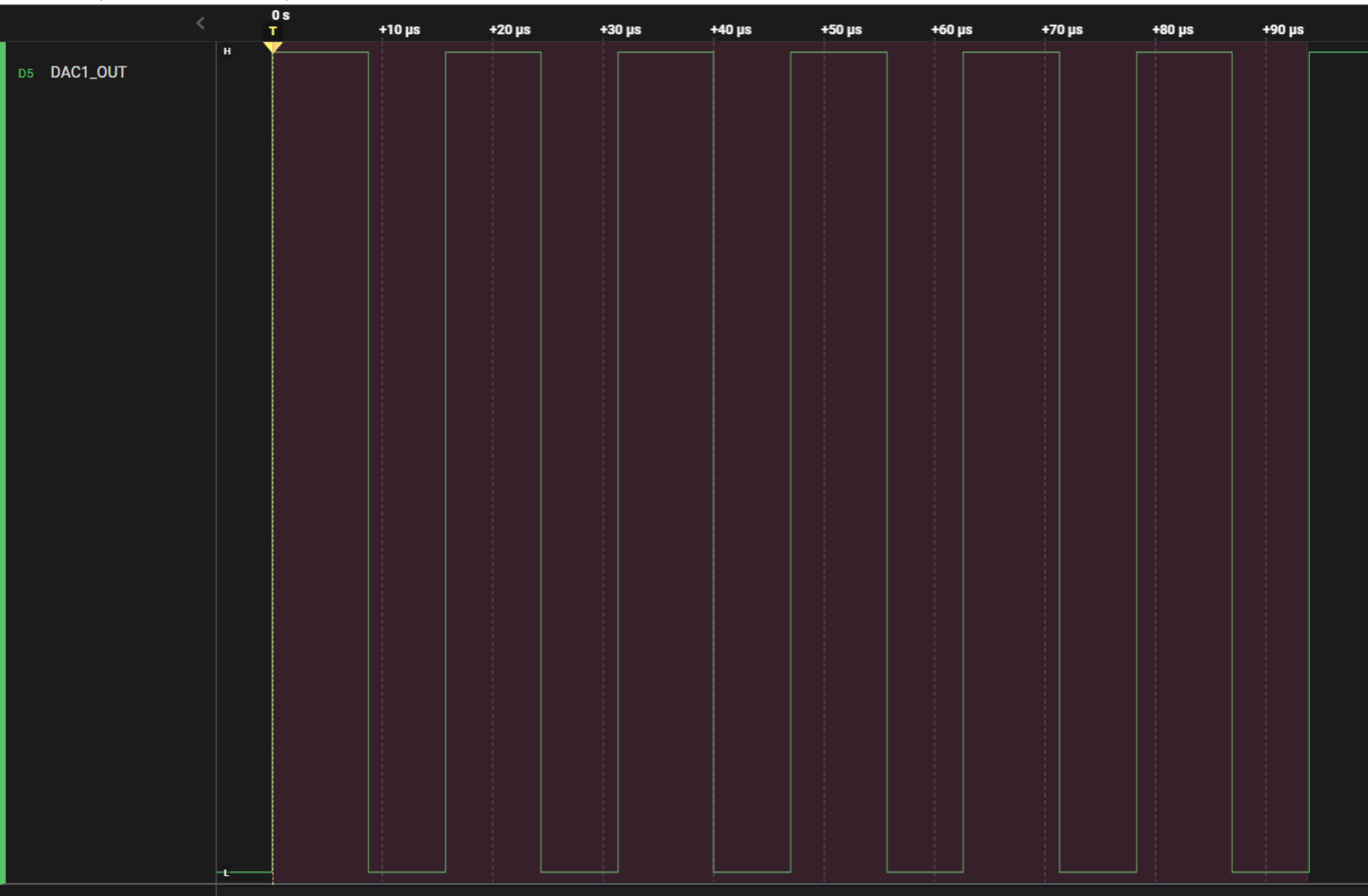
Génération d'un  
sinus à différente  
fréquence



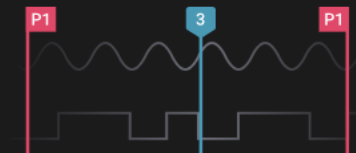
TDS 2024B - 17:36:39 01/06/2023



TDS 2024B - 17:54:34 01/06/2023



## Timing Markers ?



Click "+" to add Timing Markers

## Measurements ?

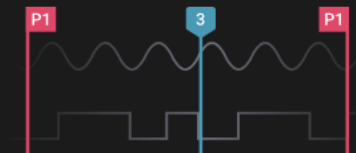
M0	→	Δ93.833 μs
N <sub>falling</sub>		6
N <sub>rising</sub>		7
f <sub>min</sub>		63.83 kHz
f <sub>max</sub>		64 kHz
f <sub>mean</sub>		63.943 kHz
T <sub>std</sub>		21.517 ns
D <sub>cycle</sub>		55.506 %

## Notes ?

Test DAC OUT sampling rate  
80 MHz / 625 = 128 kHz  
128 kHz / 2 = 64 kHz



## Timing Markers ?



Click "+" to add Timing Markers

## Measurements ?

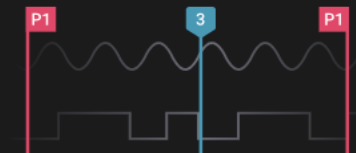
M0	→	Δ14 μs
N <sub>falling</sub>		7
N <sub>rising</sub>		8
f <sub>min</sub>		500 kHz
f <sub>max</sub>		500 kHz
f <sub>mean</sub>		500 kHz
T <sub>std</sub>		0 s
D <sub>cycle</sub>		77.083 %

## Notes ?

Test DAC OUT sampling rate  
80 MHz / 80 = 1 MHz  
1 MHz / 2 = 500 kHz



## Timing Markers ?



Click "+" to add Timing Markers

## Measurements ?

M0	→	Δ90.042 μs
N <sub>falling</sub>		9
N <sub>rising</sub>		10
f <sub>min</sub>		99.585 kHz
f <sub>max</sub>		100 kHz
f <sub>mean</sub>		99.954 kHz
T <sub>std</sub>		13.889 ns
D <sub>cycle</sub>		53.957 %

## Notes ?

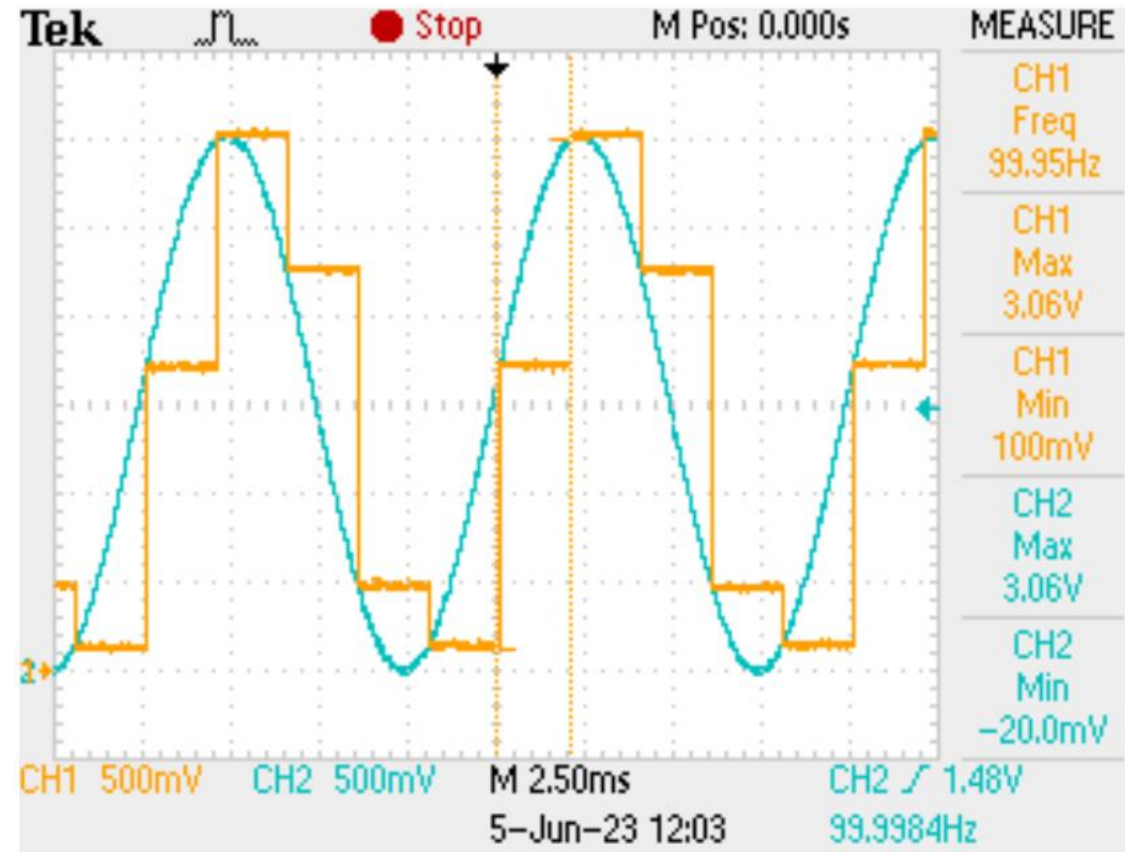
Test DAC OUT sampling rate  
 $80 \text{ MHz} / 400 = 200 \text{ kHz}$   
 $200 \text{ kHz} / 2 = 100 \text{ kHz}$

## Troisième étape L'ADC vers le DAC en mode « polling »

La méthode, la plus simple, est de transférer la conversion de l'ADC directement vers le DAC.

Cette méthode fonctionne mais on est limité en fréquence car on subit le temps de conversion de l'ADC puis le temps de conversion du DAC.

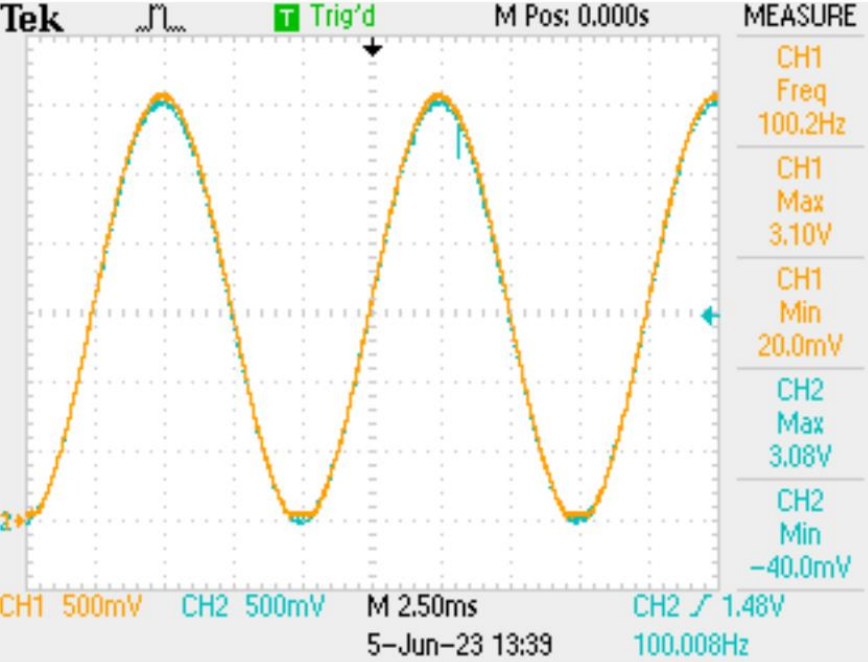
En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



TDS 2024B - 11:55:08 05/06/2023

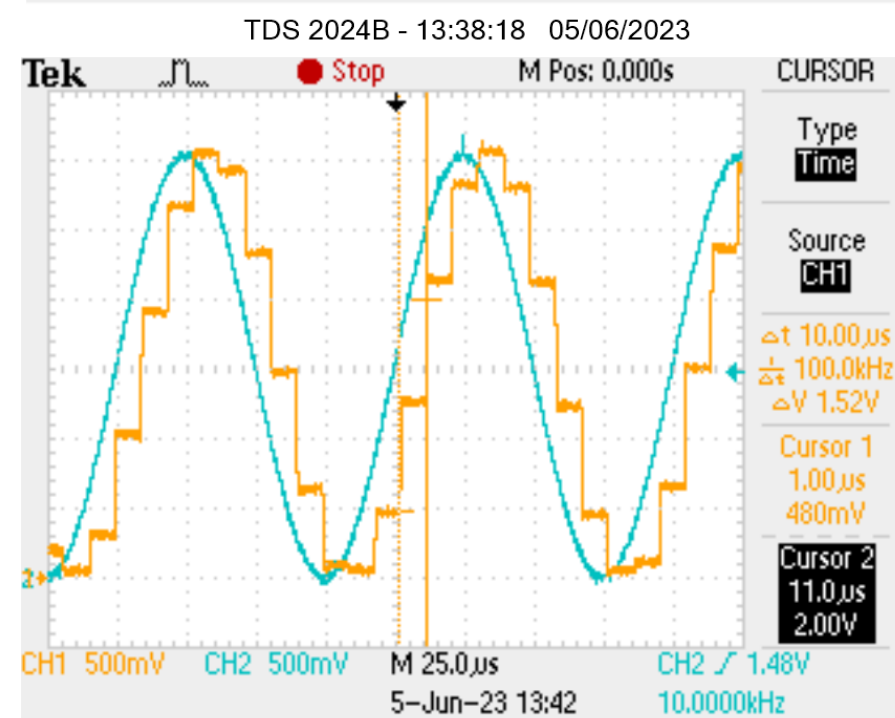
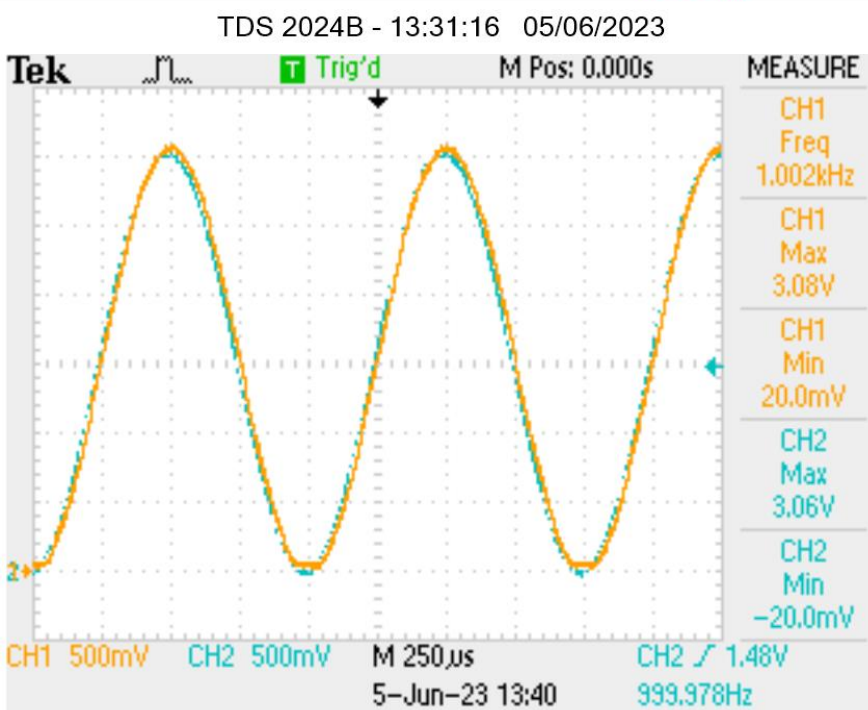
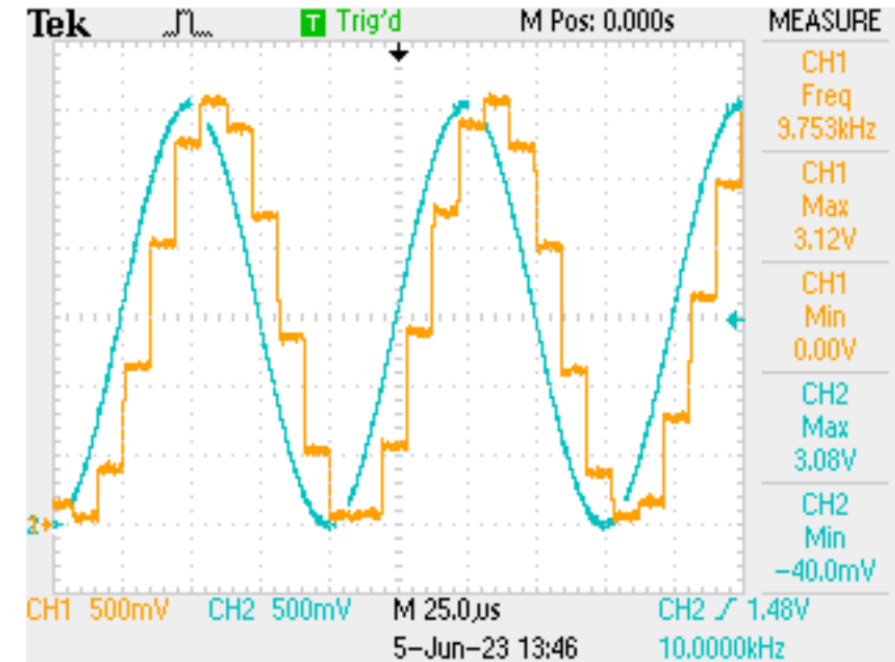
Ceci est un premier test, j'enregistre via l'ADC le signal et j'écris en sortie du DAC toutes le 10 ms.





En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC

À 10 kHz, un crénelage se forme.  
On mesure 10  $\mu$ s.  
C'est le temps pour enregistrer  
le signal avec l'ADC puis mettre  
la valeur en sortie du DAC.



TDS 2024B - 13:31:16 05/06/2023

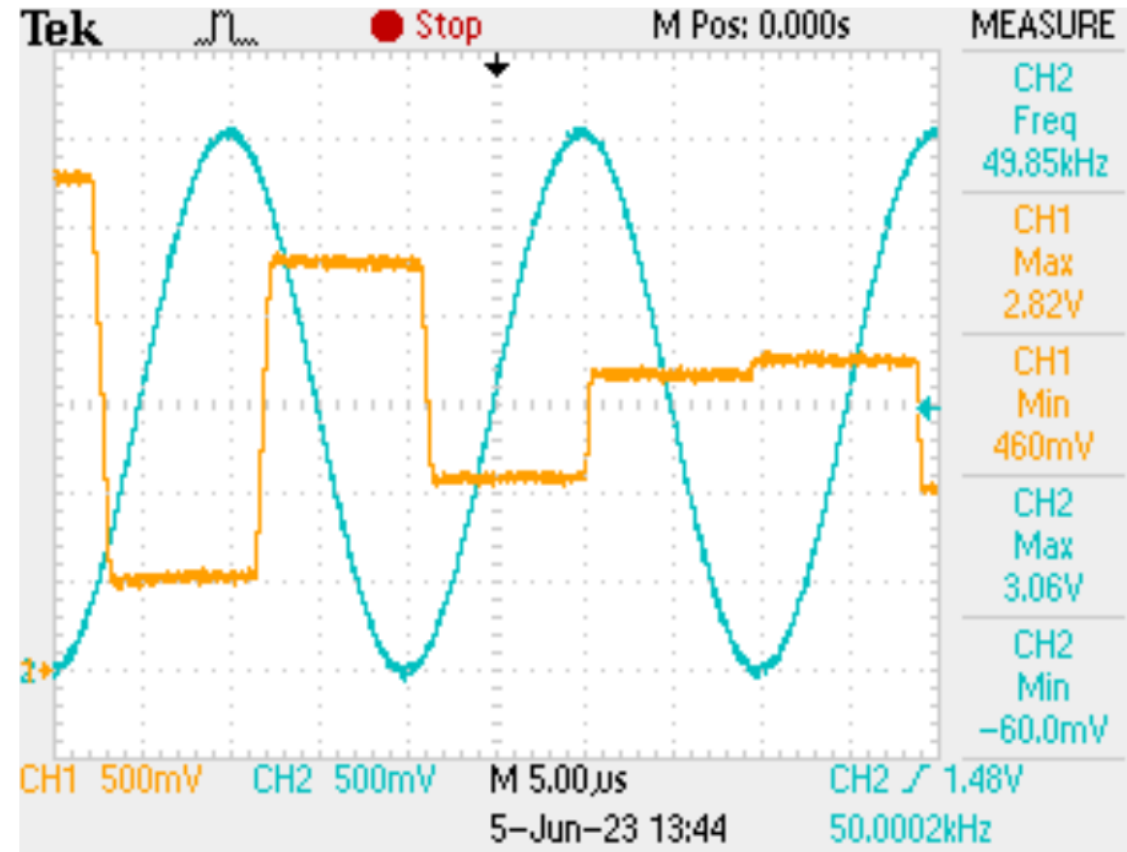
TDS 2024B - 13:38:18 05/06/2023

TDS 2024B - 13:32:23 05/06/2023

TDS 2024B - 13:34:36 05/06/2023



En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



TDS 2024B - 13:36:16 05/06/2023

En utilisant le polling, on se rend compte qu'à 50 kHz, on n'arrive plus à suivre le signal.

# Quatrième étape L'ADC vers le DAC avec DMA en utilisant un buffer ping-pong

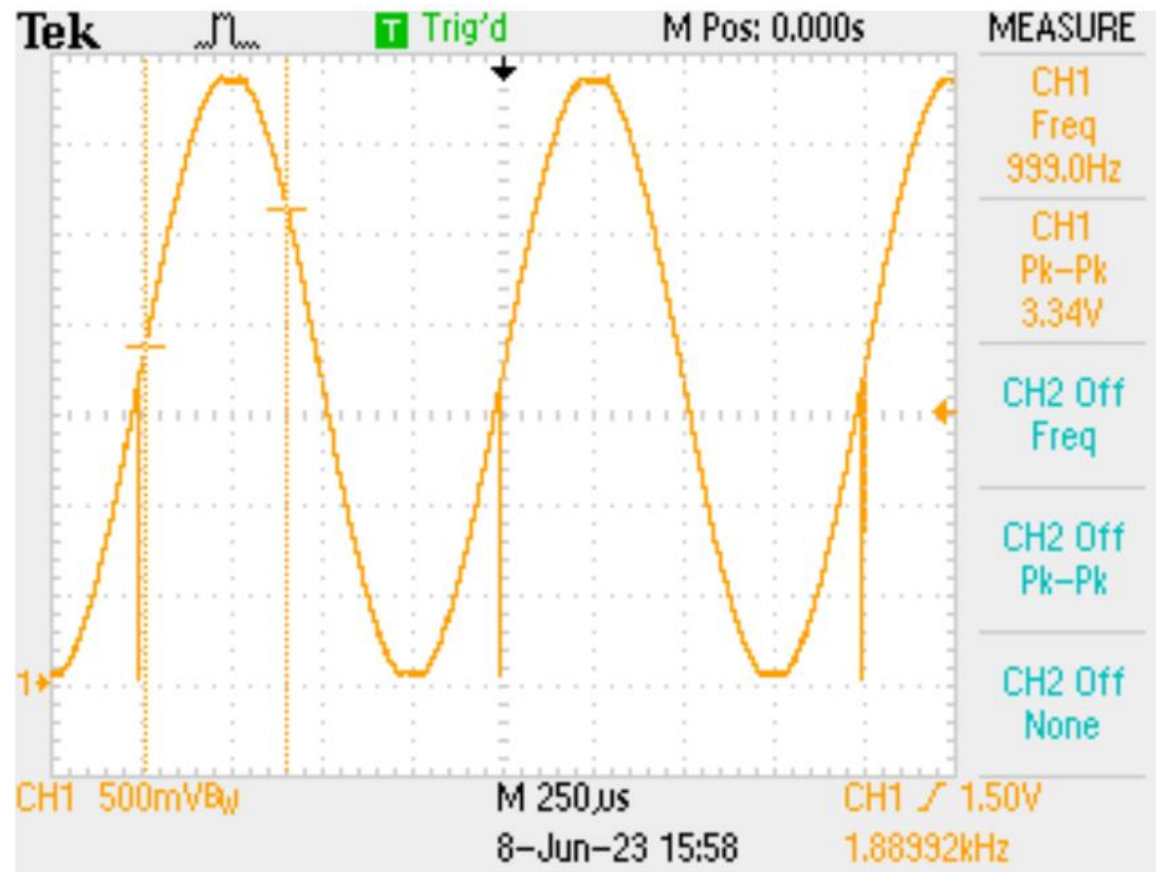
J'utilise un DMA dédié à l'ADC et un DMA dédié au DAC.

L'objectif est de les synchroniser avec un Timer pour avoir le même temps d'enregistrement (ADC) que le temps d'écriture (DAC).

Quand on a synchronisé les deux, on choisit la taille des buffers.

J'ai développé le code en deux étapes :

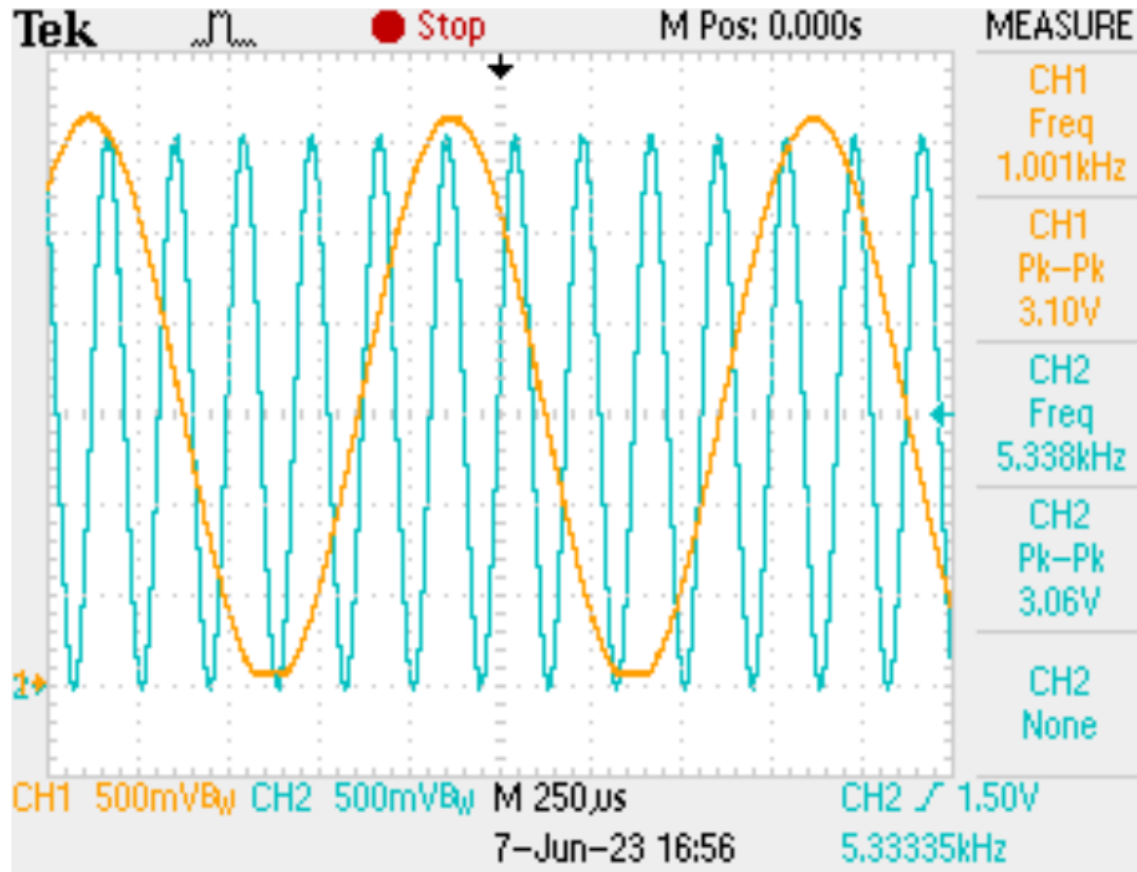
- 1) J'enregistre une période avec l'ADC puis je la joue indéfiniment avec le DAC (les 2 DMA sont séquencées)
- 2) J'enregistre mon entrée sur le buffer ADC puis je la copie sur le buffer DAC à la fin du premier remplissage (les 2 DMA fonctionnent en parallèle)



TDS 2024B - 15:50:27 08/06/2023

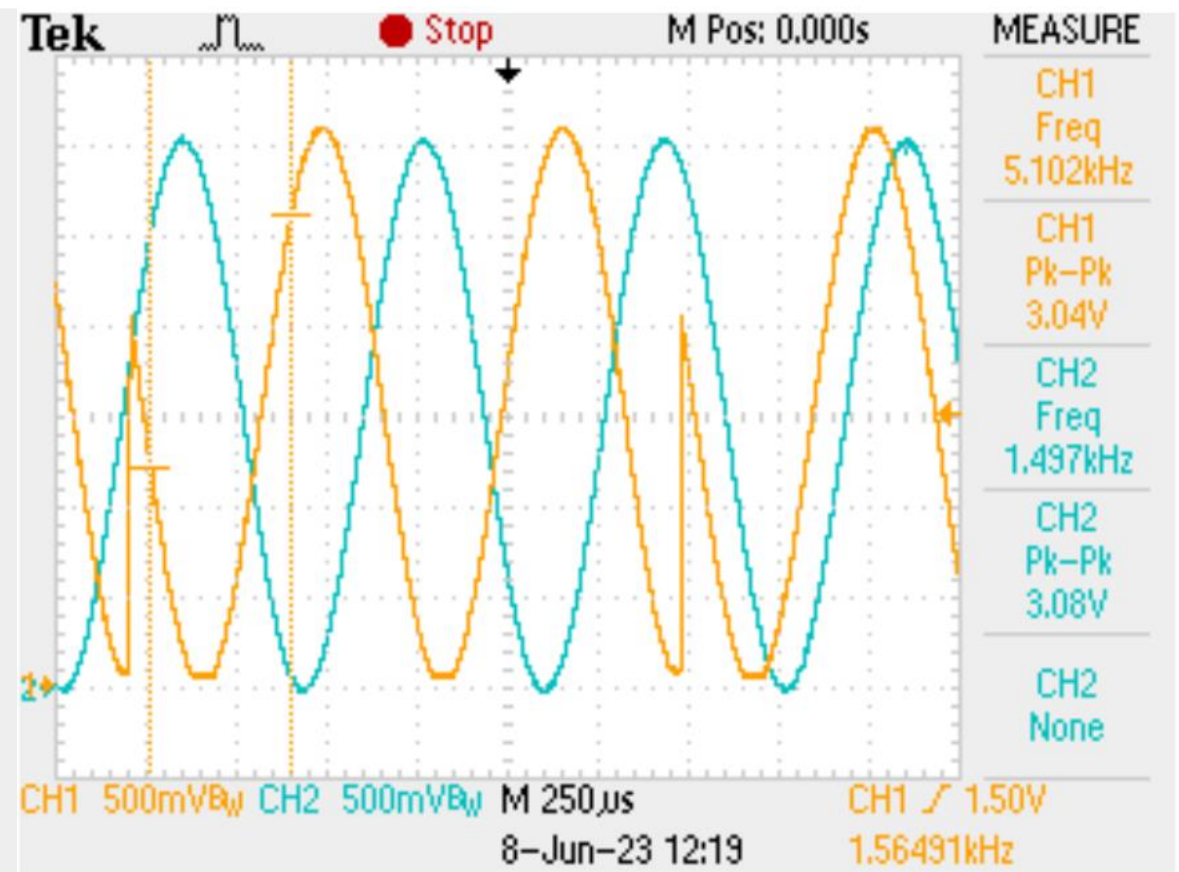
Ce signal est un test pour vérifier la sortie du DAC en lecture sur un buffer rempli.

En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



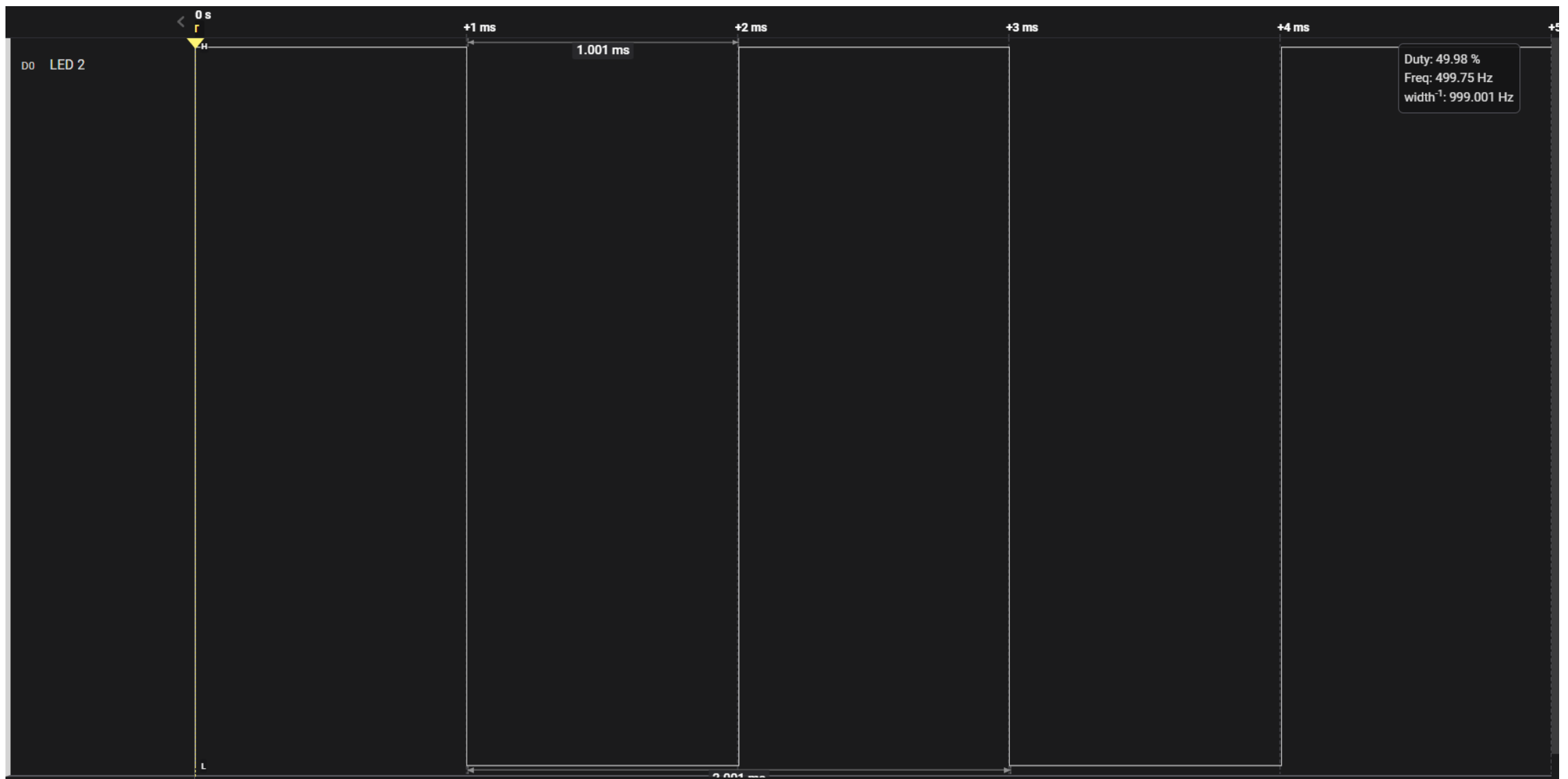
TDS 2024B - 16:47:55 07/06/2023

L'image de gauche montre en entrée de l'ADC un signal à 5 kHz.  
J'ai enregistré ce signal sur une période dans un buffer.  
J'ai ensuite rejouer à 1 kHz ce signal avec le DAC.



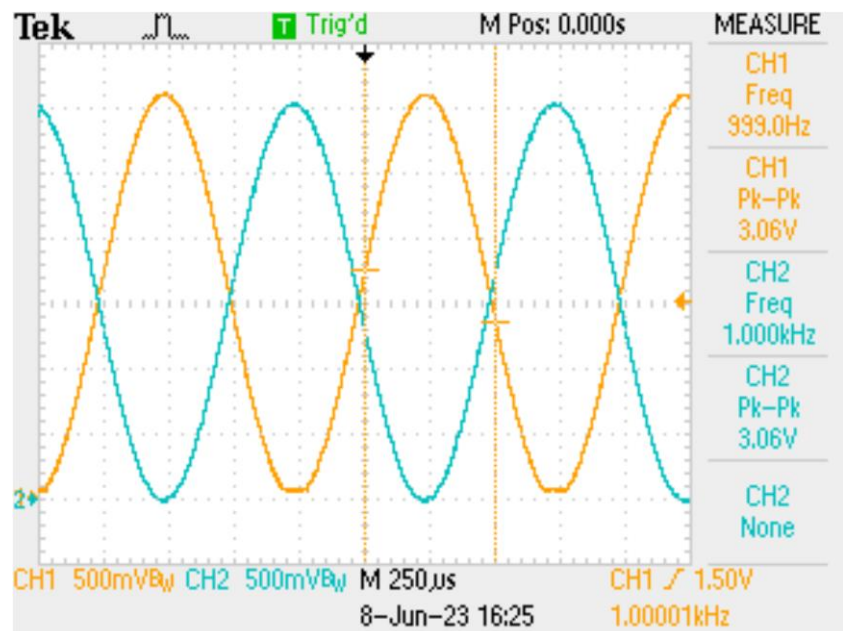
TDS 2024B - 12:11:15 08/06/2023

J'ai ensuite modifié la fréquence en entrée.  
On observe les limites du buffer.  
L'enregistrement doit-être calibré à la fréquence en entrée.

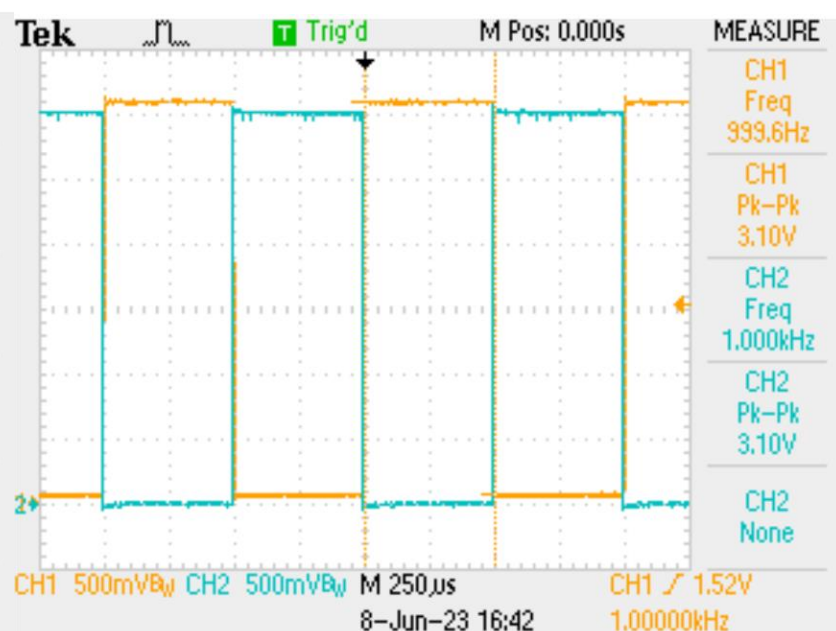


Ce signal est un « Toggle » fait à chaque fin de conversion de l'ADC.  
Je mets 1,001 ms pour remplir mon buffer de 640 éléments.  
Cela fait une fréquence d'échantillonnage de 640 kHz environ.

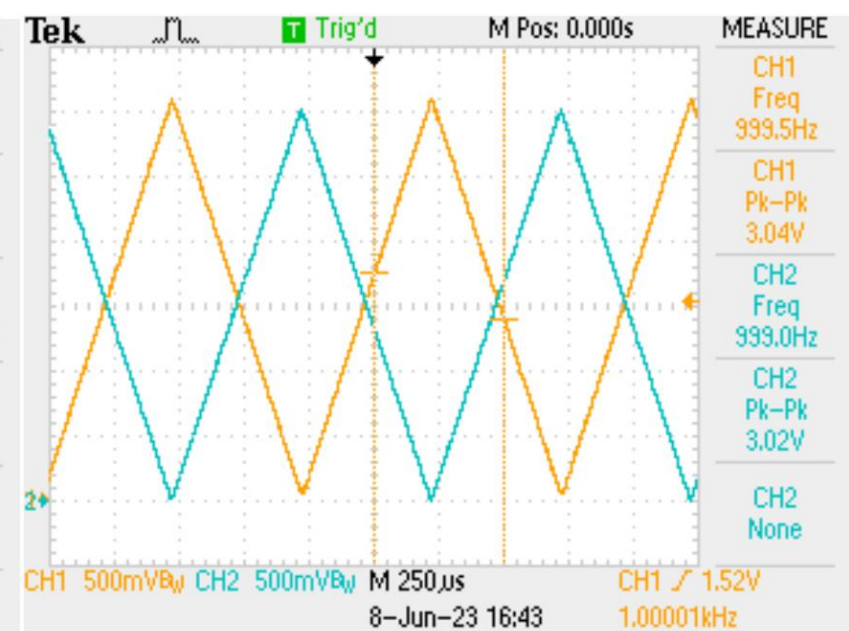
En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



TDS 2024B - 16:17:46 08/06/2023



TDS 2024B - 16:34:25 08/06/2023

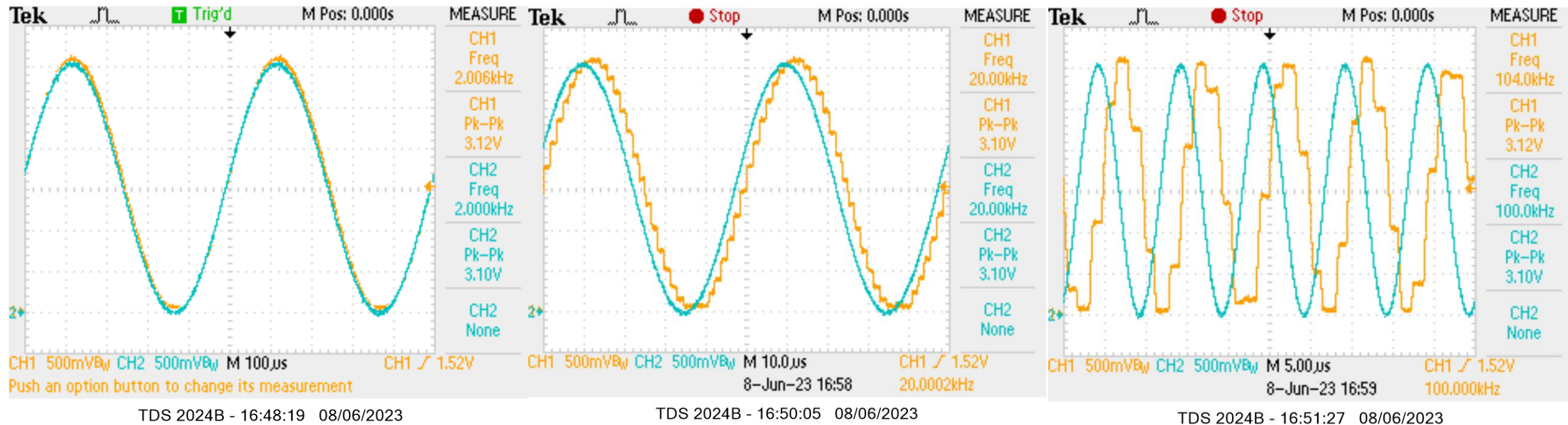


TDS 2024B - 16:35:03 08/06/2023

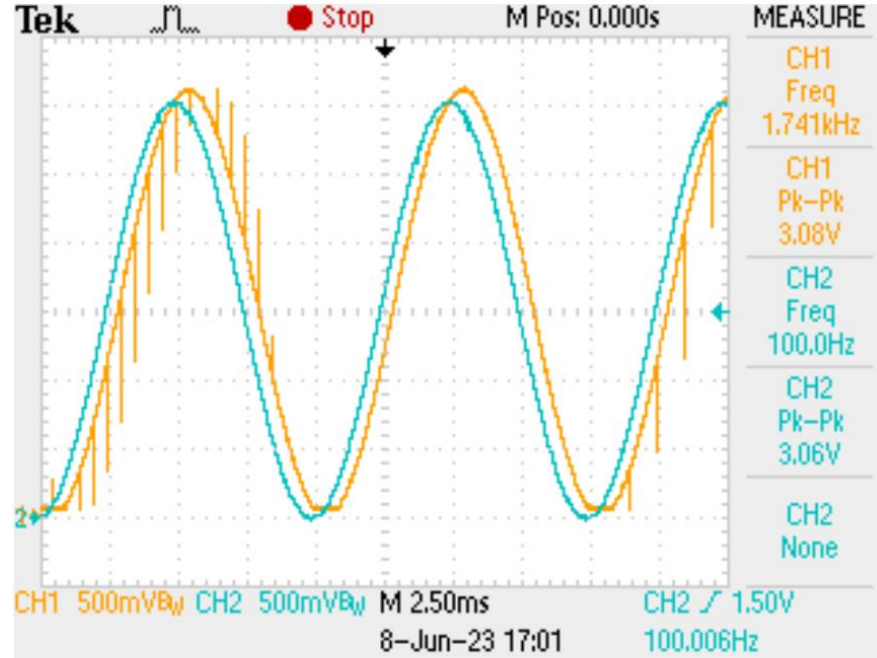
J'ai testé différentes formes de signaux en entrées pour voir la robustesse de mon code.



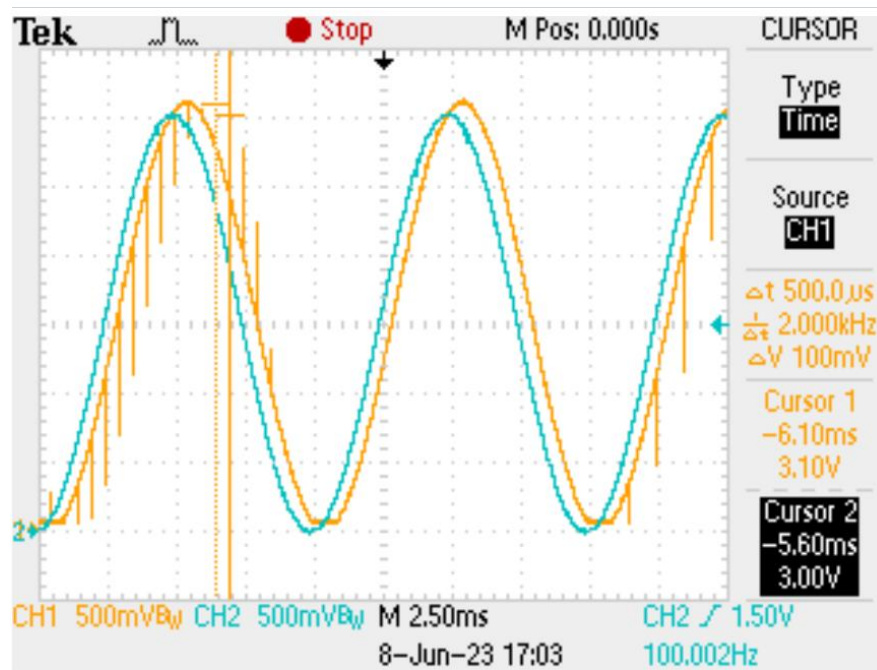
En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



J'ai testé différentes fréquences en entrées pour voir la robustesse de mon code.  
Plus, on monte en fréquence plus on observe le phénomène de sous échantillonnage.

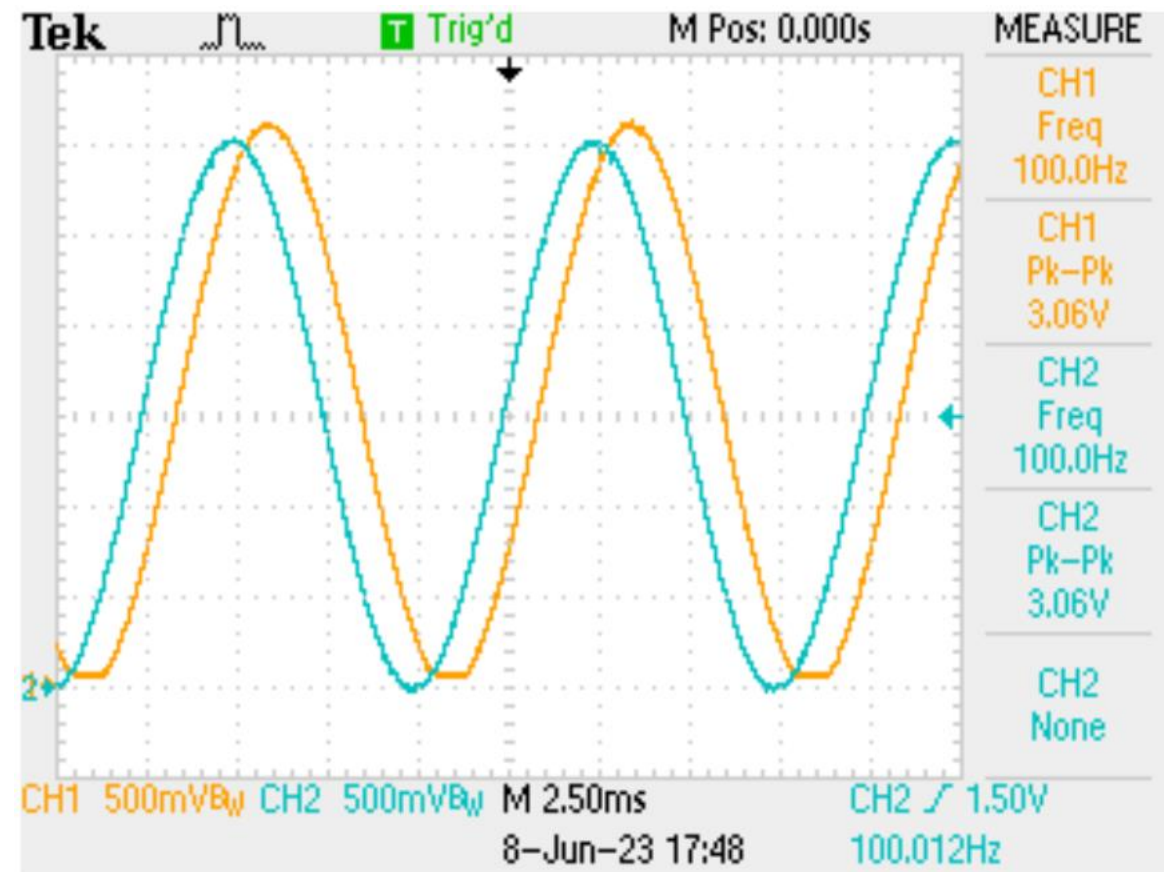


TDS 2024B - 16:53:41 08/06/2023



TDS 2024B - 16:55:41 08/06/2023

En jaune → la sortie du DAC  
En bleu → l'entrée de l'ADC



TDS 2024B - 17:40:20 08/06/2023

Sur les deux images à gauche, j'observe des pointes toutes les 500  $\mu s$ . Ce bug était dû à une mauvaise utilisation du buffer DAC. Il y avait un conflit d'écriture et de lecture au même moment. J'ai corrigé ce bug en décalant l'écriture du buffer voir image à droite.



# Conclusion

Grâce à ces exercices, j'ai appris à utiliser :

- ADC
- DAC
- Le Timer pour trigger l'ADC et le DAC
- Le mode normal de la DMA
- Le mode circulaire de la DMA
- 2 DMA en parallèle avec un double buffer