

Learning-ADC_TO_PWM

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

L'objectif est d'avoir un cycle de PWM en fonction de l'entrée de l'ADC.

Première étape

J'utilise l'ADC dans la boucle « While » puis je copie le buffer vers le registre de la PWM avec un décalage de 4 bits pour passer de 12 bit à 16 bit.

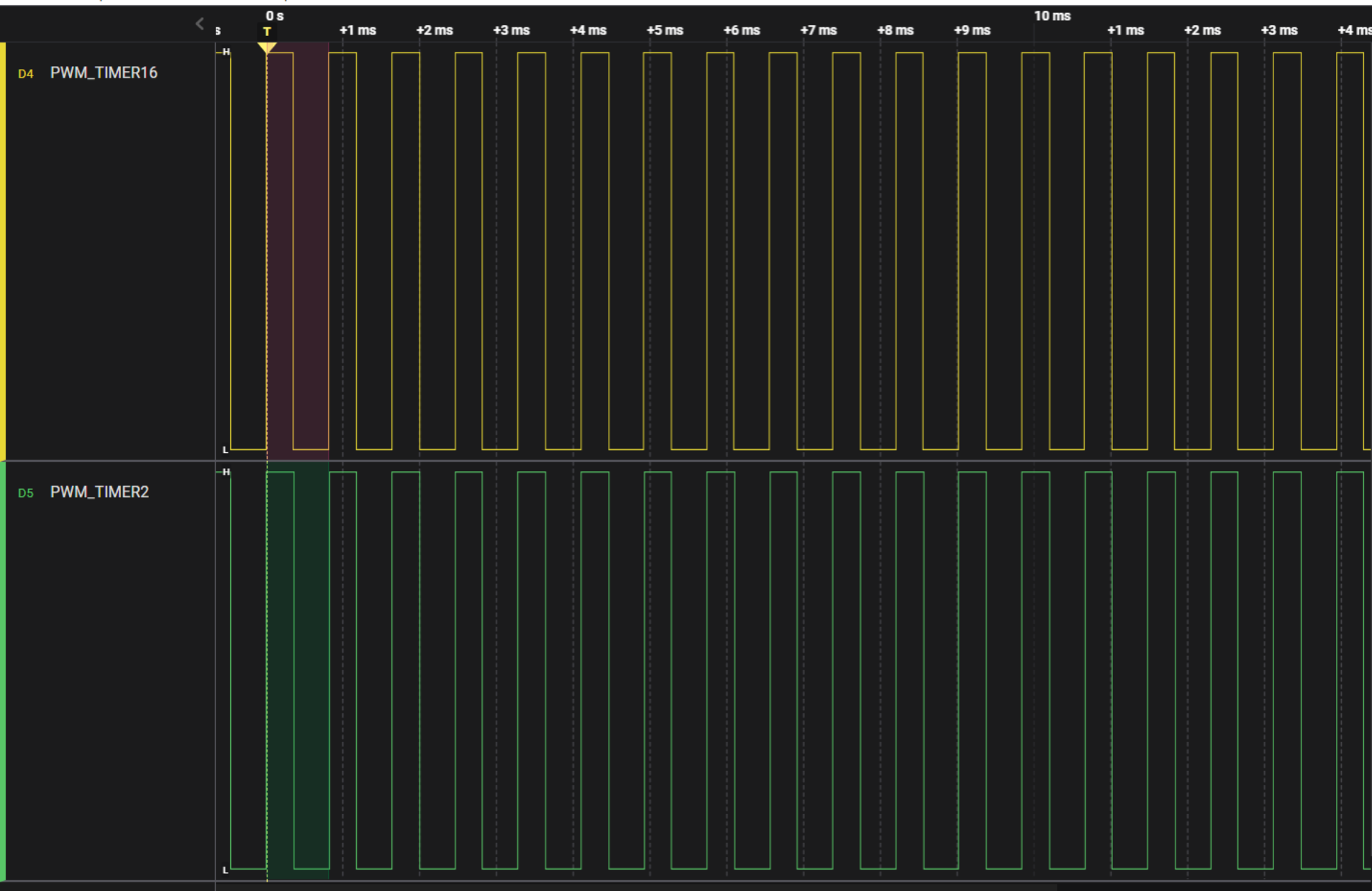
La fréquence de ma PWM est calculé comme ceci :

$$80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$$

Ensuite le rapport cyclique sera entre 100% et 0% en fonction de la valeur de l'ADC.

J'ai branché un potentiomètre au borne de l'ADC pour faire varier la tension.

Note importante : travaillant avec un Logic, je n'observe que des trame numérique. Je n'ai pas faire d'observation en entrée de l'ADC.



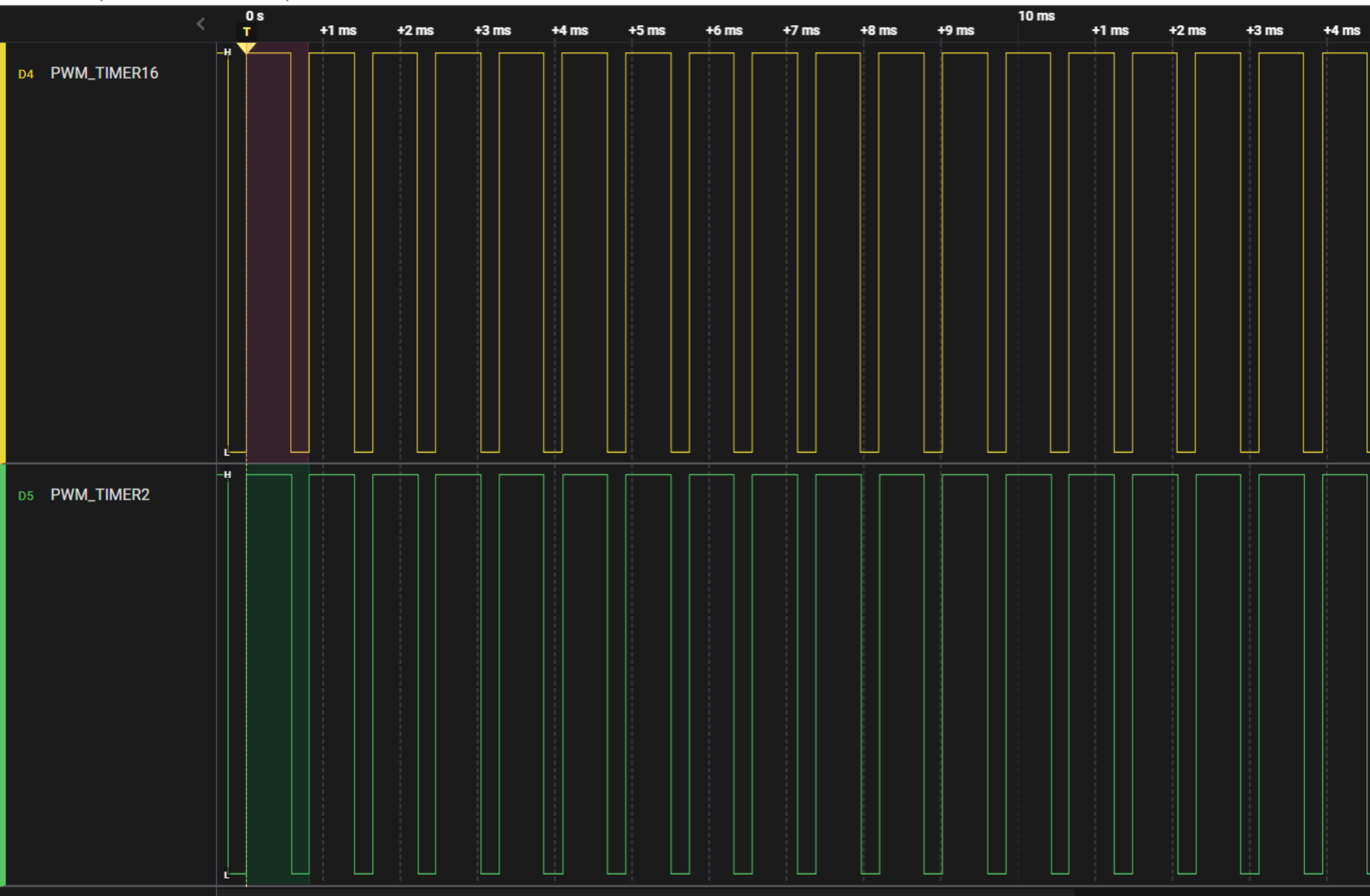
Timing Markers ?

Measurements ?

M0	→	Δ819.75 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		43.479 %
M1	→	Δ819.75 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		43.479 %

Notes ?

Test Code "Learning-ADC_PWM_DMA"
First Step: Update PWM Timer 2 and 16 based on reading on ADC1
 $80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$



Timing Markers ?

Measurements ?

M0	→	Δ819.75 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		71.363 %
M1	→	Δ819.75 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		71.363 %

Notes ?

Test Code "Learning-ADC_PWM_DMA"
First Step: Update PWM Timer 2 and 16 based on reading on ADC1
 $80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$

Deuxième étape

J'utilise l'ADC en mode interruption pour copier le buffer vers le registre de la PWM avec un décalage de 4 bits pour passer de 12 bit à 16 bit.

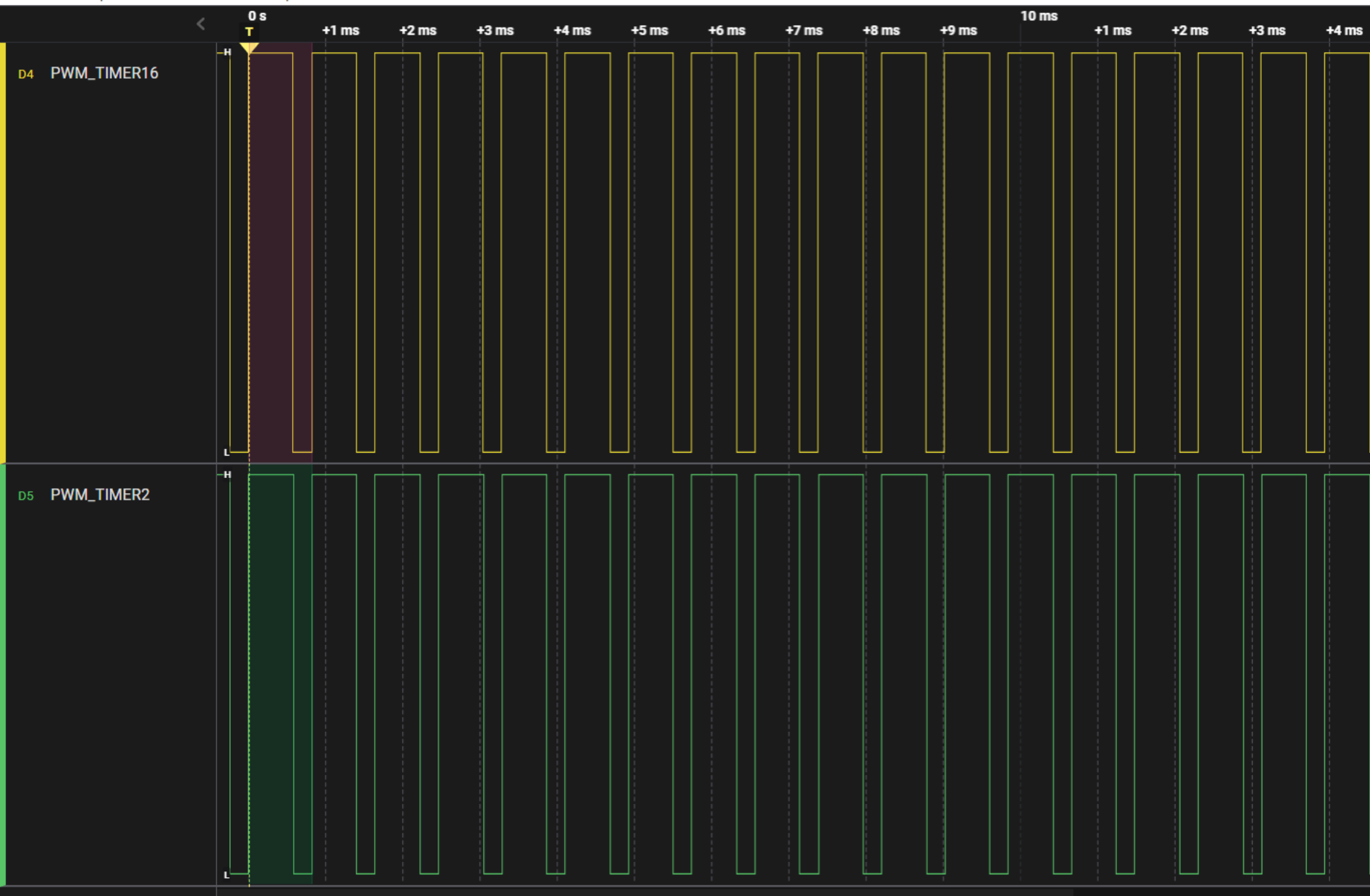
La fréquence de ma PWM est calculé comme ceci :

$$80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$$

Ensuite le rapport cyclique sera entre 100% et 0% en fonction de la valeur de l'ADC.

J'ai branché un potentiomètre au borne de l'ADC pour faire varier la tension.

Note importante : travaillant avec un Logic, je n'observe que des trame numérique. Je n'ai pas faire d'observation en entrée de l'ADC.



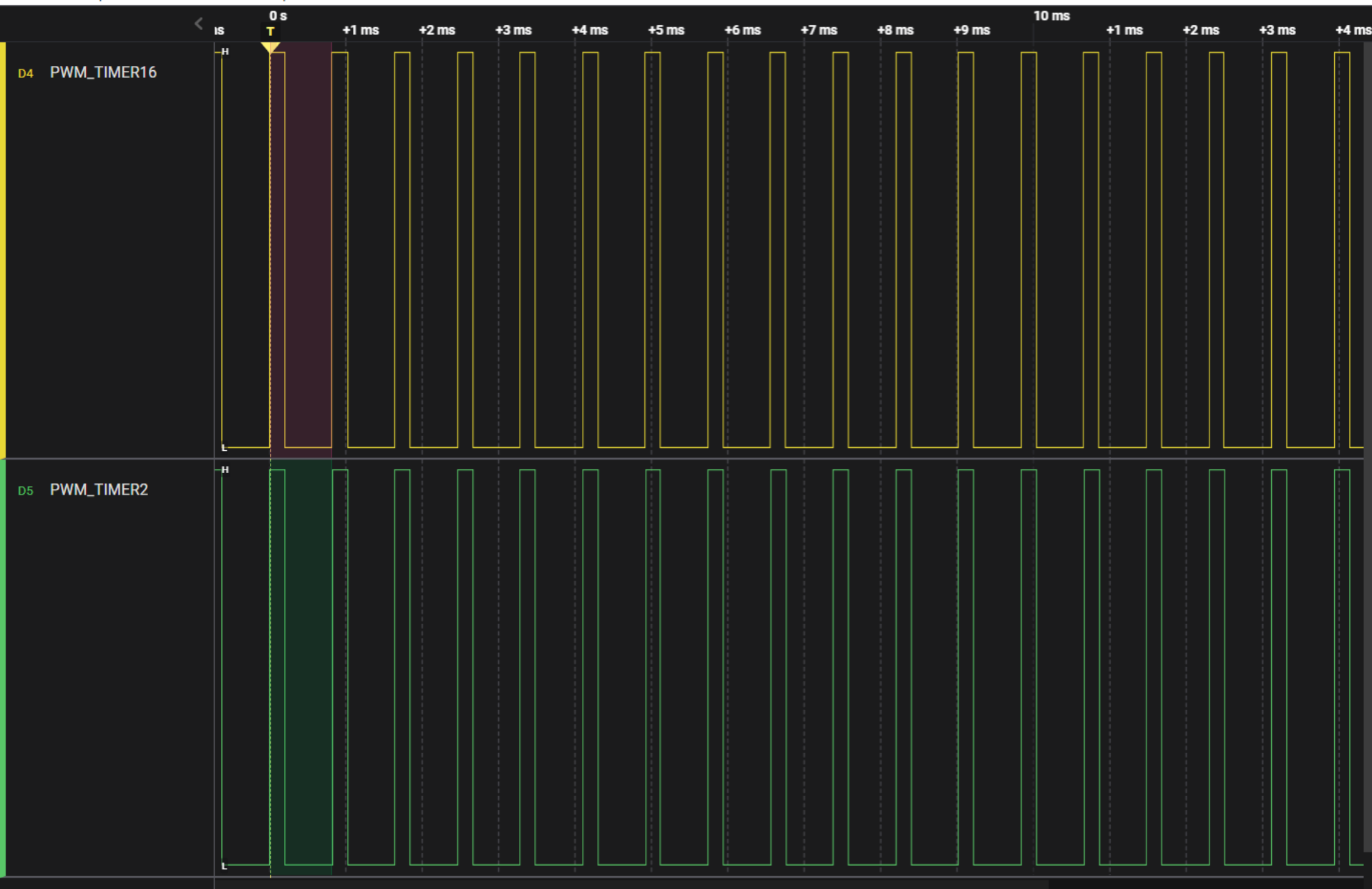
Timing Markers ?

Measurements ?

M0	→	Δ819.708 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		70.747 %
M1	→	Δ819.667 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		70.75 %

Notes ?

Test Code "Learning-ADC_PWM_DMA"
Second Step: Update PWM Timer 2 and 16
based on reading on ADC1 with interrupt
 $80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$



Timing Markers ?

Measurements ?

M0	→	Δ819.708 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		24.414 %
M1	→	Δ819.708 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.22 kHz
f _{mean}		1.22 kHz
T _{std}		N/A
D _{cycle}		24.414 %

Notes ?

Test Code "Learning-ADC_PWM_DMA"
Second Step: Update PWM Timer 2 and 16
based on reading on ADC1 with interrupt
 $80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$

Troisième étape

J'utilise une DMA en mode circulaire sur l'ADC pour copier le buffer vers le registre de la PWM avec un décalage de 4 bits pour passer de 12 bit à 16 bit.

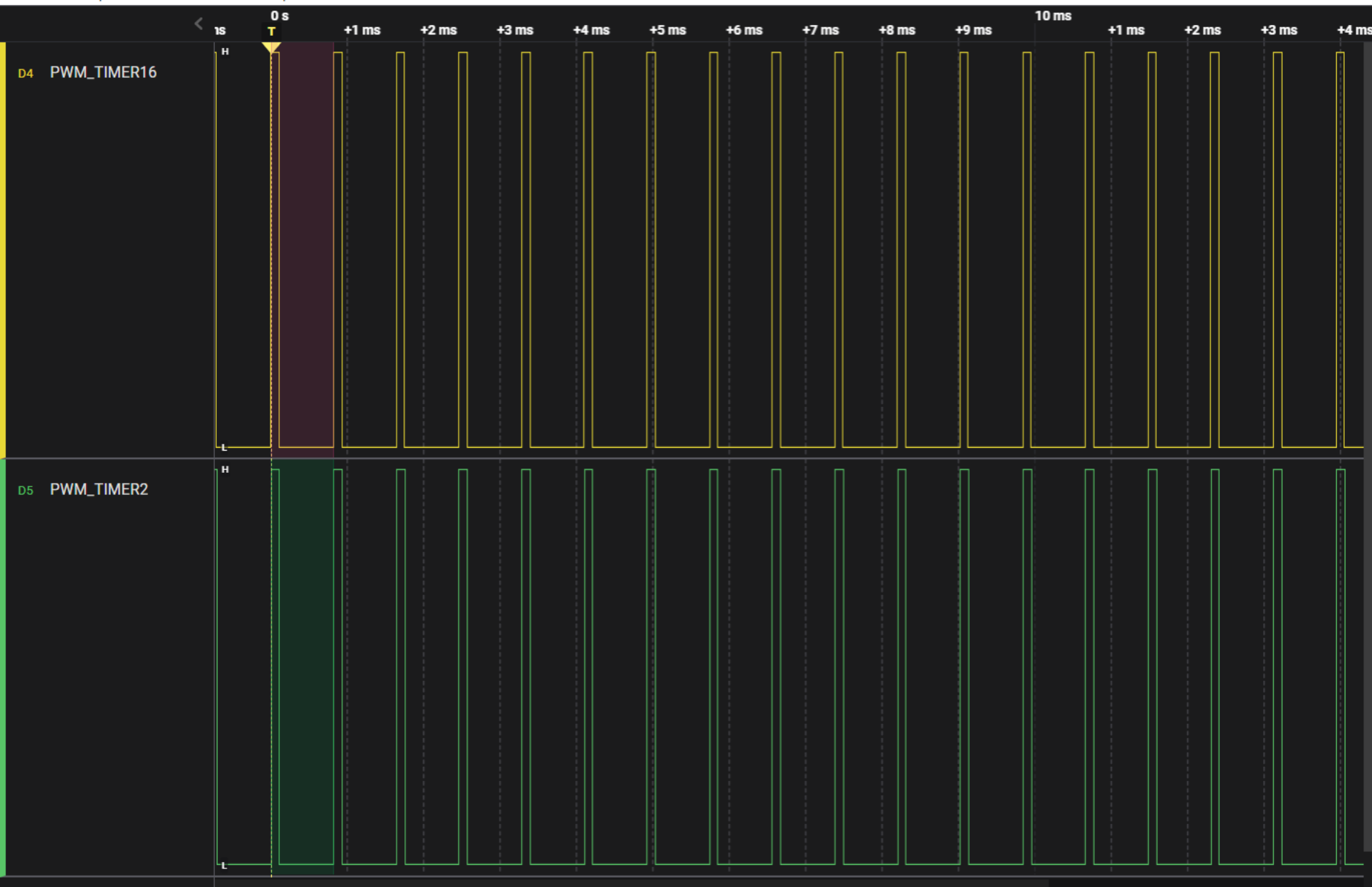
La fréquence de ma PWM est calculé comme ceci :

$$80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$$

Ensuite le rapport cyclique sera entre 100% et 0% en fonction de la valeur de l'ADC.

J'ai branché un potentiomètre au borne de l'ADC pour faire varier la tension.

Note importante : travaillant avec un Logic, je n'observe que des trame numérique. Je n'ai pas faire d'observation en entrée de l'ADC.



Timing Markers

Measurements

M0	→	Δ820.125 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.219 kHz
f _{mean}		1.219 kHz
T _{std}		N/A
D _{cycle}		13.209 %

M1	→	Δ820.083 μs
N _{falling}		1
N _{rising}		2
f _{min}		N/A
f _{max}		1.219 kHz
f _{mean}		1.219 kHz
T _{std}		N/A
D _{cycle}		13.205 %

Notes

Test Code "Learning-ADC_PWM_DMA"
Third Step: Update PWM Timer 2 and 16 based
on reading on ADC1 with DMA
 $80 \text{ MHz} / 65535 = 1,22 \text{ kHz}$

Logic 2 [Logic - Connected] [Session 0]

File Edit Capture Measure View Help

0 s

1 s

1 ms

2 ms

3 ms

4 ms

5 ms

6 ms

7 ms

8 ms

9 ms

10 ms

1 ms

2 ms

3 ms

4 ms

5 ms

6 ms

7 ms

8 ms

9 ms

D4 PWM_TIMER16

H

L

D5 PWM_TIMER2

H

L

Timing Markers ?

Measurements ?

M0 → Δ819.875 μs

N_{falling} 1

N_{rising} 2

f_{min} N/A

f_{max} 1.22 kHz

f_{mean} 1.22 kHz

T_{std} N/A

D_{cycle} 59.394 %

M1 → Δ819.875 μs

N_{falling} 1

N_{rising} 2

f_{min} N/A

f_{max} 1.22 kHz

f_{mean} 1.22 kHz

T_{std} N/A

D_{cycle} 59.399 %

Notes ?

Test Code "Learning-ADC_PWM_DMA"

Third Step: Update PWM Timer 2 and 16 based on reading on ADC1 with DMA

80 MHz / 65535 = 1,22 kHz

Session 0 × +

2 ms ^

Taper ici pour

25°C

16:14

31/05/2023

Conclusion

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

- Le logiciel Logic
- ADC
- Les Timer en mode PWM
- Le mode circulaire de la DMA
- Faire la différence entre le « polling », l'interruption, l'utilisation DMA