

---

# RAPPORT OSCILLOSCOPE

---

Jean Nanchen

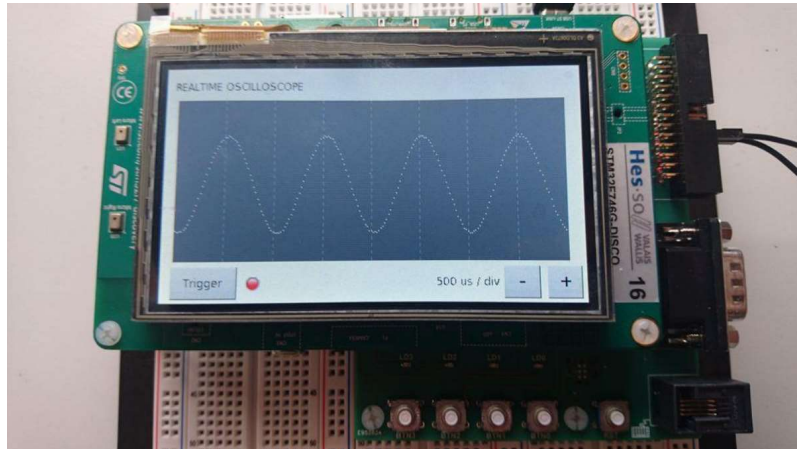
DECEMBER 20, 2020  
HES-SO – 3ÈME ANNÉE

## Table des matières

2.	<i>Introduction</i> .....	2
3.	<i>Création du projet</i> .....	2
4.	<i>Configuration du Timer 1</i> .....	3
5.	<i>Configuration de l'ADC</i> .....	5
6.	<i>Affichage</i> .....	6
7.	<i>Situation actuelle</i> .....	6
8.	<i>Questions</i> .....	7
	Configuration du timing .....	7
	Configuration de l'ADC .....	8
	Sample-Rate Tuning.....	8
9.	<i>Annexes</i> .....	9

## 2. INTRODUCTION

Le but de ce projet est d'échantillonner un signal à l'aide d'une entrée analogique et de l'afficher sur l'écran de notre discovery board.



## 3. CRÉATION DU PROJET

Le projet a d'abord été créé sur la base d'une configuration existante pour le discovery board.

De là, l'USB, le FatFS ainsi que l'Ethernet ont été désactiver.

Le RCC est configuré à 25MHz.

Les heap et le stack sont augmenté à 0x1000. L'IDE utilisé sera System Workbench.

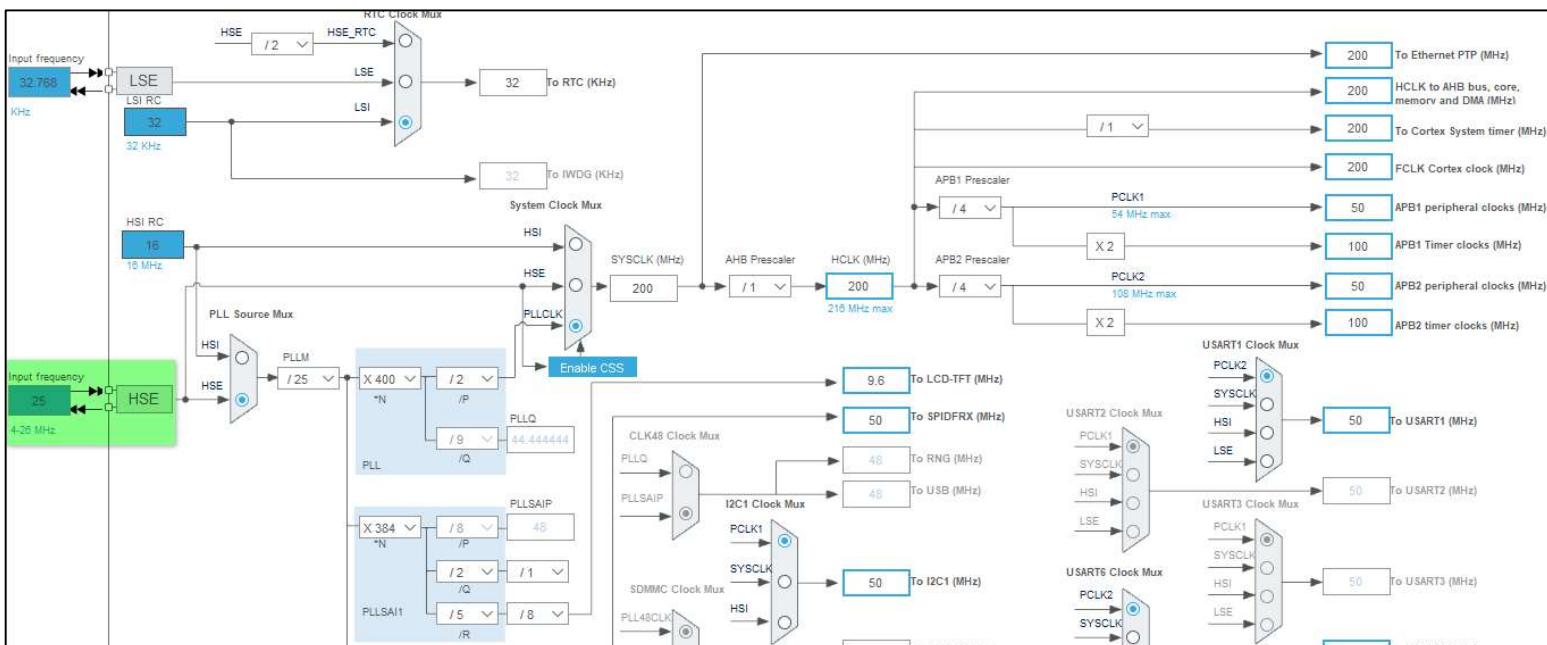


Figure 1 - Configuration du RCC

## 4. CONFIGURATION DU TIMER 1

Le timer 1 est utilisé pour lancer les conversions AD.

Nous devons échantillonner des fréquences allant jusqu'à 1kHz. La fréquence d'échantillonnage ( $f_s$ ) doit être égale ou supérieure à :

$$f_{s_{min}} = 2 * f_{in_{max}} = 2 * 1kHz = 2kHz$$

Nous choisissons une fréquence d'échantillonnage de **10kHz** pour augmenter la précision.

ABP2 est la clock qui incrémente le Timer 1. Il est cadencé à 100MHz (Figure 1). Pour générer un timeout chaque 10kHz, nous devons régler le préscaler à 100 et le counter period à 100.

$$f_{timeout} = \frac{f_{abp2}}{prescaler * counterperiod} = \frac{100MHz}{100 * 100} = 10kHz$$

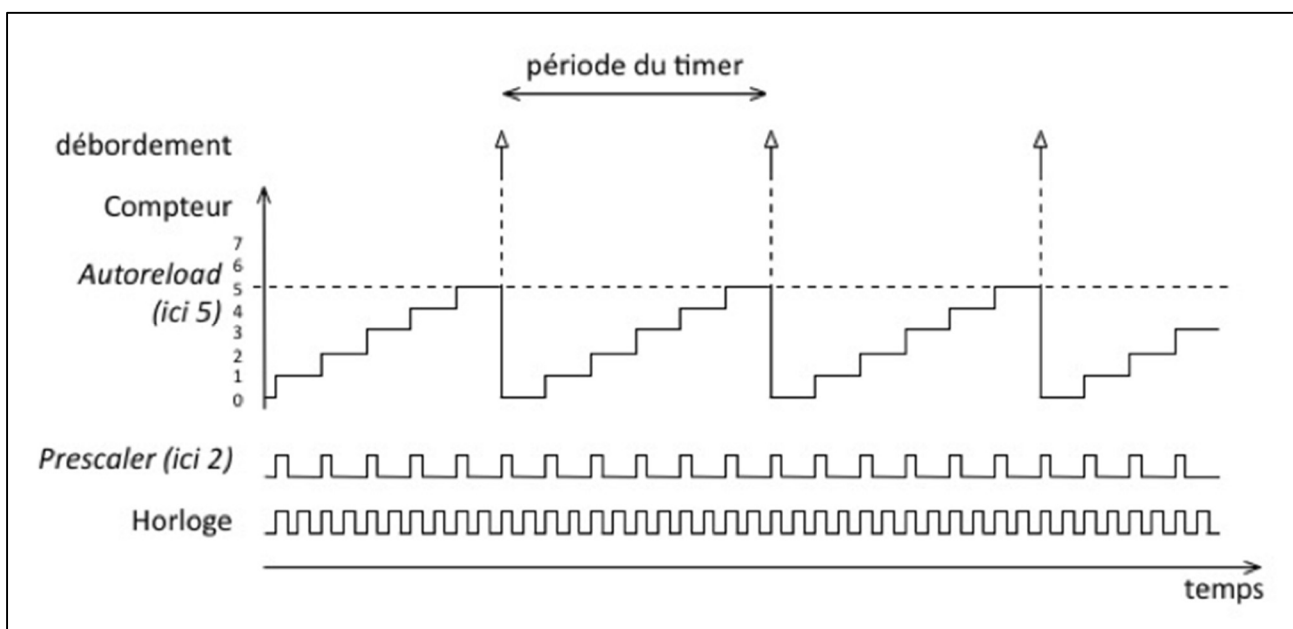


Figure 2 - Préscaler & Counter Period (Autoreload)

Slave Mode

Trigger Source

Clock Source

Channel1

Channel2

Channel3

Configuration

☒ Parameter Settings ☒ User Constants ☒ NVIC Settings ☒ DMA Settings

Configure the below parameters :

✓ Counter Settings

Prescaler (PSC - 16 bits value) 100

Counter Mode Up

Counter Period (AutoReload Register - 16 bits value) 100

Internal Clock Division (CKD) No Division

Repetition Counter (RCR - 16 bits value) 0

auto-reload preload Disable

> Trigger Output (TRGO) Parameters

> Break And Dead Time management - BRK Configuration

> Break And Dead Time management - BRK2 Configuration

> Break And Dead Time management - Output Configuration

✓ Output Compare No Output Channel 1

Mode Toggle on match

Pulse (16 bits value) 0

CH Polarity High

CH Idle State Reset

Figure 3 - Configuration du Timer 1

Sur la Figure 3 se trouve la configuration du Timer 1.

## 5. CONFIGURATION DE L'ADC

Nous avons configuré l'ADC de sorte qu'à chaque fois que le Timer 1 inverse sa sortie, une lecture ADC est lancée.

▼ ADCs_Common_Settings	
Mode	Independent mode
▼ ADC_Settings	
Clock Prescaler	PCLK2 divided by 2
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Disabled
End Of Conversion Selection	EOC flag at the end of single channel conversion
▼ ADC_Regular_ConversionMode	
Number Of Conversion	1
External Trigger Conversion Source	Timer 1 Capture Compare 1 event
External Trigger Conversion Edge	Trigger detection on both the rising and falling edges
> Rank	1
▼ ADC_Injected_ConversionMode	
Number Of Conversions	0
▼ WatchDog	
Enable Analog WatchDog Mode	<input type="checkbox"/>

Figure 4 - Configuration de l'ADC

## 6. AFFICHAGE

Pour l'affichage il a fallu coder une fonction `doShowAnalogSignal()` de la classe `OscilloscopeController`.

- Affichage du signal avec la fonction `drawGraphPoints`.
- Changement de l'axe du temps avec les boutons tactiles de l'écran
- Fonction de trigger

## 7. SITUATION ACTUELLE

Je n'ai pas eu le temps d'intégrer le RTOS à mon projet. J'ai préféré effectuer certaines des tâches supplémentaires.

Task	Résultat	Remarque
<b>Tâche 1 – Projet STM32CubeMX</b>	Fonctionnel	
<b>Tâche 2 – Éteindre USB, FatFS et Ethernet</b>	Fonctionnel	
<b>Tâche 3 – Configuration du RCC</b>	Fonctionnel	
<b>Tâche 4 – Configuration de la génératrice de code de CubeMX</b>	Fonctionnel	
<b>Tâche 5 – Build et Debug</b>	Fonctionnel	
<b>Tâche 6 – Configuration du Timing</b>	Fonctionnel	
<b>Tâche 7 – Générateur de Fréquence Externe</b>	Fonctionnel	
<b>Tâche 8 – Configuration de l'ADC (Software Triggered)</b>	Fonctionnel	
<b>Tâche 9 – Configuration du timer Hardware (TIM)</b>	Fonctionnel	
<b>Tâche 10 – Configuration du ADC (Timer Triggered)</b>	Fonctionnel	
<b>Tâche 11 – XF Integration</b>	Fonctionnel	
<b>Tâche 12 – Application</b>	Fonctionnel	
<b>Tâche 13 – Oscilloscope GUI</b>	Fonctionnel	
<b>Tâche 14 – Sample-Rate Tuning</b>	Fonctionnel	
<b>Tâche 15 – RTOS Integration</b>	Non effectué	temps
<b>FA1 : Fonction Trigger</b>	Fonctionnel	
<b>FA2: Display – Axe du temps</b>	Fonctionnel	
<b>FA3 : Enregistrer les valeurs échantillonnées à l'aide du DMA</b>		Non obligatoire

## 8. QUESTIONS

### CONFIGURATION DU TIMING

#	Component	Timing
1	Conversion du signal analogue	1 kHz ou plus
2	Rafraichissement de l'écran	20 à 60 fois par seconde

*Est-ce qu'il est possible d'exécuter le composant numéro 1 avec un XF ou bien faut-il un RTOS ? Justifiez votre réponse.*

- Non, il est impossible d'exécuter le composant numéro 1 avec un XF ou un RTOS.

*Est-ce qu'il est possible d'exécuter le composant numéro 2 avec un XF ou bien faut-il un RTOS ? Justifiez votre réponse.*

- Oui, car ce sont des fréquences plus basses.

*Si l'on combine un timer hardware avec un XF, lequel des deux doit être priorisé ? Justifiez votre réponse.*

- Le timer hardware, il doit arrêter le programme un instant pour effectuer sa ISR.



## CONFIGURATION DE L'ADC

*Combien de mesures [Samples/s] le convertisseur A/D doit-il effectuer par seconde pour pouvoir échantillonner des signaux avec des fréquences jusqu'à 1 kHz ?*

- Théorème de nyquist :  $f_s = f_{\text{min}} * 2$

*Faut-il un filtre ? Si oui, quelle sera la fréquence de coupure de ce filtre ?*

- Il faut un filtre anti-aliasing de 1kHz.

*Est-ce que la fréquence donnée par le théorème d'échantillonnage ou devrait-elle être plus élevée ?*

- Plus elle est élevée plus le signal reconstruit sera proche du signal échantillonné

*Lequel des canaux du ADC3 doit être utilisé pour pouvoir mesurer / échantillonner le signal l'aide de la broche PA0 ?*

- Le canal IN0

*Est-ce que le ADC pourrait-il éventuellement effectuer des mesures à des intervalles réguliers à l'aide de ses propres moyens ?*

- Non, uniquement en continu. Il faut un timer pour que ce soit précis.

## SAMPLE-RATE TUNING

*Quelle fréquence d'échantillonnage peut être atteinte ?*

- 20kHz

*Quel(s) composant(s) limite(nt) le système?*

Au bout d'un moment, il n'y a que des interruptions et le XF n'arrive plus à être exécuté correctement. Il faut alors un DMA qui s'occupe de ça en parallèle.

## 9. ANNEXES

**Table 1. STM32F76xxx and STM32F77xxx register boundary addresses (continued)**

Boundary address	Peripheral	Bus	Register map
0x4001 7800 - 0x4001 7BFF	MDIOS	APB2	<a href="#">Section 38.4.10: MDIOS register map on page 1470</a>
0x4001 7400 - 0x4001 77FF	DFSDM1		<a href="#">Section 17.8.16: DFSDM register map on page 555</a>
0x4001 6C00 - 0x4001 73FF	DSI Host		<a href="#">Section 20.17: DSI Host register map on page 737</a>
0x4001 6800 - 0x4001 6BFF	LCD-TFT		<a href="#">Section 19.7.26: LTDC register map on page 622</a>
0x4001 5C00 - 0x4001 5FFF	SAI2		<a href="#">Section 36.5.18: SAI register map on page 1422</a>
0x4001 5800 - 0x4001 5BFF	SAI1		<a href="#">Section 36.5.18: SAI register map on page 1422</a>
0x4001 5400 - 0x4001 57FF	SPI6		<a href="#">Section 35.9.10: SPI/I2S register map on page 1368</a>
0x4001 5000 - 0x4001 53FF	SPI5		
0x4001 4800 - 0x4001 4BFF	TIM11		<a href="#">Section 27.5.12: TIM10/TIM11/TIM13/TIM14 register map on page 1078</a>
0x4001 4400 - 0x4001 47FF	TIM10		
0x4001 4000 - 0x4001 43FF	TIM9		<a href="#">Section 27.4.13: TIM9/TIM12 register map on page 1068</a>
0x4001 3C00 - 0x4001 3FFF	EXTI		<a href="#">Section 11.9.7: EXTI register map on page 325</a>
0x4001 3800 - 0x4001 3BFF	SYSCFG		<a href="#">Section 7.2.9: SYSCFG register map on page 244</a>
0x4001 3400 - 0x4001 37FF	SPI4		<a href="#">Section 35.9.10: SPI/I2S register map on page 1368</a>
0x4001 3000 - 0x4001 33FF	SPI1		<a href="#">Section 35.9.10: SPI/I2S register map on page 1368</a>
0x4001 2C00 - 0x4001 2FFF	SDMMC1		<a href="#">Section 39.8.16: SDMMC register map on page 1529</a>
0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3		<a href="#">Section 15.13.18: ADC register map on page 483</a>
0x4001 1C00 - 0x4001 1FFF	SDMMC2		<a href="#">Section 39.8.16: SDMMC register map on page 1529</a>
0x4001 1400 - 0x4001 17FF	USART6		<a href="#">Section 34.8.12: USART register map on page 1307</a>
0x4001 1000 - 0x4001 13FF	USART1		
0x4001 0400 - 0x4001 07FF	TIM8		<a href="#">Section 25.4.27: TIM8 register map on page 958</a>
0x4001 0000 - 0x4001 03FF	TIM1		<a href="#">Section 25.4.26: TIM1 register map on page 955</a>

Figure 5 - Périphériques cadencés par APB2