

# Rapport Realtime Oscilloscope

Martin Meyer

06 Mai 2020

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Diagrammes</b>	<b>3</b>
2.1	Diagramme de timing . . . . .	3
2.2	Diagramme de classe . . . . .	4
<b>3</b>	<b>Patterns</b>	<b>5</b>
3.1	Singleton . . . . .	5
3.2	Sujet-observateur . . . . .	5
3.3	Factory . . . . .	5
<b>4</b>	<b>Questions</b>	<b>5</b>
4.1	Tâche 6 . . . . .	5
4.1.1	Question 1 . . . . .	5
4.1.2	Question 2 . . . . .	5
4.1.3	Question 3 . . . . .	5
4.2	Tâche 8 . . . . .	6
4.2.1	Question 1 . . . . .	6
4.2.2	Question 2 . . . . .	6
4.2.3	Question 3 . . . . .	6
4.2.4	Question 4 . . . . .	6
4.2.5	Question 5 . . . . .	6
4.3	Tâche 14 . . . . .	6
4.3.1	Question 1 . . . . .	6
4.3.2	Question 2 . . . . .	7
4.4	Tâche 15 . . . . .	7
4.4.1	Question 1 . . . . .	7
4.4.2	Question 2 . . . . .	7
4.4.3	Question 3 . . . . .	7
<b>5</b>	<b>Tâches facultatives</b>	<b>7</b>
5.1	FA1 - Fonction trigger . . . . .	7
5.2	FA2 - Axe du temps . . . . .	8

<b>6 Tests</b>	<b>9</b>
6.1 A . . . . .	9
6.2 B . . . . .	10
6.3 C . . . . .	11
6.4 D . . . . .	12
6.5 E . . . . .	15
6.6 F . . . . .	16
6.7 G . . . . .	18
6.8 L . . . . .	19
6.9 Summary . . . . .	20
<b>7 Conclusion</b>	<b>20</b>

## 1 Introduction

Dans ce laboratoire, nous avons créé un système temps réel réalisant un fonction d'oscilloscope. Une partie des tâches, comme l'échantillonnage, fonctionne en temps réel. D'autre tâches, comme l'affichage, n'en ont pas besoin.

Pour ce faire, nous avons travaillé avec les cartes microcontrôleur STM32f7-discovery, System Workbench for STM32 et STM32CubeMX.

Ce rapport décrit quelques parties intéressantes du programme, notamment grâce à des diagrammes UML.

## 2 Diagrammes

### 2.1 Diagramme de timing

Ci-dessous, le diagramme de timing :

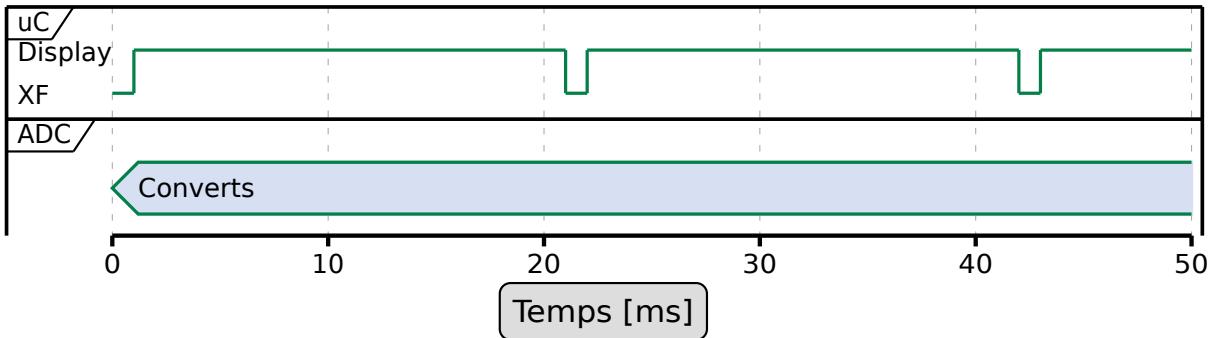


FIGURE 1 – Diagramme de timing

On peut voir sur la figure 1 que l'affichage prend environ 20ms. Il s'en suit un court passage dans le XF. L'adc est considéré comme toujours actif, car il fait une conversion toutes les 10 us.

## 2.2 Diagramme de classe

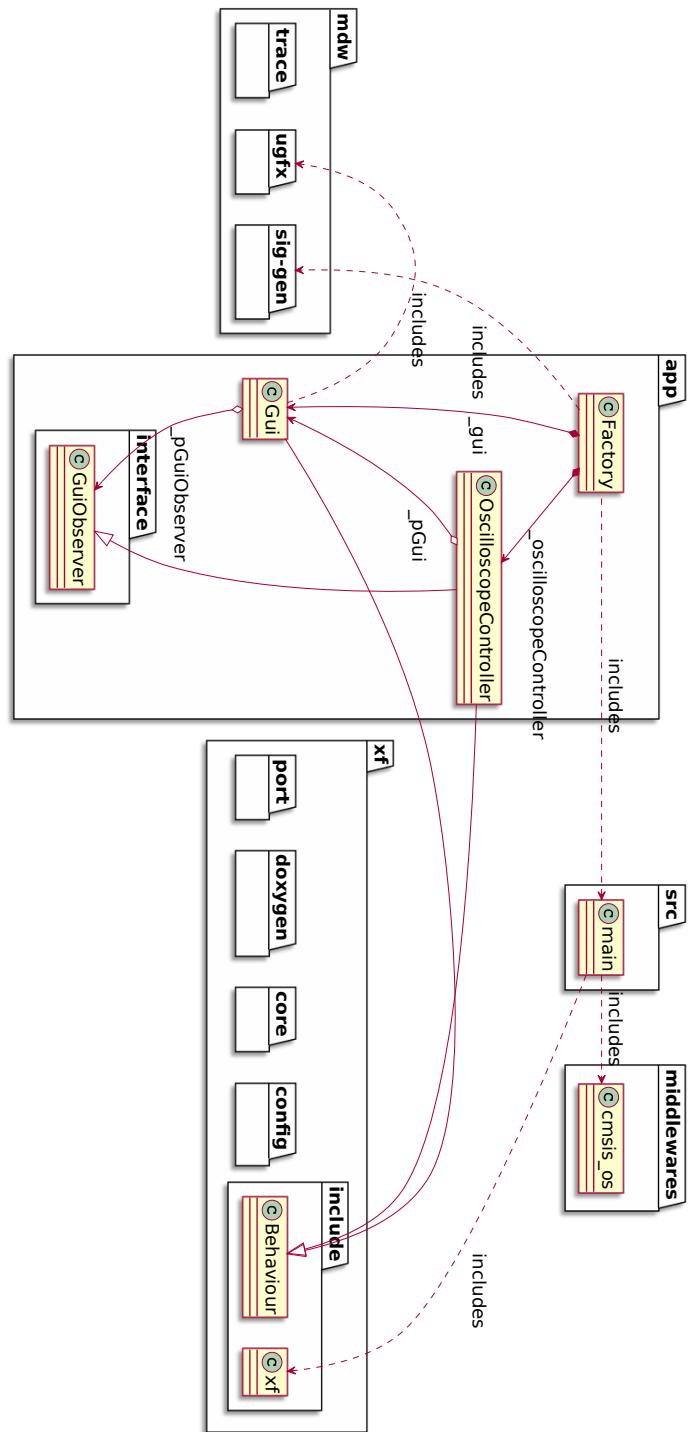


FIGURE 2 – Diagramme de classe

## 3 Patterns

### 3.1 Singleton

Le pattern singleton est utilisé dans plusieurs classes, comme la Factory ou l'OsilloscopeController.

### 3.2 Sujet-observateur

Un pattern sujet-observateur est utilisé. La classe OscilloscopeController est l'observateur et la classe Gui le sujet.

### 3.3 Factory

Une Factory est utilisée dans ce labo. Elle initialise des relations et lance le système.

## 4 Questions

### 4.1 Tâche 6

#### 4.1.1 Question 1

**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, étant donné le timing de conversion du signal analogique qui est court, le processeur passerait trop de temps dans les switching context ou dans le xf. Il est préférable d'utiliser une timer hardware dans ce cas.

#### 4.1.2 Question 2

**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, c'est possible, car le taux de rafraîchissement est suffisamment bas pour utiliser un XF ou un RTOS.

#### 4.1.3 Question 3

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

Il faut prioriser le timer hardware, car il est plus important d'avoir un échantillonnage précis plutôt qu'un affichage précis. De plus, comme un rafraîchissement de l'écran dure plusieurs milliseconds, la conversion du signal analogique serait bloquée pendant ce temps.

## 4.2 Tâche 8

### 4.2.1 Question 1

**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 ?**

Le théorème de Shannon nous dit que la fréquence d'échantillonnage doit être au moins 2 fois plus grande que la fréquence du signal à échantilloné. On devrait donc faire au moins 2kSamples/s. Cependant, pour avoir un signal lisible, il faudrait au moins avoir une fréquence de sampling de 20kSamples/s.

### 4.2.2 Question 2

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

Oui, il faut ajouter un filtre passe-bas à l'entrée du signal à mesurer avec une fréquence de coupure à 1kHz, afin d'éviter que des signaux avec une fréquence plus élevée fausse les mesures, par un phénomène de sous-échantillonnage.

### 4.2.3 Question 3

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

La fréquence doit être plus élevée (par exemple 10kSamples/s) pour avoir un signal lisible. Si l'on utilise la fréquence donnée par le théorème d'échantillonnage, on aura seulement 2 points de mesure par période.

### 4.2.4 Question 4

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

Le canal IN0.

### 4.2.5 Question 5

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

Oui il le pourrait, en mode continu. Mais on ne pourrait pas choisir l'intervalle de ces conversions, car l'ADC relancerait une nouvelle conversion automatiquement directement après qu'une conversion soit finie.

## 4.3 Tâche 14

### 4.3.1 Question 1

**Quelle fréquence d'échantillonnage peut être atteinte ?**

La fréquence d'échantillonnage maximale atteinte est de 250kSamples/s.

### 4.3.2 Question 2

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

Le temps d'interruption qui limite le système. Lorsque la fréquence d'échantillonnage est trop élevée, le processeur va être tout le temps dans l'interruption de fin de conversion ADC. Le clock et la mémoire limitent le système, car en les augmentant, on pourrait augmenter la fréquence de sampling.

## 4.4 Tâche 15

### 4.4.1 Question 1

**À quelle fréquence maximale peut-on régler l'échantillonnage ?**

On peut la régler à 250kSamples/s au maximum.

### 4.4.2 Question 2

**Quels avantages voyez-vous à utiliser FreeRTOS dans cette application ?**

Avec des systèmes qui fonctionnent à des grandes fréquences (plus de 1kHz), un RTOS devient intéressant car il permet d'utiliser des priorités entre différentes tâches. Dans notre cas, le timer serait par exemple priorisé.

### 4.4.3 Question 3

**Donnez un exemple où un RTOS serait particulièrement nécessaire ?**

Par exemple un système d'arrêt d'urgence doit être priorisé sur toutes les autres tâches.

## 5 Tâches facultatives

### 5.1 FA1 - Fonction trigger

Le but de cette fonction est d'afficher le signal d'une façon à ce qu'il reste toujours au même endroit sur l'écran et d'éviter qu'il se déplace à travers de l'écran.

Le seuil du trigger a été fixé à 2048 arbitrairement. Cette valeur correspond à la moitié de la plage de valeur de l'ADC. La fonctionnalité trigger est implémentée dans la méthode doShowAnalogSignal() de la classe OscilloscopeController.

```

1 if( TriggerOn ){
2     uint32_t temp_low=0;
3     uint32_t temp_high=0;
4     uint32_t nbMoy=5;
5     for( int i=nbMoy ; i<_adcValuesBufferSize-size-nbMoy ; i++ ){
6         for( int j=0;j<nbMoy ; j++ ){// Points average loop
7             temp_low+=_adcValuesBuffer[ i-j ];
8             temp_high+=_adcValuesBuffer[ i+j ];
9         }
10        temp_low/=nbMoy;
11        temp_high/=nbMoy;
12
13        //check the trigger value
14        if( temp_low<triggerValue&&temp_high>triggerValue ){
15            //update the pointer
16            trigPtr = &_adcValuesBuffer[ i ];
17        }
18    }
19 }
20 else{
21     trigPtr = _adcValuesBuffer ;
22 }
23
24 gui().drawGraphPoints( trigPtr , size , xScale );

```

FIGURE 3 – Trigger Code

La figure 3 montre le code du trigger. On va rechercher l’index du tableau de mesures auquel le niveau des mesures devient plus élevé que le seuil du trigger. Pour plus de précision, une moyenne de 5 points est effectuée. Une fois que le seuil est détecté, le pointer passé à la fonction drawGraphPoint est mis à jour.

## 5.2 FA2 - Axe du temps

Pour calculer le paramètre xScale de la méthode drawPoints, l’équation suivante a été utilisée :

$$\frac{600}{tDiv \cdot \frac{8}{tSample}} \quad (1)$$

Le 600 correspond au nombre de pixel en x à afficher, et le 8 est le nombre de division. Elle a ensuite été transformée pour ne pas utiliser des float avec une valeur trop petite :

$$\frac{600}{fSample \cdot \frac{8}{fDiv}} \quad (2)$$

Malgré ce calcul, l'échelle n'était pas correct. J'ai donc ajouté un facteur pour que tout fonctionne. L'équation finale :

$$\frac{600}{fSample \cdot \frac{8}{fDiv}} \cdot 1.7 \quad (3)$$

## 6 Tests

### 6.1 A

A1	Title :	Check if ADC Signal acquisition is working
Group :	Signal Acquisition	
Description :	Is buffer continuously feed with new ADC measures ?	
Initial Conditions :	Running program	
Test Results :	The buffer is continuously feed with new ADC measures.	
Final Conditions :	Running program	
Comments :		
Test Passed :	Yes	

A2	Title :	Check TIM1 continously trigger the ADC
Group :	Signal Acquisition	
Description :	Not software trigger, nor ADC continous conversion mode	
Initial Conditions :	Running program	
Test Results :	Yes, the ADC conversion is started directly from the TIM1 timeout signal.	
Final Conditions :	Running program	
Comments :		
Test Passed :	Yes	

A3	Title :	Check TIM1 trigger frequency
Group :	Signal Acquisition	
Description :	Is TIM1 trigger frequency as expected ?	
Initial Conditions :	-	
Test Results :	Test can't be done. No oscilloscope available	
Final Conditions :	-	
Comments :		
Test Passed :	-	

## 6.2 B

<b>B1</b>	<b>Title :</b>	<b>Check Sinus wave can be correctly measured</b>
Group :	Signal Frequencies	
Description :	50 Hz	
Initial Conditions :	Sampling rate : 100kSample/s, Input freq : 50Hz	
Test Results :	Test can't be done. No Sinus generator for 50Hz	
Final Conditions :	-	
Comments :		
Test Passed :	-	

<b>B2</b>	<b>Title :</b>	<b>Check Sinus wave can be correctly measured</b>
Group :	Signal Frequencies	
Description :	370 Hz	
Initial Conditions :	Sampling rate : 100kSample/s, Input freq : 370Hz	
Test Results :	The sinus is correctly displayed. The input period matches with de divisions	
Final Conditions :	-	
Comments :		
Test Passed :	Yes	

<b>B3</b>	<b>Title :</b>	<b>Check Sinus wave can be correctly measured</b>
Group :	Signal Frequencies	
Description :	500 Hz	
Initial Conditions :	Sampling rate : 100kSample/s, Input freq : 500Hz	
Test Results :	The sinus is correctly displayed. The input period matches with de divisions	
Final Conditions :		
Comments :		
Test Passed :	Yes	

<b>B4</b>	<b>Title :</b>	<b>Check Sinus wave can be correctly measured</b>
Group :	Signal Frequencies	
Description :	700 Hz	
Initial Conditions :	Sampling rate : 100kSample/s, Input freq : 700Hz	
Test Results :	The sinus is correctly displayed. The input period matches with de divisions	
Final Conditions :		
Comments :		
Test Passed :	Yes	

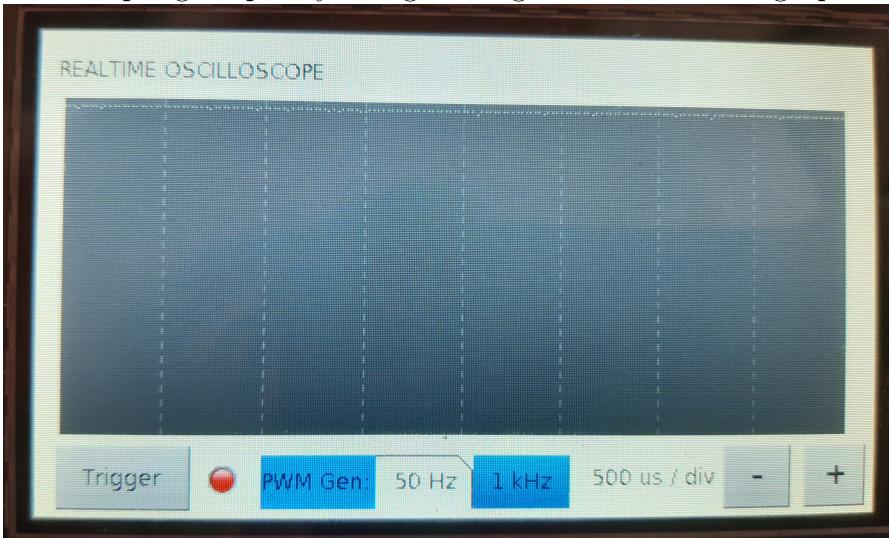
### 6.3 C

<b>C1</b>	<b>Title :</b>	<b>Check maximum sampling rate</b>
Group :	Maximum Sampling Rate	
Description :	Sample singnal with 10 k samples/s	
Initial Conditions :	No DMA, Sampling rate : 10kSamples/s	
Test Results :	The sampling works	
Final Conditions :		
Comments :		
Test Passed :	Yes	

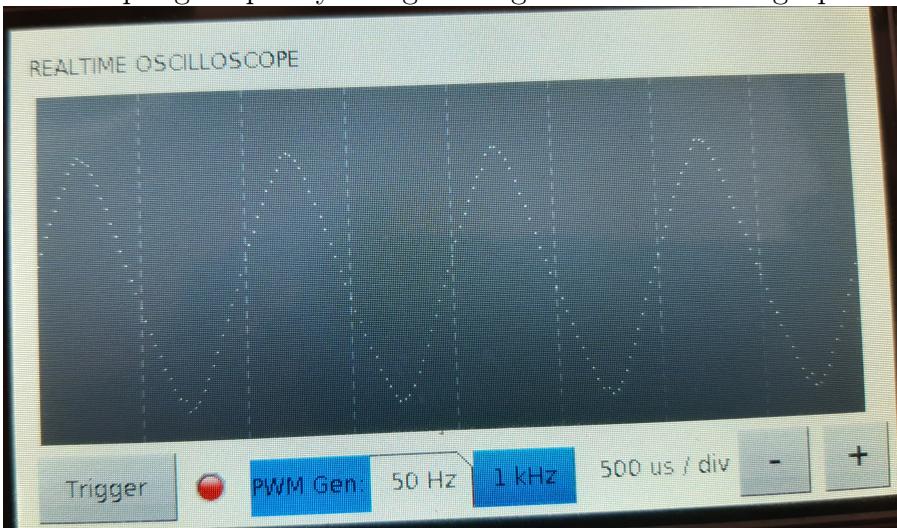
<b>C2</b>	<b>Title :</b>	<b>Check maximum sampling rate</b>
Group :	Maximum Sampling Rate	
Description :	Sample singnal with 100 k samples/s	
Initial Conditions :	No DMA, Sampling rate : 100kSamples/s	
Test Results :	The sampling works	
Final Conditions :		
Comments :		
Test Passed :	Yes	

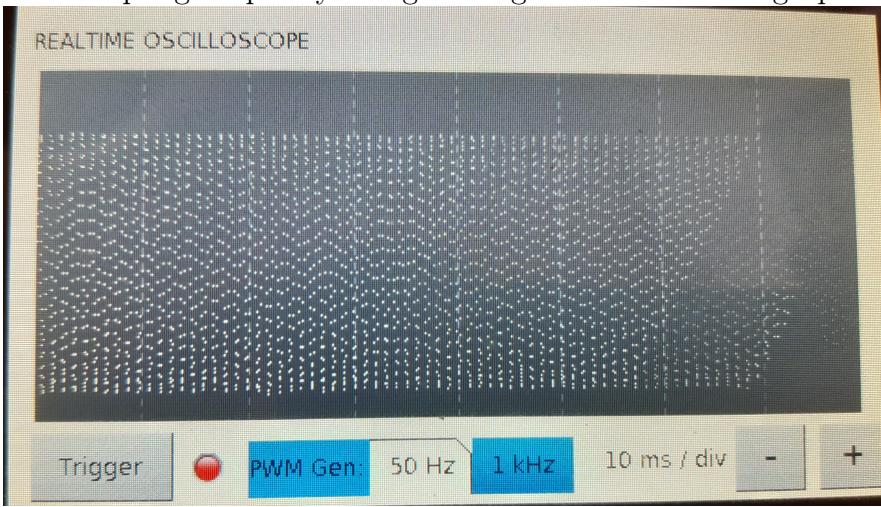
<b>C3</b>	<b>Title :</b>	<b>Check maximum sampling rate</b>
Group :	Maximum Sampling Rate	
Description :	Sample singnal with 1 M samples/s	
Initial Conditions :	No DMA, Sampling rate : 1MSamples/s	
Test Results :	The sampling doesn't work. The uC is fully used for the ADC conversion.	
Final Conditions :		
Comments :		
Test Passed :	No	

## 6.4 D

D1	Title :	Check if Sinus wave can be correctly visualised
	Group :	Signal Display
	Description :	50 Hz Signal with 500 us/div time base
	Initial Conditions :	Sampling rate : 100kSamples/s The sampling frequency is high enough. There are enough points.
	Test Results :	
	Final Conditions :	
	Comments :	
	Test Passed :	Yes

<b>D2</b>	<b>Title :</b>	Check if Sinus wave can be correctly visualised
Group :	Signal Display	
Description :	50 Hz Signal with 10 ms/div time base	
Initial Conditions :	Sampling rate : 100kSamples/s	
	The sampling frequency is high enough. There are enough points.	
Test Results :		
Final Conditions :		
Comments :		
Test Passed :	Yes	

D3	Title :	Check if Sinus wave can be correctly visualised
	Group :	Signal Display
	Description :	1 kHz Signal with 500 us/div time base
	Initial Conditions :	Sampling rate : 100kSamples/s
	The sampling frequency is high enough. There are enough points.	
		
	Test Results :	
	Final Conditions :	
	Comments :	
	Test Passed :	Yes

<b>D4</b>	<b>Title :</b>	Check if Sinus wave can be correctly visualised
Group :	Signal Display	
Description :	1 kHz Signal with 10 ms/div time base	
Initial Conditions :	Sampling rate : 100kSamples/s	
Test Results :	The sampling frequency is high enough. There are enough points. 	
Final Conditions :		
Comments :		
Test Passed :	Yes	

## 6.5 E

<b>E1</b>	<b>Title :</b>	Check maximum display refresh rate
Group :	Display Refresh Rate	
Description :	Display refresh rate : 50 Hz	
Initial Conditions :	Display refresh rate : 50 Hz	
Test Results :	The refresh is correctly done	
Final Conditions :		
Comments :		
Test Passed :	Yes	

<b>E2</b>	<b>Title :</b>	<b>Check maximum display refresh rate</b>
Group :	Display Refresh Rate	
Description :	Display refresh rate : 66 Hz	
Initial Conditions :	Display refresh rate : 66 Hz	
Test Results :	The refresh is correctly done	
Final Conditions :		
Comments :		
Test Passed :	Yes	

<b>E3</b>	<b>Title :</b>	<b>Check maximum display refresh rate</b>
Group :	Display Refresh Rate	
Description :	Display refresh rate : 100 Hz	
Initial Conditions :	Display refresh rate : 100 Hz	
Test Results :	The refresh is correctly done	
Final Conditions :		
Comments :	There are no difference between all these tests. The measured duration of a refresh is 22 ms. So the uC is refreshing all the time as fast as possible, even with the 50 Hz refresh rate.	
Test Passed :	Yes	

## 6.6 F

<b>F1</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Time Divisions Display	
Description :	From 500 us/div to 1 ms/div	
Initial Conditions :	Input signal : square 1kHz, Div : 500 us	
Test Results :	The signal changes accordingly	
Final Conditions :	Div : 1 ms	
Comments :		
Test Passed :	Yes	

<b>F2</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Time Divisions Display	
Description :	From 1 ms/div to 2 ms/div	
Initial Conditions :	Input signal : square 1kHz, Div : 1 ms	
Test Results :	The signal changes accordingly	
Final Conditions :	Div : 2 ms	
Comments :		
Test Passed :	Yes	

<b>F3</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Description :	Time Divisions Display
Description :	Initial Conditions :	From 2 ms/div to 5 ms/div
Initial Conditions :	Test Results :	Input signal : square 1kHz, Div : 2 ms
Test Results :	Final Conditions :	The signal changes accordingly
Final Conditions :	Comments :	Div : 5 ms
Comments :	Test Passed :	
Test Passed :		Yes

<b>F4</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Description :	Time Divisions Display
Description :	Initial Conditions :	From 5 ms/div to 10 ms/div
Initial Conditions :	Test Results :	Input signal : square 50 Hz, Div : 5 ms
Test Results :	Final Conditions :	The signal changes accordingly
Final Conditions :	Comments :	Div : 10 ms
Comments :	Test Passed :	
Test Passed :		Yes

<b>F5</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Description :	Time Divisions Display
Description :	Initial Conditions :	From 10 ms/div back to 5 ms/div
Initial Conditions :	Test Results :	Input signal : square 50 Hz, Div : 10 ms
Test Results :	Final Conditions :	The signal changes accordingly
Final Conditions :	Comments :	Div : 5 ms
Comments :	Test Passed :	
Test Passed :		Yes

<b>F6</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Description :	Time Divisions Display
Description :	Initial Conditions :	From 5 ms/div back to 2 ms/div
Initial Conditions :	Test Results :	Input signal : square 50 Hz, Div : 5 ms
Test Results :	Final Conditions :	The signal changes accordingly
Final Conditions :	Comments :	Div : 2 ms
Comments :	Test Passed :	
Test Passed :		Yes

<b>F7</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Time Divisions Display	
Description :	From 2 ms/div to 1 ms/div	
Initial Conditions :	Input signal : square 1kHz, Div : 2 ms	
Test Results :	The signal changes accordingly	
Final Conditions :	Div : 1 ms	
Comments :		
Test Passed :	Yes	

<b>F8</b>	<b>Title :</b>	<b>Check if changes of x-axis scaling take effect on signal view</b>
Group :	Time Divisions Display	
Description :	From 1 ms/div to 500 us/div	
Initial Conditions :	Input signal : square 1kHz, Div : 1 ms	
Test Results :	The signal changes accordingly	
Final Conditions :	Div : 500 us	
Comments :		
Test Passed :	Yes	

## 6.7 G

<b>G1-5</b>	<b>Title :</b>	<b>Check if when enabling signal triggering, the periodic signal stays still on the display</b>
Group :	Trigger Signal Display	
Description :	Enable the triggering, and change the time base	
Initial Conditions :	Input signal : 50Hz square, Div : 500us	
Test Results :	The periodic signal stay still on the display, but sometimes it moves just for 1 frame.	
Final Conditions :	Div : 10 ms	
Comments :		
Test Passed :	Partially	

<b>G6-10</b>	<b>Title :</b>	<b>Check if when enabling signal triggering, the periodic signal stays still on the display</b>
	<b>Group :</b>	Trigger Signal Display
	<b>Description :</b>	Enable the triggering, and change the time base
	<b>Initial Conditions :</b>	Input signal : 1kHz triangle, Div : 500us
	<b>Test Results :</b>	The periodic signal stay still on the display, but sometimes it moves just for 1 frame.
	<b>Final Conditions :</b>	Div : 10 ms
	<b>Comments :</b>	
	<b>Test Passed :</b>	Partially

<b>G11-15</b>	<b>Title :</b>	<b>Check if when enabling signal triggering, the periodic signal stays still on the display</b>
	<b>Group :</b>	Trigger Signal Display
	<b>Description :</b>	Enable the triggering, and change the time base
	<b>Initial Conditions :</b>	Input signal : 1kHz sinus, Div : 500us
	<b>Test Results :</b>	The periodic signal stay still on the display, but sometimes it moves just for 1 frame.
	<b>Final Conditions :</b>	Div : 10 ms
	<b>Comments :</b>	
	<b>Test Passed :</b>	Partially

## 6.8 L

<b>L1</b>	<b>Title :</b>	<b>Does the software hang after a while ?</b>
	<b>Group :</b>	Long-Term Test
	<b>Description :</b>	Check if the system is running without user interaction for about 5 minutes
	<b>Initial Conditions :</b>	With RTOS
	<b>Test Results :</b>	The system still run correctly
	<b>Final Conditions :</b>	
	<b>Comments :</b>	
	<b>Test Passed :</b>	Yes

<b>L2</b>	<b>Title :</b>	<b>Does the software hang after a while ?</b>
Group :	Long-Term Test	
Description :	Check if system is running with continuous user interaction for about 5 minutes	
Initial Conditions :	With RTOS	
Test Results :	The system still run correctly	
Final Conditions :		
Comments :		
Test Passed :	Yes	

<b>L3</b>	<b>Title :</b>	<b>Does the software hang after a while ?</b>
Group :	Long-Term Test	
Description :	Check if system is running for more than 6 hours	
Initial Conditions :	With RTOS	
Test Results :	The system still run correctly	
Final Conditions :		
Comments :		
Test Passed :	Yes	

## 6.9 Summary

<b>Total Tests :</b>	<b>31</b>
Tests passed :	27
Tests partially passed :	3
Tests not passed :	1

## 7 Conclusion

Le but de ce projet était de réaliser un oscilloscope en temps réel. On peut dire que l'objectif est atteint car la majorité des tests ont été passés. Ce labo m'a permis de mieux comprendre les notions de temps réel abordées en cours, et de mieux maîtriser des outils tel que cubeMX ou encore System Workbench for STM32.