

# TP2

Julian BARKOUDEH EI2I3-II-A

## Table des matières

<b>Tutoriels STM32F0-Peripherals Standards – DAC .....</b>	<b>2</b>
4) Génération de signaux sinusoïdaux .....	2
<b>Tutoriels STM32F0-Interruptions - Interruptions Externes.....</b>	<b>2</b>
2.3) Exemple de scrutation bloquante .....	2
2.4) Exemple de scrutation Non bloquante .....	3
3.2) Configuration du contrôleur d'interruption NVIC.....	4
3.3) Implémentation du gestionnaire d'interruption ou interrupt handler (ISR).....	5
3.4) Gestion du délai d'exécution des ISR .....	5
<b>Tutoriels STM32F0-Interruptions - Interruptions Timer .....</b>	<b>6</b>
<b>Tutoriels STM32F0-Application Filtre Numérique .....</b>	<b>7</b>
3.4) Implémentation du suiveur.....	7
4.1) Arithmétique virgule flottante .....	7
4.2) Arithmétique virgule fixe .....	8

#### 4) Génération de signaux sinusoïdaux

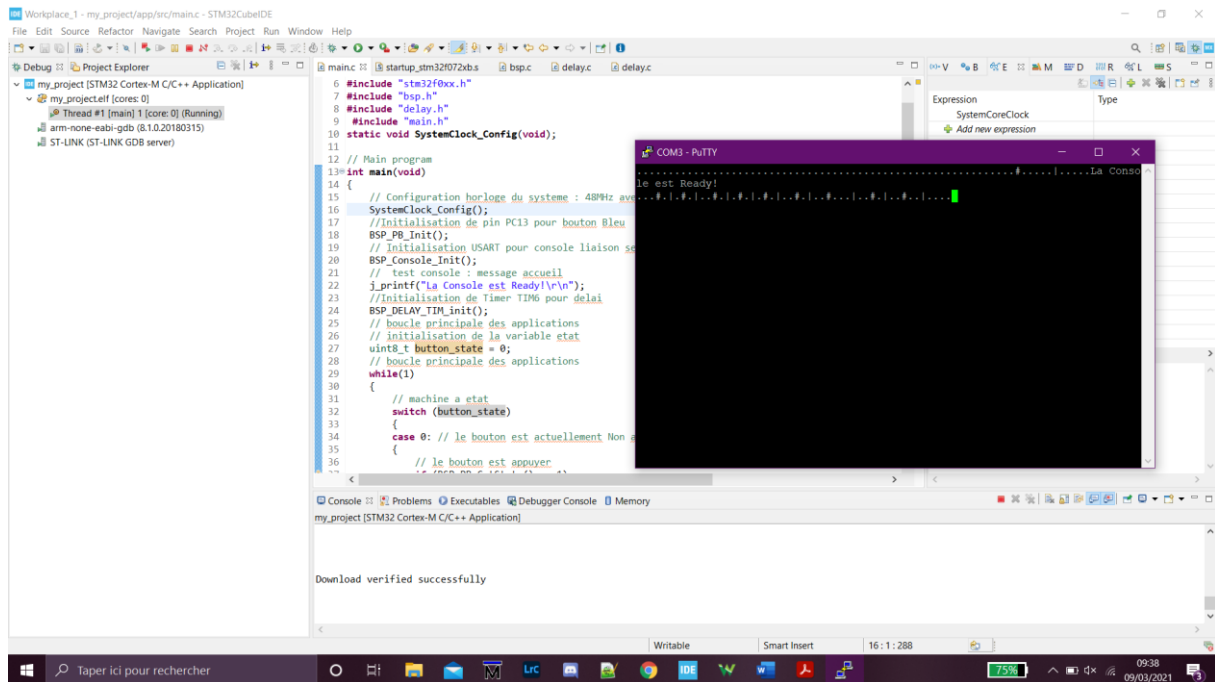


### 2.3) Exemple de scrutation bloquante

Download verified successfully

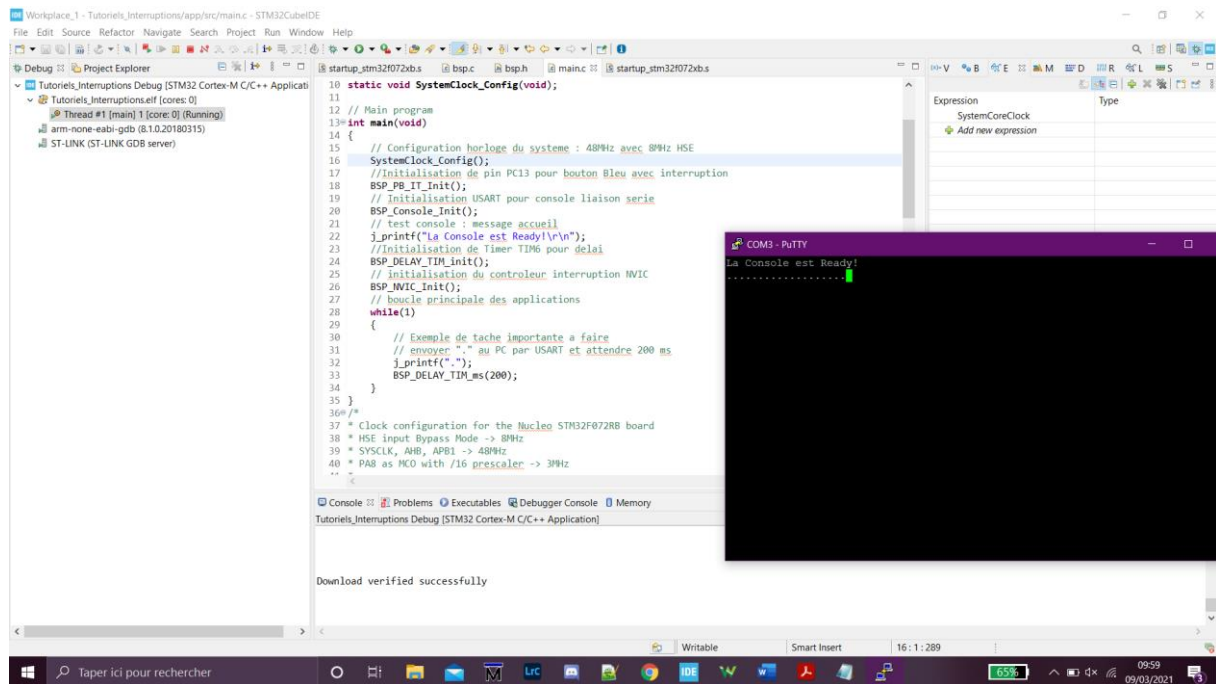
- ➔ Le programme n'affiche qu'une fois « |#. » avec l'appui et le relâchement du bouton à cause de l'utilisation des boucles while. En effet le programme reste dans les boucles While tant que le bouton est appuyé ou non appuyé et n'exécute pas le reste du programme.

## 2.4) Exemple de scrutation Non bloquante

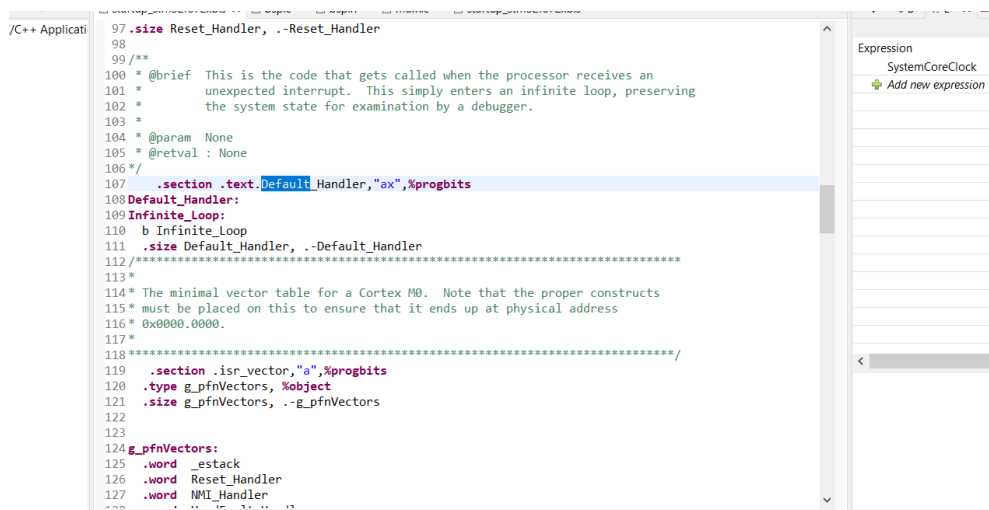


-> On remarque qu'en changeant la valeur de x de 200 à 500 et après à 1000, que l'affichage des points se passe plus lentement. En effet, avec x plus grand moins de d'appuis de boutons sont détectés.

### 3.2) Configuration du contrôleur d'interruption NVIC



On remarque qu'en appuyant sur le bouton l'exécution du programme est interrompue, et ne peut pas reprendre qu'en appuyant sur Reset.



The screenshot displays an IDE with two windows:

- Main Editor:** Contains C code for configuring the STM32F072RB. The code includes comments in French and English, defining pin numbers, initializing the UART, setting up a timer delay, and entering a loop where it sends a message over the console.

```

10 static void SystemClock_Config(void);
11
12 // Main program
13 int main(void)
14 {
15     // Configuration horloge du systeme : 48MHz avec 8MHz HSE
16     SystemClock_Config();
17     //Initialisation de pin PC13 pour bouton Bleu avec interruption
18     BSP_PB_Init();
19     // Initialisation USART pour console liaison
20     BSP_Console_Init();
21     // test console : message accueil
22     j_printf("La Console est Ready!\r\n");
23     //Initialisation de Timer TIM6 pour delai
24     BSP_DELAY_TIM_init();
25     // Initialisation du controleur interruption
26     BSP_NVIC_Init();
27     // boucle principale des applications
28     while(1)
29     {
30         // Exemple de tache importante a faire
31         // envoyer "." au PC par USART et attendre
32         j_printf(".");
33         BSP_DELAY_TIM_ms(200);
34     }
35 }
36 /*
37 * Clock configuration for the Nucleo STM32F072RB
38 * HSE input Bypass Mode -> 8MHz
39 * SYSCCLK, AHB, APB1 -> 48MHz
40 * PAB as MCO with /16 prescaler -> 3MHz

```

- COM3 - PuTTY:** Shows the output of the program. It displays "La Console est Ready!" followed by a series of dots, indicating the successful execution of the code.

On remarque qu'en augmentant  $x$  le délai d'interruption est plus long.

The screenshot displays the STM32CubeIDE environment. The main editor window shows the following C code:

```

25  BSP_DELAY_TIM_init();
26  // initialisation du controleur interruption NVIC
27  BSP_NVIC_Init();
28  // boucle principale des applications
29  while(1)
30  {
31      // Traitement de la tâche liée à la détection du front descendant du bouton
32      if (button_irq == 1)
33      {
34          j_printf("#");
35          button_irq = 0;
36      }
37      // Exemple de tâche importante à faire
38      // envoyer "." au PC par USART et attendre
39      j_printf(".");
40      BSP_DELAY_TIM_ms(200);
41  }
42  }
43  /*
44  * Clock configuration for the Nucleo STM32F072RB
45  * HSE input Bypass Mode -> 8MHz
46  * SYSCLK, AHB, APB1 -> 48MHz
47  * PA8 as MCO with /16 prescaler -> 3MHz
48  *
49  */
50  /*
51  * Clock configuration for the Nucleo STM32F072RB
52  * HSE input Bypass Mode -> 8MHz
53  * SYSCLK, AHB, APB1 -> 48MHz
54  * PA8 as MCO with /16 prescaler -> 3MHz
55  */

```

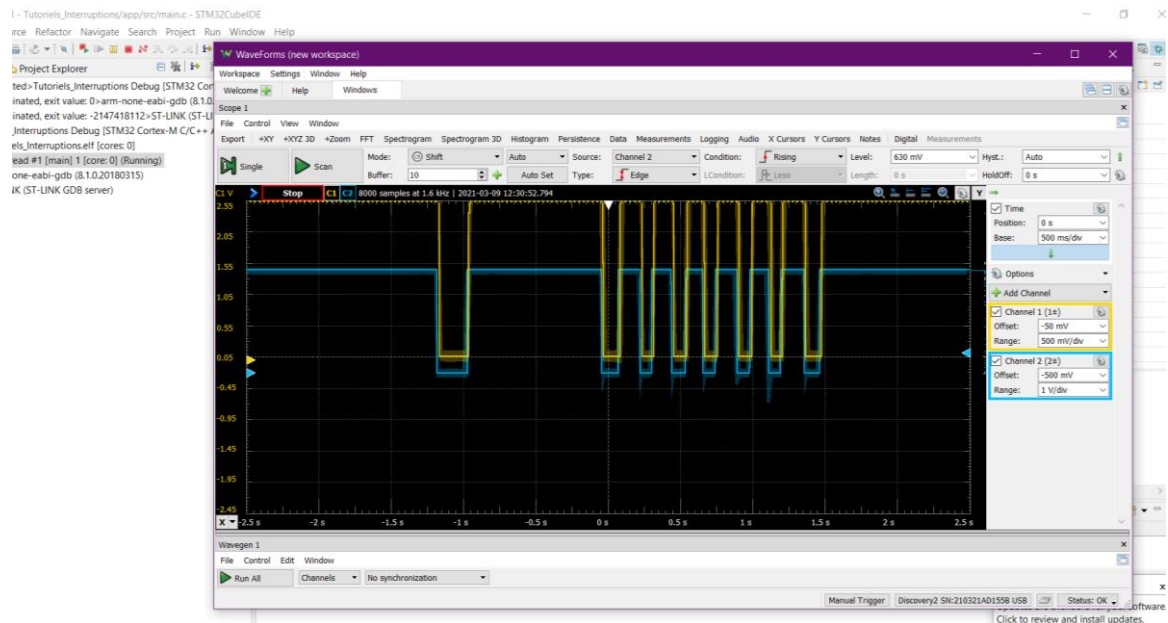
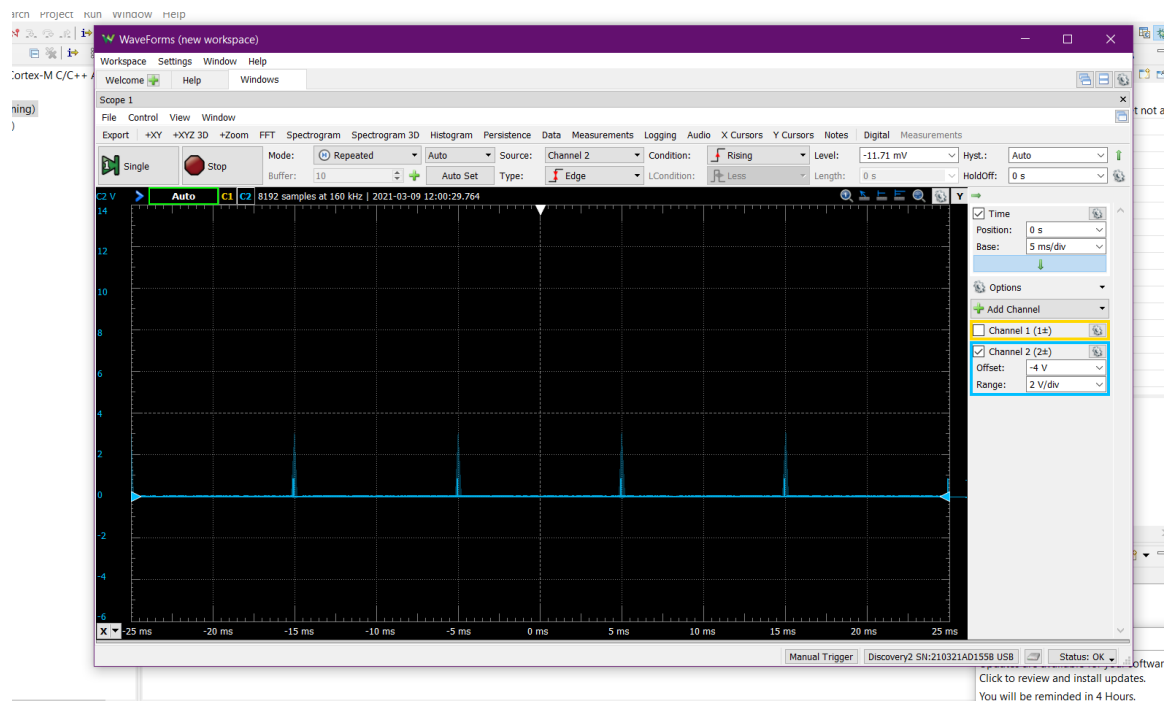
On the right, the 'Expression' pane shows 'SystemCoreClock' with a value of 'Error: Target not av'. Below it, a 'COM3 - PuTTY' window displays the console output: 'La Console est Ready!' followed by a series of dots and a green cursor.

At the bottom, the 'Console' pane shows the text: 'Tutoriels\_Interruptions Debug [STM32 Cortex-M C/C++ Application]'.

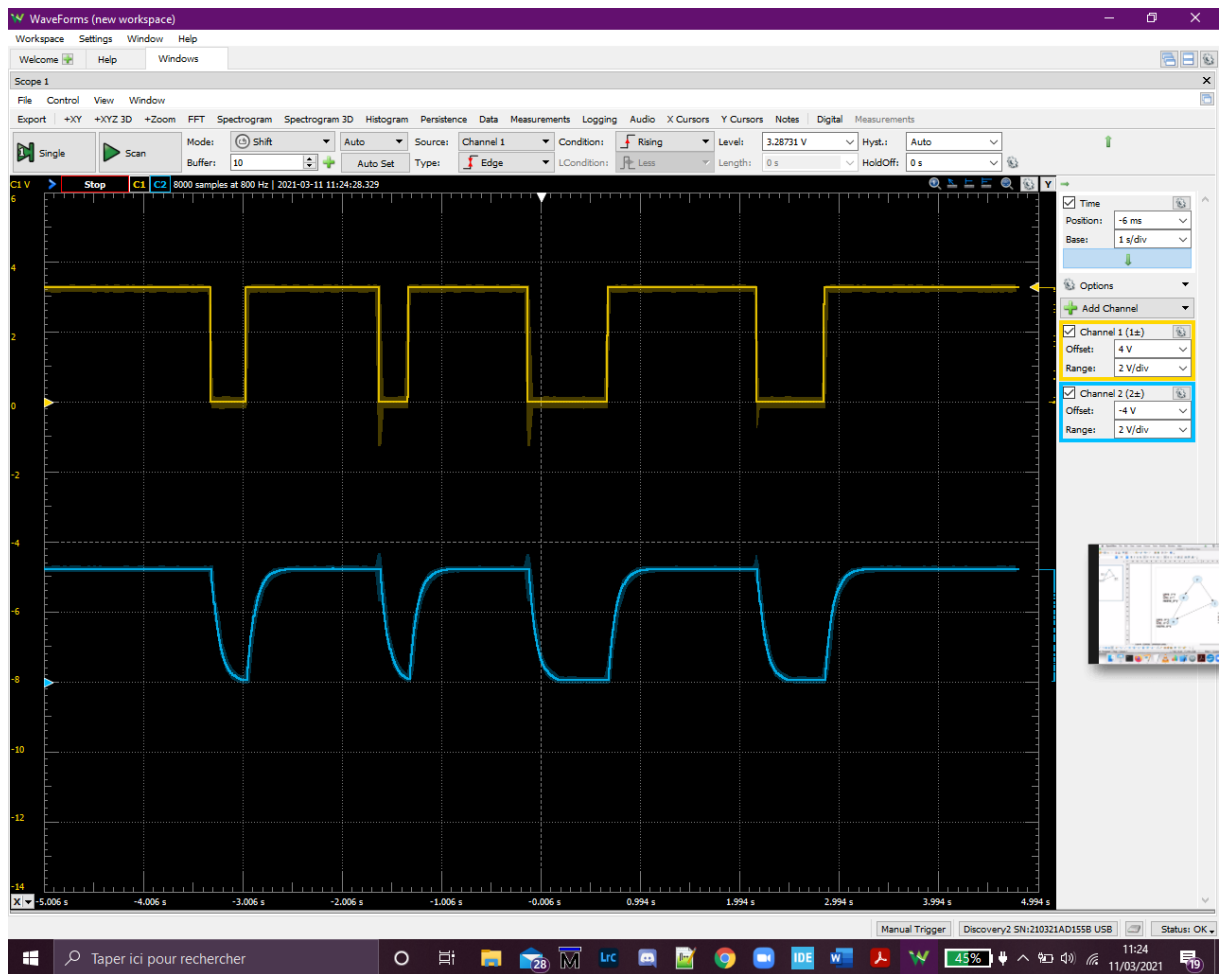


# Tutoriels STM32F0-Application Filtre Numérique

## 3.4) Implémentation du suiveur



## 4.1) Arithmétique virgule flottante

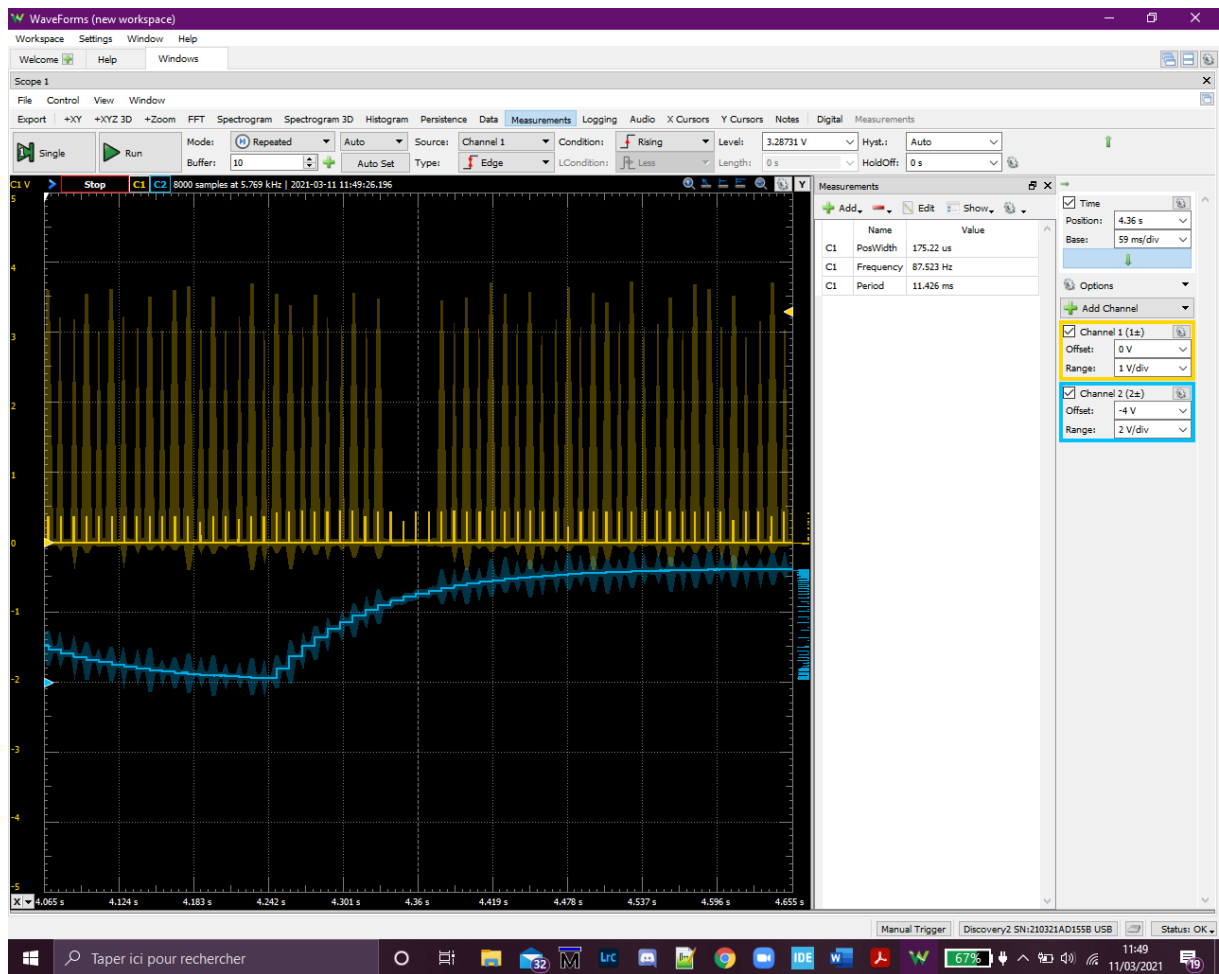


Avec Channel 1 : Pc1 et channel2 : PA4.

On mesure bien le temps de montée de la courbe de channel 2 à :  $3.09 - 2.853 = 0.237$  s.

#### 4.2) Arithmétique virgule fixe





Avec Channel 1 : PA4 et Channel 2 : PA5