# **Test of Delay between Two PWMs**

## **Delay between Two PWMs**

### **Condition**

1. Two PWMs have the same frequency and pulse width

### **Description**



1. p1 = p2 and w1 = w2
2. Pulse of PWM 1 is output and after delay, d, pulse of PWM2 is output
3. w1(or w2) + d < p1(or p2)/2, so w1+d is less than duty ratio 50% of PWM1 (or PWM2)

## **Test Conditions**

### **Target**

1. Search how to implement delay between two PWMs
2. Compare with features of implementation methods

### **Hardware Conditions**

1. STM32 NUCLEO-L412RB-P. MCU model is STM32L412RBP
2. System clock is 80MHz, timer counter resolution is 1 MHz (prescaler is 80)
3. PWM frequency is 100Hz (period is 10ms), and duty ratio is 10 % (pulse width is 1 ms)

### **Software Conditions**

* STM32CubeIDE Version: 1.10.1

### **References**

1. RM0394 Reference manual (STM32L41xxx/42xxx/43xxx/44xxx/45xxx/46xxx advanced Arm®-based 32-bit MCUs)
2. STM32L412xx datasheet
3. STM32 NUCLEO-L412RB-P schematic
4. User manual STM32 NUCLEO-L412RB-P

## **Method 1**

### **Concepts**



1. PWM1: TIM2 CH1
2. OC2: Output compare of TIM2 CH2. Pulse could not be pulse in really.
3. PWM2: TIM1 CH1
4. TIM1 and TIM2 are synchronized timers, TIM2 is master and TIM1 is slave. Trigger of TIM1 start is OC2 matching

### **Description**

1. TIM1 and TIM2 are independent
2. PWM1 and PWM2 are configure as the same conditions (frequency and duty ratio)
3. Matching time of OC2 is w1 + d
4. Delay is handled by HW

## **Method 2**

### **Concepts**



1. PWM1: TIM2 CH1
2. OC2: Output compare of TIM2 CH2. Pulse could not be pulse in really.
3. PWM2: TIM1 CH1
4. Start PWM2 in interrupt handler of TIM1 start is OC2 matching

### **Description**

1. TIM1 and TIM2 are independent
2. PWM1 and PWM2 are configure as the same conditions (frequency and duty ratio)
3. Matching time of OC2 is w1 + d
4. Delay is handled by SW

## **Method 3**

### **Concepts**



1. PWM1: TIM2 CH1
2. OC2 and PWM2: TIM2 CH1
3. Compare mode of OC2 is toggle mode
4. OC2 is used as PWM2. OC2 is not PWM accurately but PWM pulses are output by SW

### **Description**

1. TIM2 is used only. and OC2 interrupt is used
2. When start (Ps), OC2 match time is w1+d.
3. At first matching (P1), OC2 match time is w2+d.
4. At second matching (p2), OC2 match time is w1+d. After update event, TIM2 counter is start from 0, so P1 event occurs and next P2.

## **Method 4**

### **Concepts**

* Concepts and configurations are the same as Method 3. But DMA is used instead interrupt to handle OC2 matchings

## **Comparison of Methods**

### **Pros and Cons**

|  |  |  |
| --- | --- | --- |
| **Method** | **Advantage** | **Disadvantage** |
| 1 | 1. Controlled by HW | 1. Settings are complex and not intuitive  2. Pulse width and delay cannot be changed during PMWs run. |
| 2 | 1. Concept is simple | 1. Pulse width and delay cannot be changed during PMWs run.  2. Controlled by SW, so process delay must be considered |
| 3 | 1. Pulse width or delay can be changed during PMWs run.  2. Controlled by SW, but process delay can be ignored | 1. Concept is complex and not intuitive  2. Deep consideration is required to handle process  3. Two interrupts are required every period of PWM |
| 4 | 1. Pulse width or delay can be changed during PMWs run. | 1. Concept is complex and not intuitive  2. Deep consideration is required to handle process |

### **Resource Required**

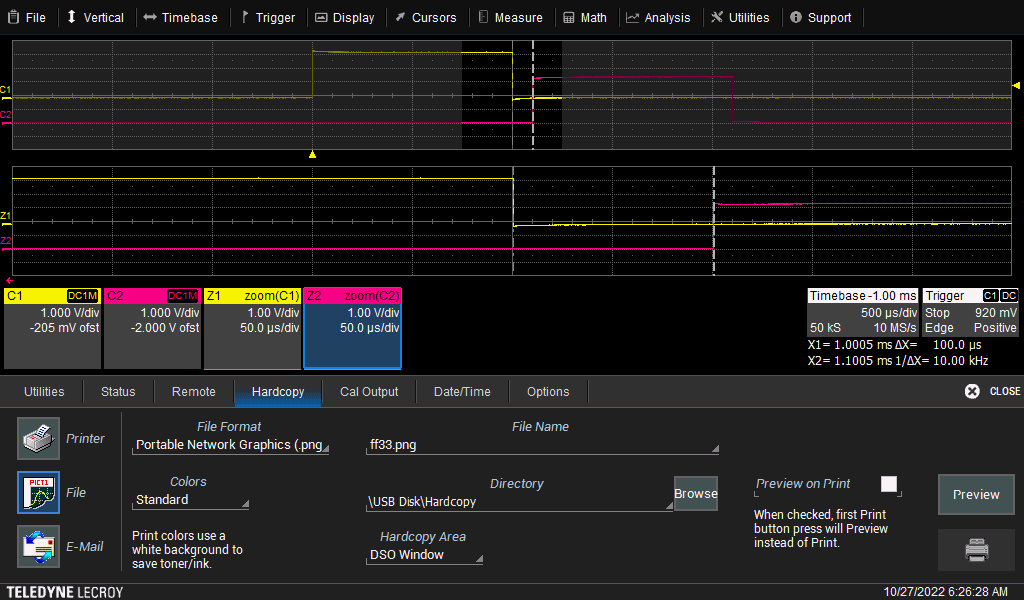
|  |  |
| --- | --- |
| **Method** | **Resources Required** |
| 1 | 2 Timers and 2 GPIOs |
| 2 | 2 Timers and 2 GPIOs |
| 3 | 1 Timer and 2 GPIOs |
| 4 | 1 Timer and 2 GPIOs |

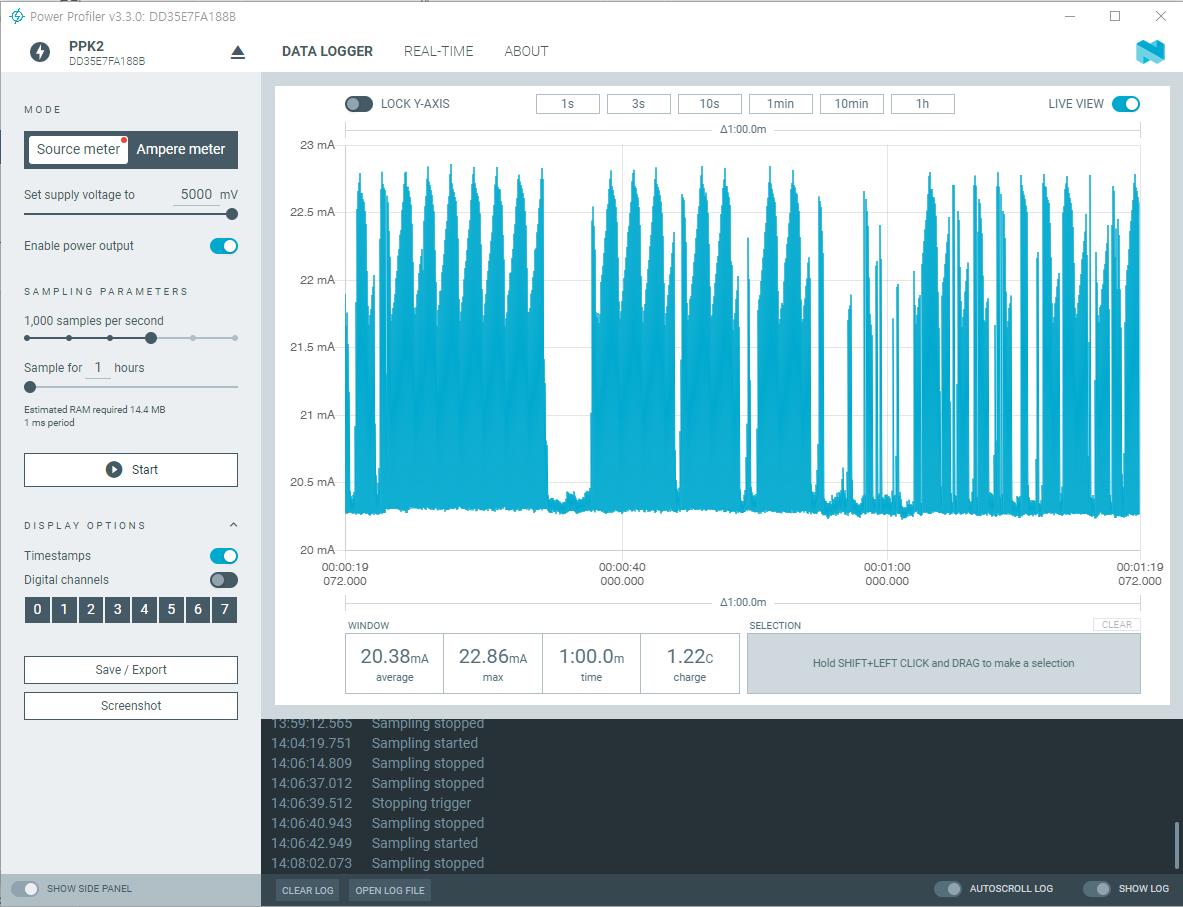
### **Process Delay Time and Power Consumption**

|  |  |  |
| --- | --- | --- |
| **Method** | **Process Delay (us)** | **Power Consumption (AVG mA)** |
| 1 | 0 | 20.38 |
| 2 | 2.5 | 18.03 |
| 3 | 0 | 19.92 |
| 4 | 0 | 19.91 |

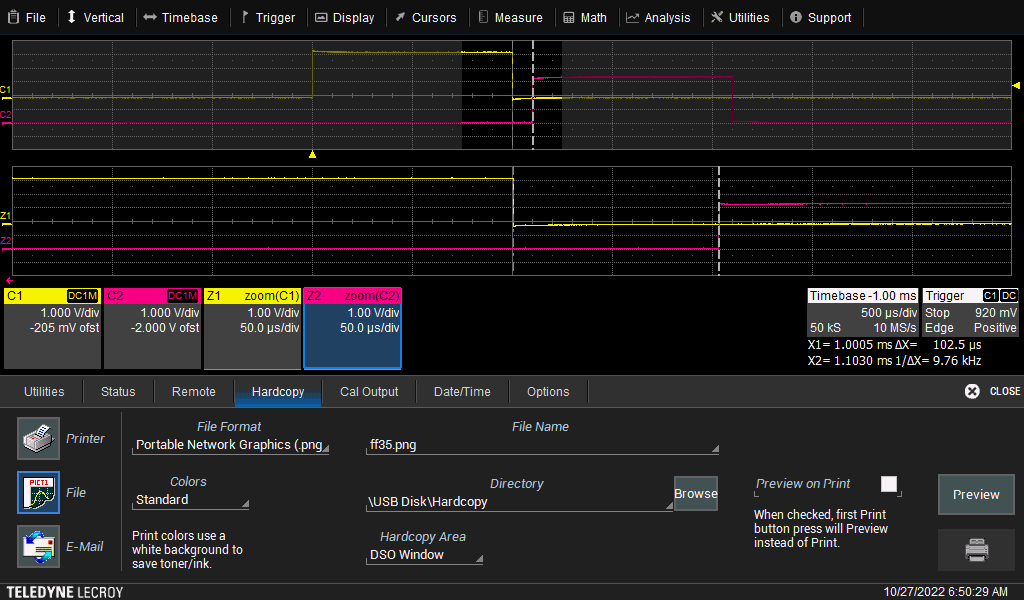
※ 1 LED is used

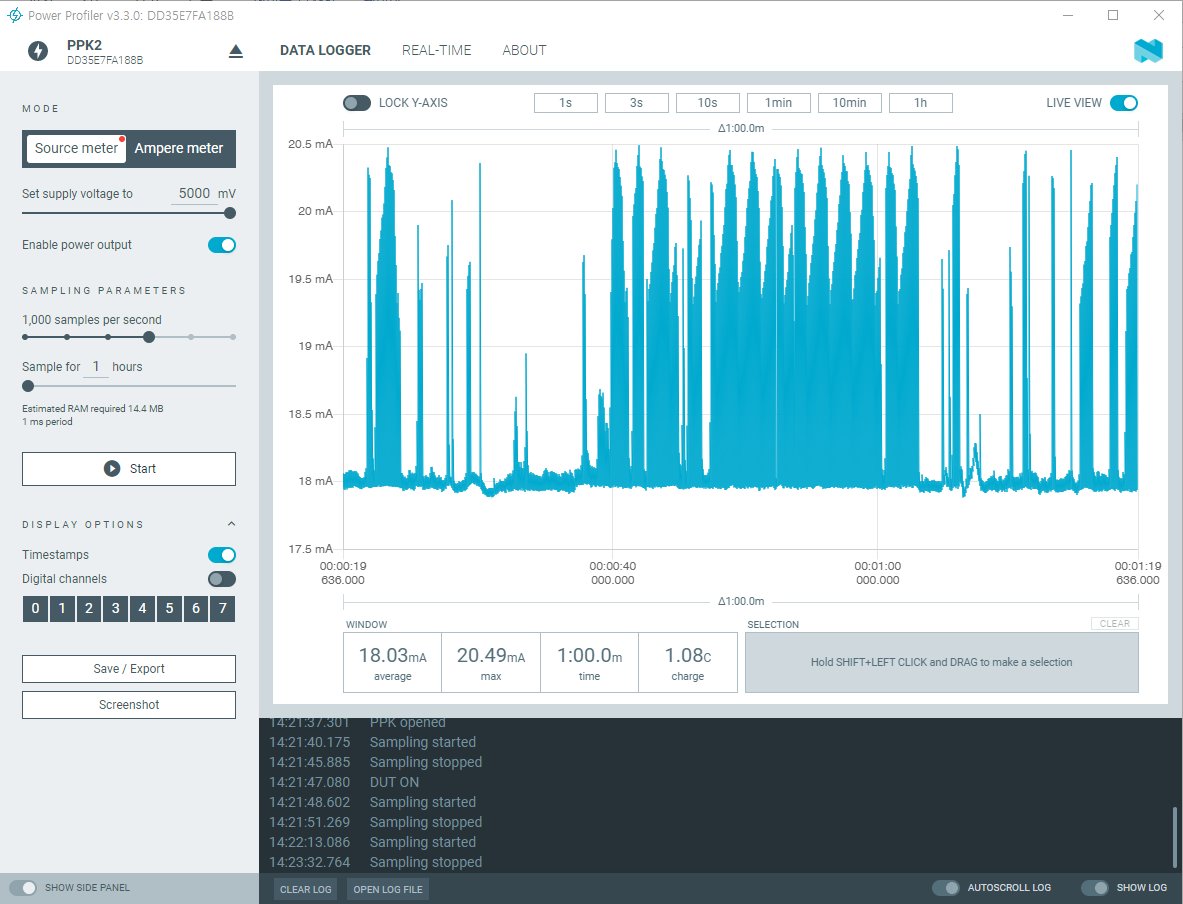
### **Reference – Method 1**



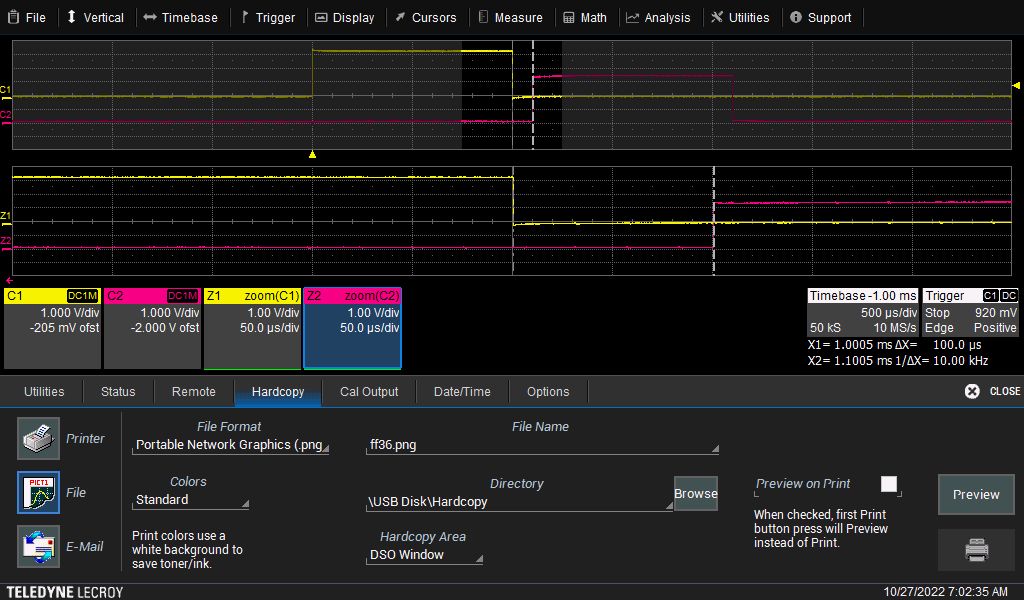


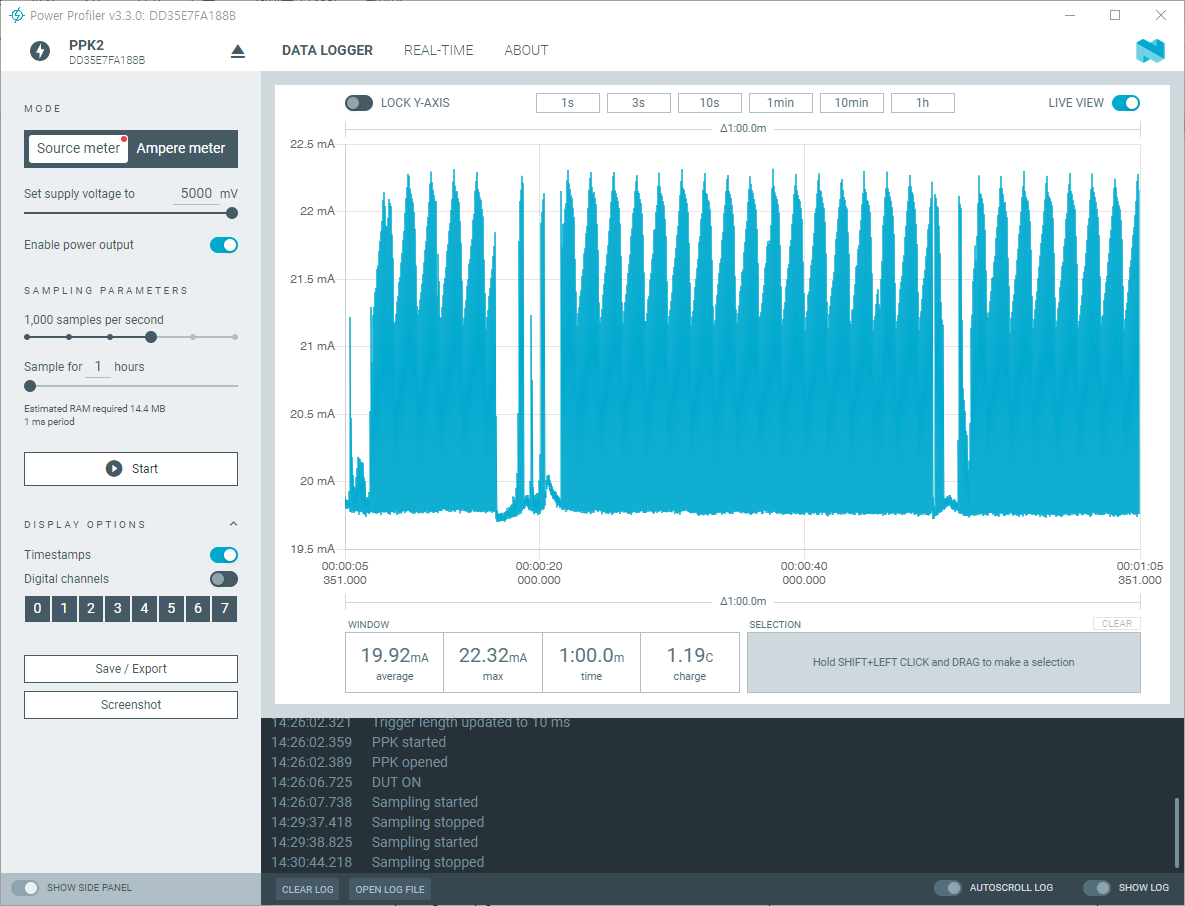
### **Reference – Method 2**

****



### **Reference – Method 3**





### **Reference – Method 4**

