EIE3810 Microprocessor System Design Laboratory

Laboratory Report #5

Name: Tengfei Ma

Student ID: 121090406

Date: Nov. 24, 2023

The Chinese University of Hong Kong, Shenzhen

- Experiment 1: Using TIM3 and interrupt handler
- Experiment 2: Using TIM4 and TIM3 together
- Experiment 3: Setup SYSTICK with a 10ms periodic tick
- Experiment 4: Drive an LED with PWM
- Experiment 5: Read JOYPAD key input based on TIM3

1. **Experiment 1**

1.1 Result

figure1: TIM3 controls DS0 to have a flashing rate of 1Hz.

figure2: Change DS0 to have a flashing rate of 5Hz.

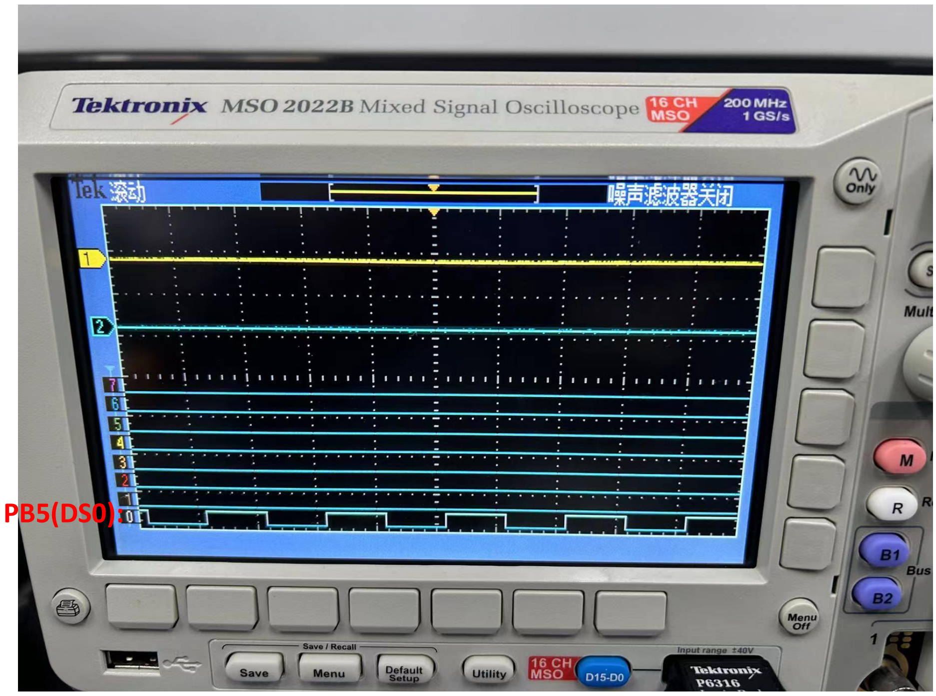

figure 1 DS0~1Hz

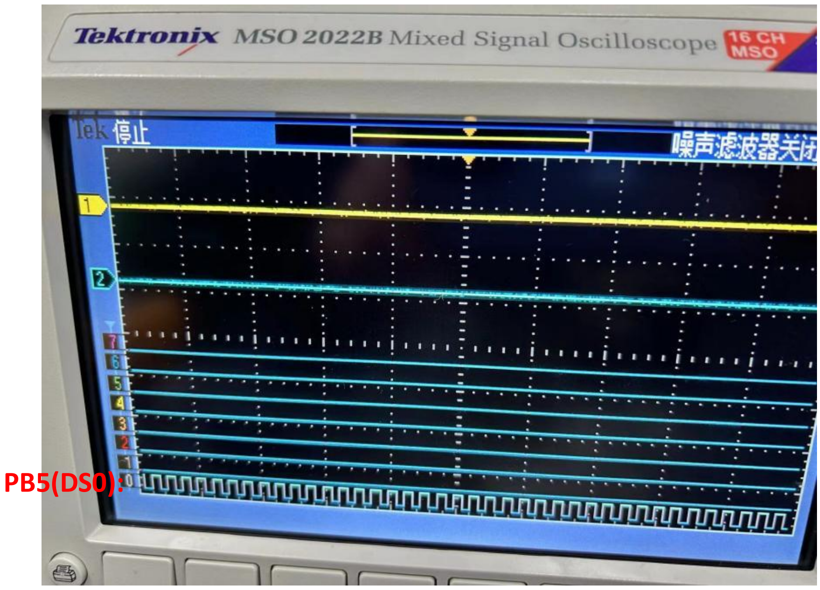

figure 2 DS0~5Hz

1.2 Questions

1.2.1 What is the relation between the LED flashing period, pre-scalar value, and clock counter value?

$$\frac{1}{LED\ flashing\ period} = \frac{system\ clock\ frequency}{pre\_scalar\_value \times clock\ counter\ value},$$

pre_scalar_value = psc+1 ,

clock counter value = arr+1

2. Experiment 2

2.1 Result

figure3: Set up TIM4 to control DS1 to have a flashing rate of 4Hz, and TIM3 still controls DS0 to have a flashing rate of 1Hz.

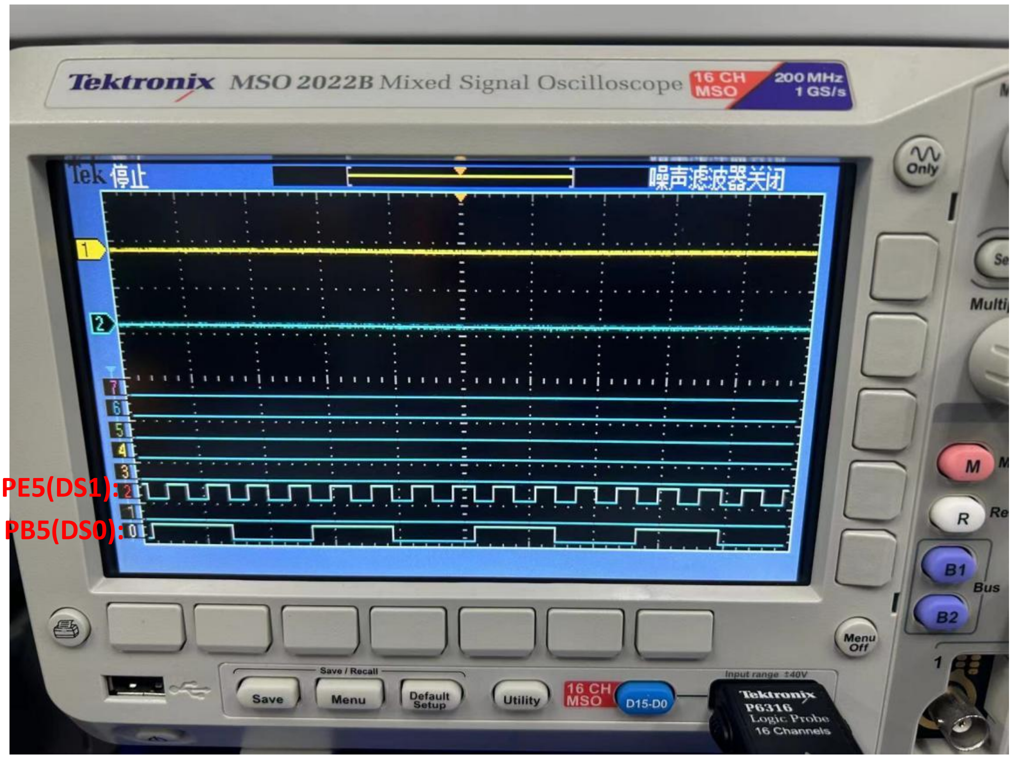

figure 3 DS0~1Hz, DS1~4Hz

figure 4: Keep DS0 flashing at 1Hz, change DS1 to flash oppositely with DS0.

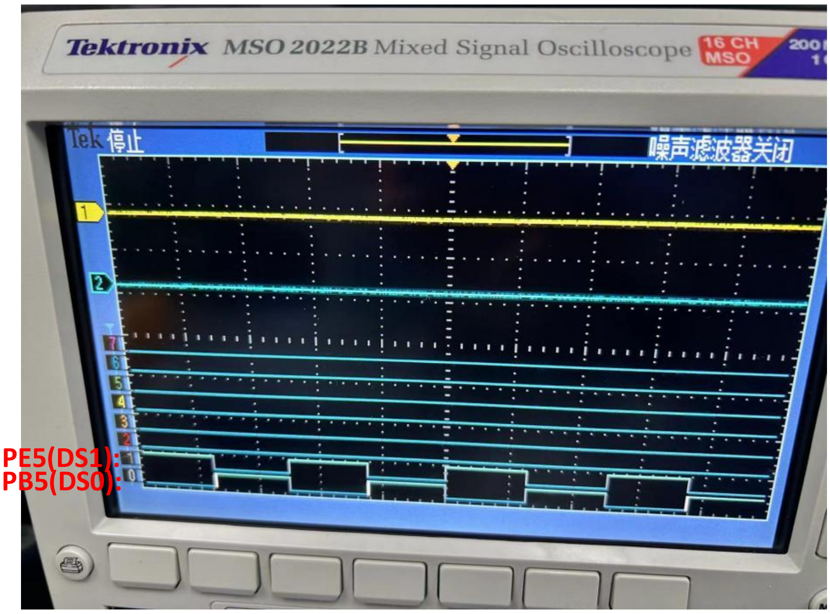

figure 4 DS0~1Hz,DS1 flashes oppositely

3. Experiment 3

3.1 Design

3.1.1 Elaborate in details how I realize step 4).

Period of SysTick is 10ms, and the frequency of SysTick is 100Hz. Triggering handler means flipping LED status. So triggering handler twice means one cycle of flashing. To set DS0 as 5Hz, task1HeartBeat should be triggered every $\frac{100}{5\times2}=10$ SysTick_handler. To set DS1 as 2Hz, task2HeartBeat should be triggered every $\frac{100}{2\times2}=25$ SysTick_handler. So the code should be as follows.(figure5)

```
int main(void)
{
  EIE3810_clock_tree_init();//initialize clock tree
  EIE3810_LED_Init();//initialize LED
  EIE3810_NVIC_SetPriorityGroup(5);//set PRIGROUP as 5
  EIE3810_SYSTICK_Init();//initialize systick

  GPIOE->ODR |=1<<5;// turn off DS1
  GPIOB->ODR |=1<<5;// turn off DS0
  //GPIOE->BSRR=1<<5;
  //GPIOB->BSRR=1<<5;

  while(1)
  {
    if (task1HeartBeat>=10)
    {
      GPIOB->ODR ^=1<<5;//flip PB5
      task1HeartBeat=0;//load task1HeartBeat as 0
    }
    if (task2HeartBeat>=25)
    {
      GPIOE->ODR ^=1<<5;//flip PE5
      task2HeartBeat=0;//load task2HeartBeat as 0
    }
  }

}
```

figure 5 Set DS0, DS1 as 5Hz, 2Hz

3.2 Result

Set DS0 and DS1 to flash at 5Hz and 2Hz respectively.

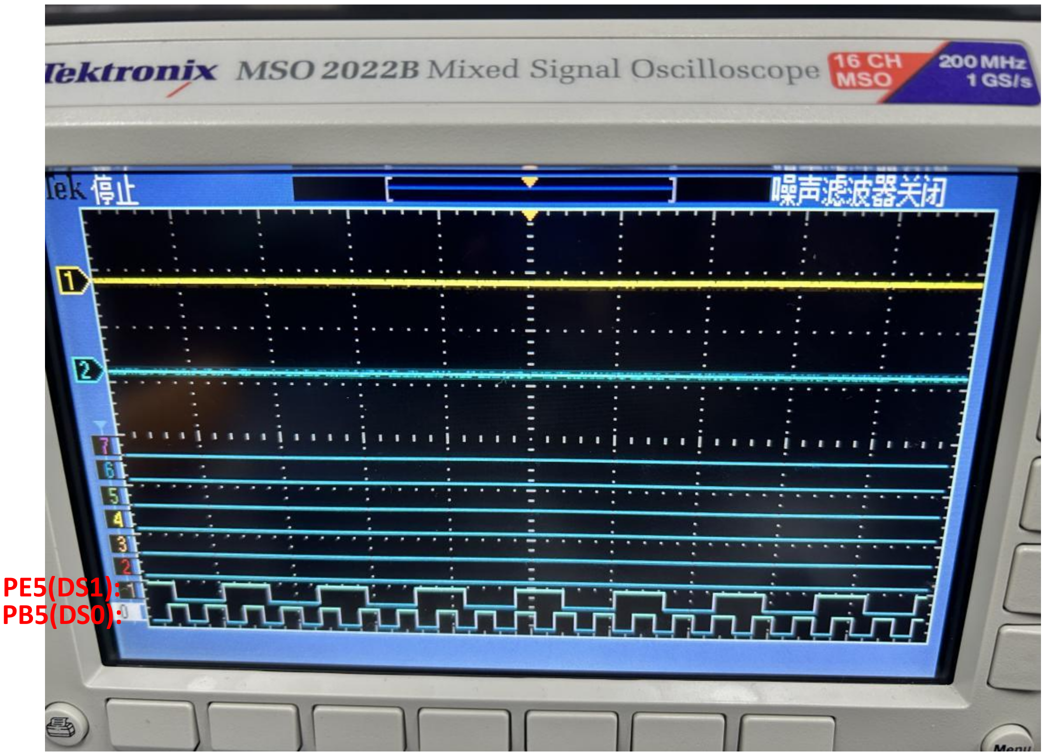

figure 6 DS0~5Hz,DS1~2Hz

3.3 Questions

3.3.1 How to change the SysTick frequency to 200 Hz?

To change SysTick frequency from 100Hz to 200Hz, we can halve the reload value, which triggers SysTick_Handler twice frequently. So change SysTick->LOAD from 89999 to 44999.(eg. figure 7) By the way, I explain how to get 90000: $clock\_counter\_value = \frac{system\_clock\_frequency}{pre\_scalar\_value \times LED\_flashing\_frequency} = \frac{72MHz}{8 \times 100Hz} = 90000$

Notice that CLKSOURCE=0 (by default) means FCLK/8,namely,pre_scalar_value=8.

```
//SysTick->LOAD=90000-1;
SysTick->LOAD= 45000-1;/
```

figure 7 change SysTick_LOAD from 89999 to 44999

4. Experiment 4

4.1 Result

Use a PWM signal that is generated by TIM3 to drive DS0 with various brightness.

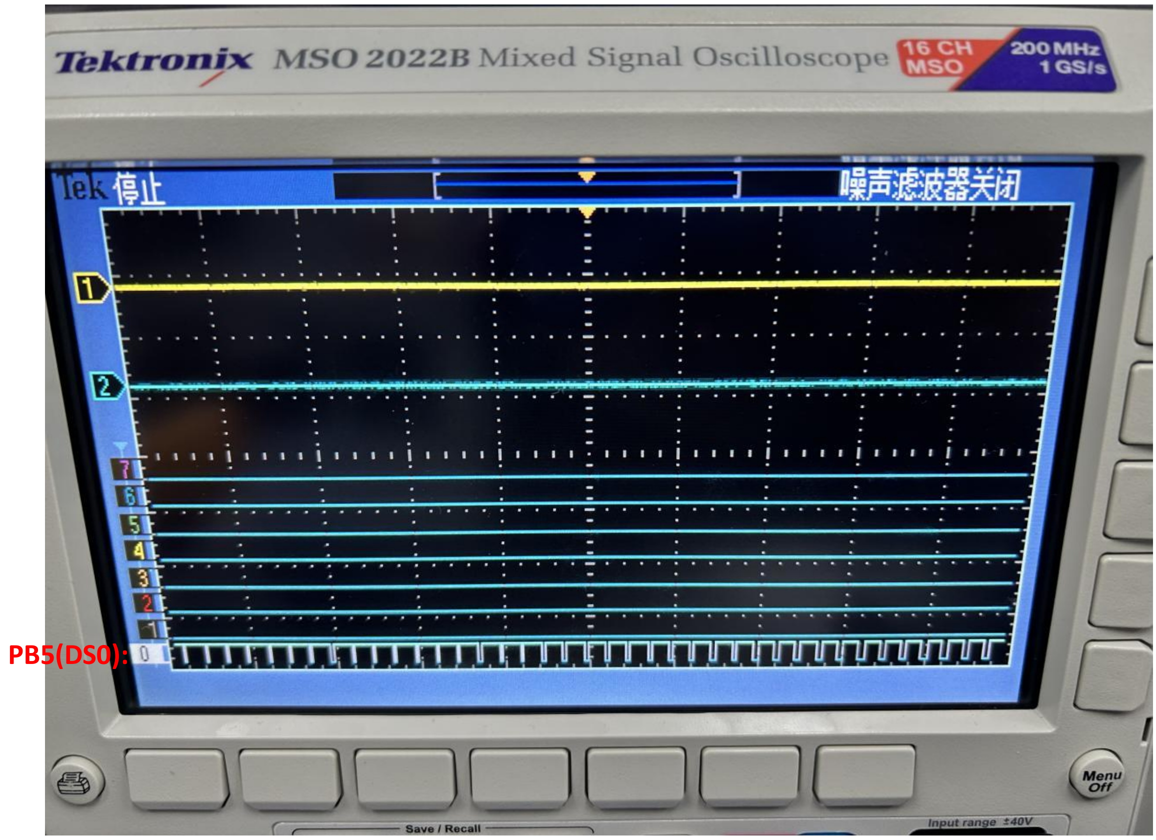

figure 8 DS0 waveform (1)

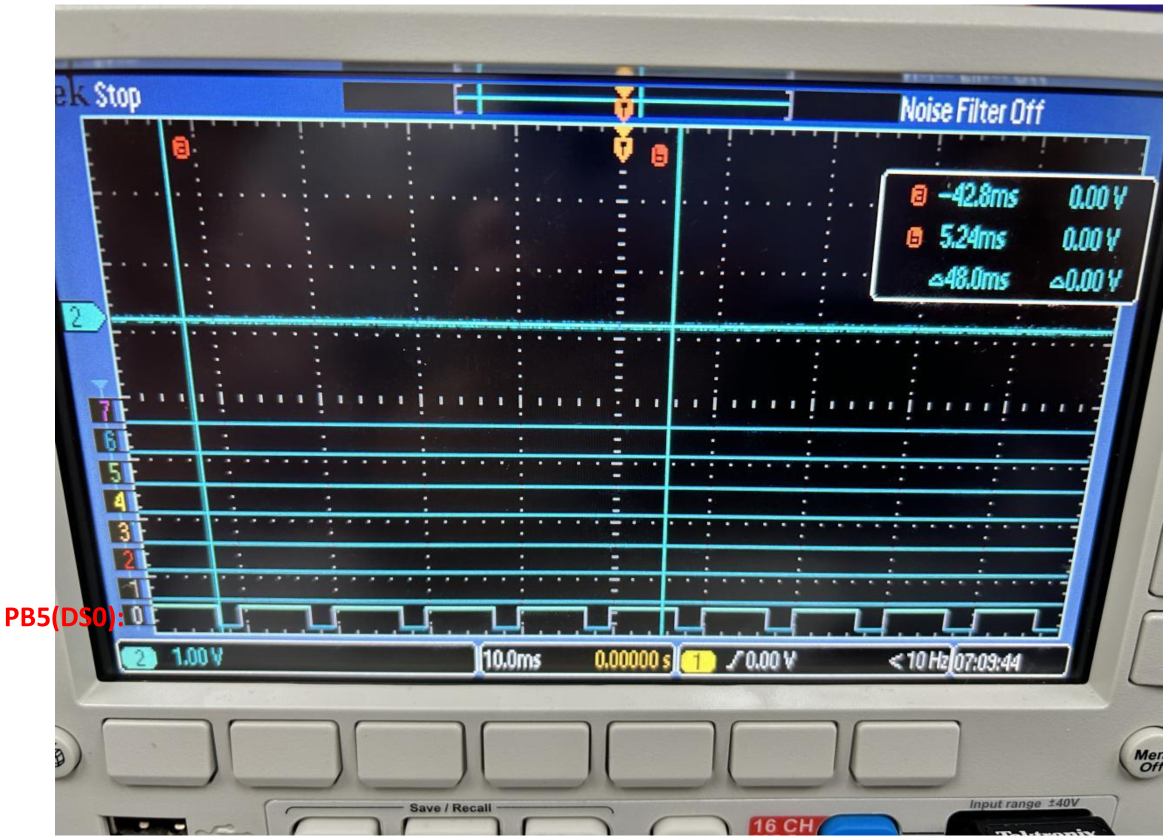

figure 9 DS0 waveform (2)

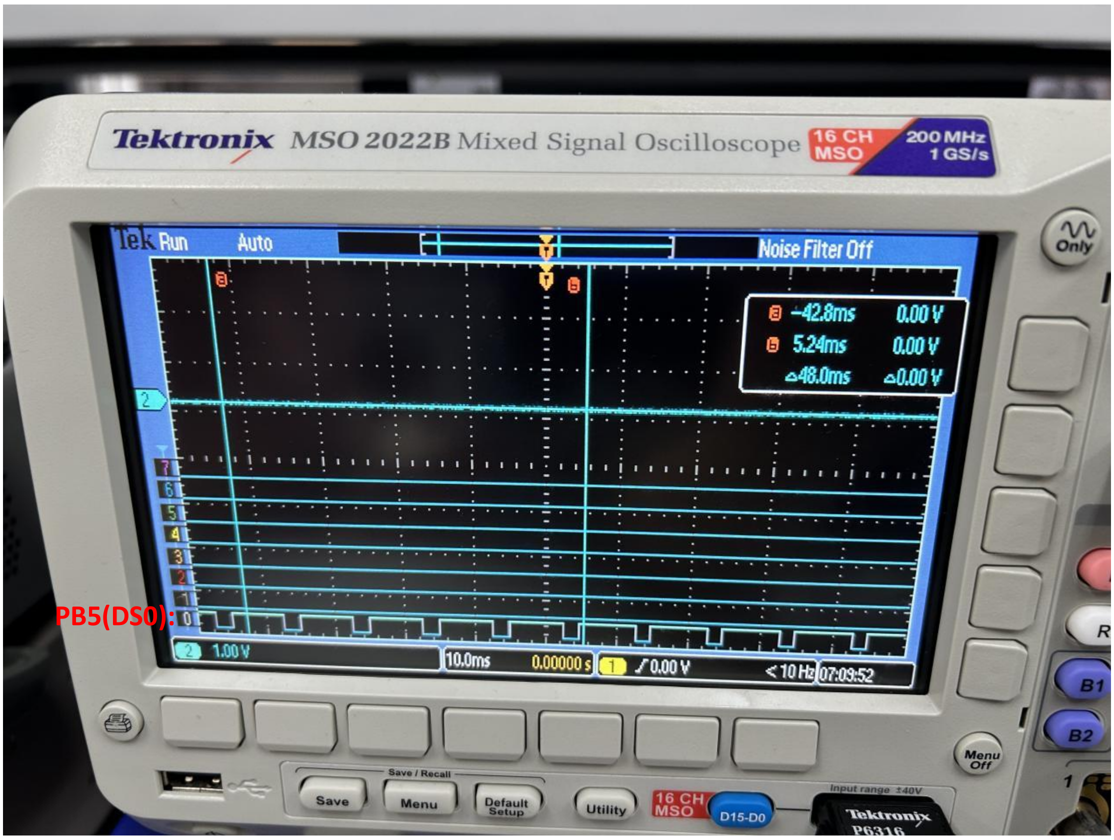

figure 10 DS0 waveform (3)

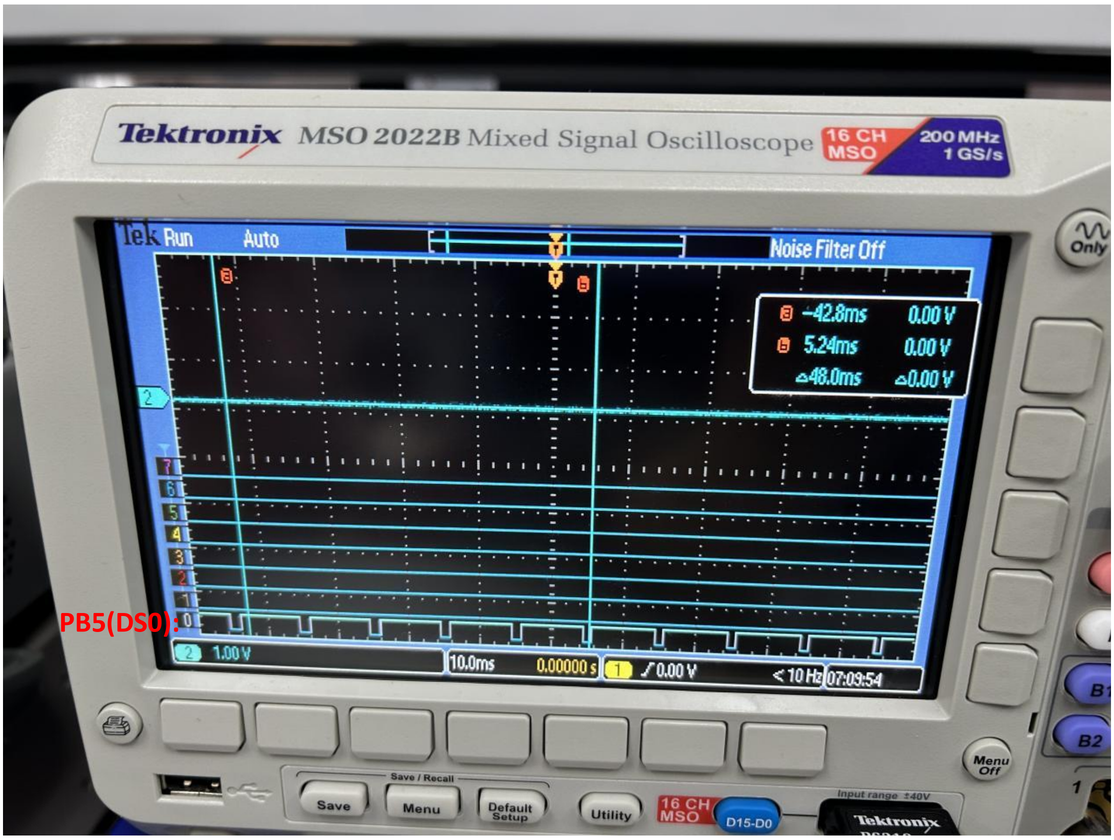

figure 11 DS0 waveform (4)

4.2 Questions

4.2.1 If LED0_PWM_VAL is 0, 9999, or 10000, what happens to the DS0 signal in an oscilloscope, and what happens to LED0?

LED0_PWM_VAL = 0:

DS0(LED0) is always High. And DS0(LED0) is always off.

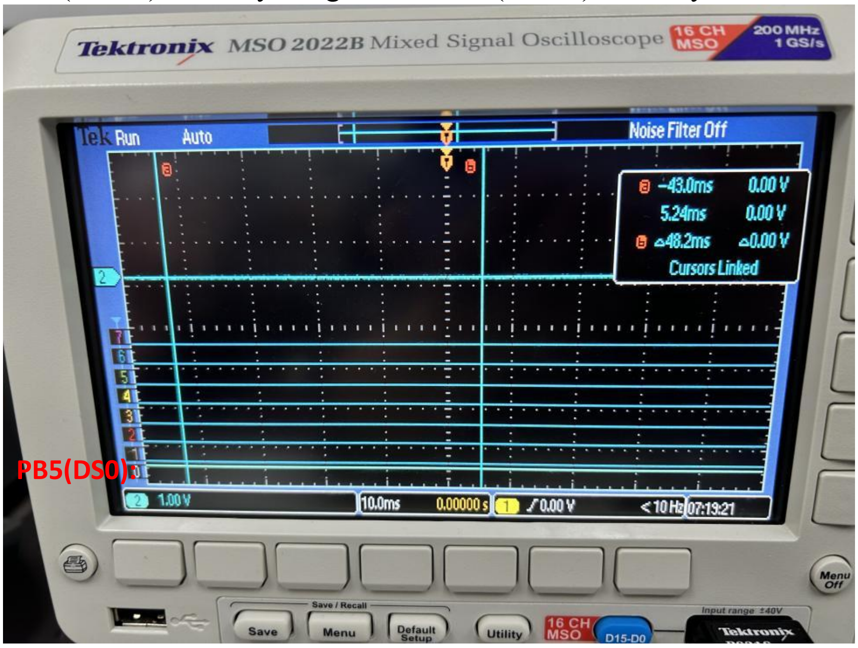

figure 12 LED0_PWM_VAL=0

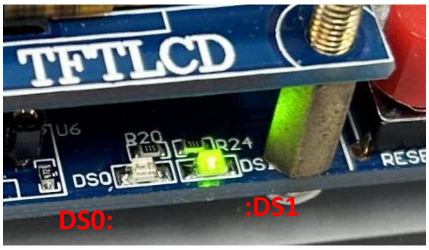

figure 13 LED0_PWM_VAL=0

LED0_PWM_VAL = 9999:

DS0(LED0) is Low except a very short period of High periodically. And DS0(LED0) looks like always on(Theoretically, brightness is reduced extremely slightly periodically).

figure 14 LED0_PWM_VAL=9999

figure 15 LED0_PWM_VAL=9999

LED0_PWM_VAL = 10000:
DS0(LED0) is always Low. And DS0(LED0) is always on.

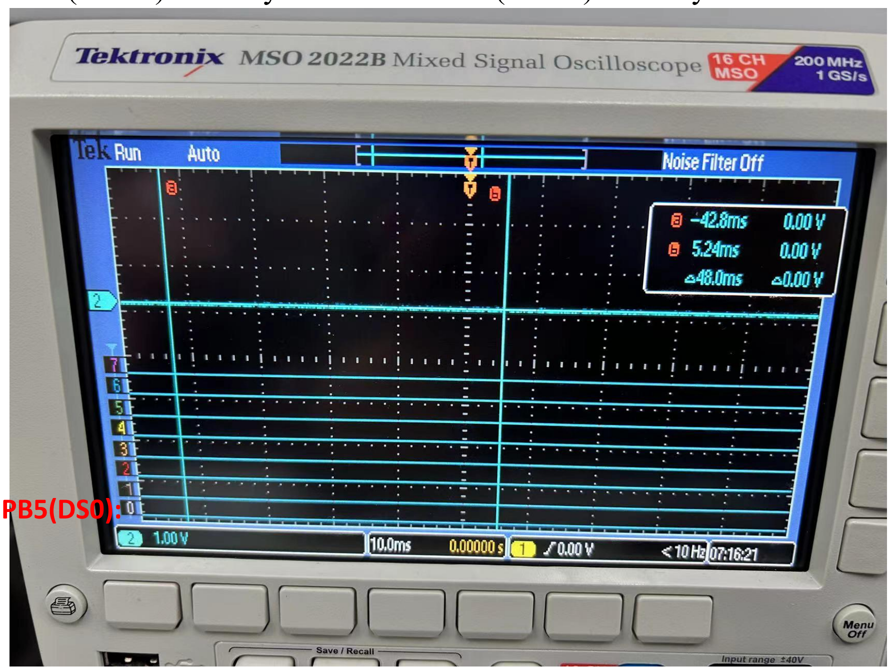

figure 16 LED0_PWM_VAL = 10000

figure 17 LED0_PWM_VAL=10000

5. Experiment 5

5.1 Result

figure 18 Push LEFT

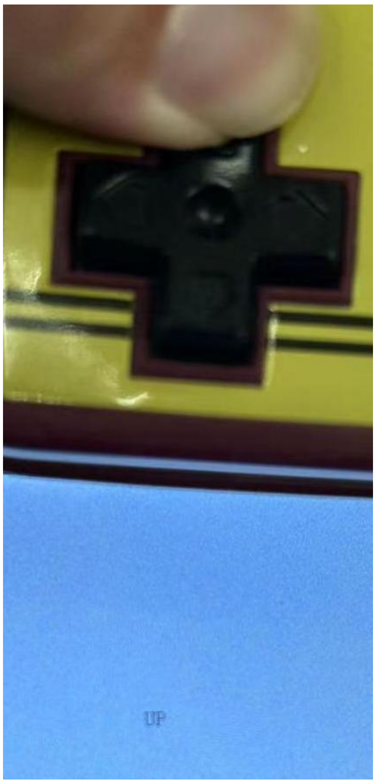

figure 19 Push UP

figure 20 Push RIGHT

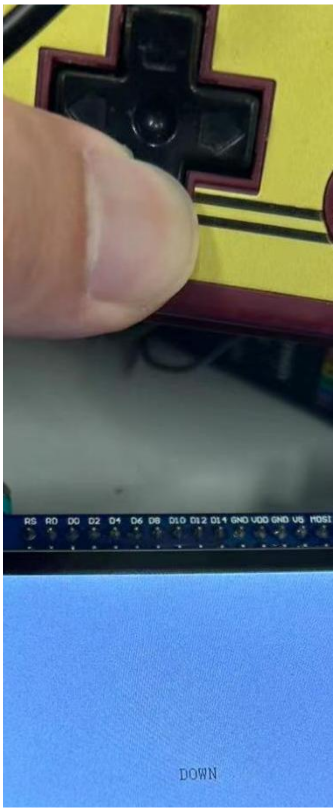

figure 21 Push DOWN

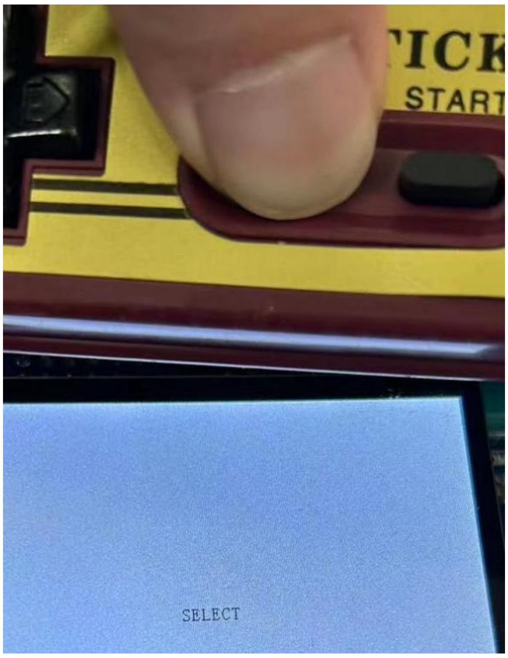

figure 22 Push SELECT

figure 23 Push START

figure 24 Push B

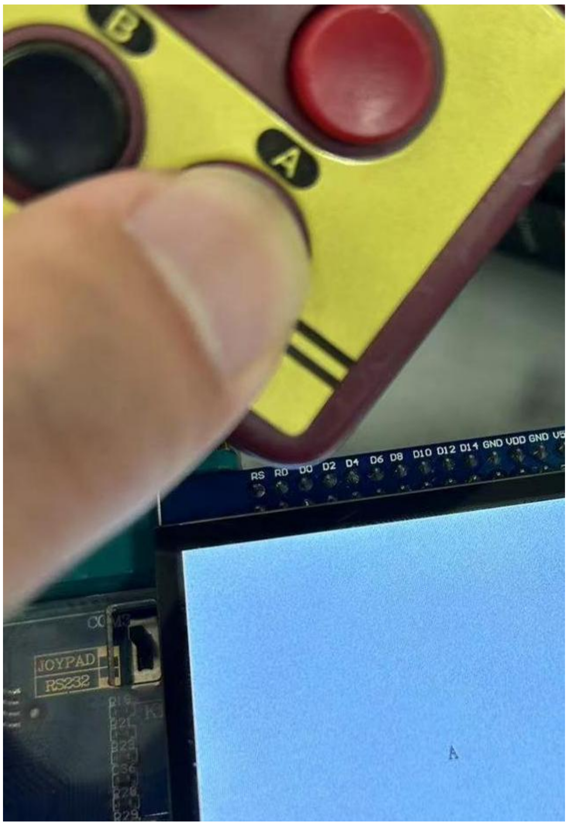

figure 25 Push A

6. Conclusion

I have learned how to use the General Purpose Timer of Cortex-M3 and how to use the SYSTICK Timer of Cortex-M3 and the generation of a Pulse Width Modulation signal and use timer to acquire input from JOYPAD.