



# **MICROCHIP**

---

***Regional Training Centers***

**Bonus Labs**

# Lab2 Bonus Lab

- Press BT3(PA15) to toggle below scenario:
- State 0 :
  - ▢ LED1 / LED2 toggle mutually.
  - ▢ BT2(PA14) Pressed -> LED3 Light, Released -> LED3 Dark
- State 1 :
  - ▢ LED2 / LED3 toggle mutually.
  - ▢ BT2(PA14) Pressed -> LED1 Light, Released -> LED1 Dark

■ **Let's go!**

# Lab2 Bonus Lab

## Step 1

### **a** change coding style to state machine mode

```
uint32_t i = 0;
uint32_t State = 0;

int main (void)
{
    SYS_Initialize ( NULL );

    while(1)
    {
        if( i++ > 200000 )
        {
            i = 0;

            if( State==0 ) { LED1_Toggle(); LED2_Toggle(); }
            else           { LED2_Toggle(); LED3_Toggle(); }
        }

        ...
    }
}
```

# Lab2 Bonus Lab

## Step 2

### **a** change coding style to state machine mode

```
int main (void)
{
    ...
    while(1)
    {
        if( i++ > 200000 ) { ... }

        if( BT2_Get() )
        {
            if( State==0 ) LED3_Clear();
            else          LED1_Set();
        }
        else
        {
            if( State==0 ) LED3_Set();
            else          LED1_Clear();
        }
    }
}
```

# Lab2 Bonus Lab

## Step 3

### **a** change coding style to state machine mode

```
int main (void)
{
    ...
    while(1)
    {
        if( i++ > 200000 ) { ... }

        if( BT2_Get() ) { ... }
        else { ... }

        if ( !BT3_Get() )
        {
            while( !BT3_Get() );

            State = !State;

            LED1_Set();
            LED2_Clear();
            LED3_Clear();
        }
    }
}
```

# Lab6 Bonus Lab

- Try use three LEDs to implement the Binary Counting.
- The MSB LED is LED1 and toggle in one second period

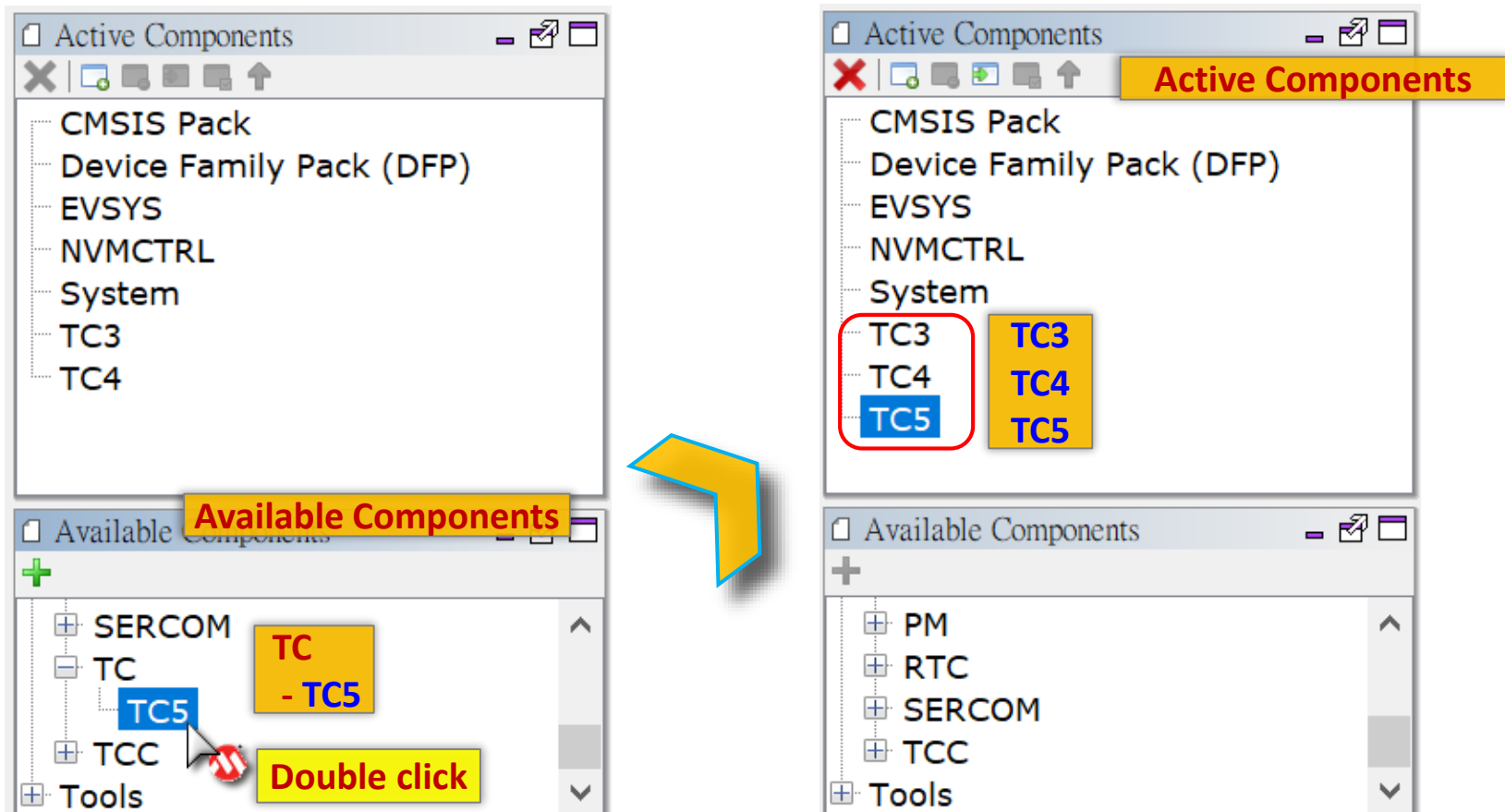
 **Let's go!**

		LED1	LED2	LED3
<div><div></div><div></div><div></div><div></div></div> <div>1 sec</div>	0	0	0	0
	1	0	0	1
	2	0	1	0
	3	0	1	1
<div><div></div><div></div><div></div><div></div></div> <div>1 sec</div>	4	1	0	0
	5	1	0	1
	6	1	1	0
	7	1	1	1

# Lab6 Bonus Lab

## Step 1

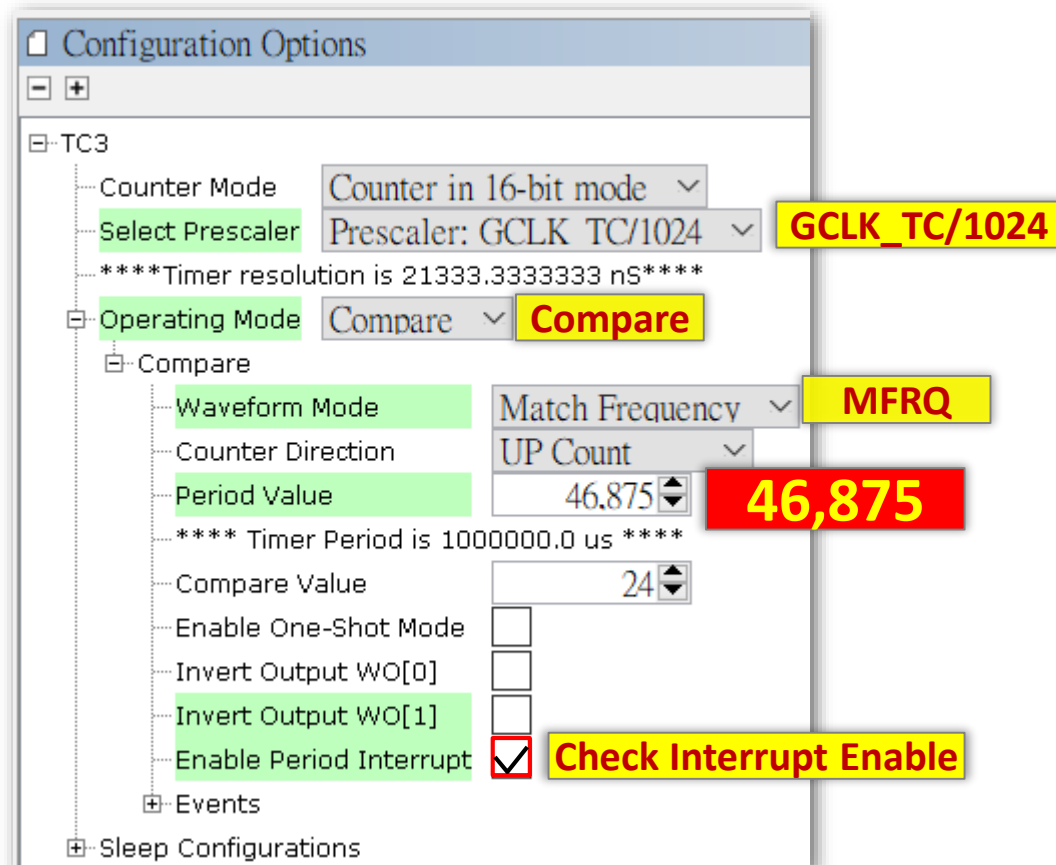
- ❖ Add extra **TC5** component to project, now you have three TC components there are **TC3**, **TC4** and **TC5** currently.



# Lab6 Bonus Lab

## Step 2

- Configure **TC3 Period** to **1 second** for **MSB LED1** toggle.
- Enable Period Interrupt** of TC3 in configuration windows.

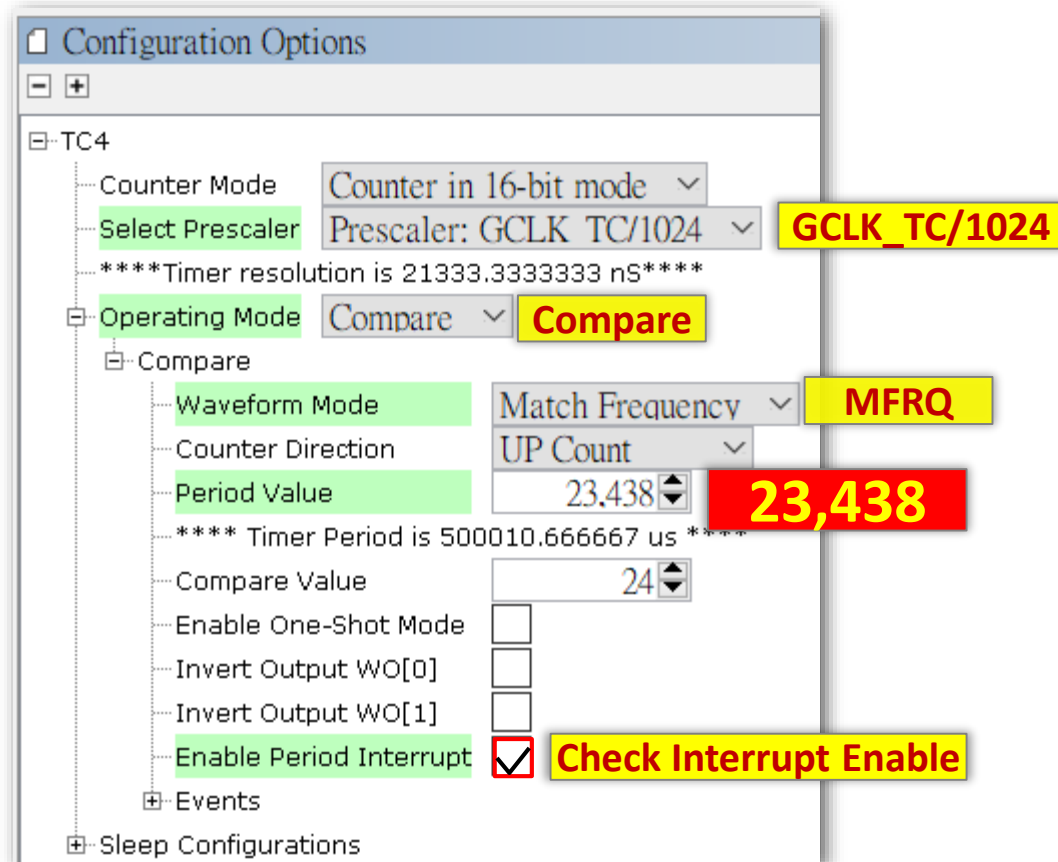




# Lab6 Bonus Lab

## Step 3

- Configure **TC4 Period** to **1/2 second** for **LED2 toggle**.
- Enable Period Interrupt** of TC4 in configuration windows.



# Lab6 Bonus Lab

## Step 4

- Configure **TC5 Period** to **1/4 second** for **LSB LED3 toggle**.
- Enable Period Interrupt** of TC5 in configuration windows.

The screenshot shows the 'Configuration Options' window for TC5. The settings are as follows:

- Counter Mode:** Counter in 16-bit mode
- Select Prescaler:** Prescaler: GCLK TC/1024 (Annotated with **GCLK\_TC/1024**)
- \*\*\*\*Timer resolution is 21333.3333333 nS\*\*\*\***
- Operating Mode:** Compare (Annotated with **Compare**)
- Compare Section:**
  - Waveform Mode:** Match Frequency (Annotated with **MFRQ**)
  - Counter Direction:** UP Count
  - Period Value:** 11,719 (Annotated with **11,719**)
  - \*\*\*\* Timer Period is 250005.333333 us \*\*\*\***
  - Compare Value:** 24
  - Enable One-Shot Mode:** ☐
  - Invert Output WO[0]:** ☐
  - Invert Output WO[1]:** ☐
  - Enable Period Interrupt:** ☒ (Annotated with **Check Interrupt Enable**)
- Events:** ☐
- Sleep Configurations:** ☐

# Lab6 Bonus Lab

## Step 5

- Configure initial LEDs status for counting start from **0 0 0**.

Pin Settings

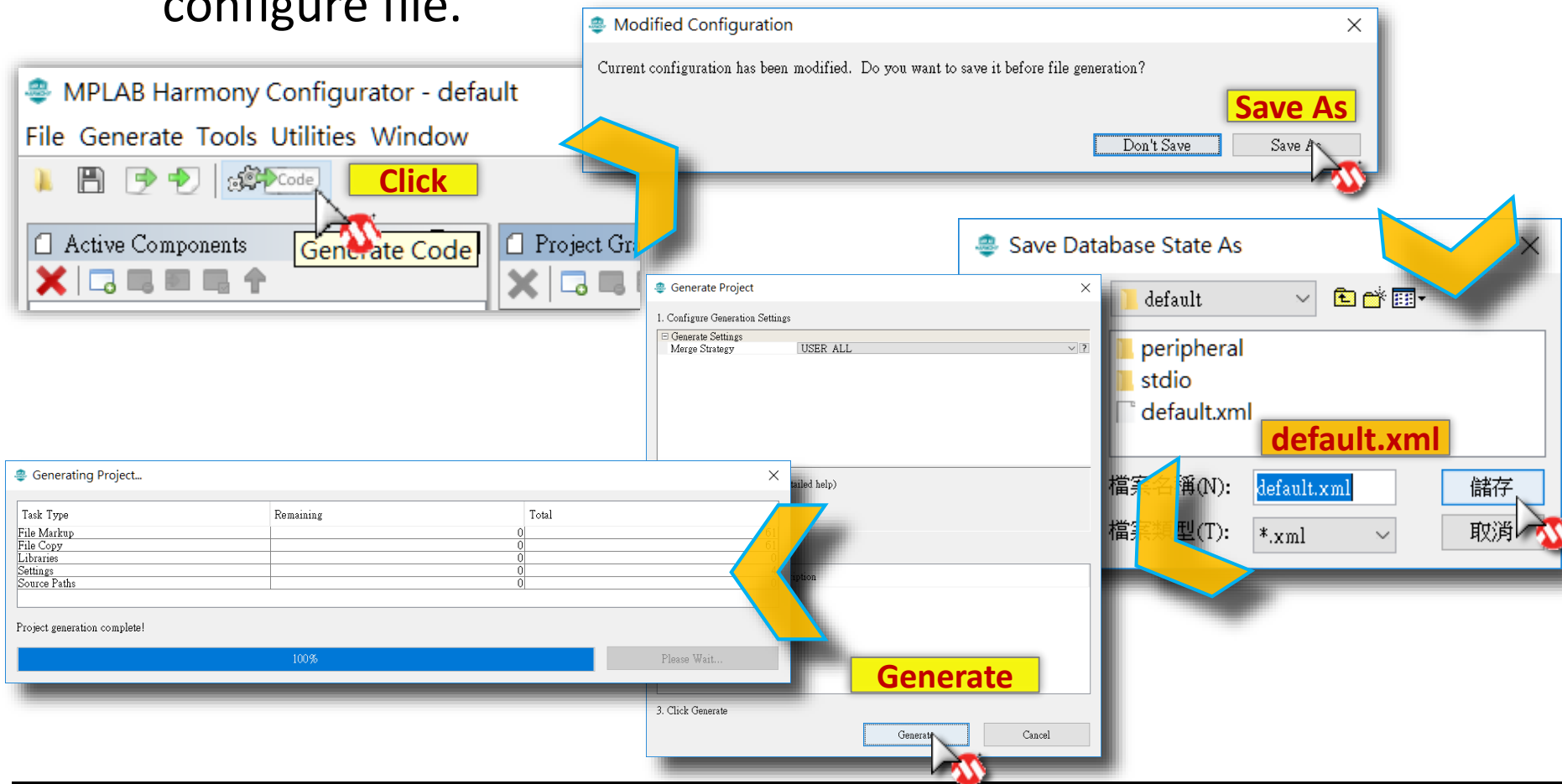
Order: Pins Table View Easy View

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull
13	PA08		Available	Di...	High ...	Low		
14	PA09	LED2	GPIO	Di...	Out	High	High	
15	PA10		Available	Di...	High ...	Low		
16	PA11		Available	Di...	High ...	Low		
17	VDDIO			Di...	High ...	Low		
18	GNDIO			Di...	High ...	Low		
19	PB10		Available	Di...	High ...	Low		
20	PB11		Available	Di...	High ...	Low		
21	PA12		Available	Di...	High ...	Low		
22	PA13		Available	Di...	High ...	Low		
23	PA14	BT2	GPIO	Di...	In	Low		
24	PA15		Available	Di...	High ...	Low		
25	PA16		Available	Di...	High ...	Low		
26	PA17	LED3	GPIO	Di...	Out	Low	Low	
27	PA18	TC3_WO0	TC3_WO0	Di...	High ...	n/a		
28	PA19	TC3_WO1	TC3_WO1	Di...	High ...	n/a		
29	PA20	LED1	GPIO	Di...	Out	High	High	
30	PA21		Available	Di...	High ...	Low		

# Lab6 Bonus Lab

## Step 6

- Click  to Generate Code and save changes to MHC configure file.



# Lab6 Bonus Lab

## Step 7

### **a** change coding style to state machine mode

```
void TC3_Overflow(TC_COMPARE_STATUS status, uintptr_t context)
{
    if( status & TC_INTFLAG_OVF_Msk )
        LED1_Toggle();
}

void TC4_Overflow(TC_COMPARE_STATUS status, uintptr_t context)
{
    if( status & TC_INTFLAG_OVF_Msk )
        LED2_Toggle();
}

void TC5_Overflow(TC_COMPARE_STATUS status, uintptr_t context)
{
    if( status & TC_INTFLAG_OVF_Msk )
        LED3_Toggle();
}

int main (void)
{
    ...
}
```

# Lab6 Bonus Lab

## Step 8

### **a** Add code segment to your main loop

```
...
int main (void)
{
    SYS_Initialize ( NULL );
    ...
    TC3_CompareCallbackRegister( TC3_Overflow, (uintptr_t )NULL );
    TC3_CompareStart();
    TC4_CompareCallbackRegister( TC4_Overflow, (uintptr_t )NULL );
    TC4_CompareStart();
    TC5_CompareCallbackRegister( TC5_Overflow, (uintptr_t )NULL );
    TC5_CompareStart();

    while( true )
    {
        SYS_Tasks ( );
        // Remove all polling mode TCx control code segment
        // Remove BT2 control LED3 code segment
    }
}
```

### **b** Program firmware to target board then observe result.

# Lab6 Bonus Lab

## Step 9

TC3			TC4			TC5		
	LED1		LED2		LED3			
0	0	1 sec	0	1/2 sec	0	1/4 sec	1/4 sec	
1	0		0		1			
2	0		1	1/2 sec	0			
3	0		1		1			
4	1	1 sec	0		0			
5	1		0		1			
6	1		1		0			
7	1		1		1			

# Lab11 Bonus Lab

- Input “LED1” in TeraTerm will toggle LED1
- Input “LED2” in TeraTerm will toggle LED2
- Input “LED3” in TeraTerm will toggle LED3

• **Let's go!**



# Lab11 Bonus Lab

## Step 1



Create UART TxRx Callback function, related buffer and

```
...
uint8_t USART5_ReceiveData[5];
...
int main (void)
{
    ...
    SERCOM5_USART_Read( USART5_ReceiveData, 4 );
    ...

    while(1)
    {
        if (TC3_IsOverflow)
        { ...
          // LED1_Toggle();
        }

        if (TC4_IsOverflow)
        { ...
          // LED2_Toggle();
        }

        ...
    }
}
```

# Lab11 Bonus Lab

## Step 2

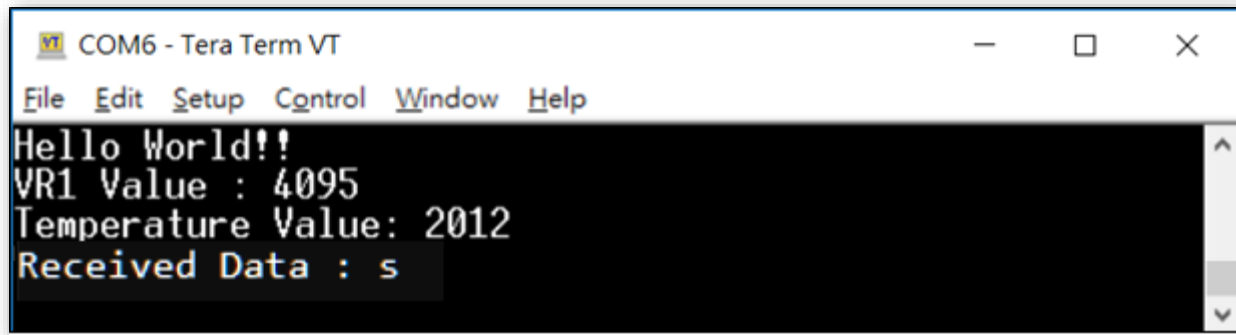
### **a** Create UART TxRx Callback function, related buffer and

```
int main (void)
{
    ...
    while(1)
    {
        ...
        if( USART5_IsReceived )
        {
            printf( "\r\nReceived Data : %s\r\n", USART5_ReceiveData );
            if( USART5_ReceiveData[0] == '1' &&
                USART5_ReceiveData[1] == 'e' &&
                USART5_ReceiveData[2] == 'd' )
            {
                if( USART5_ReceiveData[3] == '1' )    LED1_Toggle();
                else if( USART5_ReceiveData[3] == '2' )    LED2_Toggle();
                else if( USART5_ReceiveData[3] == '3' )    LED3_Toggle();
            }

            SERCOM5_USART_Read( USART5_ReceiveData, 4 );    }
        }
    }
}
```

# Lab13 Bonus Lab

- Use VT100 command to control TeraTerm UART output as



```
COM6 - Tera Term VT
File Edit Setup Control Window Help
Hello World!!
VR1 Value : 4095
Temperature Value: 2012
Received Data : s
```

```
printf("\033[2J");    // Clear screen
printf("\033[?251");  // Hide cursor
printf("\033[y;xH");  // Set cursor to (x,y)
                      // Left-up corner is from (1,1)
```

 **Let's go!**

# Lab13 Bonus Lab

## Step 1

### **a** Add code segment to your main.c

```
...

int main (void)
{
    ...
    printf("\033[2J");
    printf("\033[?25l");

    while(1)
    {
        ...
        if( TC3_IsOverflow )
        {
            ...
            printf("\033[1;1H");
            printf( "Hello World!!\r\n" );
        }

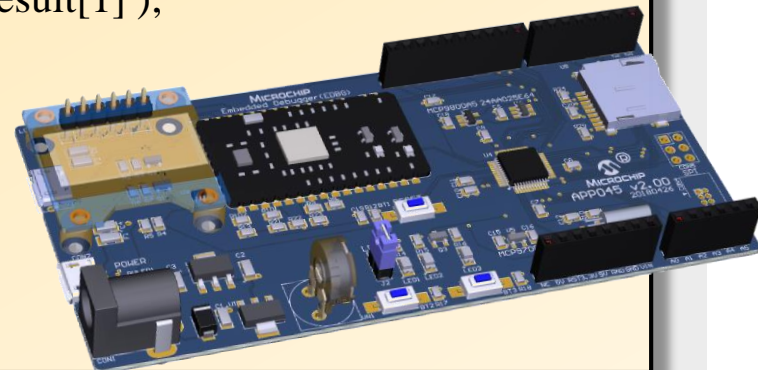
        if( USART5_IsReceived )
        {
            ...
            printf("\033[4;1H");
            printf( "Received Data : %1c\r\n", USART5_ReceiveData[0] );
        }
    }
}
```

# Lab13 Bonus Lab

## Step 2

### a Add code segment to your main.c

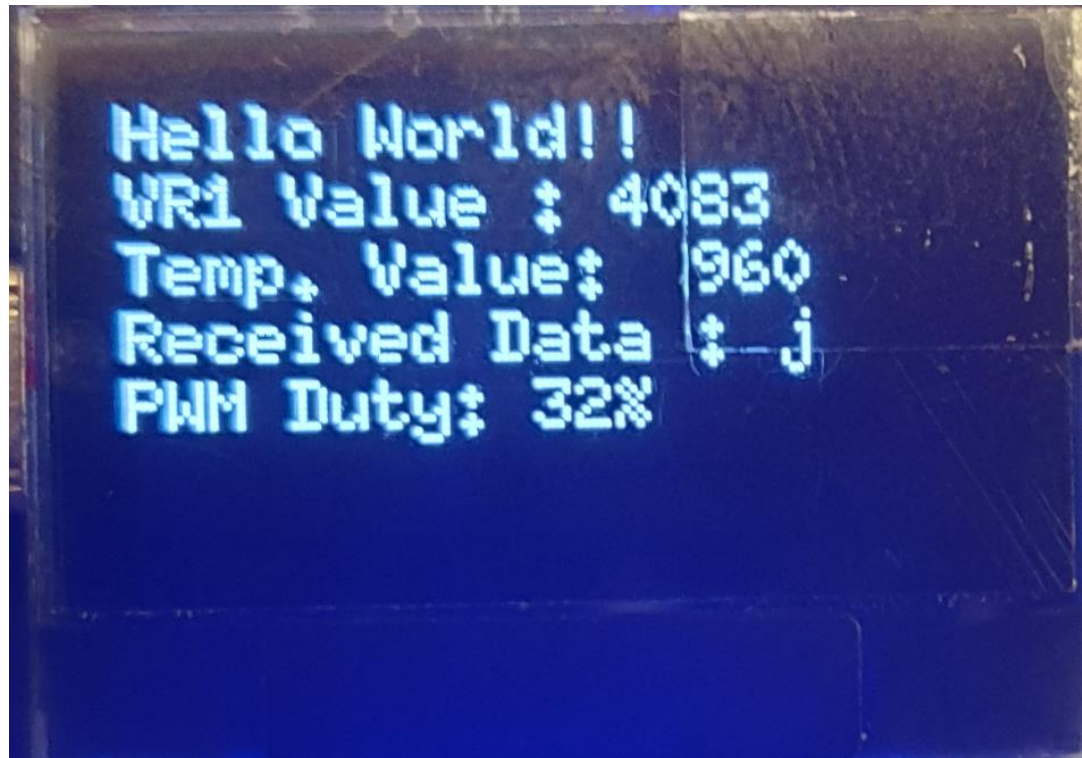
```
int main (void)
{
    ...
    while(1)
    {
        ...
        if( ADC_IsCompleted )
        {
            ...
            printf("\033[2;1H");
            printf( "VR1 Value : %4d\r\n", ADC_Result[0] );
            printf("\033[3;1H");
            printf( "Temperature Value : %4d\r\n", ADC_Result[1] );
        }
        ...
    }
}
```



### b Program firmware to target board then observe result.

# Lab17 Bonus Lab

- Try add extra row on OLED to shows PWM duty as below.



- Let's go!**

# Lab17 Bonus Lab

## Step 1

### **a** Add code segment to your project

```
uint8_t DisplayBuffer[5][64];  
...  
int main (void)  
{ ...  
    while ( true )  
    { ...  
        if (TC3_IsOverflow)  
        { ...  
            do  
            {  
                u8g_DrawStr( &OLED_Instance, 0, 0, DisplayBuffer[0] );  
                u8g_DrawStr( &OLED_Instance, 0, 10, DisplayBuffer[1] );  
                u8g_DrawStr( &OLED_Instance, 0, 20, DisplayBuffer[2] );  
                u8g_DrawStr( &OLED_Instance, 0, 30, DisplayBuffer[3] );  
                u8g_DrawStr( &OLED_Instance, 0, 40, DisplayBuffer[4] );  
            }  
        }  
    }  
}
```

# Lab17 Bonus Lab

## Step 2

### **a** Add code segment to your project

```
...
int main (void)
{
    ...
    while ( true )
    {
        ...
        if (TC4_IsOverflow)
        {
            ...
            Duty += DutyDistance;
            if (Duty >= 100 || Duty <= 0)
                DutyDistance = -DutyDistance;

            sprintf( (char *) DisplayBuffer[4], "PWM Duty: %d%%", Duty );

            if (Duty >= 100)
                TCC2_PWM16bitDutySet( 1, TCC2_PWM16bitPeriodGet() + 1 );
            ...
        }
    }
}
```