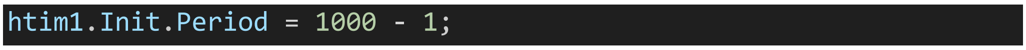
Lab6 report

B09901055 楊倢綺

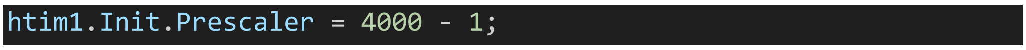
1. Change

   (a)trigger frequency



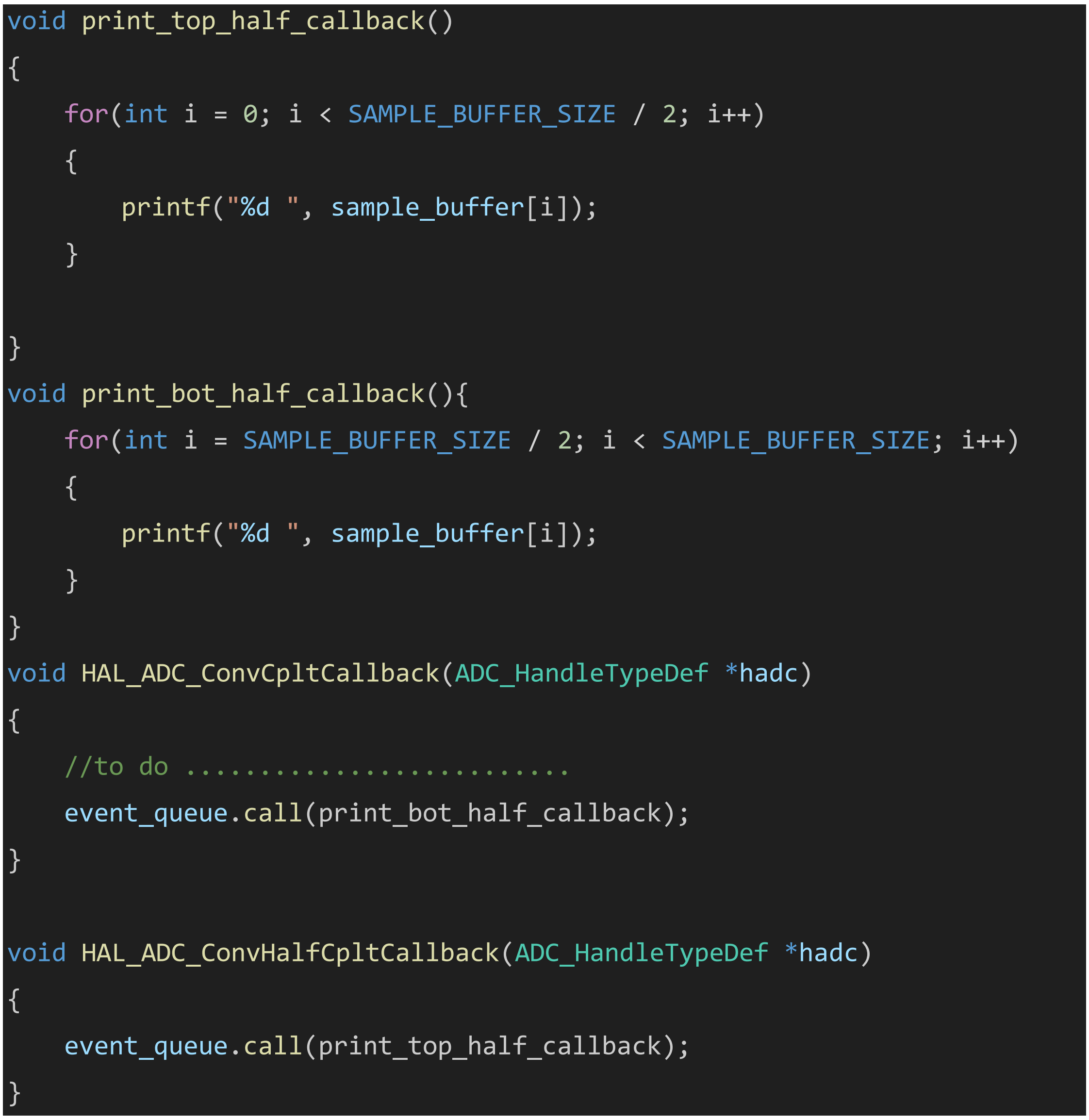
```
htim1.Init.Period = 1000 - 1;
```

   (b)sampling time of ADC



```
htim1.Init.Prescaler = 4000 - 1;
```

2. Setups DMA to transfer the data from ADC data register to a specific buffer when each conversion finishes. When the top half of buffer is filled, the interrupt will be generated and print all data in the top half of buffer. When the bottom half of buffer is filled, the interrupt makes all data in the bottom half of buffer printed.



```
void print_top_half_callback()
{
    for(int i = 0; i < SAMPLE_BUFFER_SIZE / 2; i++)
    {
        printf("%d ", sample_buffer[i]);
    }

}
void print_bot_half_callback(){
    for(int i = SAMPLE_BUFFER_SIZE / 2; i < SAMPLE_BUFFER_SIZE; i++)
    {
        printf("%d ", sample_buffer[i]);
    }
}
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    //to do ..........................
    event_queue.call(print_bot_half_callback);
}

void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef *hadc)
{
    event_queue.call(print_top_half_callback);
}
```

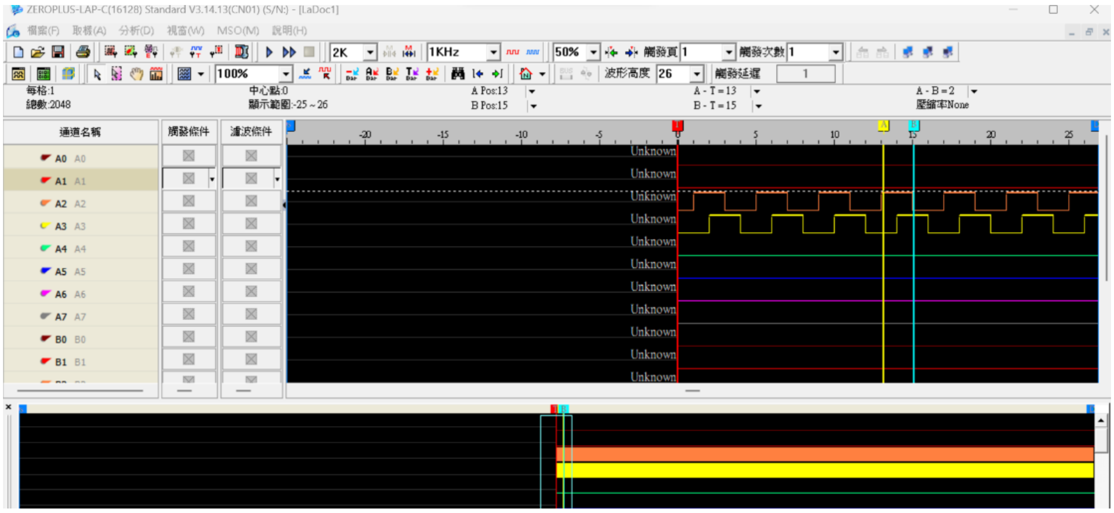
In our code, when the buffer is half-full or all-full, we let the event_queue to schedule the function to print all the value in the buffer. Since HAL_ADC_ConvCpltCallback( ) is an IRQ, we can’ t directly do the I/O operation in it.

The period of each printing would be:
System clock: 80MHz
80,000,000/4000/1000 = 20 Hz
Buffer size = 256
Period = 256/2/20 = 6.4s

3.

```
void tophalf_full_callback(){
    pin1 = !pin1;
    // printf("top full\n");
}
void bothalf_full_callback(){
    pin2 = !pin2;
    // printf("bot full\n");
}
```

In our code, we call the pin-togger function when the buffer is full.



2K
1KHz
50%
觸發頁 1
觸發次數 1
100%
波形高度 26
觸發延遲 1

每格:1
總數:2048
中心點:0
顯示範圍:-25 ~ 26
A Pos:13
B Pos:15
A - T = 13
B - T = 15
A - B = 2
壓縮率None

Period = 2ms
Frequency = 500Hz