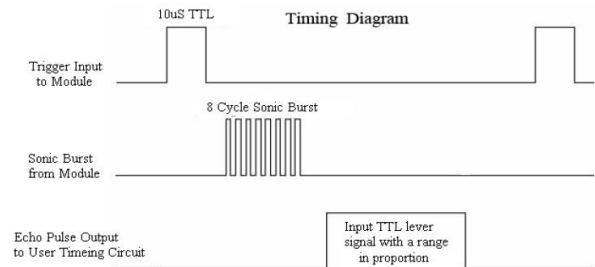


一、模組簡介

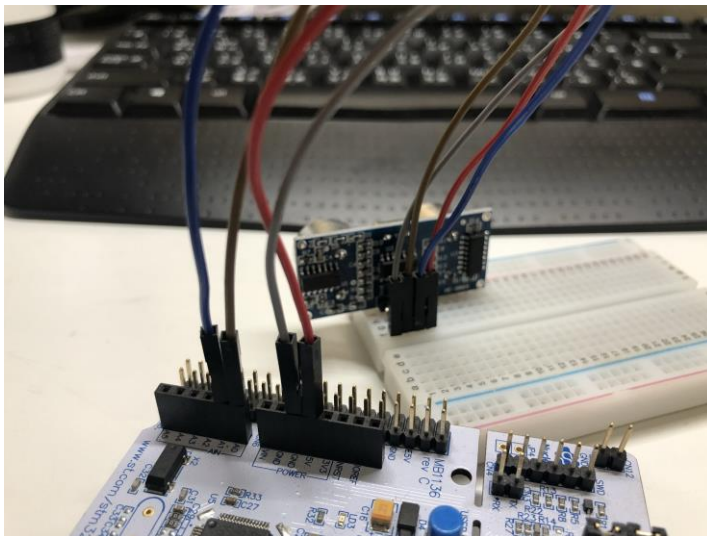
此模組有四個腳位，Vcc, Trig, Echo, Gnd。工作電壓為 5v。送一個超過 10us 的高電壓給 Trig，模組會發出 8 個 40KHz 的方波來探測距離。若有接收到返回的訊號，Echo 會輸出高電位，其持續時間為超聲波從發射到返回的時間。



<http://www.circuitdb.com/?p=1162>

而聲音傳播 1cm 所需時間，若以音速 340 m/s 來計算，大約是 29us。因此我們只要測量 Echo 高電位持續的時間 T，距離即為 $T / 29 / 2$ (cm)。

二、while-loop 做法



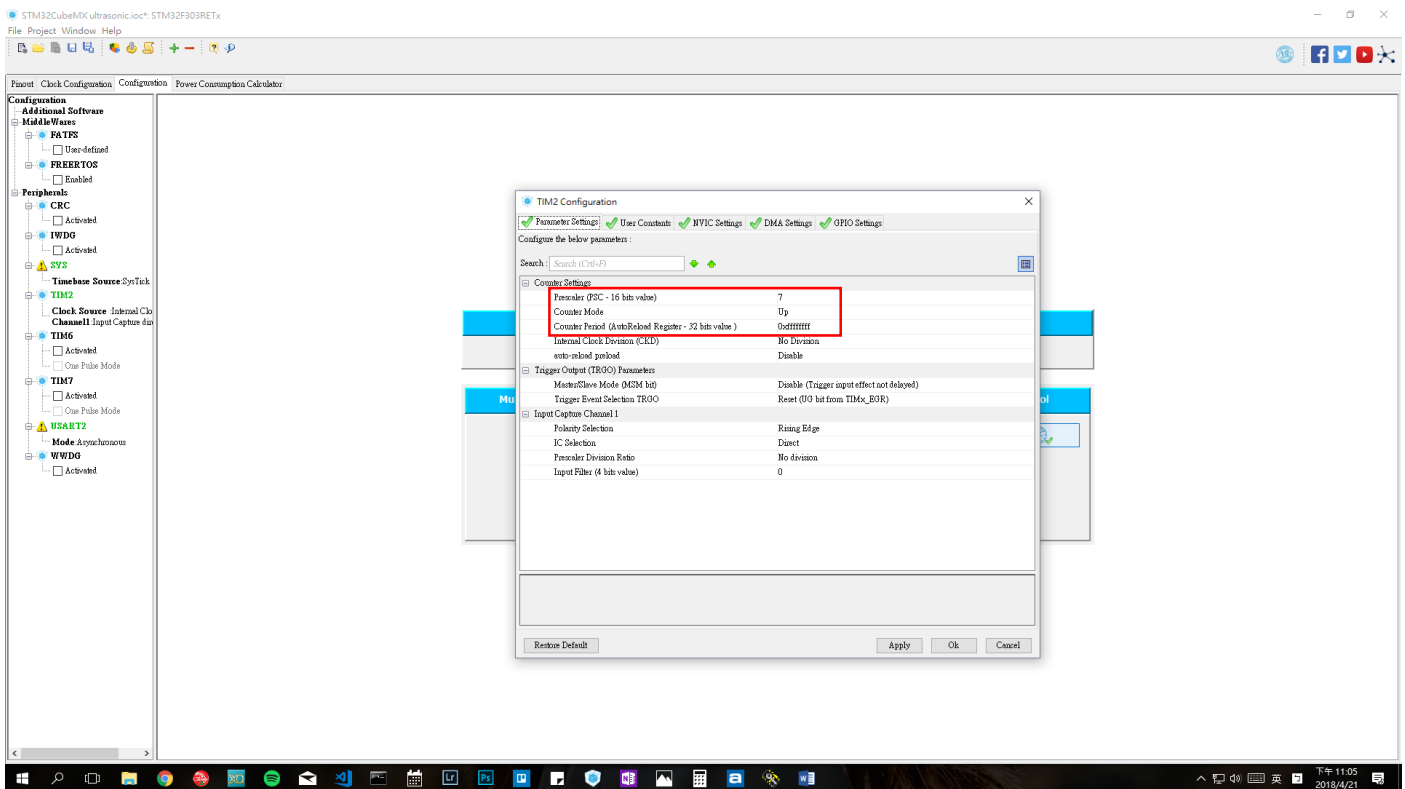
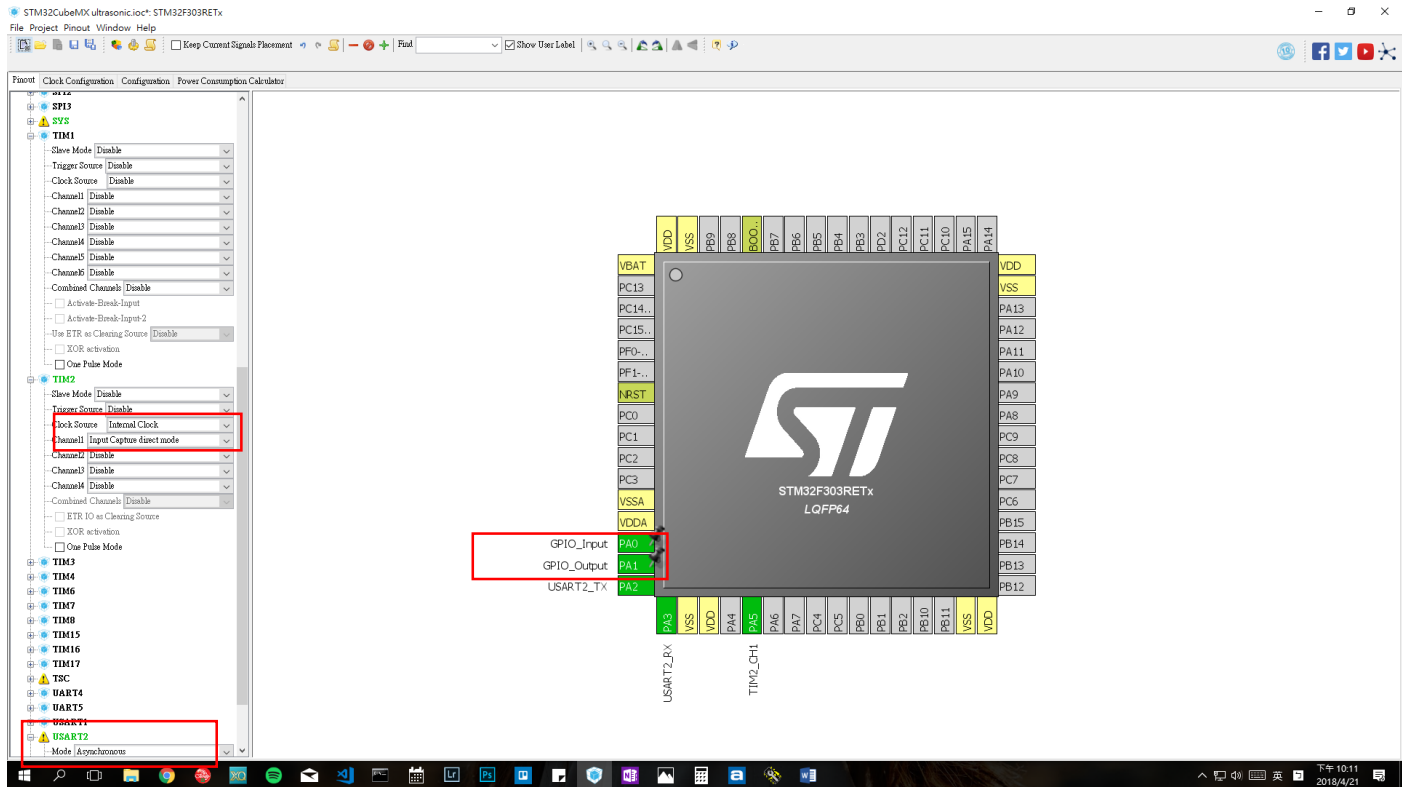
[Module] ↔ [EVB]

Vcc ↔ 5V

Gnd ↔ Gnd

Trig ↔ PA_1

Echo ↔ PA_0



main while loop

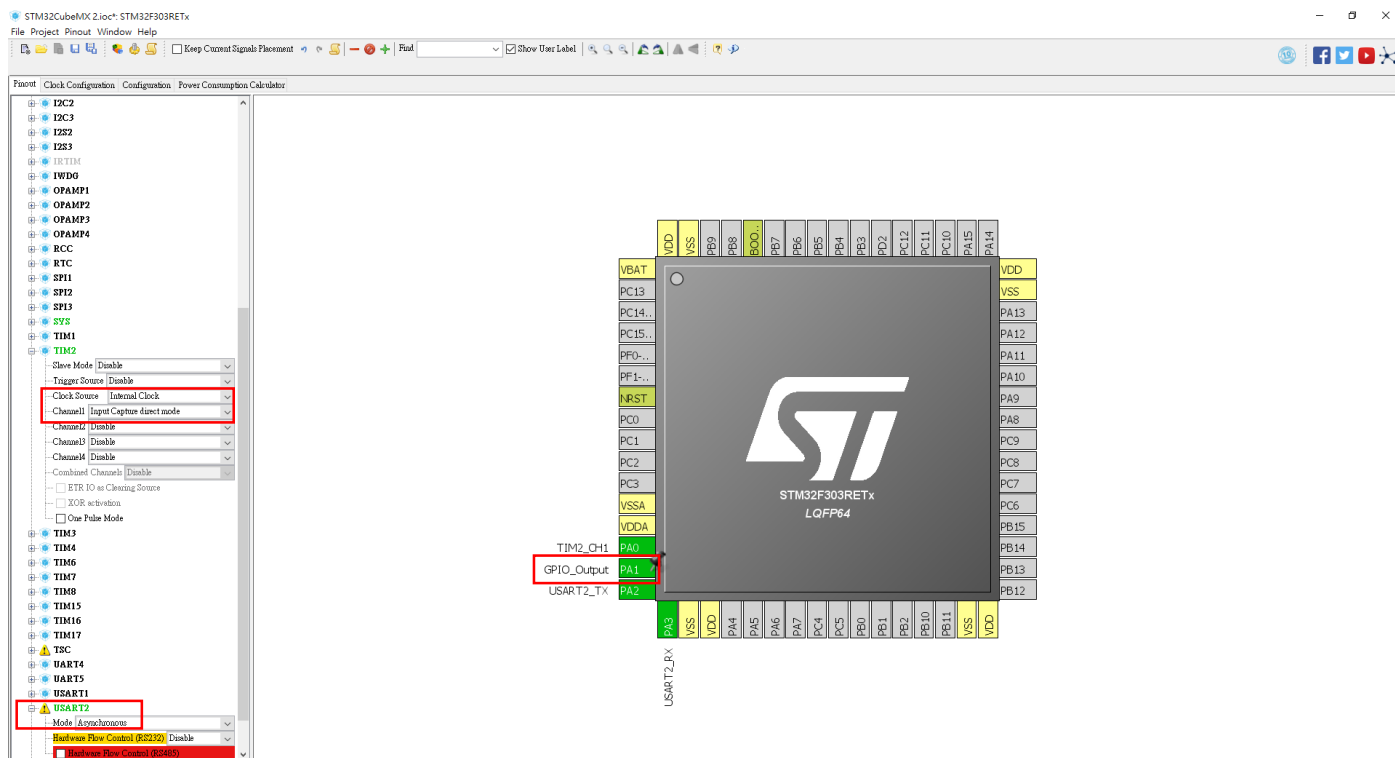
```
/* USER CODE BEGIN WHILE */
while (1)
{
    while(HAL_TIM_Base_Start(&htim2)!=HAL_OK); //等待 TIM 開啟完成
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,1); //輸出高電位至 Trig
    HAL_Delay(1); //持續至少 10us，這裡直接設為 1ms
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,0); //關閉 Trig 輸出
    while(!HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)); //等待 Echo 拉到高電位
    uint32_t t1=HAL_TIM_ReadCapturedValue(&htim2,TIM_CHANNEL_1); //Echo 拉到高電位時，將計數器值存入
t1
    while(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)); //等待 Echo 回到低電位
    uint32_t t2=HAL_TIM_ReadCapturedValue(&htim2,TIM_CHANNEL_1); //Echo 回到低電位時，將計數器值存入
t2

    double distance=0;
    if(t2>t1){
        //若 t1 t2 在同個計數週期內，直接算出時間差(因為 prescaler 為 7，一個計數間隔即為 1us)，再根據公式算距離
        distance=(t2-t1)/(double)58;
    }else{
        //若 t1 t2 不在同個計數週期內，使用__HAL_TIM_GET_AUTORELOAD 取出上個週期最後一個值，再算出時間差及距離
        distance=(__HAL_TIM_GET_AUTORELOAD(&htim2)-t1+1+t2)/(double)58;
    }
    //受 C 的 printf 與法限制，以下將整數部份及小數部份分開傳送
    int integer=(int)distance;
    int point=(int)((distance-integer)*100);
    char tosend[20]={0};
    sprintf(tosend,"%d.%02d\r\n",integer,point);
    HAL_UART_Transmit(&huart2,tosend,sizeof(tosend),0xffff);
    //等待至少 400ms
    HAL_Delay(400);
/* USER CODE END WHILE */

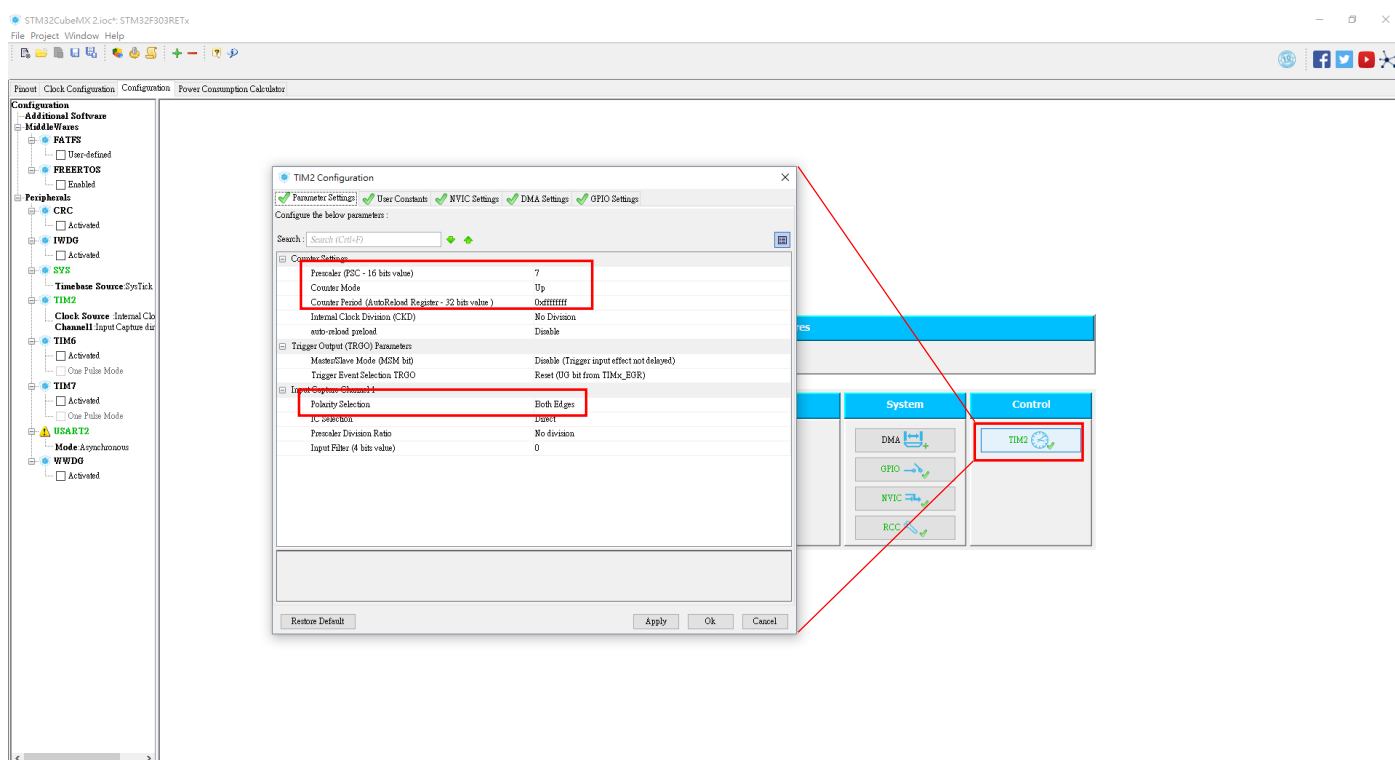
/* USER CODE BEGIN 3 */
}
}
```

完整程式請見 [Github](#)

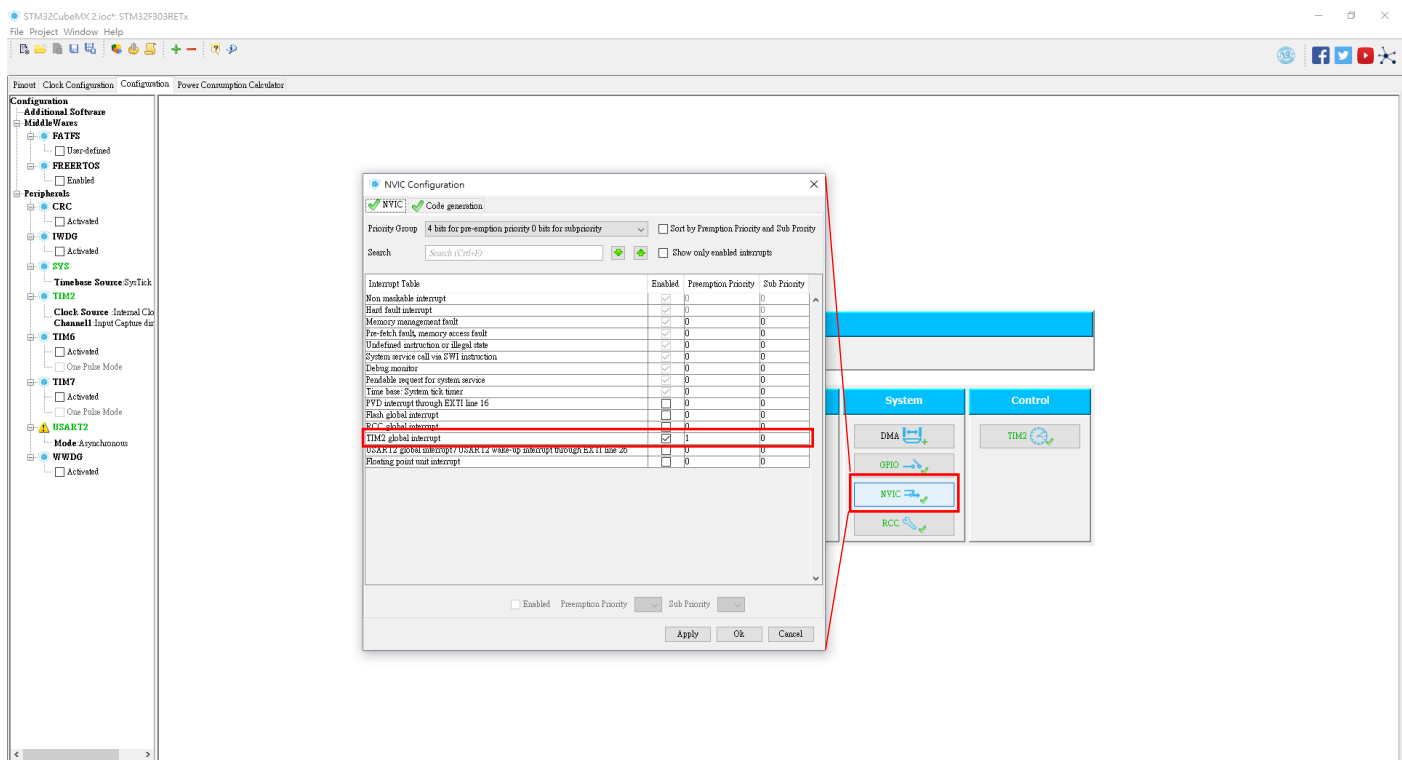
二、較好的做法 - TIM Input Capture(IC) Interrupt



與上個範例不同: PA0 為 TIM2_CH1 的 Input capture



下方 Polarity selection 為 Both edge，讓上升沿和下降沿都能觸發 IC。



由於我們等等會在 TIM2 中斷裡面使用 HAL_Delay，所以我們把這個中斷的優先權降低為 1

```

/* USER CODE BEGIN 0 */
uint32_t t1=0,t2=0,tmp=0;
//此中斷會在 TIM2_CH1 接到的 PIN 腳有 rising edge 或 falling edge 時被呼叫
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef* htim){
    if(htim->Instance==TIM2){
        tmp=HAL_TIM_ReadCapturedValue(&htim2,TIM_CHANNEL_1);
        //若是上升沿，記錄至 t1
        if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)==1){
            t1=HAL_TIM_ReadCapturedValue(&htim2,TIM_CHANNEL_1);
        }else{
            //若是下降沿，記錄至 t2
            t2=HAL_TIM_ReadCapturedValue(&htim2,TIM_CHANNEL_1);
            double distance=0;
            if(t2>t1){
                distance=(t2-t1)/(double)58;
            }else{
                distance=(__HAL_TIM_GET_AUTORELOAD(&htim2)-t1+1+t2)/(double)58;
            }
            int integer=(int)distance;
            int point=(int)((distance-integer)*100);
            char tosend[20]={0};
            sprintf(tosend,"%d.%02d\r\n",integer,point);
            HAL_UART_Transmit(&huart2,tosend,sizeof(tosend),0xffff);
            //進行下次測距的 Trig
            HAL_Delay(400);
            HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,1);
            HAL_Delay(1);
            HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,0);
        }
    }
}
/* USER CODE END 0 */

```

```
/* USER CODE BEGIN 2 */  
//開始等待 TIM IC 的中斷  
HAL_TIM_IC_Start_IT(&htim2,TIM_CHANNEL_1);  
//進行第一次測距的 Trig  
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,1);  
HAL_Delay(1);  
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,0);  
/* USER CODE END 2 */
```

完整程式請見 [Github](#)