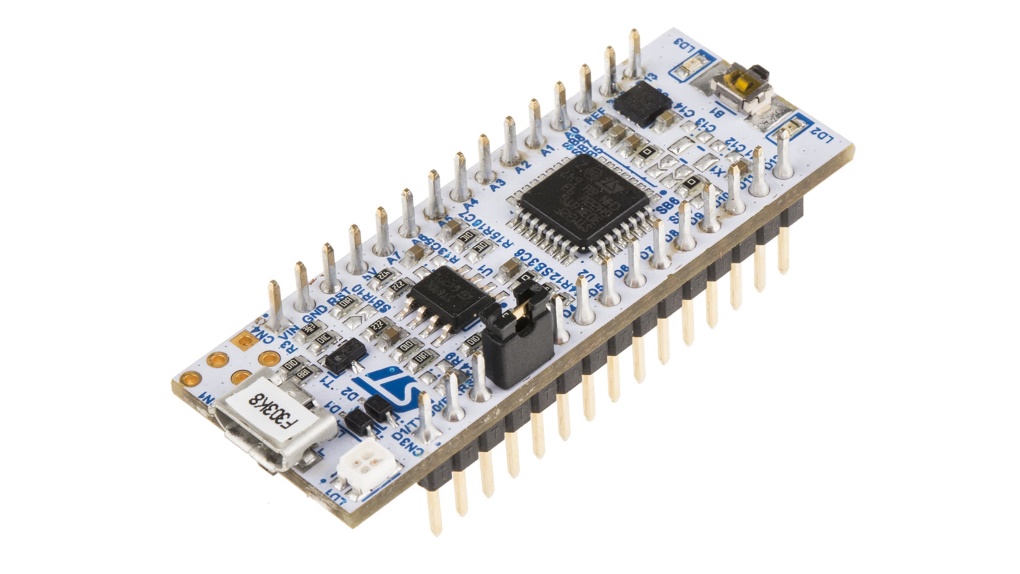
淡江大學電機工程學系

微處理機實驗結報



班 級：電資三

姓 名：李俊逸

學 號：411440521

日 期：2024年11月10日

1. **實驗名稱**

期中考試(三題)

1. **實驗目的**
2. 了解STM32微控制器與系統環境建置。
3. 熟悉 EXTI (外部中斷) 與 Timer PWM 的設定及使用。
4. **實驗材料**

* **軟體:** STM32CubeIDE 1.16.0
* **硬體:** STM32開發版

1. **實驗結果**

* 第一題

程式流程:

1. 使用按鍵中斷偵測按鍵是否按住，若按住則開始閃爍效果。
2. Timer 中斷每秒切換 LED 的亮或暗，並通過 PWM 調整 LED 亮度。
3. 按鍵釋放時將 LED 亮度設定為 0（暗）。

流程圖:

一張含有 文字, 螢幕擷取畫面, 圖表, 設計 的圖片

自動產生的描述

實作部分:

* main.c from EXIT.c -> HAL\_GPIO\_EXTI\_Callback()

|  |
| --- |
| if(HAL\_GPIO\_ReadPin(B1\_GPIO\_Port, B1\_Pin)) { // PC13 down  HAL\_TIM\_PWM\_Start(&htim3, TIM\_CHANNEL\_3);  } else {  HAL\_TIM\_PWM\_Stop(&htim3, TIM\_CHANNEL\_3);  } |

* Timer\_PWM.ioc

|  |
| --- |
| * + Button   GPIO -> PC13 -> User Label -> B1   * + Timer 3   Prescaler 42000-1  Counter Peroid 6000-1  Pulse 2000 |

* 第二題

程式流程:

1. 按鈕按下時觸發中斷，將 PWM 亮度逐步調高，達到峰值後逐步調低。
2. 在 PWM 佔空比達到最小時結束呼吸燈效果。

流程圖:

**一張含有 文字, 螢幕擷取畫面, 圖表, 設計 的圖片

自動產生的描述**

實作部分:

main.c from Timer\_PWM.c -> main()

|  |
| --- |
| void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin){  uint16\_t i;  /\*tim3 LED1\*/  for(i=0;i<1000;i++)  {  htim3.Instance->CCR3 = i;  \_\_HAL\_TIM\_SetCompare(&htim4, TIM\_CHANNEL\_2, i);  HAL\_Delay(1);  }  for(i=1000;i>=1;i--)  {  htim3.Instance->CCR3 = i;  \_\_HAL\_TIM\_SetCompare(&htim4, TIM\_CHANNEL\_2, i);  HAL\_Delay(1);  }  HAL\_GPIO\_WritePin(GPIOB, LD3\_Pin, GPIO\_PIN\_SET);  } |

Timer\_PWM.ioc

|  |
| --- |
| * Button   GPIO -> PC13 -> NVIC -> Enable   * Interrupt Priority   NVIC -> EXTI line[15:10] interrupts -> Enable  NVIC -> EXTI line[15:10] interrupts -> Preemption Priority -> 2 (數字越小，優先級越高) |

* 第三題

流程圖

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

自動產生的描述

我沒做出來

LED1 : 把1000改成500 變成半秒為周期

LED2 : 把第一題的做法拿過來用，並在Timer Interrupt 中寫入開啟、關閉Timer的code來控制。

* **Prescaler 42000-1**

**84M / 42000 = 2000為1秒**

* **Counter Peroid 2000-1**

**一個周期1秒**

* **Pulse 1000-1**

**高電位0.5秒**

**實驗心得:**

通過此次實驗，我們熟悉了如何運用 STM32 的 Timer 中斷與 PWM 控制 LED 的亮度與閃爍行為。以 EXTI 和 Timer 結合 PWM 的設計思路，實現了 LED 的閃爍燈和呼吸燈效果。透過不使用 for 迴圈和 delay 函數，我們有效地提升了程式的運行效率與反應速度，這也讓我們深入理解了 STM32 的硬體中斷、定時器以及 PWM 在實際應用中的靈活性。