

LAB 07 - Requirement Description

- Youtube 影片

- Interrupts: <https://youtu.be/YqjC44qt4QU>
- Timer: <https://youtu.be/-NRGM66ue6k>

- Hackmd 連結

- Link: https://hackmd.io/@CM_CBTvtTuKNjJZ1yF2ptw/SyhtzeGM1l

- Lab requirements

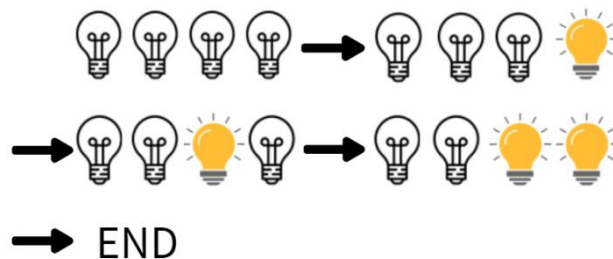
1. 基本題(70%)

- 題目敘述

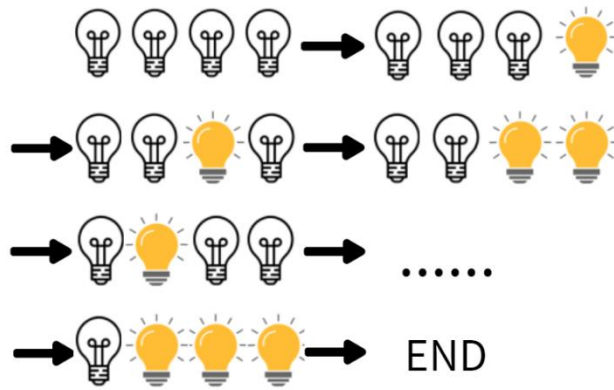
以四個 LED 燈泡，分別與 **RA0,RA1,RA2,RA3** 腳位連接，由左到右排列，使其代表一組 4 位元的二進位數值的 0-3 bit，並且將一按鈕與 **RB0** 腳位連接，當按下按鈕時，亮燈順序將切換狀態，分為狀態 1、狀態 2，與狀態 3 依序轉換。

- 狀態展示

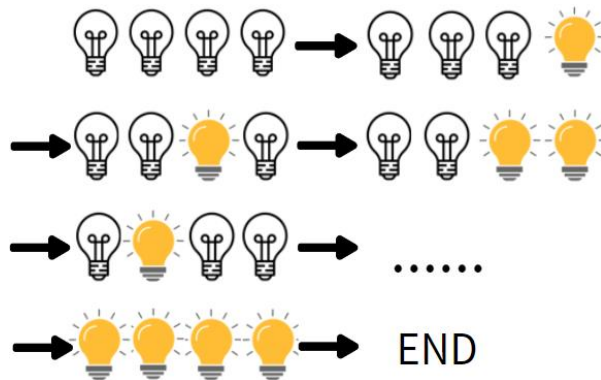
狀態 1(0 到 3 計數器):



狀態 2(0 到 7 計數器):



狀態 3(0 到 15 計數器):



➤ 操作流程

按下按鈕 -> 狀態 1
 按下按鈕 -> 狀態 2
 按下按鈕 -> 狀態 3
 按下按鈕 -> 狀態 1.....

➤ 評分標準

1. 使用 **ISR**，並在按下按鈕時觸發使燈泡改變狀態
2. 燈泡與按鈕需對應正確腳位
3. 以 **Assembly** 編寫

2. 進階題(30%)

➤ 題目敘述:

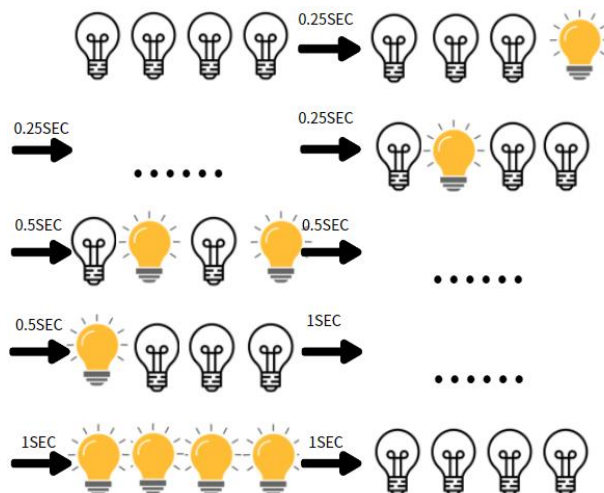
以四個 LED 燈泡，分別與 RA0,RA1,RA2,RA3 腳位連接，由左到右排列，使其代表一組 4 位元的二進位數值的 0-3 bit，並設計一 0 到 15 的環形計數器，每次間隔時間隨計數而變化，需使用 TIMER2 設計正確間隔時間，詳細間隔時間如下：

0-3:間隔 0.25 秒

4-7:間隔 0.5 秒

8-15:間隔 1 秒

➤ 過程示意



➤ 評分標準

1. 禁止使用 **DELAY** macro，其餘方法不限
2. 燈泡需對應正確腳位
3. 計數器須持續，而非做完一次流程即停止
4. 以 **Assembly** 編寫

3. 加分題(20%)

➤ 題目敘述:

以四個 LED 燈泡，分別與 **RA0,RA1,RA2,RA3** 腳位連接，由左到右排列，使其代表一組 4 位元的二進位數值的 0-3 bit，並設計一環形計數器，並且將一按鈕與 **RB0** 腳位連接，當按下按鈕時，計數器亮燈及間隔時間狀態均改變，詳細狀態如下

亮燈:

狀態 1:0 到 7 計數器

狀態 2:0 到 15 計數器

狀態 3:0 到 15 倒數計數器

時間間隔:

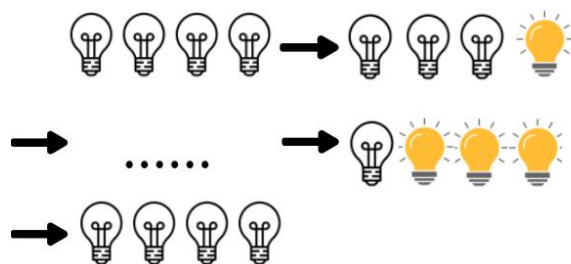
狀態 1:間隔 0.25 秒

狀態 2:間隔 0.5 秒

➤ 狀態示意圖

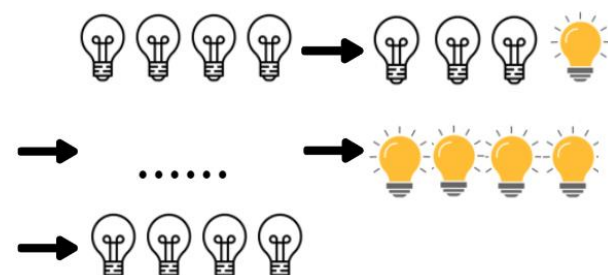
狀態 1

(間隔 0.25 秒,0 到 7 計數器)



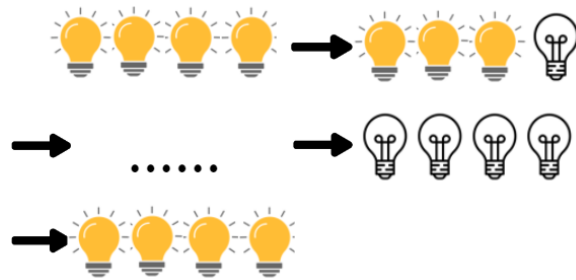
狀態 2

(間隔 0.5 秒, 0 到 15 計數器)



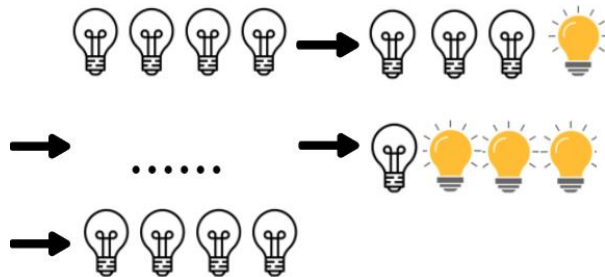
狀態 3

(間隔 0.25 秒, 0 到 15 倒數計數器)



狀態 4

(間隔 0.5 秒, 0 到 7 計數器)



.....

➤ 提示

總共狀態不只 4 種

➤ 評分標準

1. 使用 **ISR**，並在按下按鈕時觸發使燈泡改變狀態
2. 燈泡需對應正確腳位
3. 計數器須持續，而非做完一次流程即停止
4. 以 **Assembly** 編寫

LAB 07 - Requirement Description

- Youtube video

- Interrupts: <https://youtu.be/YqjC44qt4QU>

- Timer: <https://youtu.be/-NRGM66ue6k>

- Hackmd

- Link: https://hackmd.io/@CM_CBTvtTuKNjJZ1yF2ptw/SyhtzeGM1l

- Lab requirements

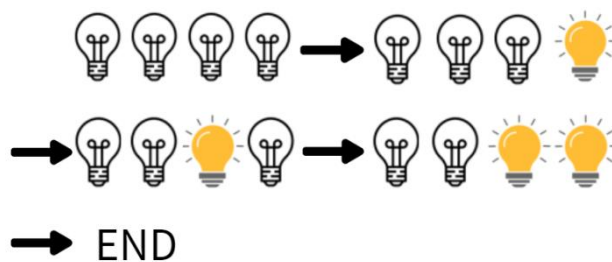
1. Basic (70%)

- Description:

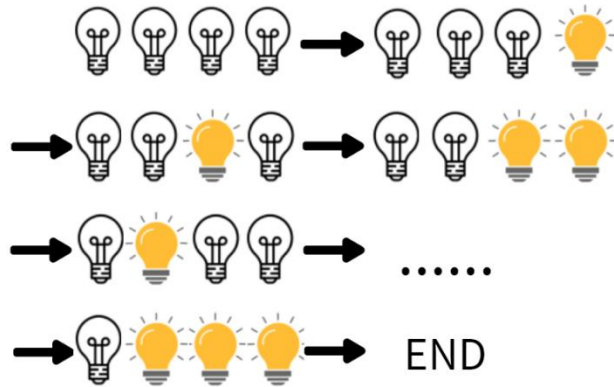
You will need four LED lights and one button, with the LEDs connected to the pins **RA0**, **RA1**, **RA2**, and **RA3** in sequence from left to right, representing the bits 0 to 3 of a 4-bit binary value. Additionally, a button is connected to the pin **RB0**. When the button is pressed, the state of the LEDs should be able to toggle.

- State demonstration

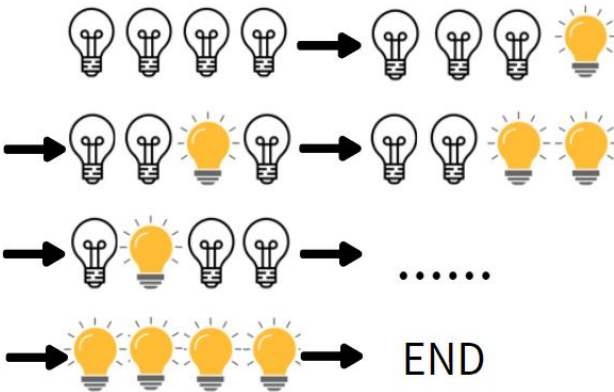
State1(0 to 3 counter):



State2(0 to 7 counter)



State3(0 to 15 counter)



➤ **Criteria:**

1. Use an **ISR** to trigger a change in the light bulb's state when the button is pressed.
2. The light bulb and button must correspond to the correct pins.
3. Write in **Assembly**.

2. Advanced (30%)

➤ Description:

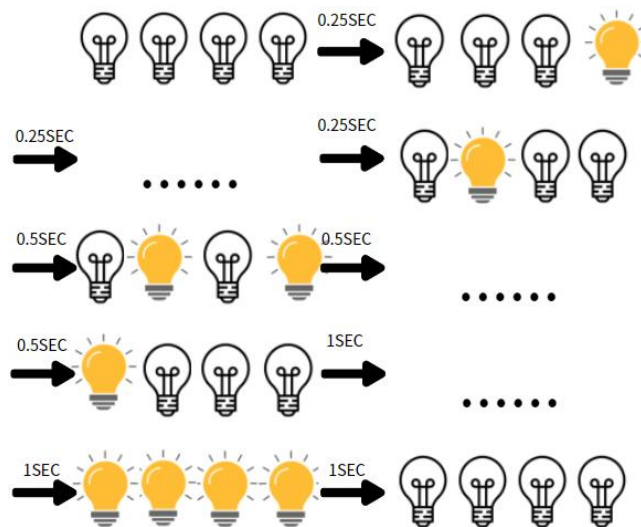
You will need four LED lights and one button, with the LEDs connected to the pins **RA0**, **RA1**, **RA2**, and **RA3** in sequence from left to right, representing the bits 0 to 3 of a 4-bit binary value. Design a 0 to 15 cyclic counter. The time interval between each count should vary according to the count value. **TIMER2** must be used to design the correct time intervals. The detailed intervals are as follows:

0-3: 0.25 seconds

4-7: 0.5 seconds

8-15: 1 second

➤ Demonstration:



➤ Criteria:

1. Don't use **DELAY** macro in this program, other methods are not restricted.
2. The light bulb must correspond to the correct pins.
3. The counter must run continuously, rather than stopping after completing one cycle
4. Write in **Assembly**.

3. Bonus (20%)

➤ Description:

You will need four LED lights and one button, with the LEDs connected to the pins **RA0**, **RA1**, **RA2**, and **RA3** in sequence from left to right, representing the bits 0 to 3 of a 4-bit binary value. Design a ring counter. Additionally, connect a button to the **RB0** pin. When the button is pressed, both the counter's lighting pattern and time intervals will change. The detailed states are as follows:

Lighting pattern:

State 1: 0 to 7 counter

State 2: 0 to 15 counter

State 3: 0 to 15 countdown counter

Time intervals:

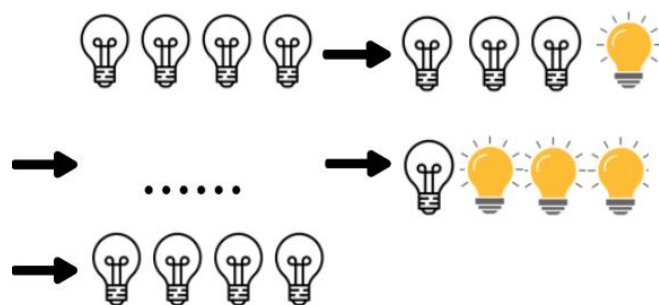
State 1: 0.25 seconds

State 2: 0.5 seconds"

➤ State demonstration:

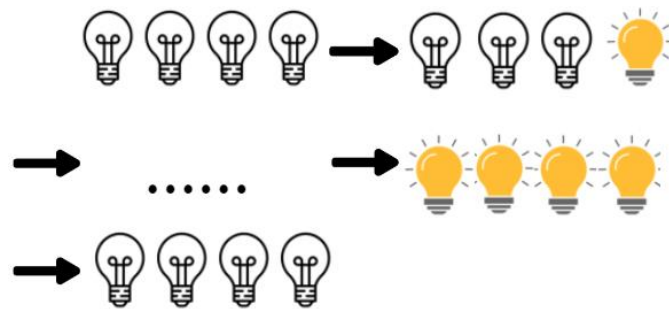
State1

(time interval:0.25sec,0 to 7 counter)



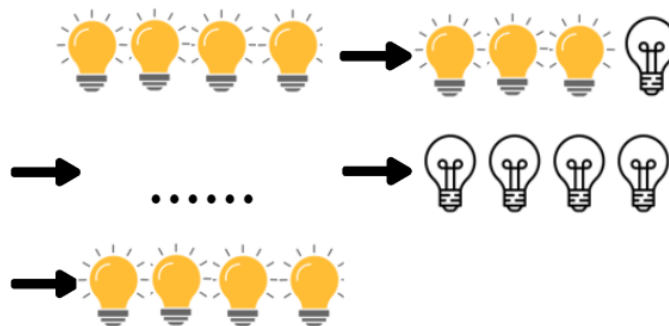
State2

(time interval:0.5sec, ,0 to 15 counter)



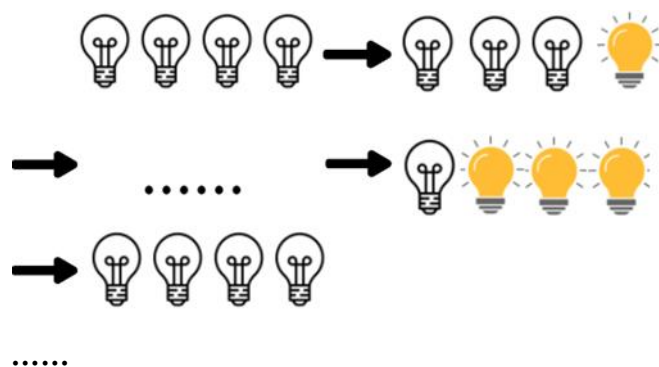
State3

(time interval:0.25sec, ,0 to 15 countdown counter)



State4

(time interval:0.5sec,0 to 7 counter)



➤ Hint

There are more than 4 states

➤ Criteria

1. Use an **ISR** to trigger a change in the light bulb's state when the button is pressed.
2. The light bulb and button must correspond to the correct pins.
3. The counter must run continuously, rather than stopping after completing one cycle
4. Write in **Assembly**.