

開關遊戲

Lights Out Game

第五組

1103309 王聖允

1101454 梁文璟

1101547 陳鈺橦

1101408 張鈞閔

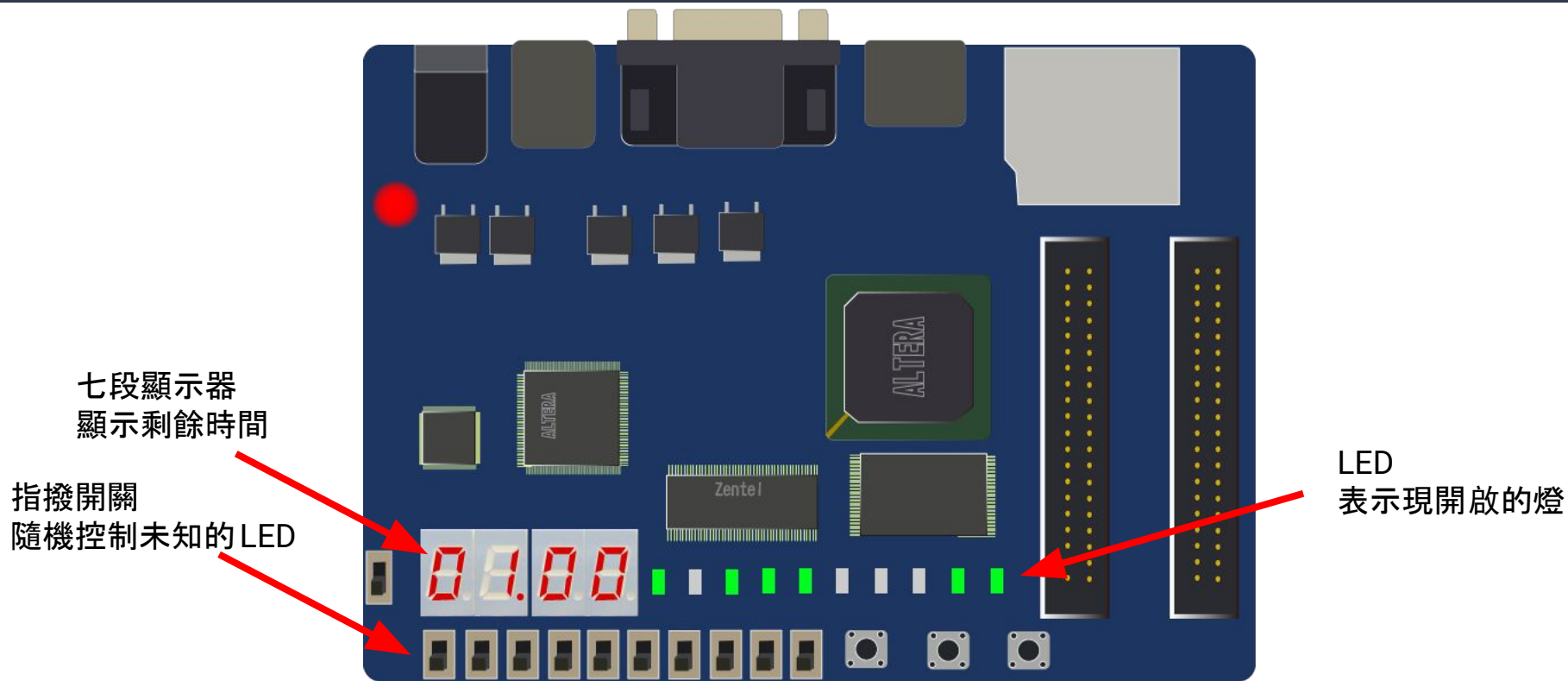
靈感來源與規則Demo



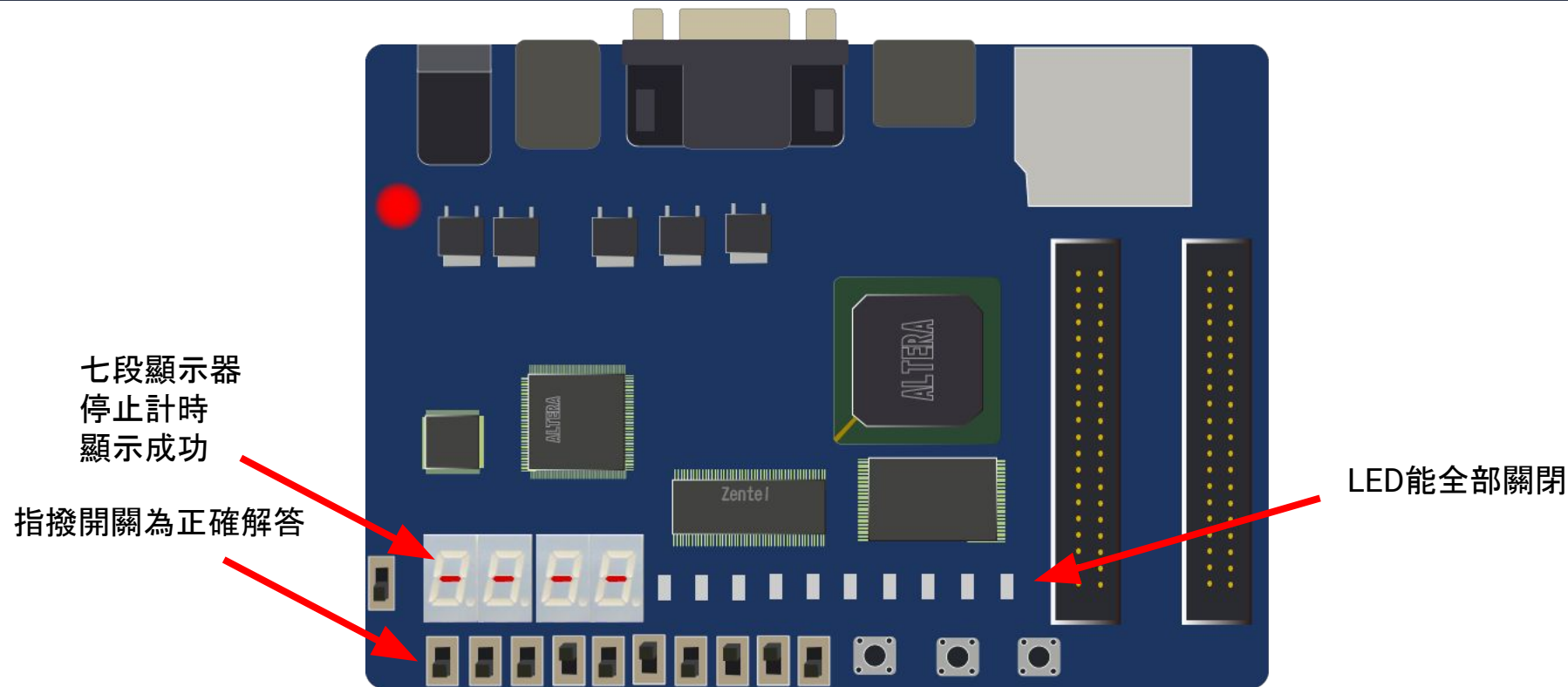
功能

1. 隨機產生題目 或 固定題目 兩種模式
2. 30秒破關時間
3. 使用七端顯示器呈現秒數 以及是否過關
4. 切換開關 變更多個LED之亮暗(對號電燈和左右相鄰的)
5. LED全亮為失敗, 全滅為成功

DE0 BOARD - 初始狀態

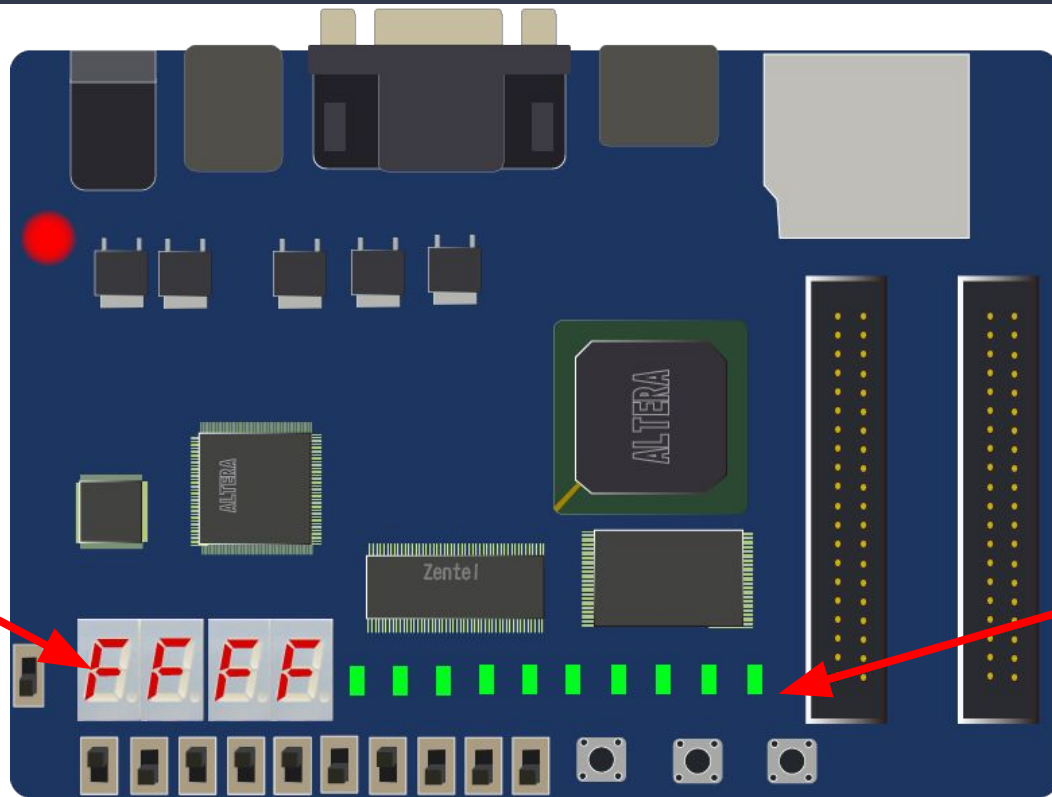


DE0 BOARD - 成功破關



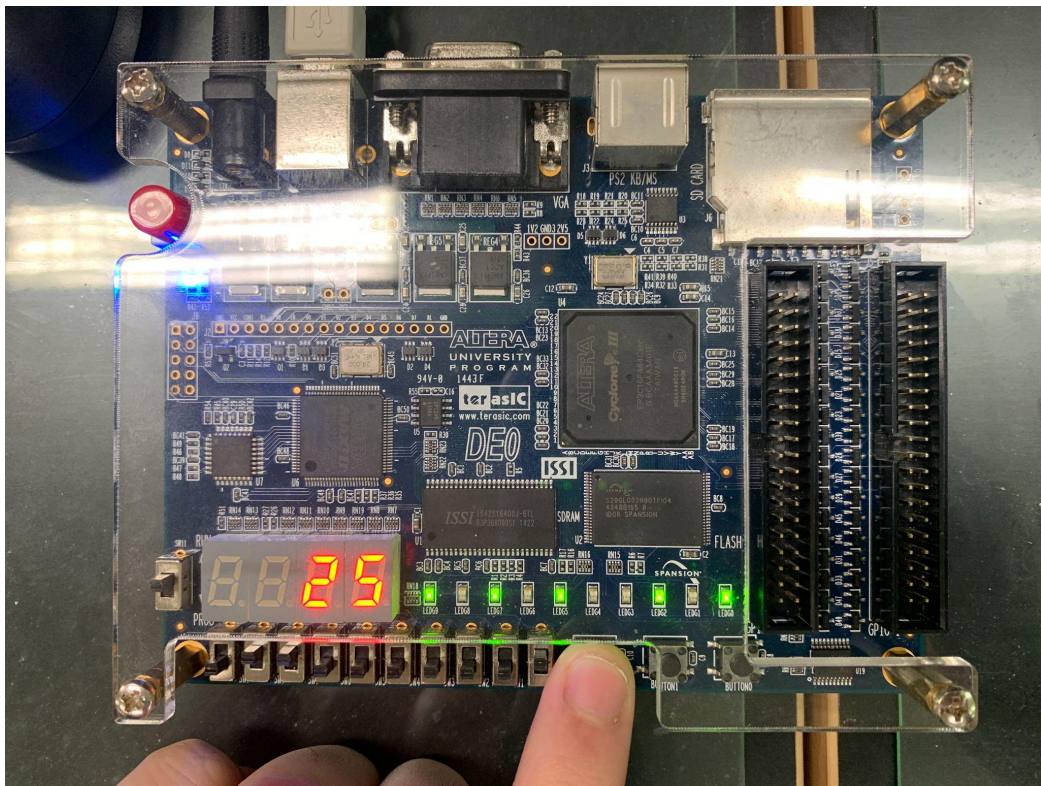
DE0 BOARD - 失敗或時間到

七段顯示器
停止計時
顯示失敗

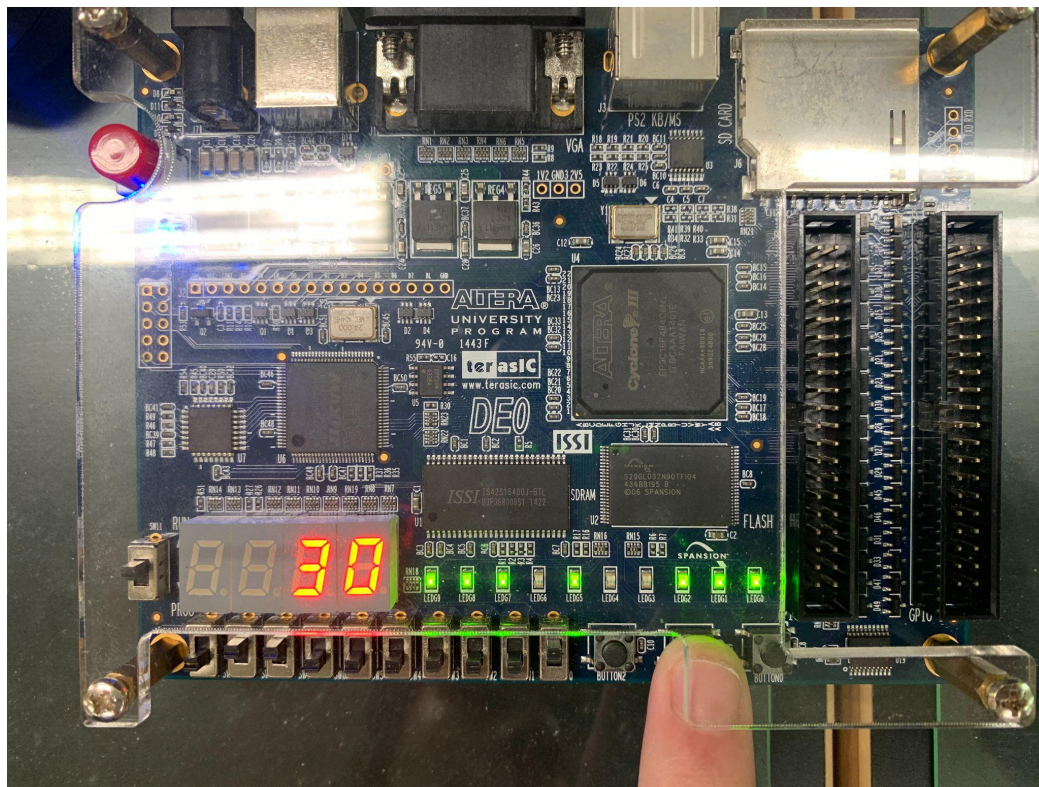


LED全部亮起時..

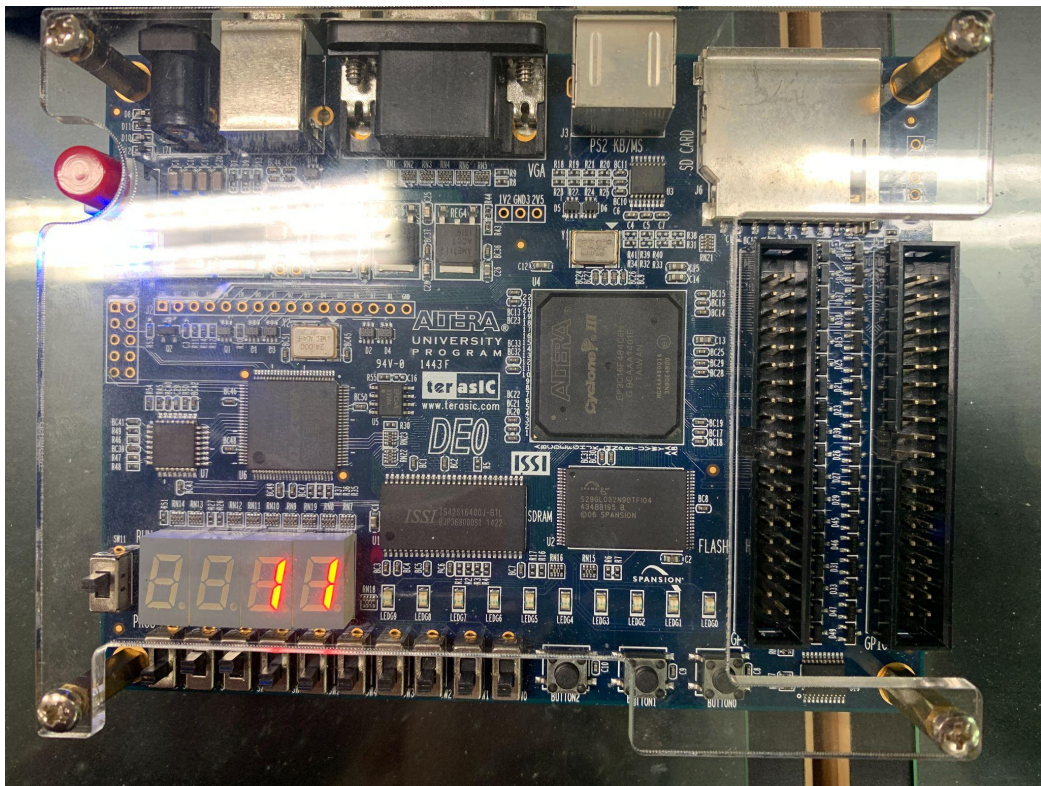
實際照片：固定題目解題中



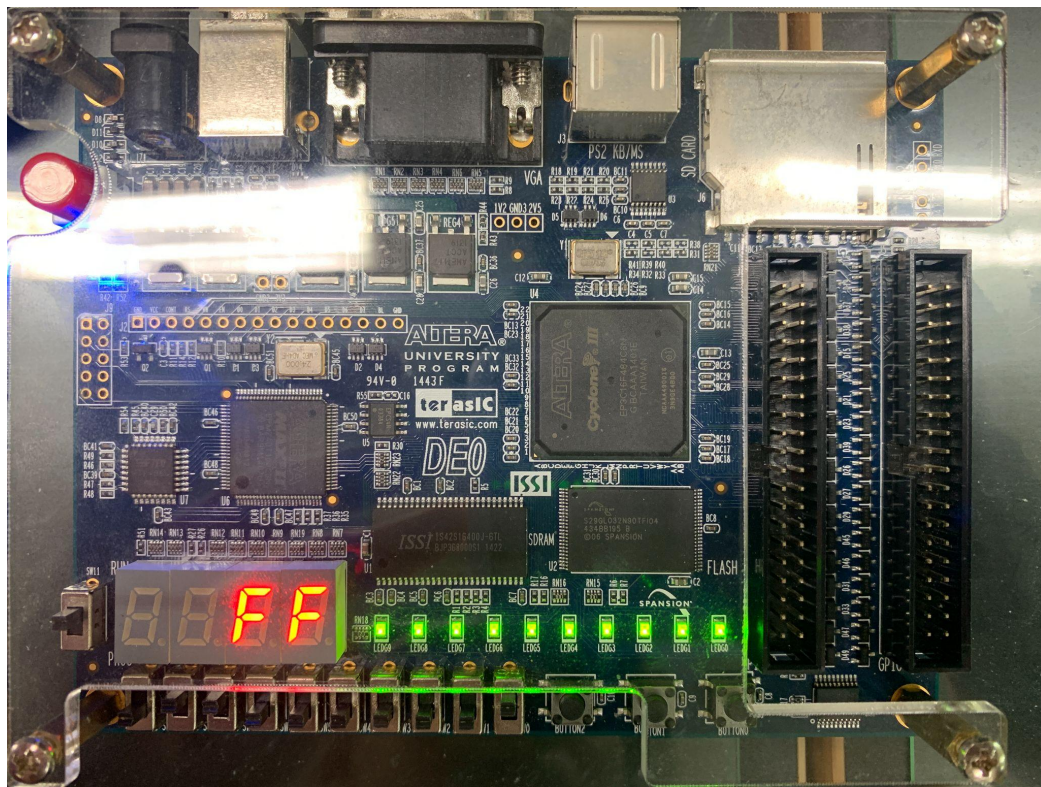
實際照片：隨機題目解題中



實際照片:成功解題(停止)



實際照片：解題失敗 (FF)



邏輯原理解釋

為了用一個開關控制多個電燈，並且要讓目前電燈狀態反向
(例如:「開關開」要變成「關開關」)，我們用XOR來達成

目前電燈: 10-bit Array來記錄 (名稱為 lights)
假設SW4要控制LED3、LED4、LED5

那麼偵測到SW4被切換時
 $\text{lights} = \text{lights} \text{ XOR } 10'b0000111000$

這樣即可將這幾個電燈的狀態達到反向

邏輯原理解釋

如何達成隨機題目？

從上一頁的解釋可以知道

每個開關都有對應的一串bits, 在撥動開關時會去跟目前狀態做XOR

(例如: SW4對應的bits就是10'b0000111000)

隨機題目時, 為了確保一定有解, 所以我們用程式模擬 隨機撥動幾個開關

也就是 隨機跟幾串開關對應的bits做XOR

這樣玩家遊玩時, 只要撥到剛剛程式模擬撥動的所有開關, 一定可以通關

實際上程式中達成方式:

建立一個10bits array, 在50MHz clock posedge加一(快速變動)

在Random出題鈕按下時取下這個array的值, 並看每一個bit

如果該bit是1的話, 就拿對應的bits串去做XOR

(例如: array[4]為1, 就將lights與10'b0000111000做XOR)

這樣就達成有效Random, 且一定有解

問題解決與優化

一開始我們讓開關與電燈切換時機與50MHz的posedge綁定
結果發生過度頻繁XOR造成燈號不穩定(有時開關可能會造成震盪)
後來我們改成只要某switch有變動時, 將對應變數設成 1
並在一段時間後 統一將需要變動的電燈一次改變
讓電燈狀態穩定許多

謝謝聆聽