

我做的是電子密碼門，目的是為了加強居家的防護，以及可以讓使用者不用攜帶鑰匙，避免鑰匙不見而無法開門的窘境。

以下是我使用到的元件：

1. 步進馬達
2. 超音波 sensor
3. 遙控器
4. 紅外線 sensor
5. 蜂鳴器
6. LCD
7. Button

功能說明：

當超音波 sensor 感測到附近有人時，便會將 LCD 亮起，並要求使用者輸入密碼。使用者透過遙控器對著紅外線 sensor 輸入密碼，若輸入正確步進馬達就會轉動來讓門開啟，並且在 LCD 上顯示 Pass!。

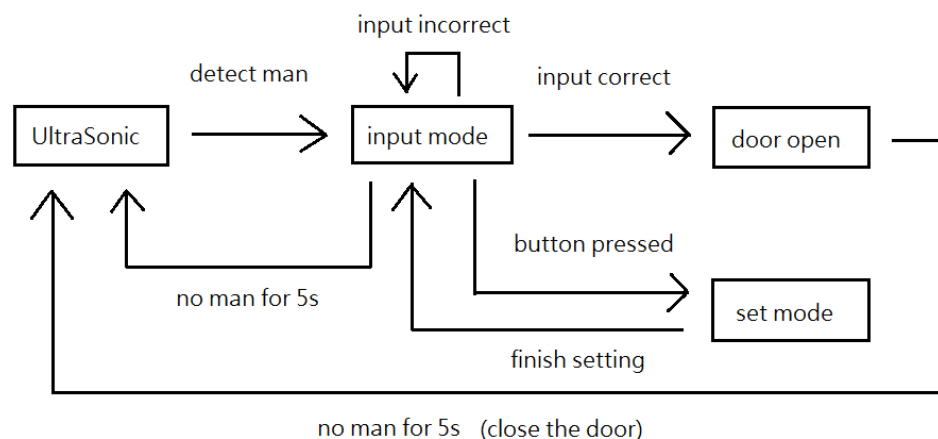
若輸入錯誤則會在 LCD 上要求再次輸入，如果連續錯誤 5 次，蜂鳴器便會響起，並且有一段時間無法輸入密碼。

而當超音波 sensor 發現附近沒人超過 5 秒，便會將 LCD 關閉，同時如果門是開著的，就會把門關上。

使用者可以透過門後面的 button 設定密碼，按下 button 進入 setting mode，使用者就可以設定新密碼。

密碼是由 0~9 組成的，長度為 1~16，+號是確認鍵，每次輸入完要按下+號。

以下是用圖來表示大概的結構：



其中 input mode 和 set mode 的轉換，我是透過 RTOS 的 priority 的差異來達成。Set mode 的 priority 較 input mode 高，但在 setup 的時候便將 set mode suspend，

然後藉由 **button interrupt** 的方式，在一按下 **button** 的時候便將 **set mode** 用 **resume** 來喚醒，然後在 **set** 結束時再次 **suspend set mode**。

以下是這次遇到的困難：

1. **IRremote** 和 **buzzer** 的 **tone** 有衝突，好像是因為 **timer** 用到同一個，因此我將 **IRremoteInt.h** 做了修改，裡面原本就有備用的 **timer**，所以我只是做了 **comment** 和 **uncomment**，在第 70 和 71 行。

修改前：

```
// #define IR_USE_TIMER1    // tx = pin 9
#define IR_USE_TIMER2      // tx = pin 3
```

修改後：

```
#define IR_USE_TIMER1    // tx = pin 9
// #define IR_USE_TIMER2      // tx = pin 3
```

2. 我原本想將超音波 **sensor** 的部分獨立為一個 **task**，**priority** 設為和 **input mode** 一樣，可是這樣做了以後，**run** 起來發現進不了超音波 **sensor** 和 **input mode** 的 **task**。最後只好將超音波 **sensor** 的部分放到 **input mode** 的地方一起做。

3. 原本輸入密碼的部分我想使用薄膜鍵盤，可是發現 **port** 會不夠用，所以才改用遙控器的方式。