《Learn To Use “Buttons”》

按鈕這個元件看似簡單，但是其實也是大有學問。

首先是因為裡面有彈簧，因此每次按下去以後，由於彈簧的震動效應，下方的電線依然可能反覆被接通；因此，我所認定的一次，可能對arduino而言是3~4次輸入；因此我們必須使用debounce技術使得每次點擊的次數和arduino接到的訊號是相同的，而debounce換言之就是讓arudino接到第一次訊號後小睡片刻，因此之後因為彈簧效應的接通就不會被讀取到。

然而，這樣卻又引發一個問題，如果我今天需要多次點擊，該怎麼辦？網路上許多的範例即使能做到double-click，但假使我要更多像是penta-click之類的怎麼辦，而且每次點擊也都必須濾掉個自的bounce訊號。

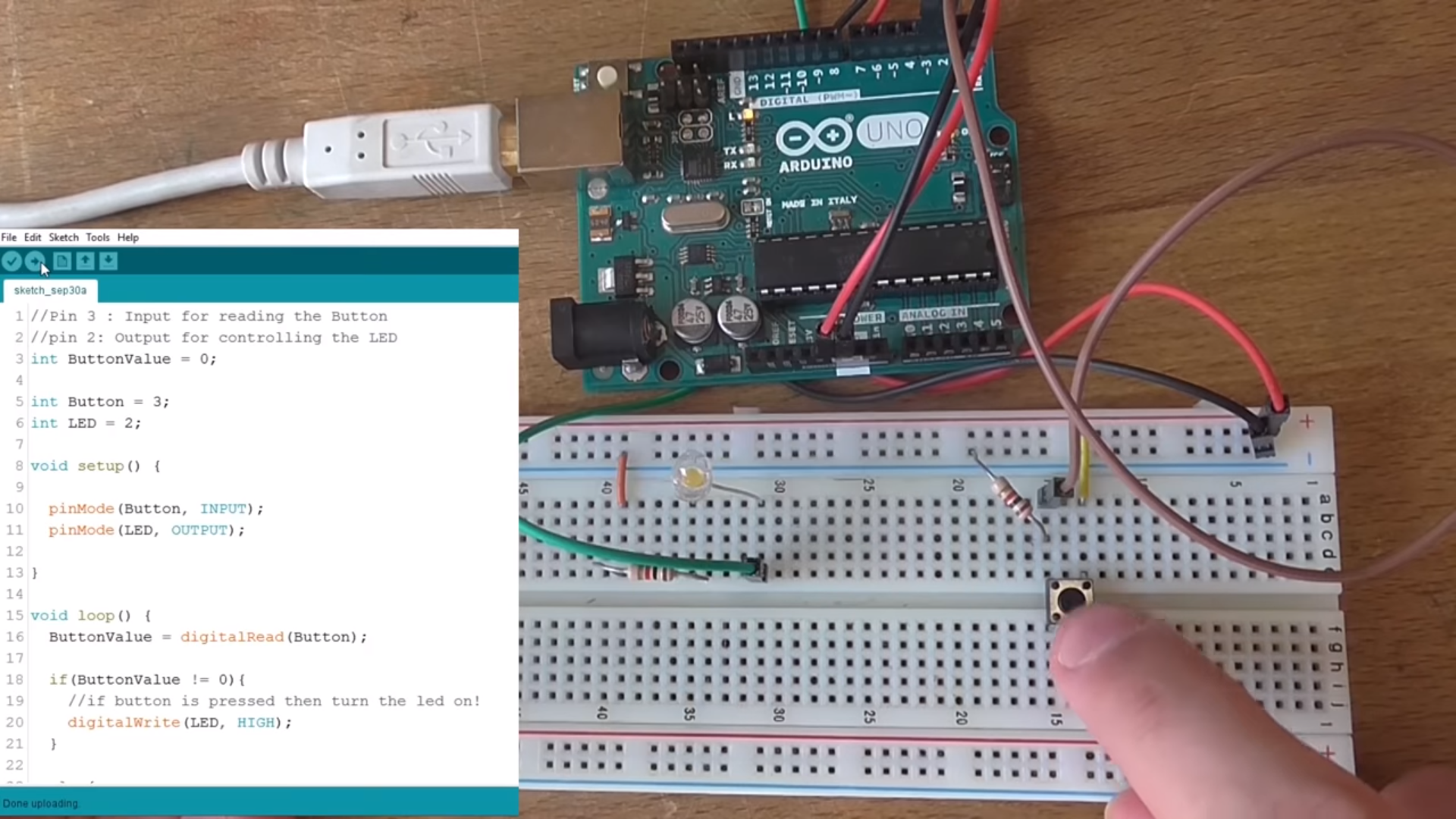
再者，有了多次點擊自然少不了長按hold。長按也可分為稍微hold一下，跟我真的hold很久。

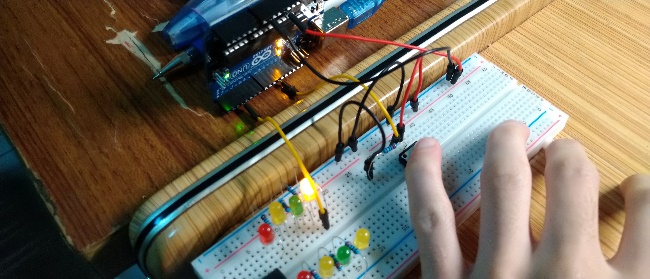
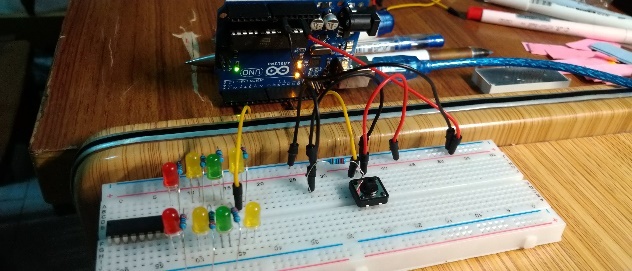
之後，最後還可以考慮一個，就是與瞬時的multiple-click不同，也和長時間的hold不同的；所謂的continuously toggle，也就是說，我可能每x秒每x秒固定連續點擊按紐，在連續點擊y次後開始作用該功能(為了區分(many times mono-click))。

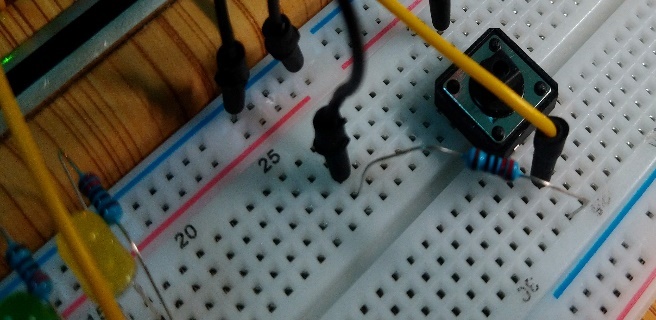
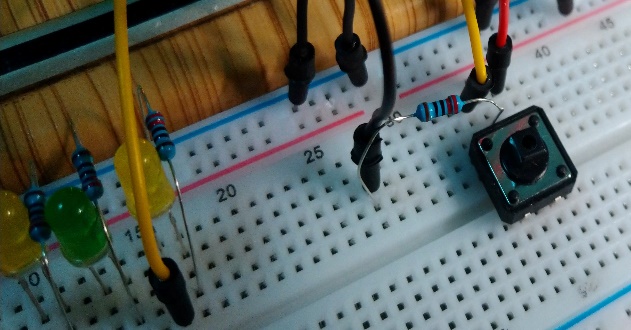
最後，人的每次點擊總會有一定的誤差，因此我也要設計一個簡單的時間間格區間，來表達哪一個區間內的連續點擊是double-click和triple-click的區分；哪一個區間內的間隔是toggles而非many times mone-clicks；還有如果我一下toggle一下multiple-click一下hold是否有辦法區分出來？

**因此接下來的專案將詳細探討這些可能的技術層面，已達到少少的按扭、多多的功能；讓不起眼的按扭也可以在arduino中有了大放異彩的可能。不過屆時會依照自任專案完成度來決定是否終止計畫，也預防自己可能會太過鑽牛角尖在細節上。**

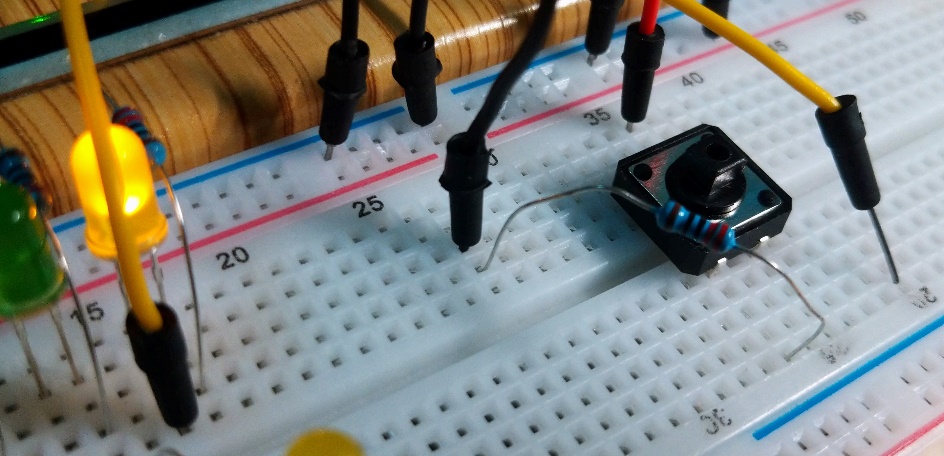
“\*Practice\_One.ino” learn to use button element

<https://www.youtube.com/watch?v=ksNbEuhO4fU>

簡單的第一個範例，先學習怎麼使用button



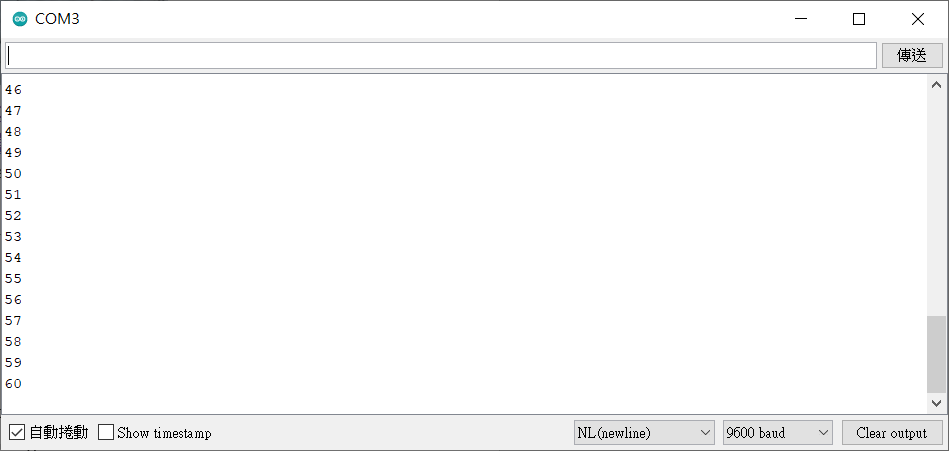
由此可知我的按紐內部是怎麼樣的構造



**奇特的是，當電路呈現斷路時(PIN 11應該沒有電流流入)，但是結果顯示digitalWrite(11)時，卻是有電流，所以燈亮了。**

“\*Practice\_Two\_v1.ino” to check for bouncing phenomenon

再來是，為了確保真的有所謂的bounce現象，我們做一個簡單的機制。每當按鈕按下便記數一次，等到10就亮LED，只要在燈亮的時候，我們既的次數少於10就可以知道有所謂的bounce現象。



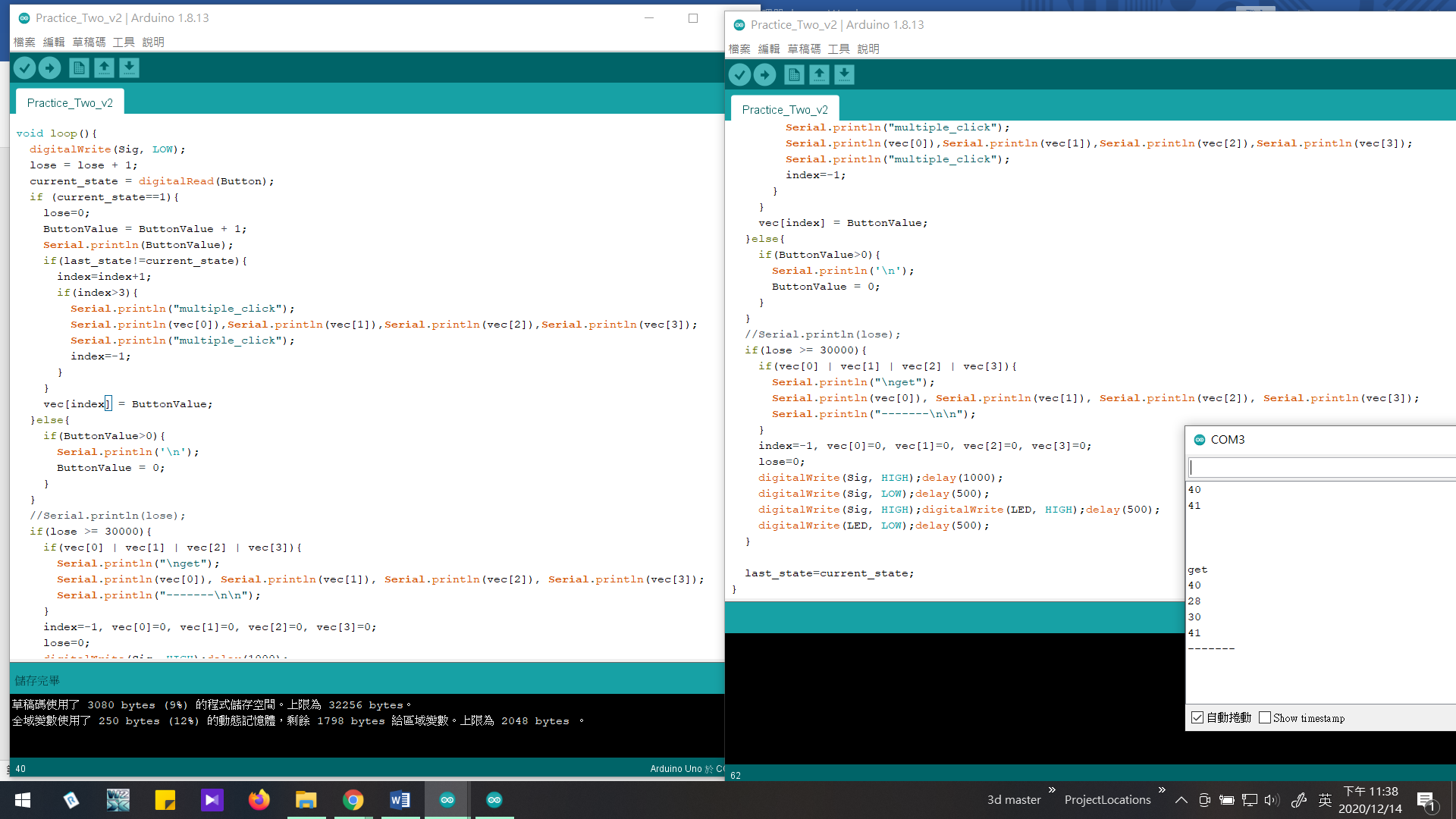
而有一次呢，我長按了一下，arduino卻認為我按了60下。



當我認為我以迅雷不及掩耳的速度按下的時候，arduino覺得我按了20下。

**然而我認為此現象並非所謂的bounce現象，而是因為當我按下按鈕，電路接通就開始記數。所以此現象應該是，電流跑得太快了的問題。而真正的bounce現象，我認為應該要用儀器才能測量出來。**

不過，不論哪個原因，現在要解決的是這個問題。

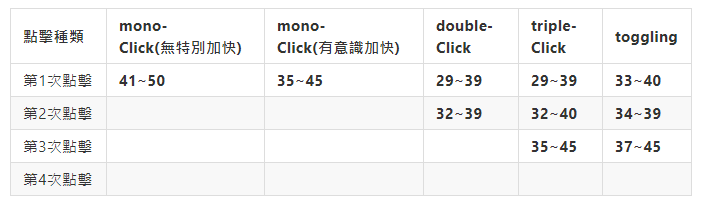


“\*Practice\_Two\_v2.ino” to analysis the counts of each press(up to 4)(extensable)

在一連串的腦力激盪後，想出了一個方法，可以記錄每次點擊時arduino的記數數目。然後為了使每次的紀錄可以記錄在陣列中，做了一個外部迴圈計時器，並以LED燈號為信號。每當信號跑至O時， (黃亮紅亮/黃暗/紅暗/O/紅亮/紅暗)，必須立刻開始按按鈕，如此就可以精確測出3次點擊中的記數的數目(第四次的會不准)。若要測第4次，擇要擴大陣列大小以及外部計時迴圈的數字。

然後也做了個簡單的統計，目前可以記錄4次連擊中，每種不同的點擊數字區間。也因為有了這篇程式的測量數據，我認為加以調適，應該可以做出各種multiple-click的判斷器；再者，hold短則150長則200~300，因此也能夠做出各種hold的判斷器；toggle通常會有一個特別的現象，尤其是在要求連續toggle多次後再開始做動作，每次的點擊上，通常記數數量不會相差太多，甚至可能會有重複，因此也比較好判斷。

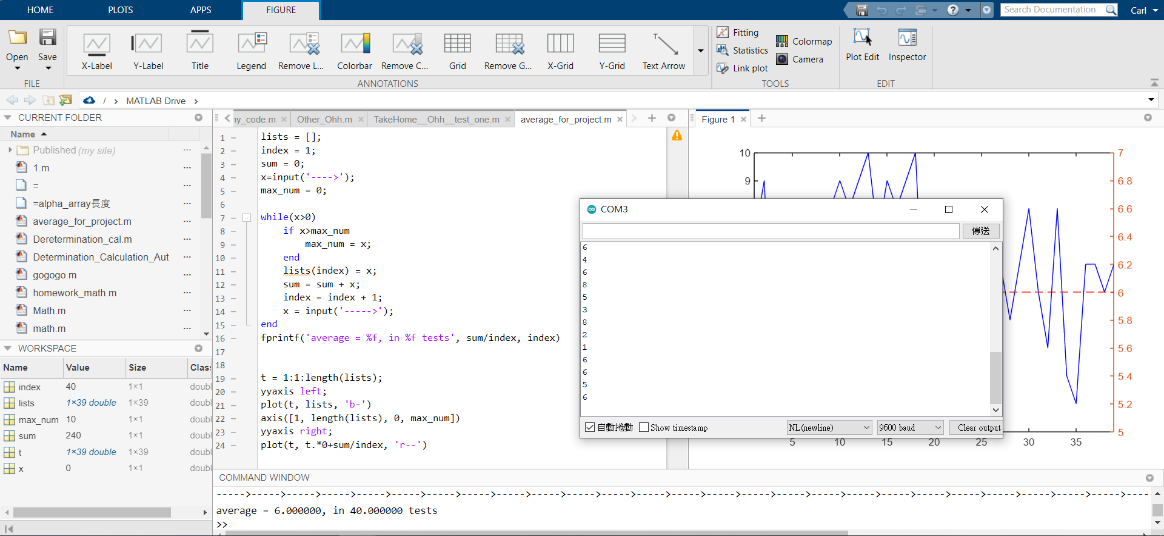
簡單來說，這次實驗到這裡，已經差不多了。雖然起初進階的要求自己沒有達到，但我相信這是時間的問題了，畢竟方向已經出來了。接下來，我想要找找看是否有已經內建好的按鈕模組，以省下我自己去做filtering的動作。

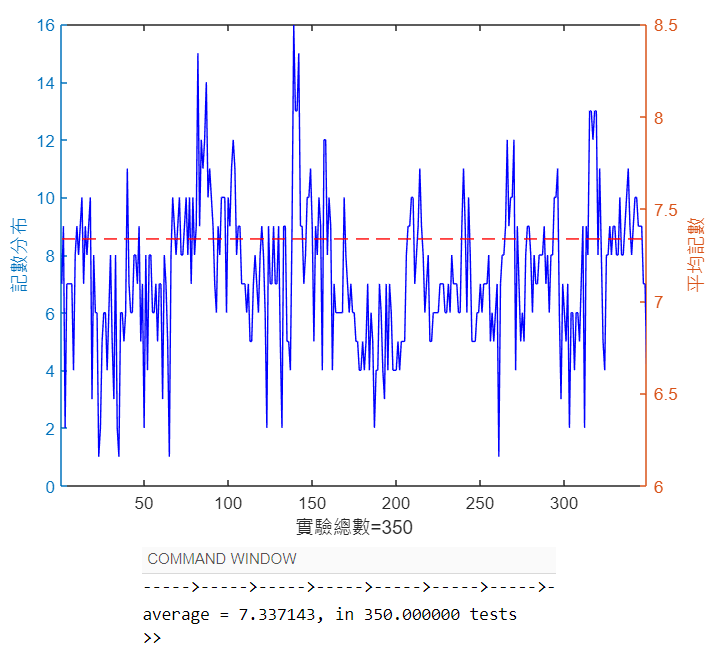
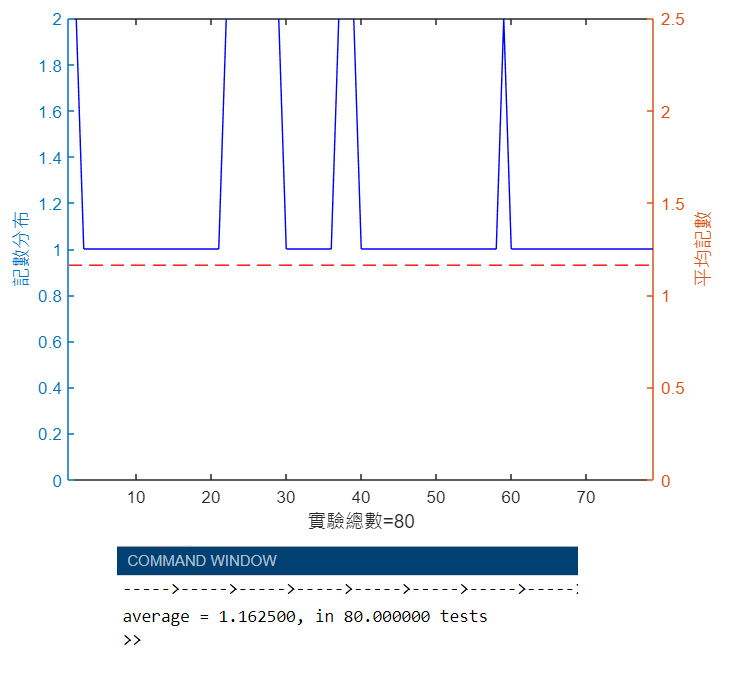


圖表：【註】實驗都只以我的手速為準，且實驗次數低，僅做大致參考。

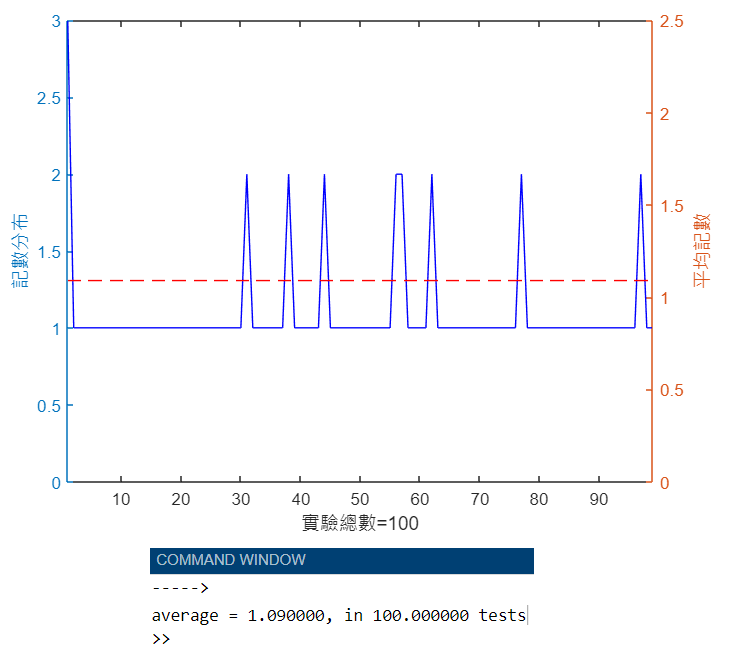
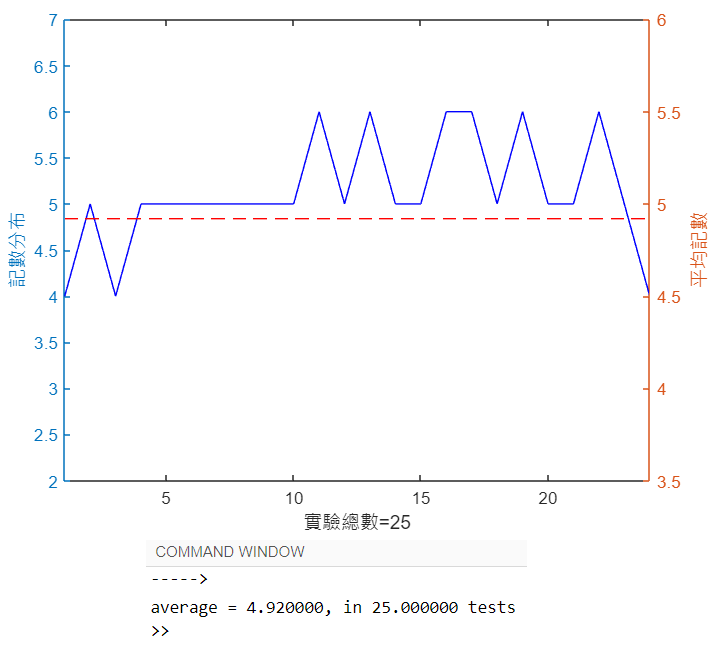
“Practice\_Three.ino” try to input a number to Arduino

運用arduino每行輸出搭配matlab可以快速畫出圖表



此時delay(10) 此時delay(100)

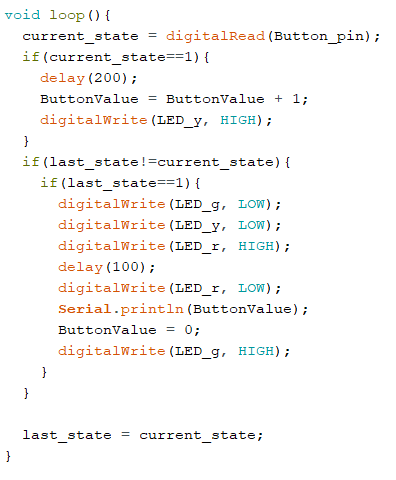
此時delay(200)

(左) 除非真的按住特別久，不然基本上不會有問題，都會是記數為1。

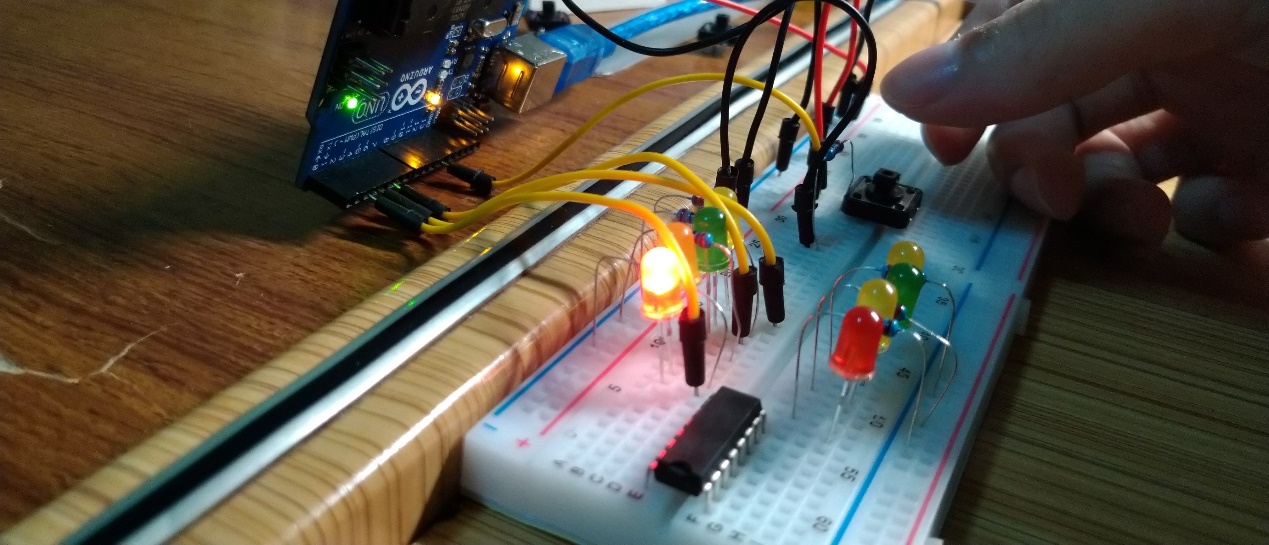
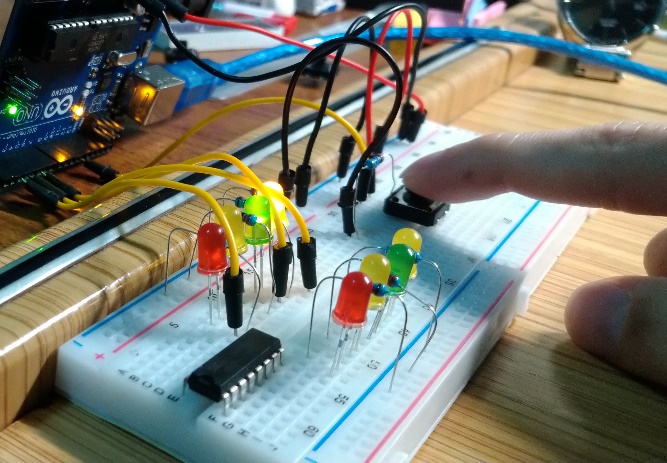
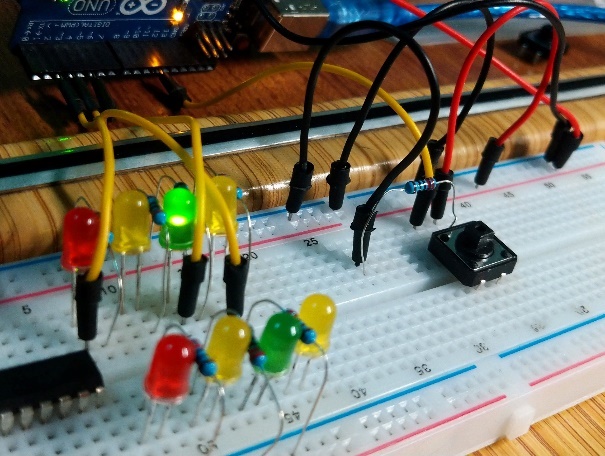
(右)平感覺長按，大多在5上下；可以做跳號功能。

如此一來已經可以做到簡單的記數字了。

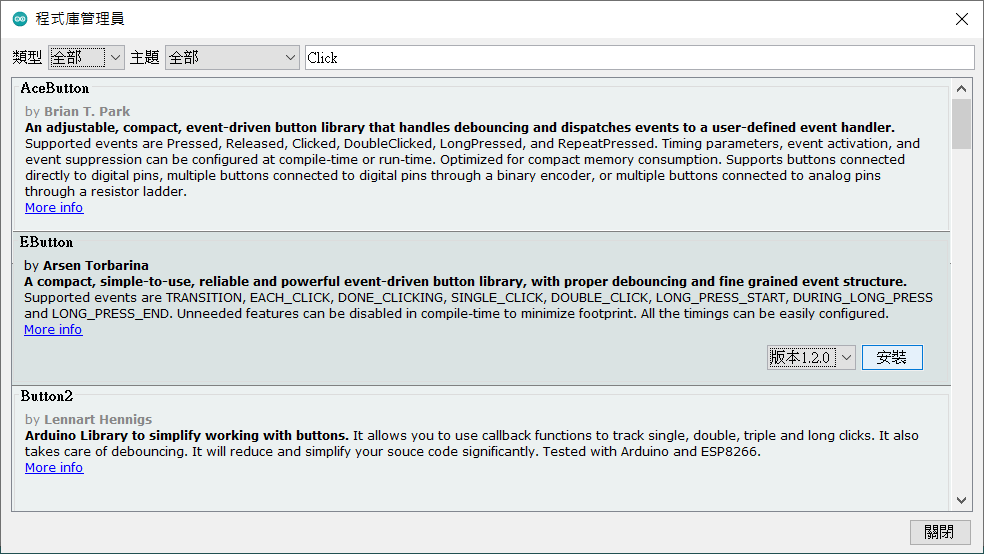
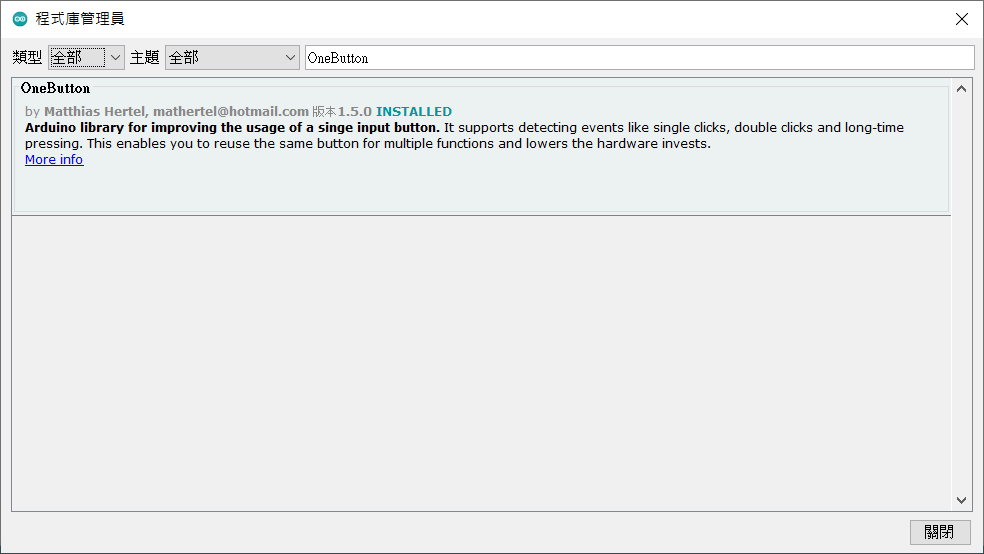
**利用按鈕按數字到4-digit-7segement-LED-display裡面，並結合SN74HC595N以省下arduino腳位，正是”番茄鐘”(即本專題)的第一個功能。**

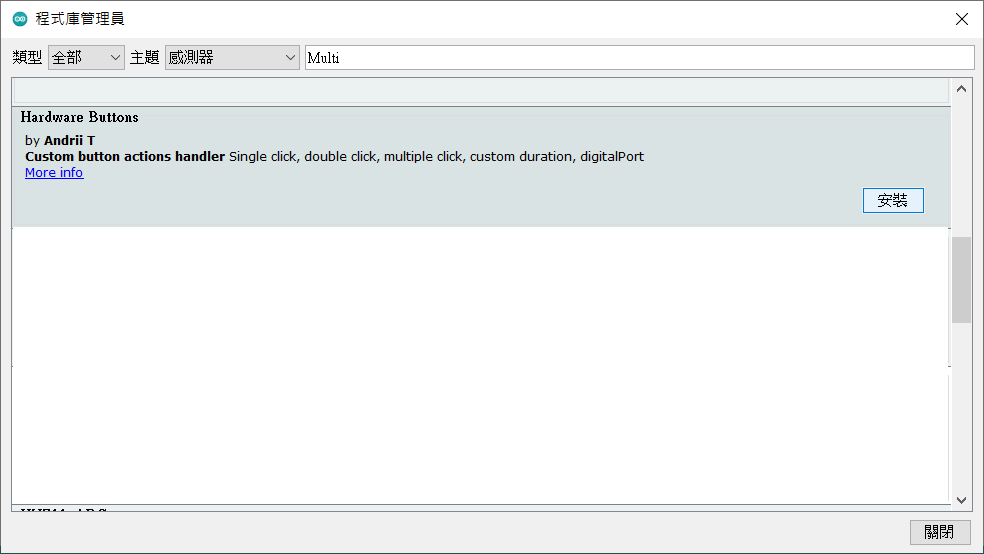
**此段程式給足了需要的控制力**

上文的delay(100)改為delay(200)有很棒的展示效果。



“Practice\_Four.ino” try to control button with others external libraries





安裝了以上5個library來進行測試。

心得：

我認為HardwareButton在Button的宣告上做了相當好的一個設計，它提供了按鈕物件一般使用時的簡單宣告用法也提供了搭配指標的作法，對於一個大型專案來說，如果要把許多小物件歸納為一組class來控管的話。我認為使用HardwareButton Library可以在搭配上有非常大的幫助。

至於AceButton的話，我認為加入了所謂的EventHandler的功能更是讓只用者對Button的設計可以運用得更充分。大致瀏覽一番，如果我沒搞錯的話，應該是對於每個Button物件都可以customized一組各自獨立的events( buttonConfig->setFeature )去監控不同事件。我認為這個專案可以把按鈕玩到相當出色，每個按鈕依據需求加入不同features然後依照EeventHandler中流程去執行運算。

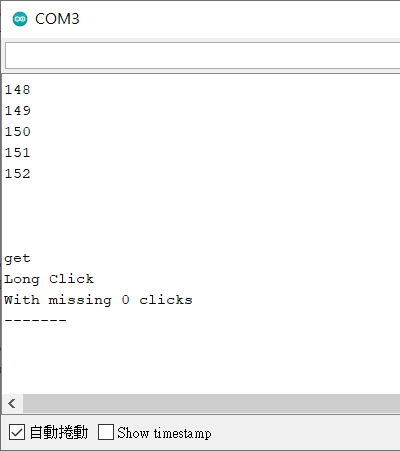
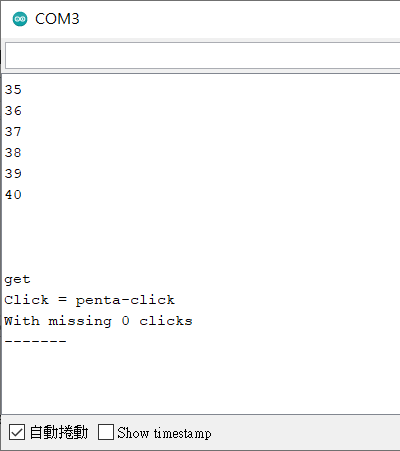
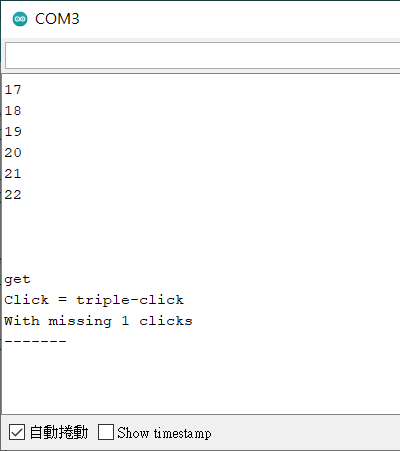
而我最後選擇的則是使用OneButton來做測試。因為我認為OneButton把程式碼化簡得相當簡單，而且和HardwareButton相同的是，他可以把自訂函數的位置傳入另一個函數並一起觸發。對於此刻上不需要把按鈕玩得花里胡哨的我而言，OneButton可以說是首選。

<https://github.com/mathertel/OneButton>

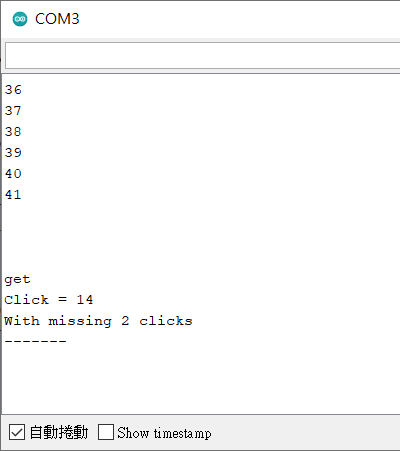
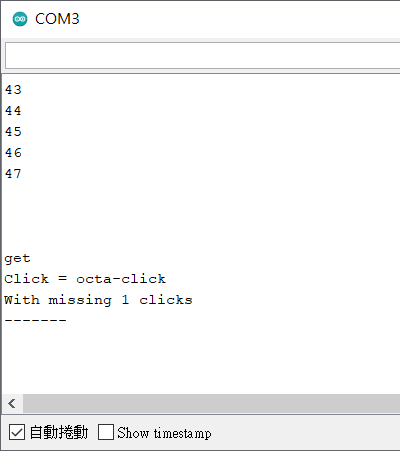
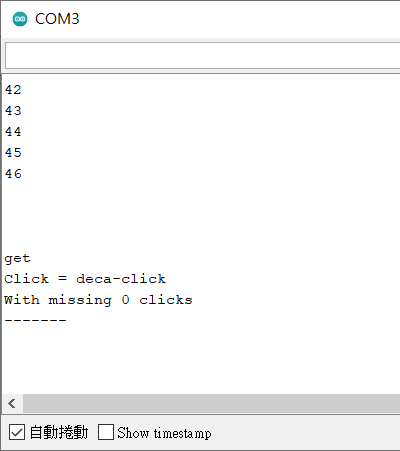
**然而令人感到相當失望與生氣的是，放在git-hub上面的竟然有點不如我的預期。萬念俱灰下，我覺得既然是自己的專案，就不應該偷懶；況且我在” Practice\_Two\_v1.ino”自己的練習程式中所做的state\_detection還更勝一籌，因此我有信心可以做出我夢想中的button\_controller。**

“Practice\_Four.ino” using OneButton Library to control my button 🡪 **Failed For External Library Bugs**

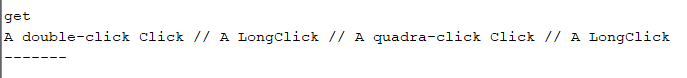
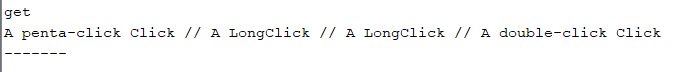
“Practice\_Five.ino” using my own way to design a simple controller ( and publish on my Git and Arduino Library as other contributors do? )



(左) 3次點擊 (中)5次點擊 (右)長按判斷(可以超長按判斷)



(左)10次點擊(中) 8次點擊(右)16次點擊，因為點得急，有輛次被判斷為bounce

2次點擊，1次長按，4次點擊，1次長按

5次點擊，1次長按，1次長按，2次點擊

multiple-click點擊偵測理論上可測到無上限(直到記憶體耗盡，因為她是用toggle的方式偵測)；但是若是要用真正連擊條件來說(1秒內完成的)，那arduino在這方面的記數可能會有問題。

後來也加入了，一下長按一下連擊的按法，也成功判斷，但是這部分有時候會有小錯，要多次實驗才會有正確值，實務上不建議大量使用。

Toggle的部分還沒加入偵測，但是以我目前的程式架構而言，是可以輕易做到的。

**因此這次實驗算得上大獲成功**。結果放入” Practice\_Five\_final.ino”中。

**因為實驗的成功，日後要使用連擊的時候便可以直接引用” Practice\_Five.ino”，或是把” Practice\_Five.ino”變成” Practice\_Five.h/.cpp”直接include也可以。**

**詳細資料請上個人GitHub頁：**[**https://github.com/IceStarDragon707/ArduinoProject/tree/main/Button\_Practices**](https://github.com/IceStarDragon707/ArduinoProject/tree/main/Button_Practices)