

王陽香

彈珠台



## 一、材料

使用材料	用途
木材、鐵釘、冰棒棍、資料夾、軟尺	彈珠檯本體
LED 燈(紅、綠、藍、黃)	裝飾
微動開關	偵測彈珠經過
按鈕開關	重啟鍵
跳線	連接 FPGA
漆包線	連接各電子零件
電火布、透明膠帶	保護漆包線連結處



led 燈



微動開關

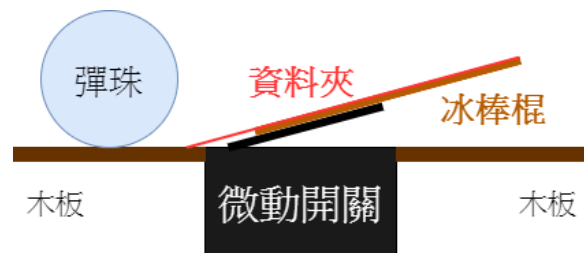


按鈕開關

## 二、構造

我們想要再本體的檯面上加上燈光效果，但又不希望她影響到彈珠的走向，所以 led 的部分我們使用釘子打穿後再把洞挖大，把 led 燈埋藏在平面以下。香腸王三個字分別用三個訊號來控制，如果得到 1 分就會亮「香」一個字，以此類推，當獲得 4 分時，三個字會開始有閃爍的效果。當打到超過 4 分，三個字的燈會全暗，需要重新 reset。中間香腸的部分是接 3V3，所以只要接上 fpga 就會亮。四組燈都是採用並聯的方式，避免其中一顆燈燒毀就整串都不會亮。

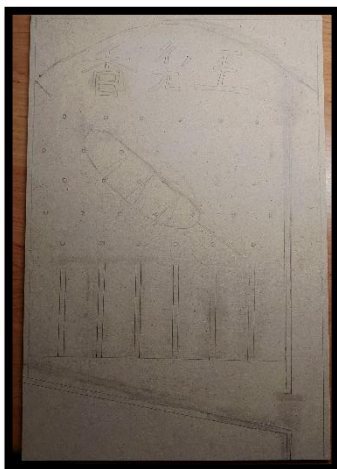
微動開關的部分也是直接嵌在木板上，因為其感應的範圍很小，所以我們用資料夾將其加長加寬，並在底下黏冰棒棍，使得資料夾不會太軟而失去其作用。如果直接黏冰棒棍，會造成檯面與冰棒棍之間的差距太大導致彈珠滾不上去，我們的構造如右圖。



按鈕開關埋在左下角的冰棒棍箱子裡，利用零錢投入縫中按壓到我們的開關，執行 reset 的功能。左邊的冰棒棍片並沒有直接黏在箱子上，而是用線將它固定在上頭，做成一個可以開關的門，方便我們取出投入的零錢。



### 三、組裝過程



#### STEP 1

利用木工鋸裁切木條以模擬框架



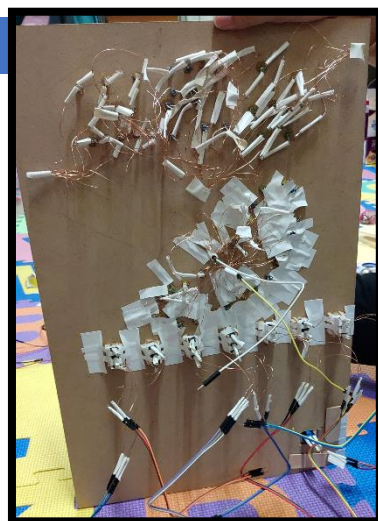
#### STEP 2

於底板繪製設計稿以確認各物品相對位置



#### STEP 3

相應點利用鐵槌與鐵釘打洞，遷入用漆包線、電火布、透明膠帶連結跳線的開關及 LED 燈



#### STEP 4

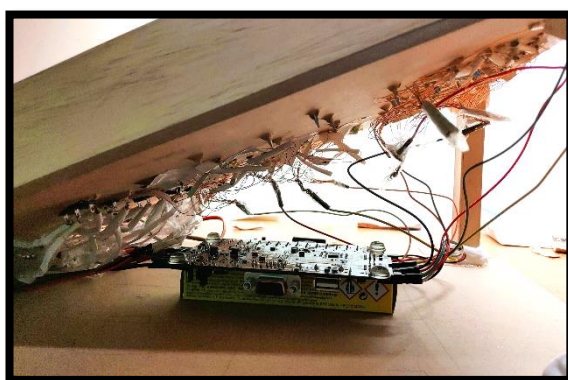
測試開關及 LED 燈連結是否穩當



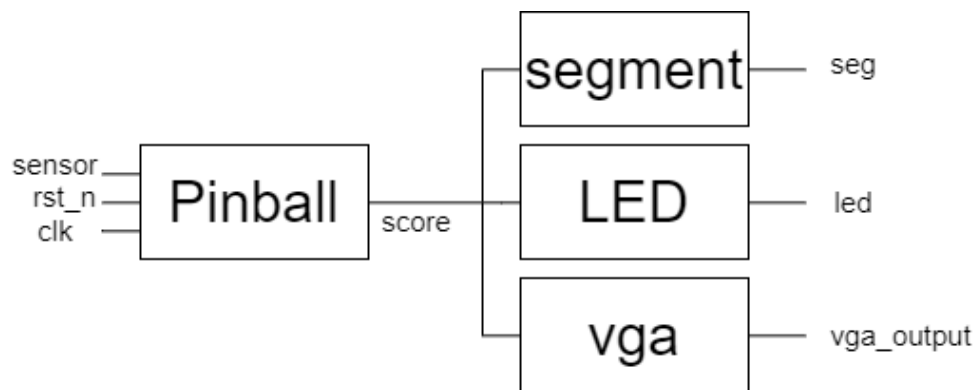
#### FINISH

#### STEP 5

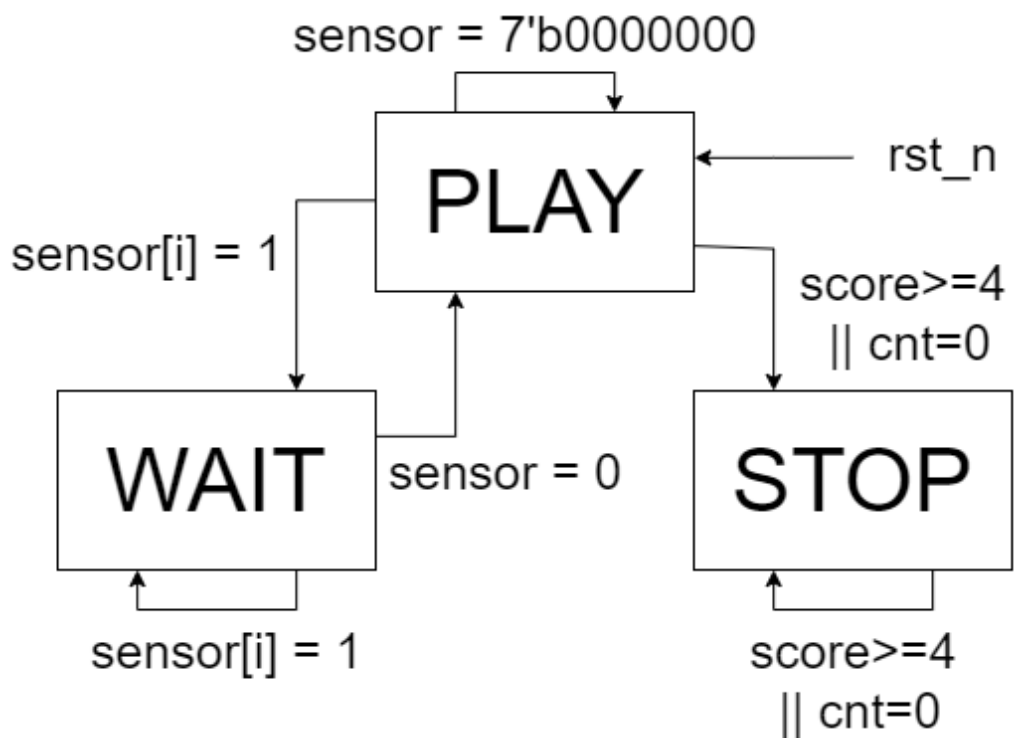
經過測試調整，將彈珠台本體組裝完成，並將跳線與 FPGA 連結後架置底座



#### 四、Block Diagram



#### 五、State Transition Diagram

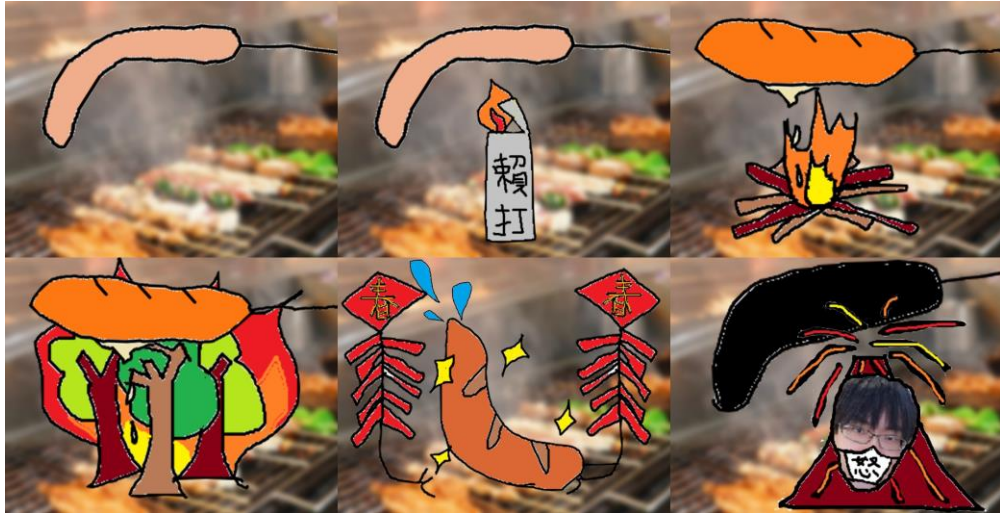


## 六、設計

主要分成三個檔案，top.v 處理 sensor 傳送進來的訊號，收到後再判斷 output 的結果應該要是什麼，chip2chip.v 是因為旁邊的 pmod 數量不夠，所以多接一塊板子，vga.v 是根據 top.v 的 output 做出相對應的變化。

function Pinball 分成 3 個 state，PLAY，WAIT 跟 STOP。一開始 reset 後的 state 為 PLAY，分數是 0，cnt=3。在 PLAY 的 state 中，如果感應到其中一個 sensor 變成 1，next\_state 就會變成 WAIT，cnt 減一，score 根據 sensor 位置加不同的分數。之所以要進入 WAIT 是因為不知道買到的感應器是否有做 debounce 跟 onepulse，所以直接寫成當其中一個感應器感應到 1，就先進去 WAIT 裡等待，等到所有的 sensor 都變成 0 後，再回去 PLAY 的 state。只有當 score 大於等於 4 或 cnt=0，state 才會進入 STOP，score 跟 cnt 都不改變，等下一次 reset 被觸發後再重新開始。

function segment 主要是用來 debug，Pinball 輸出的 score 會顯示在 fpga 上，四個 seven segment 都會顯示一樣的 (AN=4' b0000)。function LED 則是連接 led 燈，score 也會送進來當 input，用 case 來決定現在要亮什麼。



利用不同分數決定顯示的圖片，如上圖由左至右、上至下為 0~5 分。

在 `mem_addr_gen` 中，設立常數 `H1`、`H2`、`H3` 作為各列圖片之最左點，常數 `V1`、`V2` 作為各行圖片之最上點，由以上兩種參數來選擇圖片(如右圖)。同時將 `vga_controller` 產生之 `input`，`h_cnt`、`v_cnt` 分別除以三及二，以將螢幕顯示範圍控制在該圖片內，使螢幕中的六格顯示之圖案為同一像素，結果導致畫素降低。

## 七、遭遇困難與解決

### 1. 跳線不穩

一開始以並聯方式串接 LED，並連結 FPGA 之 3V3 與 GND 時，LED 一直不亮。起初是以為漆包線沒磨好，導致電流不通，但一看到磨過的部分出現了生鏽的情形，我們便懷疑起連結的跳線是否出現了問題，經過了測試，果然如此。

### 2. 訊號干擾

將所有的電子零件嵌於木板上後，發現偶爾會出現數據亂跳的問題，例如觸動分數為 0 的微動開關，卻會有 RESET 鍵的效果。由於在尚未嵌入木板時，各個 sensor 的數據傳遞是順利的，推斷不是程式的問題。誤打誤撞間，我們發現當我們將裸露的漆包線利用電火布包起時，數據的傳遞便逐漸穩定，於是我們推定，雖然漆包線原有一層外漆，但若漆包線彼此接觸仍會互相干擾。而為了確保數據傳輸穩定，我們除了使用電火布外，還再用了透明膠帶將漆包線完整的纏起來。



### 3. 微動開關觸發困難

由於沒有良好的挖洞工具，我們本來不打算將微動開關嵌入木板內，想使用間接觸發的方式來觸發微動開關，同 STEP4 的那張圖，利用上方較陡的冰棒棍製造與下方延長微動開關施力臂的冰棒棍之間位能差距，使得彈珠的重力能夠順利地觸動微動開關，卻發現冰棒的厚度阻止了彈珠的順利進行。最終，我們只好回到原本的方案，將微動開關嵌入木板中，而由於微動開關的觸動部分較小，我們同樣使用冰棒棍來延長施力臂，並使用了資料夾製造滑順的平面，同 STEP6。

### 4. 訊號電壓不足

在連結 LED 燈時，我們預設每加一分多亮一個圖案，分別是"香"、"腸"、"王"以及香腸的圖案，在三個字的部分 LED 的亮起是還可以觀察的到的，但香腸的圖案卻無法順利地亮起，我們懷疑是因為香腸圖案的 LED 連結的太多了，所以電壓不足。於是我們將 LED 再分四部分連結，卻發現無法解決這個問題，於是為了圖案的美觀性，我們決定直接將香腸圖案連結 3V3，不對香腸圖案做控制。為補償這件事，當 4 分時，我們將三個字("香"、"腸"、"王")做閃爍的處理。

## 5. 燒 flash

當我們完成了所有的測試，我們想說要將程式燒進 flash 中，使用時直接接行動電源即可，才不會顯得太笨重，卻不想燒完了 flash，數據亂跳的情形又出現了，而且當將 slave 的電源關掉再打開，燒進去的標示便會消失，在 Vivado 中，該板子顯示，他含有燒進去的 bin 檔卻是 Unprogrammed 的狀態。無法解決這個問題，我們決定照舊燒 bit 檔。