

# **Final Project Report**

**Team25**

組長：蔡登瑞、組員：蔡政諺

# Outline

- **Introduction**
- **Motivation**
- **System Specification(explanations, diagrams)**
- **Experimental Results**
- **Conclusion**
- **Contribution List**

## ● Introduction

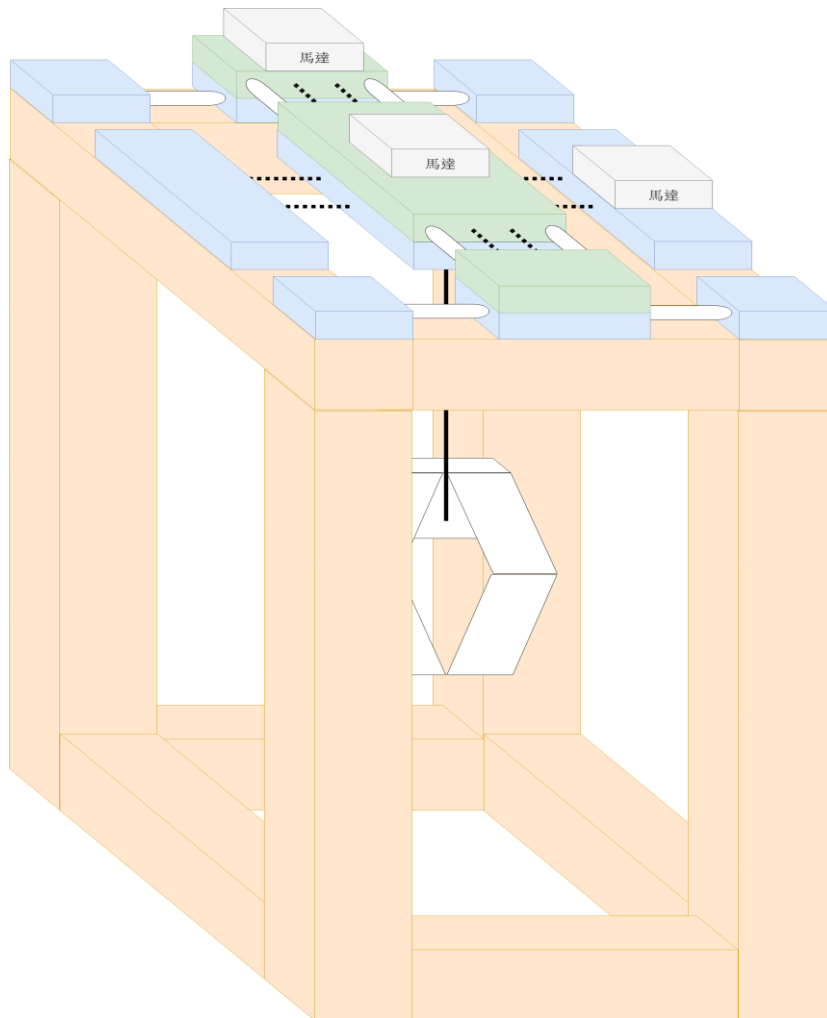
夾娃娃機是一種市面上常見的遊戲機台。遊戲過程為玩家藉由按鈕或搖桿控制機械手臂，抓取箱內物品至取物口做為獎品。我們的 Project 即是一個小型的夾娃娃機(長 25cm、寬 25cm、高 40cm)，並將 Verilog code 燒進 FPGA 板，連結鍵盤操作爪子移動、抓取，並在每次遊戲期間進行 30 秒的倒數計時(秒數會顯示在 FPGA 板的 7-segment 上)，在時間內玩家可以自由控制爪子抓取物品，倒數結束後爪子會自動歸位。



## ● Motivation

我們一開始因為背景知識只有 VGA，考慮了很多電子遊戲(如小精靈、貪食蛇等)。但用 Verilog 寫電子遊戲並接 VGA 顯示於螢幕上，限制相當地多，還不如用 C++寫；加上我們在社群網站(巴哈姆特)爬文，看到有人曾經成功用 Basys3 FPGA 板做出夾娃娃機，於是決定挑戰自我，將這門課的 Final Project 主題定為夾娃娃機。

設計圖



## ● System Specification

這次的 Verilog Design 綜合了我們學習過的 7-segment、Keyboard、Audio，也嘗試用 FPGA 連接伺服馬達。使用到的 Module 如下：

### ■ Claw\_Machine

這次 Design 的 Finite State Machine。總共有五個 State：IDLE、MOVE、CATCH、BACK、HACK，詳見數後頁的 State Transition Diagram。

IDLE：閒置狀態，此時無論按什麼按鍵，馬達都不會動。若按下 start，會根據 switch 有沒有拉起，進入 MOVE(switch==0、遊戲模式)或 HACK(switch==1、開發者模式)。

MOVE：移動狀態，可以按下設定好的四個按鍵，使 x、y 軸的兩顆馬達正、反轉。此處我設定了這兩個馬達的轉動上限，當 counter 數到設定的值時，就不能再往該方向轉動，以免旋轉過度造成設備受損。此處我也設定了 30 秒的時間限制，剩餘秒數會顯示在 7-segment 上。若按下 catch 按鍵、或者時間到，就會進入下一個 state，CATCH。

CATCH：夾取的狀態，流程為夾子張開→繩子往下→夾子夾緊。達到設定時間後會進入下一個 state，BACK。

BACK：重置的狀態，在 MOVE 和 CATCH 的 state 中記錄了四顆馬達的轉動量(正轉會記錄為正值，反轉會記錄為負值)，在這個 state 會根據四顆馬達轉動的數據，將四顆馬達轉回遊戲開始前的角度。四顆馬達都歸位後，就會鬆爪、並回到 IDLE。

HACK：開發者模式，可以不受限制地任意控制四顆馬達正、反轉，用於校正四顆馬達的座標。在此處若按下 catch 按鍵，就會回到 IDLE。

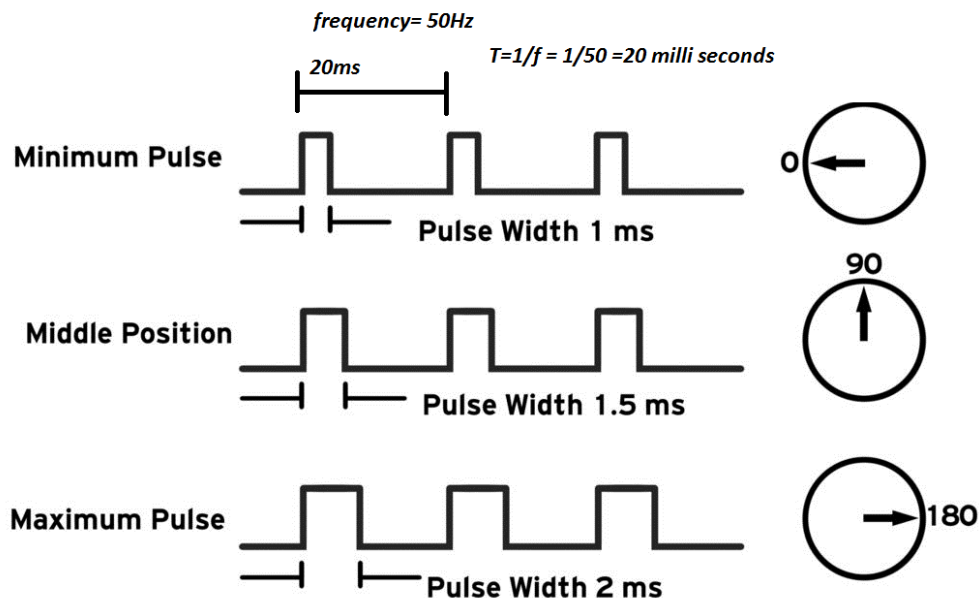
### ■ Servo\_Motor\_PWM\_Gen

用於產生伺服馬達的 PWM 訊號(50Hz、20ms)。這個模組會根據輸入的 dir，改變輸出訊號的 Duty cycle，以決定要正轉(高電位佔波長的 1ms)、反轉(高電位佔波長的 2ms)、或者不轉動(高電位佔波長的 1.5ms)。

```
338 module Servo_Motor_PWM_Gen (clk, rst, dir, signal);
339     input clk, rst;
340     input [1:0] dir;
341     output reg signal;
342     reg [29:0] count;
343     parameter POS = 30'd10_0000, NEG = 30'd20_0000, STOP = 30'd15_0000, WAVELENGTH = 30'd200_0000;
344     reg [29:0] dir_count;
345
346     always @ (posedge clk) begin
347         if(rst) begin
348             count <= 30'd0;
349             signal <= 1'b0;
350         end
351         else begin
352             count <= (count < WAVELENGTH) ? count + 1'b1 : 30'd0;
353             signal <= (count < dir_count) ? 1'b1 : 1'b0;
354         end
355     end
356
357     always @ (*) begin
358         case(dir)
359             2'b01: begin
360                 dir_count = POS;
361             end
362             2'b10: begin
363                 dir_count = NEG;
364             end
365             default: begin
366                 dir_count = STOP;
367             end
368         endcase
369     end
370 endmodule
```

◆ 伺服馬達背景知識

伺服馬達原本有 180 度的轉動上限，根據 PWM 訊號的 Duty cycle，決定角度(0 度~180 度)，如下圖。我們用的是改裝後的 360 度伺服馬達，PWM 的意義於是變為 Pulse width 小於 Middle position(1.5ms) 的為正轉、Pulse width 大於 Middle position 的為反轉。



■ Debounce

利用串接 DFF，除去訊號在高低電位之間的異常跳動。

■ Onepulse

將維持在高電位的訊號轉變為只有一個 Clock cycle 的訊號。

■ AN\_Replace

使 AN 交互切換，以肉眼無法感覺的頻率( $100\text{MHz} \div 2^{16}$ )輪流顯示四個 7-segment，以達到四個 7-segment 同時顯示的錯覺。

■ Segment\_Display

根據 timer 的值(剩餘秒數)、以及 AN 目前切換到哪，決定哪些 segment 要亮燈、哪些要暗燈。

■ KeyboardDecoder

根據鍵盤輸入，輸出 last\_change、key\_valid、key\_down，這邊我們只有用到 key\_down。key\_down 在按下去時為 1、放開時為 0，我們根據這個特性，將 key\_down 作為 Servo\_Motor\_PWM\_Gen 模組的 dir。

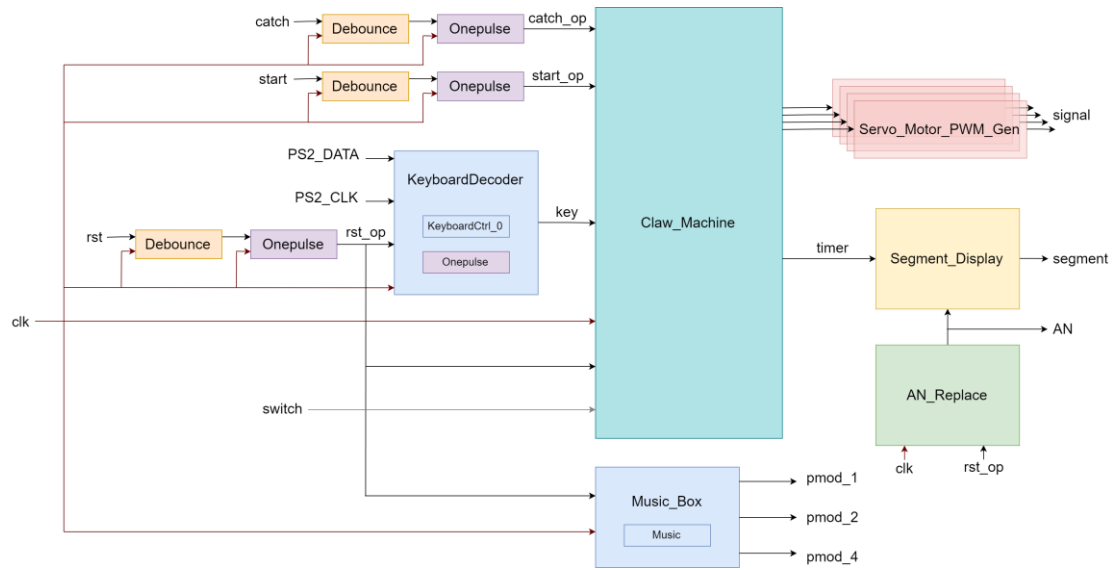
■ Music\_Box

輸出提供音效模組使用的 pmod\_1、pmod\_2、pmod\_4。

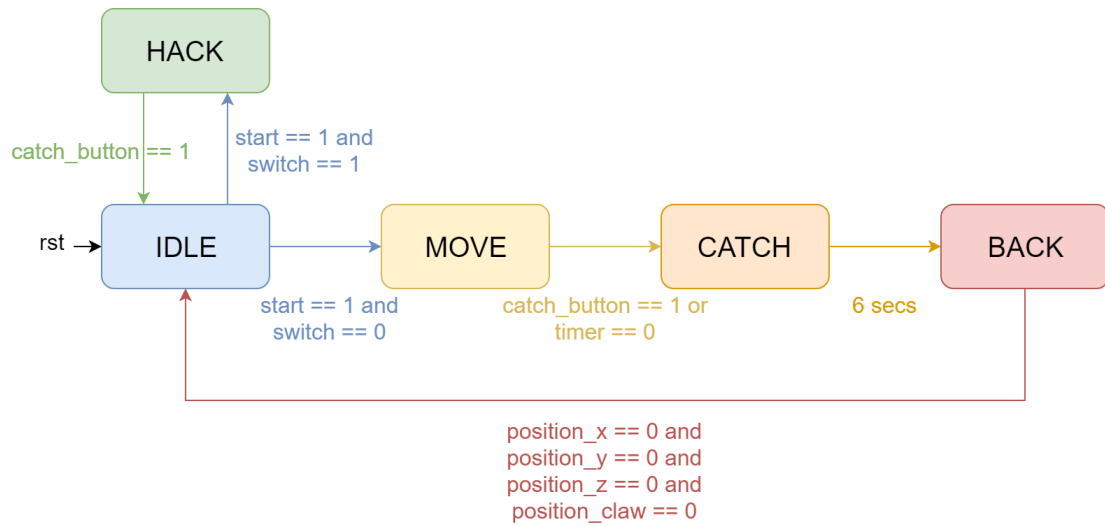
■ Music

手刻的音樂。曲名為〈愛江山更愛美人〉，調性為 G 大調。

## Block Diagram



## State Transition Diagram



## ● Experimental Results

### ■ Physical design Process

在夾娃娃機實體部分，我們一開始是想用木材為基底，並利用 3D 列印機列印出其他控制零件，可是後來借不到 3D 列印機，所以改成用壓克力板來製作零件。為了使夾娃娃機可以進行  $x,y,z$  三軸移動，分別使用了三顆馬達來操控，此外，還用了一顆馬達來操控爪子。為了可以移動，我們上網訂購了四根光軸還有買了八顆軸套，並在完成之後上潤滑油，以讓移動更順暢。

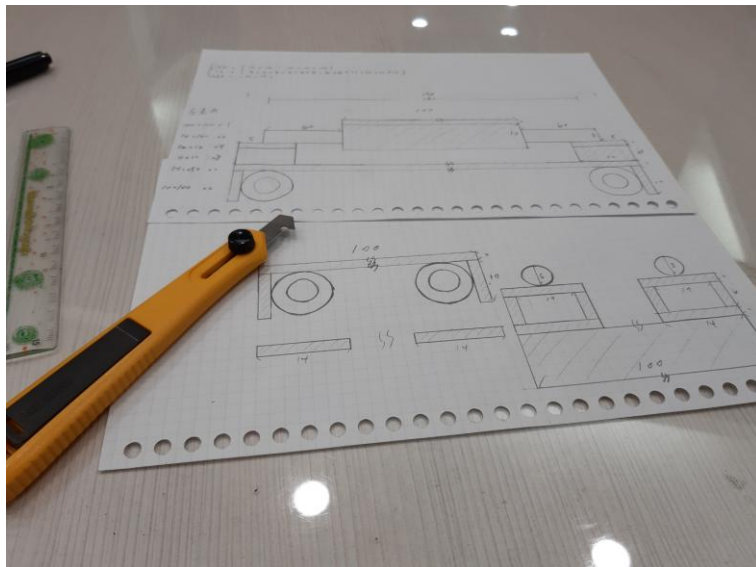
#### ◆ 基底部分

使用的是松木角材，質感比較脆，易釘。

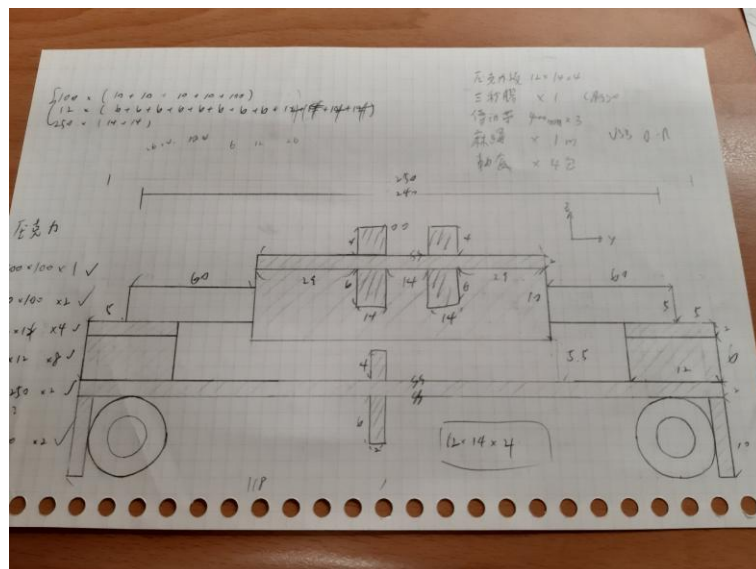
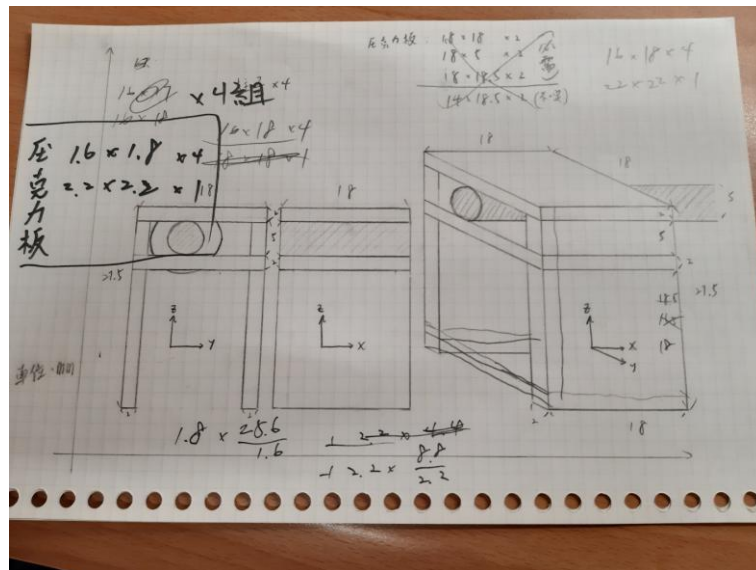


#### ◆ 零件部分

起初想使用 3D 列印機來製作，可是發現可以借到的機率有點低，所以我們改成用壓克力板來手工製作零件，但由於將壓克力板裁切成特殊的圖形會有些困難，因此對於原本的設計圖就得要做一些更改。

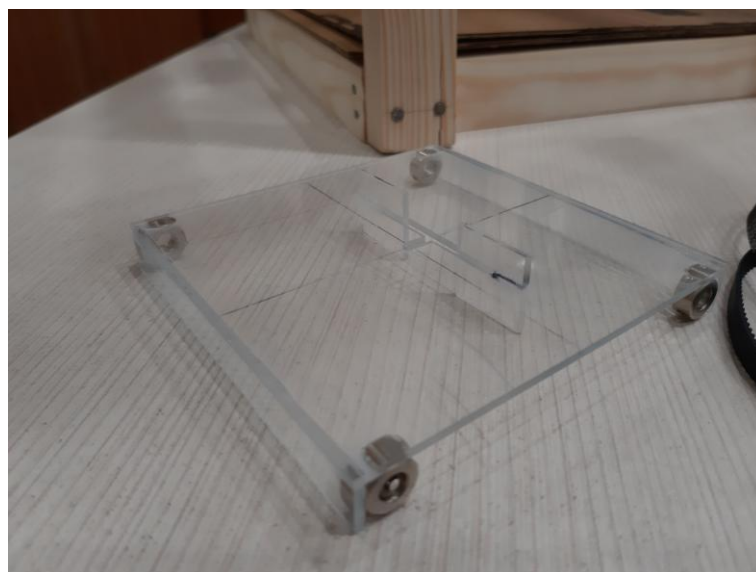


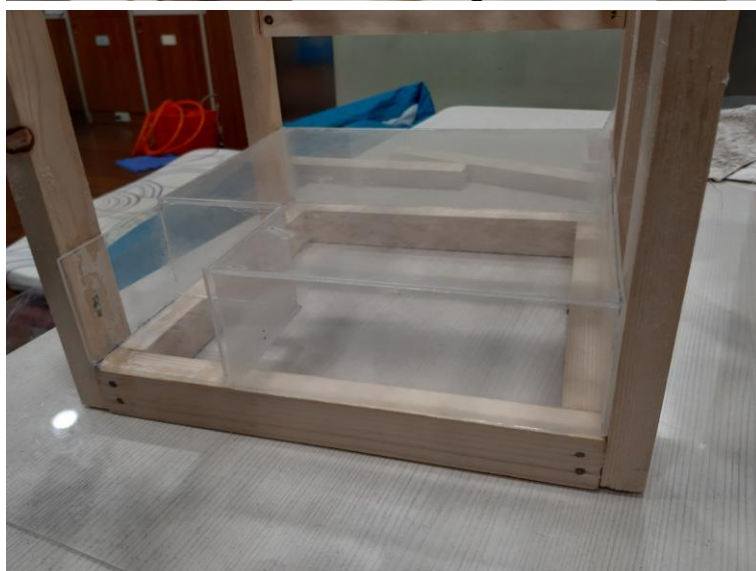
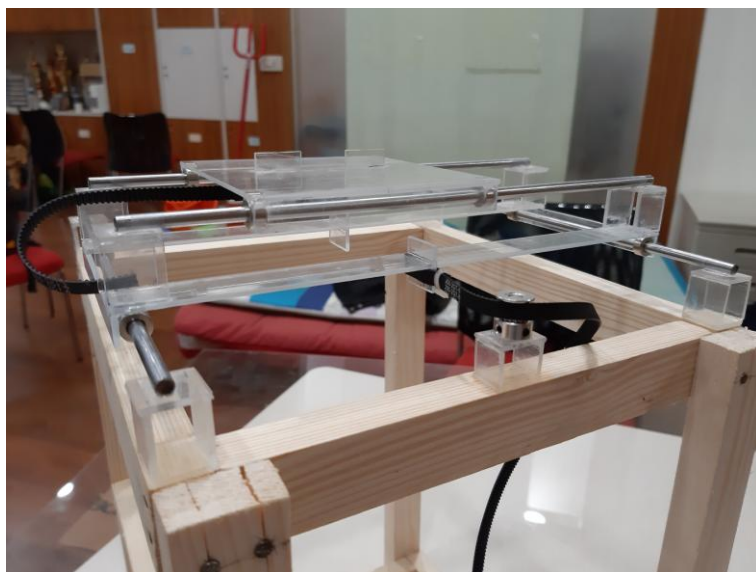




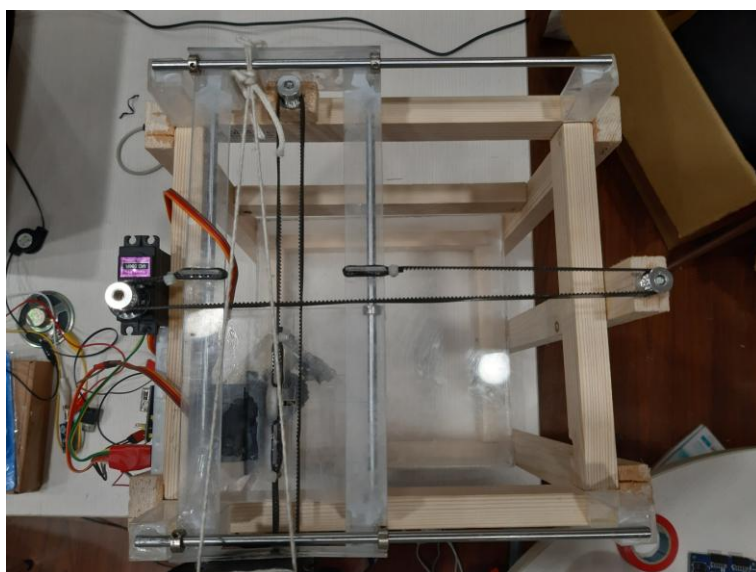
## ◆ 製作及組裝

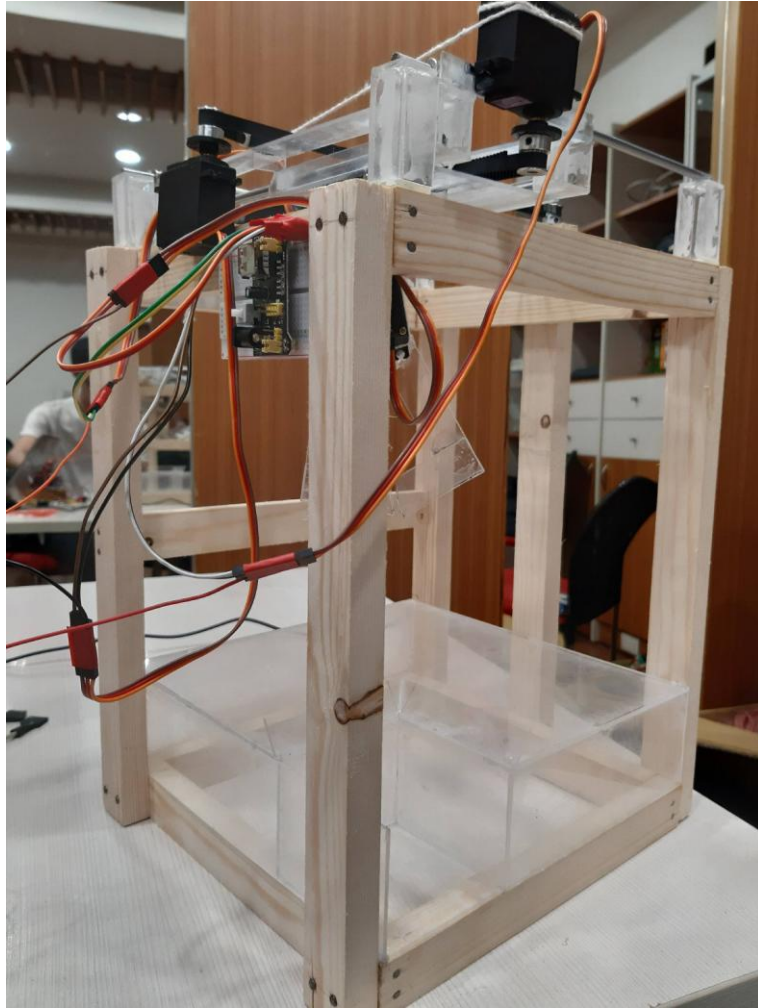
使用壓克力切割刀裁切壓克力板，並且用三秒膠來黏





◆ 成品





## ■ Problems

### ◆ 線接錯、電源供電模組燒壞、電流不足、線太短

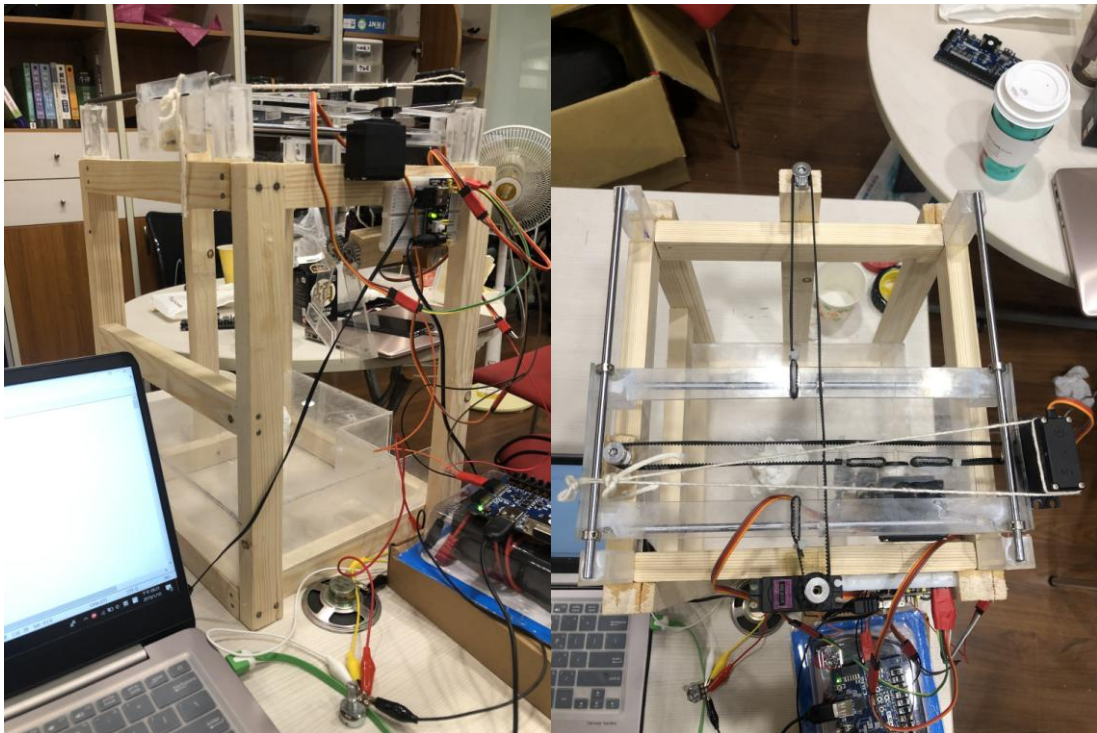
當我們解決馬達的線要怎麼接之後便開始測試四顆馬達是否能運作，一開始我們是用電腦接 USB 對 USB 的線給電源供應模組，再接到馬達上，然而馬達卻遲遲無法反轉，只會震動並發生嗡嗡的聲音。我們起初不知道原因，因為這些東西是第一次接觸，我們有寄信給賣家並詢問她原因，賣家猜測是供電電流不足而導致的。而有一次再報告進度的課堂上，我們有詢問了老師，老師是說可能是麵包板壞掉。後來我們就去再買一次麵包板跟電源模組，經過比對之後發現是原本的電源模組壞掉，然後我們想到可能是一開始馬達接線的時候接錯，導致電源模組燒壞。可是馬達反轉的時候會有 delay，這個問題困擾了我們很久，我們以為是線本身可能有折到、受損，demo 前一天跟別人借了幾條線來測試，發現是線太長的問題！所以後來除了控制線以外，電源線跟接地線都是用比較短的杜邦線來連接。到現在我們還是不知道為什麼杜邦線太長會有問題。



## ● Conclusion

寫 Verilog 的邏輯真的與以往寫軟體語言相去甚遠，但經過一個學期的淬鍊，我們對於三種描述語言(Gate Level、Data Flow、Behavioral)已相當熟悉，也在 12 次 Lab 裡，多次練習過 Finite State Machine，因此我們在這次 Final Project 的 Coding 方面並沒有遭遇太大的瓶頸。相對地，硬體設備常常不聽使喚，零件的切割、組裝過程也相當繁複，反而是這次 Final Project 最令我們費心的部分。不過最後成品完成度滿符合我們所預期，真的覺得收穫滿滿，也很有成就感。

### 成品



(左：平視，右：俯視)

夾娃娃機操作影片雲端連結：

<https://drive.google.com/open?id=1UrqcaX7U5OILt5PPBI6BAK501oIZN8fG>

## ● Contribution List

- ❖ 蔡登瑞
  - 畫設計圖
  - 零件切割、組裝
- ❖ 蔡政諺
  - 畫 block diagram
  - 畫 state transition diagram
- ❖ 共同完成
  - 寫 code
  - 寫 report