

FINAL PROJECT REPORT

Vending Machine
之信不信我拆你屋頂

GROUP14

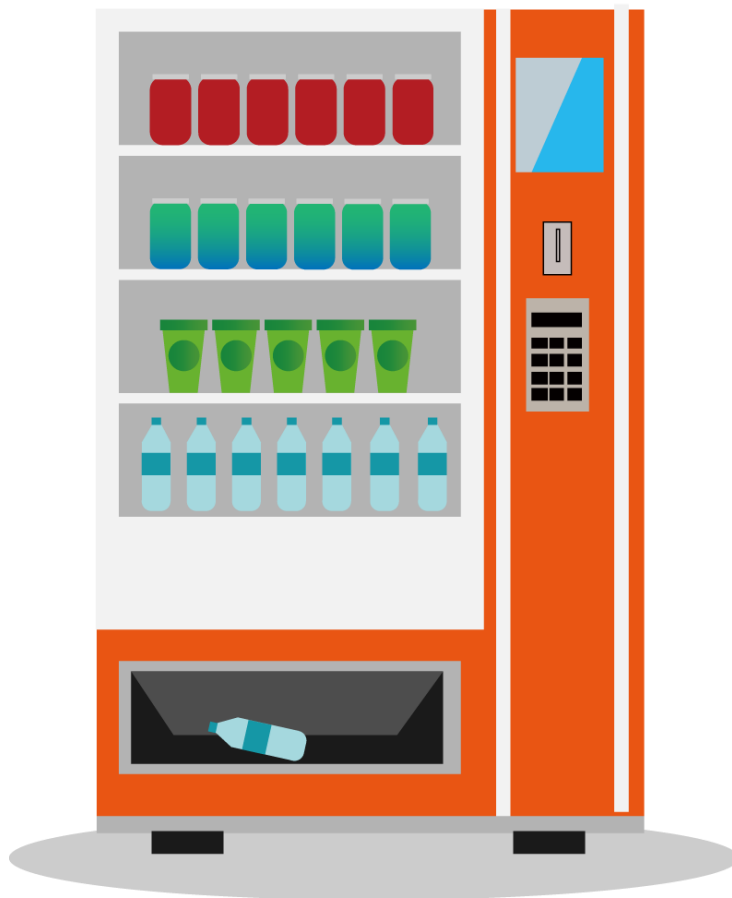
組長:王領崧 組員:鍾皓崙

Outline

- Motivation
- Introduction
- FPGA I/O port
- Experimental Completeness
- Problem & discussion
- Reflection
- Contribution List
- System Specification

● Motivation

組長王同學考完期中考以後，覺得口有點渴，就想去樓下販賣機買個飲料，生性懶惰的他覺得每次都要爬下樓買飲料非常的麻煩，於是突發奇想決定利用 Final project 的機會，手做一個 vending machine 放在房間內，搭配 VGA 與音樂做出一台動感的自動販賣機。

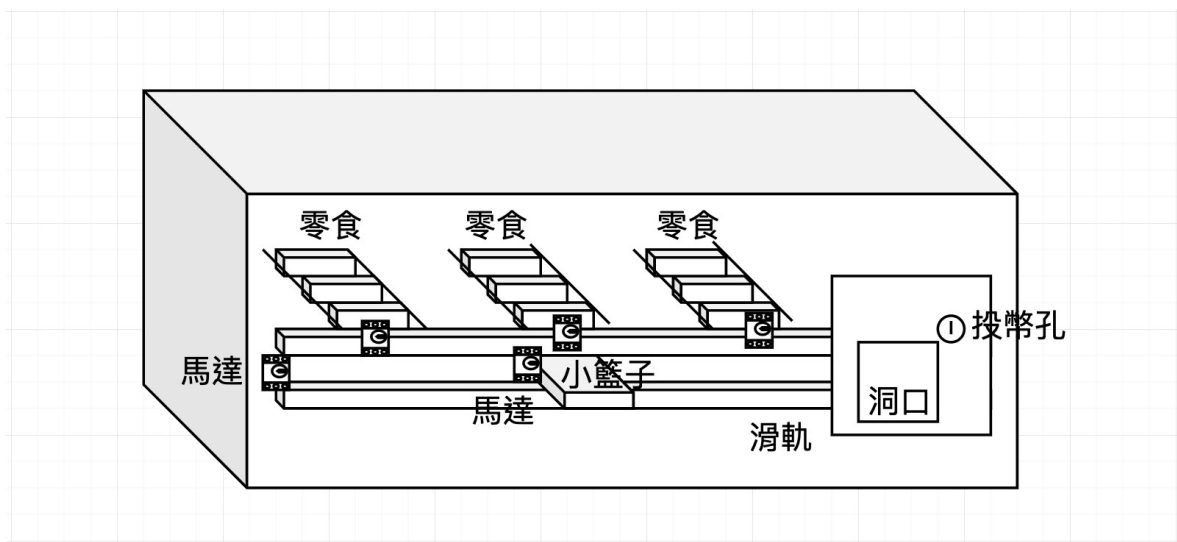


● Introduction

自動販賣機 (vending machine) 在我們的生活中隨處可見，是繁忙現代人補充水分或熱量的幫手。自動販賣機的發明者，一般相信是生於一世紀亞歷山大港的希羅，他是一位希臘工程師、數學家。他製造的機器接收了硬幣後會給顧客一定數量的聖水。現今的自動販賣機有多種形式，常見的有兩種，第一種為一般封閉式的飲料販賣機，另一種為透明玻璃能看見運送機構的販賣機。當然也有許多特殊形式的販賣機，販賣的物品更是無奇不有，例如炸薯條拉麵或甚至是汽車。

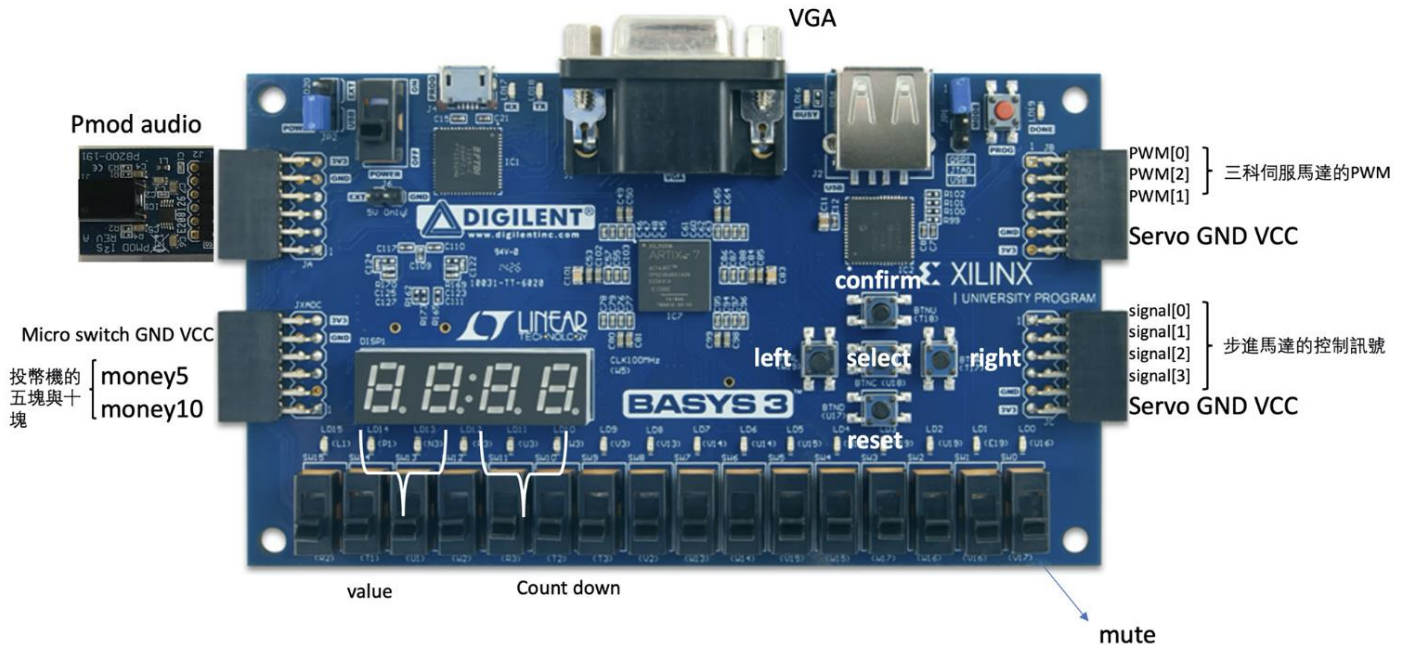
自動販賣機通常有幾個要素，它能判斷購買者所投入的錢金額是多少 (投幣機)，它能顯示金額和能購買的商品品項 (七段顯示器)，它提供簡潔的使用者介面供客人選擇商品 (push button & VGA)，它能正常出貨，提供客人所需的商品 (機械結構)。有些甚至會有一些與購買者特殊的互動機制，例如特殊音效或是燈光 (audio & LEDs)。

Vending Machine 設計圖



- FPGA I/O port

Basys3: Pmod Pin-Out Diagram



- Experimental Completeness

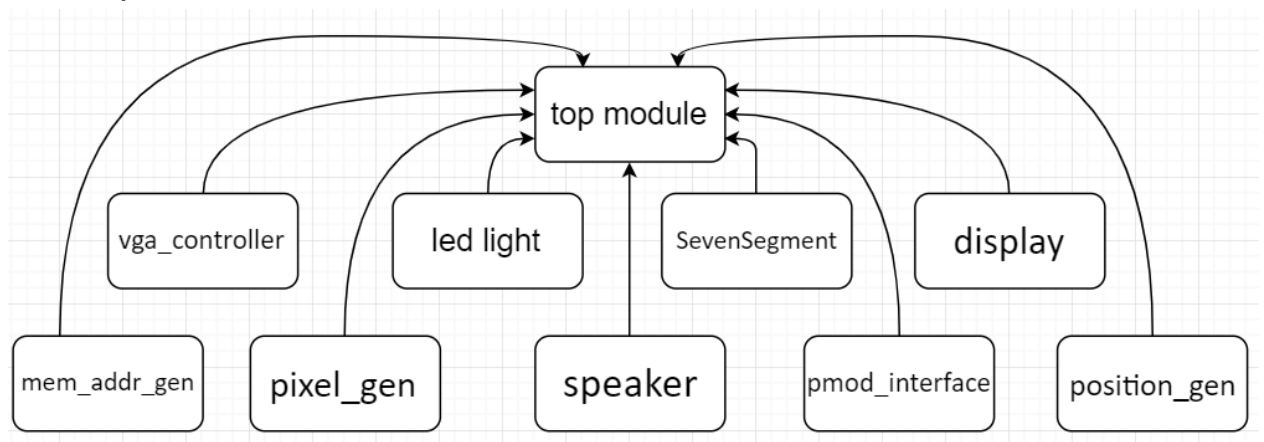
以下為此次 project 所完成的

- VGA 選單資訊
- Audio 背景音效、購買成功音效
- LED 購買成功特效
- Switch (Mute)
- Button 左右移動鍵 選擇鍵 確認鍵
- 7-segment(已投入金額、倒數計時)
- 步進馬達(傳輸帶)、伺服馬達(閘門)
- 投幣機(觸動開關)
- 組裝、與 FPGA 整合

與 Proposal 的 spec 不同為開關機與開關機的特效，因為發現販賣機不太需要開關機的功能，所以相關特效全部捨棄，改為購買成功的特效與 LED 特效，switch 開關機則改成音樂靜音的功能。

● System Specification

✓ Top Module 示意圖



✓ Pmod_interface

➤ pmod_step_driver

分別用來控閘們的伺服馬達與不同步帶的步進馬達。

➤ Servo_interface

Servo_interface 中又會有四個小的 module 作用為輸出一個波長為 20ms (50Hz) 的脈衝波，伺服馬達會依據每個波為高電位的時間來判斷要旋轉多少角度，若持續給馬達一樣的脈衝波，馬達會轉到該角度後停下來，並鎖死。以下是馬達轉角度與輸入之 PWM 的關係圖。PWM 產生方法是利用一個 counter 數是否已經累加到了 1.5ms (產生 90 度的 PWM)，若到了 1.5ms 就把 PWM 輸出為 0，如果 counter 累加到了 20ms 代表上一個 cycle 已經結束，把訊號拉起來 counter 重新開始計算。



➤ pmod_step_driver

步進馬達的控制則是有運用 state transition，分為以下幾個 state (Initial、Go_buy、Wait_buy、Back_buy)。

Initial:

在這個 state 中會等待外部輸入的指令來決定是否要開始移動並判斷要進入 go_buy1,2,3 (分別代表三個不同的飲料)。

Go_buy:

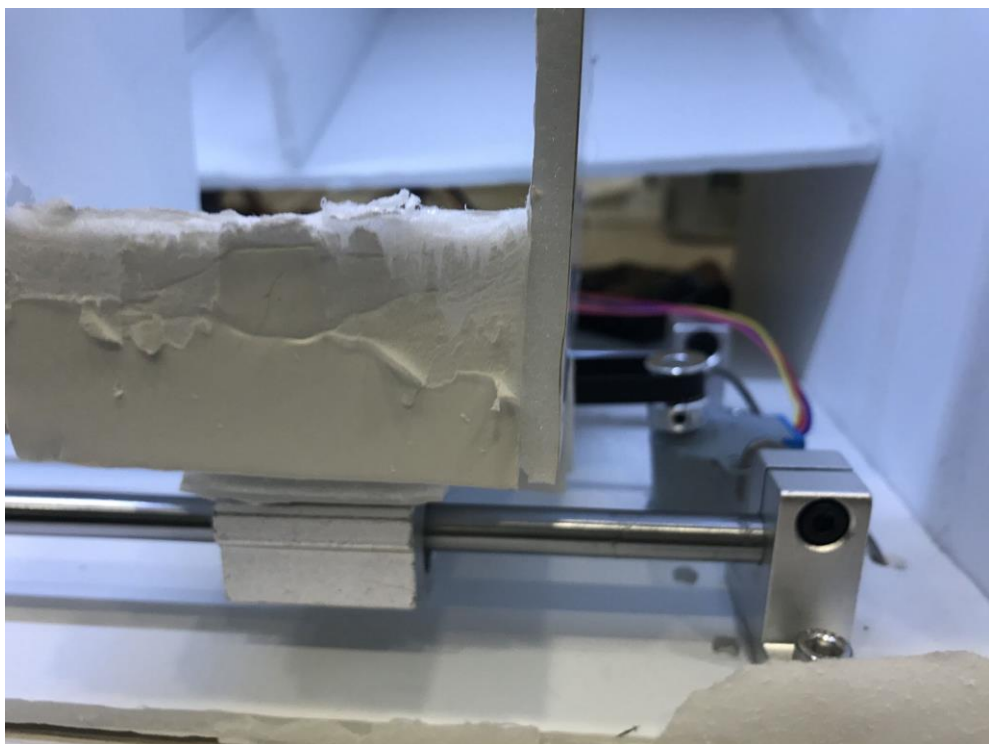
進入 go_buy 後會利用一個 counter 計算小籃子走到哪個地方，這麼一來就能準確的將飲料接住。接著進入 wait_buy。

Wait_buy:

在 wait_buy 中也會利用 counter 計算，等待一段固定時間讓 Servo_interface 將閘門打開並接住飲料。

Back_buy:

這個 state 跟 go_buy state 大同小異，也是利用一樣的原理將小籃子移回出貨口。並且再次進入 initial 等待下一次的指令。



➔ 上圖為步進馬達同步帶與光軸的結構

✓ VGA

➤ mem_addr_gen

將可樂、雪碧、啤酒芬達所組成的圖片(.coe 檔)載入進來，因為圖片不用做任何位移，因此完全不需要額外的 counter 之類的東西。

➤ pixel_gen

pixel_gen 利用 state 的不同，繪製出不同的圖案，底下用三種 state 個別說明(WAIT、PAY、SEND_PROCESS)。

WAIT:

WAIT state 中只有繪製一個橘色正方形圓框，並且只有對應位置的飲料圖案會顯示，其他部分則為全黑。default 狀態在中間的位置，並依據 POS 的不同，往左往右、選擇不同的飲料。

PAY:

PAY state 中，圖片會維持在所選擇的物品上面，比起 WAIT state，橘色外框下多了條藍色線條在不斷地做繞圈移動。

SEND_PROCESS:

SEND_PROCESS state 中，圖片仍然維持在所選擇的物品上面，橘色框框仍舊還在，只是沒有藍色線條的繞圈特效，取而代之的是在對應位至下方，多了一個笑臉，其目的是希望使用者能耐心等待。

✓ Seven Segment

為 FPGA 上數字顯示的部分，只有在 PAY state 才会有功用，前面兩位是用來表示已投入的金額，後面兩位是用來倒數時間的(40 秒)，在其餘的 state 中，會顯示----。

✓ position_gen

根據 FPGA 上的按鈕(left、right)，POS 會切換至不同的 state(LEFT、MIDDLE、RIGHT)，用來讓其他 module 判斷，產生相對應飲料所需執行的動作。

✓ Display

為整個 Design 的 finite state machine，總共分為三個 state(WAIT、PAY、SEND_PROCESS)，底下會分別介紹，詳見於後面的 state transition diagram。

WAIT:

為使用者選擇飲料的 state，利用 FPGA 上的左右鍵來左右移動，按下選擇的 select 鍵以後，會跳入 PAY state。

PAY:

PAY state 為投入金額的 state，使用者須在 40 秒內，達到足夠的投入金額，我們利用從投幣機接過來的兩個觸動開關來增加金額(5 塊、10 塊)，如果使用者按下 confirm 後，判斷金額有無達到飲料價錢，有的話則進入 SEND_PROCESS 的 state，如果沒有的話，則繼續維持在 PAY state，等待使用者投入更多的金額。如果倒數計時為 0 時，判斷已投入金額是否達到飲料價錢，是的話則自動進入 SEND_PROCESS 的 state，如果沒有的話，則直接回到 WAIT state，重新等待使用者選擇新的飲料。

SEND_PROCESS:

SEND_PROCESS state，會輸入訊號給 Pmod_interface module，讓販賣機去拿取相對應的飲料，當販賣機完成拿取的動作後，會傳回一個 complete(經過 Create Large Pulse 轉換)的訊號，這樣才會回到 WAIT state，否則的話，停留在原 state，等待拿取動作的完成。

✓ Clock_divider

將 clk 調整為適當的 $\text{clk}/2^{16}$ 、 $\text{clk}/2^{13}$...，用來實作 state transition 和 seven_segment 等等功能。

✓ Led light

為購買成功(販賣機拿取飲料)時，產生的 LED 特效，在其他 state 時，為全暗的情況。

✓ Onepulse

將高電位的訊號轉成只有維持一個 cycle 的訊號。

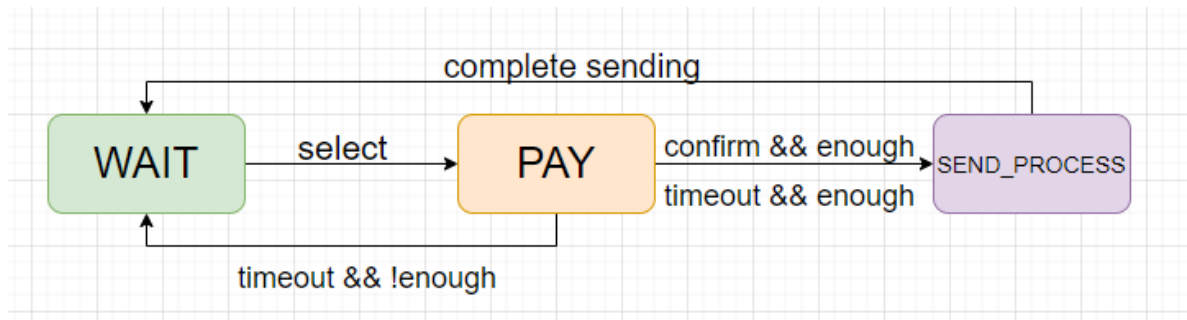
✓ Debounced

穩定訊號，除去訊號在高低電位之間的異常跳動。

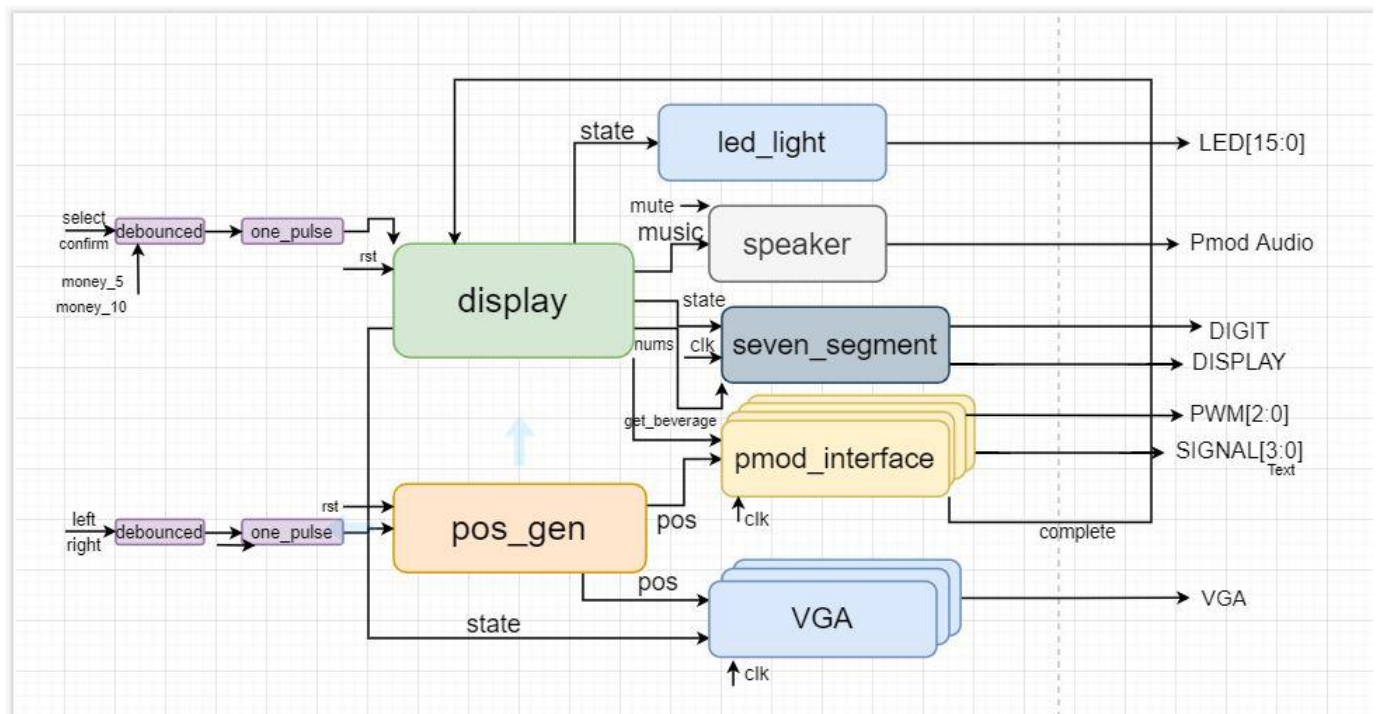
✓ Speaker

手刻的音樂，背景音樂為<你的酒館對我打了烺>，投入金額時的音效為<全家歡迎光臨>。因為 music 所使用 module 基本上都是採用助教 lab8 中所提供的，因此在這邊不做詳細的介紹。

State transition diagram



Block diagram



● Problem & discussion

➤ 伺服馬達：

伺服馬達在實作上有極大的困難，伺服馬達分為兩種，可連續旋轉與不可連續旋轉，我們購買驅動電壓及電流都比較大的可連續旋轉的伺服馬達，在實作過程中發現伺服馬達遲遲不會旋轉。經過測試和對照馬達 spec 後發現是我們的電流太小導致馬達不會旋轉。多方考慮後因購買外接電源成本過高，我們決定利用步進馬達代替伺服馬達來驅動同步帶，在實作完成後發現步進馬達對比伺服馬達有更好的操控性與精準性。但扭力與轉速較不足，不過在我們增加輸入電壓後情況有明顯改善。

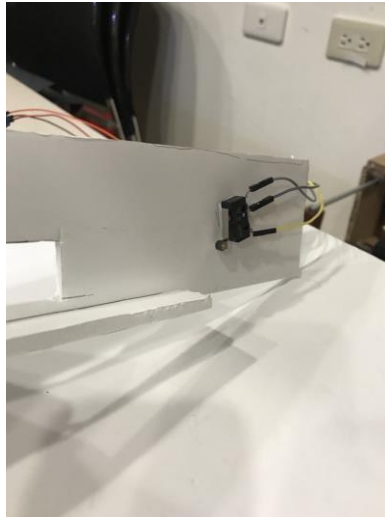
➤ 販賣機結構：

販賣機本體是以珍珠板和熱熔槍實作完成的，因此整體的機械強度非常不足（因為販賣鐵鋁罐，重量比較重），時常發生粘接處龜裂，板子斷裂或下陷，這樣的問題我們在成品完成後都無法完全解決。比較簡陋的解決辦法就是用粘性更強的保麗龍膠或是強力膠做結構補強。承載飲料的平台會因為飲料數目增加而下陷導致飲料運送過程發生失誤，解決辦法就是在承載平台下方多增加一個支撐座，效果非常的顯著。我們在最後檢討認為，材料才是問題的癥結點，像是這種需要承載物品的結構應該要使用木板和木條才會穩固。我們一開始就不該選用珍珠版進行實作。

➤ 投幣機：

一開始我們是利用在軌道上，開一個僅 10 塊會落下的凹洞，而 50 塊會因為錢幣面積較大，無法從凹洞掉落，而順順的滾至底端，藉此來分辨 10 塊錢、50 塊錢。但是，實作測試後，發現滾動的軌道需要做得十分陡，才能使 50 塊不會卡在洞口，但是這樣子 10 塊又會因為滾動速率太快，導致它不會從原先設計的凹洞處落下。後來上網查詢資料後，發現可以利用在軌道側邊牆

上，挖一個 10 塊錢直徑大小的長方形，搭配將軌道偏向那一邊，這樣 50 塊因為體積大，會忽略挖洞處、仍視整體為一面牆，進而解決之前因為速率不夠，而卡在凹洞處的問題。



➤ 程式整合:

因為馬達控制都是用 `clk` 來驅動，但是本體 `state transition` 卻是用 `clk_16` 來驅動，一開始因為沒注意到 `clk` 不同步的問題，導致販賣機都會拿取飲料數次才會停止，後面使用了老師 `final` 所提供的 `CreateLargePulse(clk->clk_16)` 與 `onpulse(clk16->clk)`，才使兩邊 `clk` 同步，順利解決問題。

● Reflection

原本我們是想利用 pmod 模組的三軸感測器，實作手勢偵測的相關應用，但是深入了解後，發現收集數據需要經過複雜困難的數學轉換，才能變成可讀的加速度資料供我們使用，難度過大，所以再經過思考討論後，我們決定實作難度適中的自動販賣機，它也剛好符合 project 的規則，可以展現 coding 之餘，也能表現出我們硬體實作的能力，完美符合邏輯設計實驗的主旨與精神。

製作的過程中，在軟體方面，經過這一學期老師細心指導與 8 個 lab 的磨練下，並沒有遇到什麼太大的問題，進行的還算是十分順利。在販賣機本體的製作上，因為材料的選擇、不同型別馬達的控制，導致困難重重，但經過勤奮不懈的努力與上網查詢相關資料，所有的問題都一一解決，我們也在時間充裕的情況下，又額外完成了原本 spec 沒開的投幣機，組出一個有模有樣的自動販賣機，讚！

經過這學期 lab 與最後的 final project 的洗禮，我們都成長了不少，體驗了邏設實驗之美，謝謝老師與助教一學期的指導與幫忙。

● Contribution List

王領崧:投幣機、程式整合、VGA 選單、FPGA

鍾皓崑:音樂、投幣機、硬體設備製作、馬達操控

共同完成:demo PPT、Proposal、Report