

主題：

發牌機器人

組長：黃鈺涵

組員：竇賢祐、陳冠霖、趙容

1. 系統功能與原理說明：

實作一個能夠自動發牌的機器人，讓機器人能夠自動發出指定的牌數量並且轉向來發下一堆牌。

以紅外線感測器偵測是否有牌發出，若有就將計數-1，到0時停止發牌，並且輸入指令來轉向，並開始下一次的發牌。

牌的下方利用減速馬達來轉動輪子，讓牌可以發出。

機器人底部後方也有一個輪子用來控制方向，如此可以發出不同堆牌。

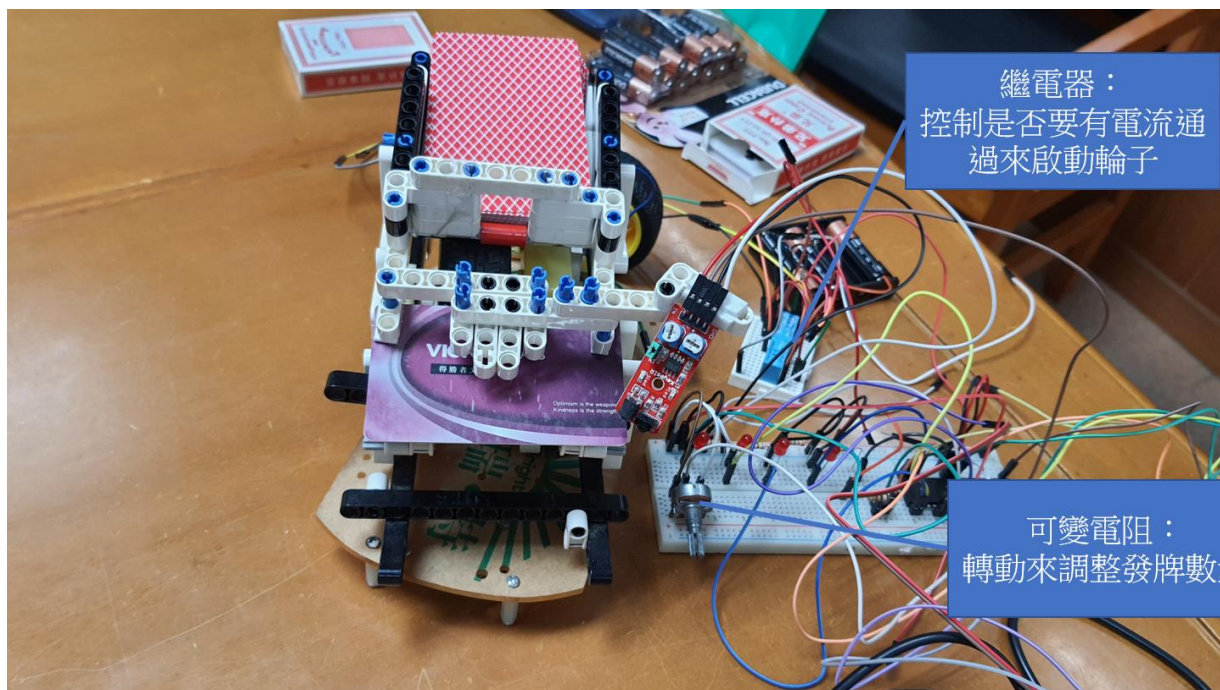
2. 系統使用環境及對象：

每次打撲克牌都覺得發牌很麻煩，有時可能手滑不小心多發一張或少發一張，導致最後每個人的牌數量不一致，因此我們可以利用這個機器人自動發出牌，保證每堆牌的數量都是一樣的，不必擔心手滑等問題導致大家牌數量不同。

.

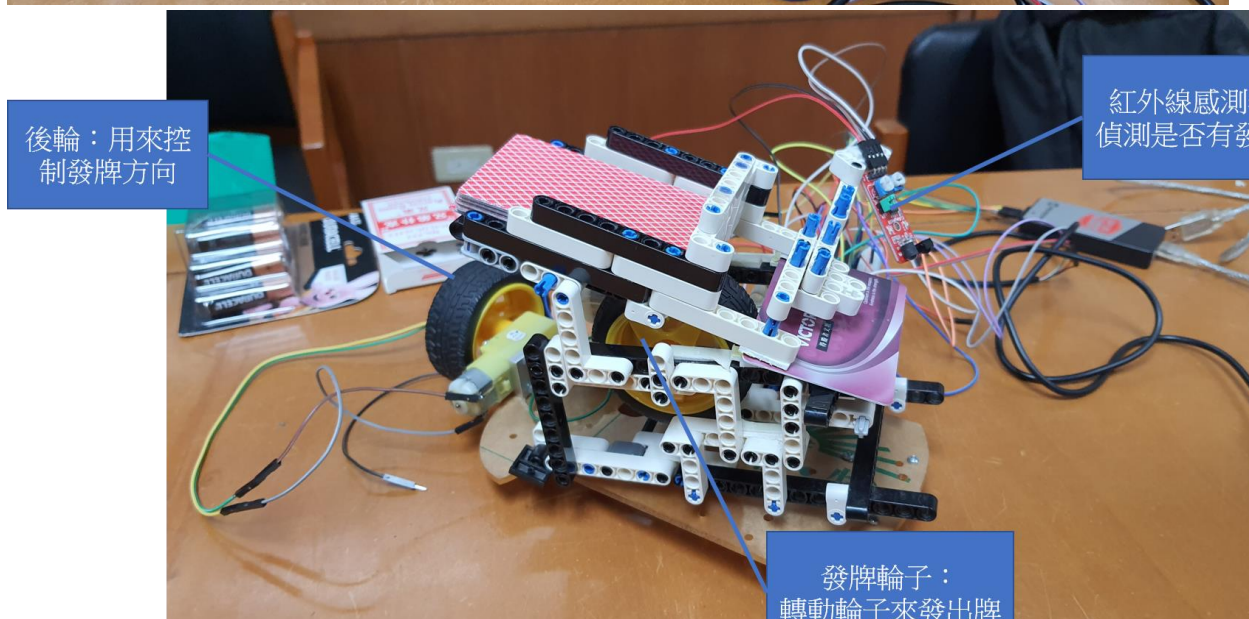
3. 系統完整架構圖、流程圖、電路圖、設計

流程架構:



繼電器：
控制是否要有電流通
過來啟動輪子

可變電阻：
轉動來調整發牌數量



後輪：用來控
制發牌方向

紅外線感測器：
偵測是否有發出牌

發牌輪子：
轉動輪子來發出牌

4. 系統開發工具、材料及技術

開發工具：MPLAB(C code)

材料：樂高、撲克牌、輪子、減速馬達、pic18f4520、紅外線障礙物感測器、pickit 4、繼電器、3號電池、可變電阻、LED燈

技術：須了解繼電器運作原理、紅外線感測器的輸出

設計好能夠支撐整體的支架、利用單一輪子來有不同的發牌堆

ADC -> 設計好Tad的時間等，才能正確抓到轉換完的數值

UART-> baud rate的計算，設定好相關register

interrupt->ADC時因需抓取轉換的內容，因此轉換完畢會跳進interrupt來取得轉換的數值

ADC及UART都會使用到內部震盪器的頻率，所以設定頻率時需兩兩搭配，確定兩邊皆可正常使用這個頻率，才不會導致其中一邊出錯。

5. 周邊接口或library及api使用說明：

輸出：

RD0、RD1、RD2、RD3：LED燈泡

RD5：旋轉馬達(控制後輪)

RD4：發牌馬達(撲克牌底部的馬達)

輸入：

AN7：可變電阻(ADC轉換)

RB5：紅外線偵測(接收紅外線感測器的訊號)

使用之前助教給的UART_Write，來顯示terminal上的輸出。

6. 實際組員之分工項目

黃鈺涵：購買材料(泡棉膠、撲克牌)、code撰寫、電路接線測試

趙容：提供材料(樂高、繼電器、輪子、馬達)、code撰寫、電路接線測試、打報告、研究繼電器使用

陳冠霖：code撰寫、電路接線測試、DEBUG大師

竇賢祐：打報告、研究紅外線感測器使用、電路接線測試、協助DEBUG

7. 遇到的困難及如何解決

1. 馬達選用：

原本最初我們使用的是直流電刷馬達，透過不同大小的齒輪組成一個扭力較大、速度較慢的驅動模組。但是後來發現這樣組起來的馬達雖然可以成功發射撲克牌，穩定性卻很差，無法保證每一次都可以有足夠的扭力將撲克牌送出來。因此我們最後改成使用扭力輸出很穩定的減速馬達來驅動，問題就解決了。

2. 發牌機架構：

原本最初我們想要用簡單的積木、或是紙盒等等的組合來當作我們的系統支架，但是後來發現這幾種方式都不太穩定，因為我們的設計要求需要很精確，稍微有一點不吻合就可能導致一次發出多張撲克牌、或是撲克排無法直線射出等等的問題。所以最後我們選擇花時間(+花錢)用樂高組出一個能夠穩定支撐發牌機支架。

3. 電源供應：

很顯然我們的馬達沒辦法直接使用TTL傳輸出來的電力，電壓不足不能推動馬達，因此需要使用電池盒，使用其他供電源，但又為了讓電池盒不會跟PIC18f4520上的電路打架，所以使用了繼電器控制，讓電流不會互相衝突。

4. 電線短路：

線路沒注意好，不小心正負極直接接在一起，導致電線短路，線路熔掉。

影片及CODE連結:

https://drive.google.com/drive/u/1/folders/16RMdtLbuemv7xy8IS3QGgEcQoPSISu_c?fbclid=IwAR2NverjSY2EhY_X8H8iGA5J2FJ2DP97wAf8WgjLz4pDUtHZQgCDyA3-dio