

微算機期末專題 洪英豪 邱繼揚 張凱盛

原始碼下載連結：

https://drive.google.com/file/d/14qASu6LvjmXsyvq_DW_PWqE8zAzn5s0V/view?usp=sharing

影片檔觀看連結：

https://drive.google.com/file/d/1E7fg595dm2SltZMzip_pxmMGmvyGeBRm/view?usp=sharing

• 系統功能與原理說明

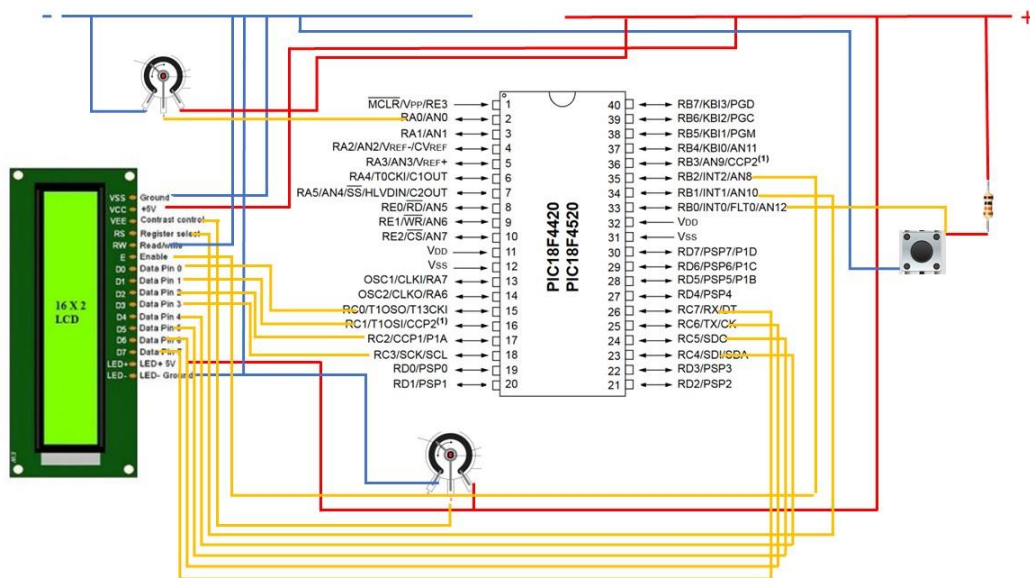
設計一個 LCD 螢幕，透過旋轉可變電阻選擇角色，進入遊戲後透過按鈕控制遊戲角色跳躍躲避障礙物，存活時間越久分數越高，撞到障礙物後顯示 GAME OVER 回到遊戲初始畫面。

• 系統使用環境及對象

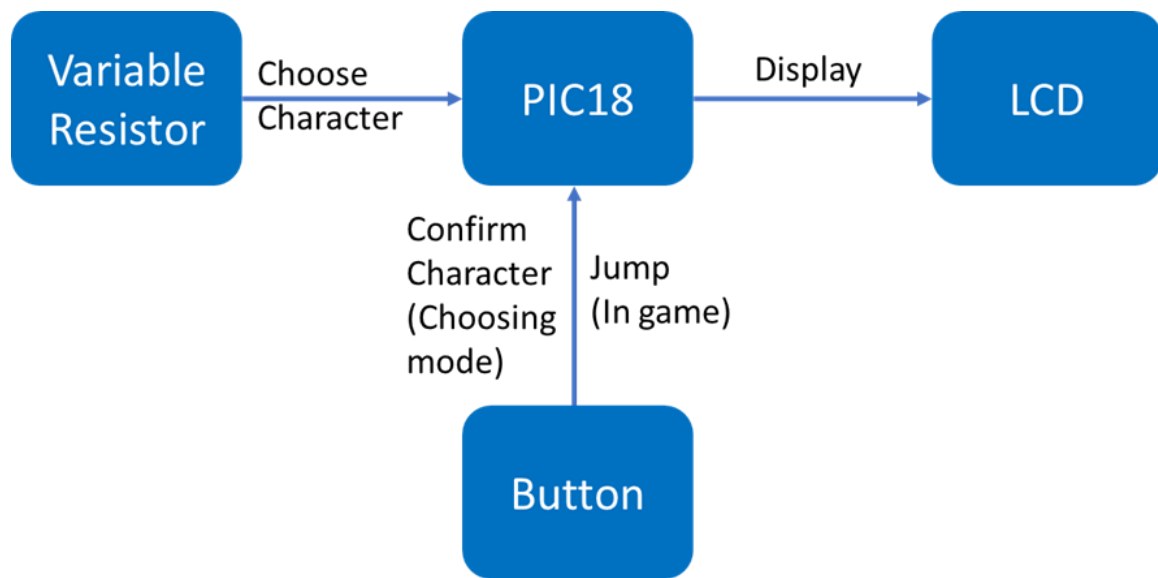
所有人都可以玩的小遊戲

• 系統完整電路圖、架構圖、流程圖、設計

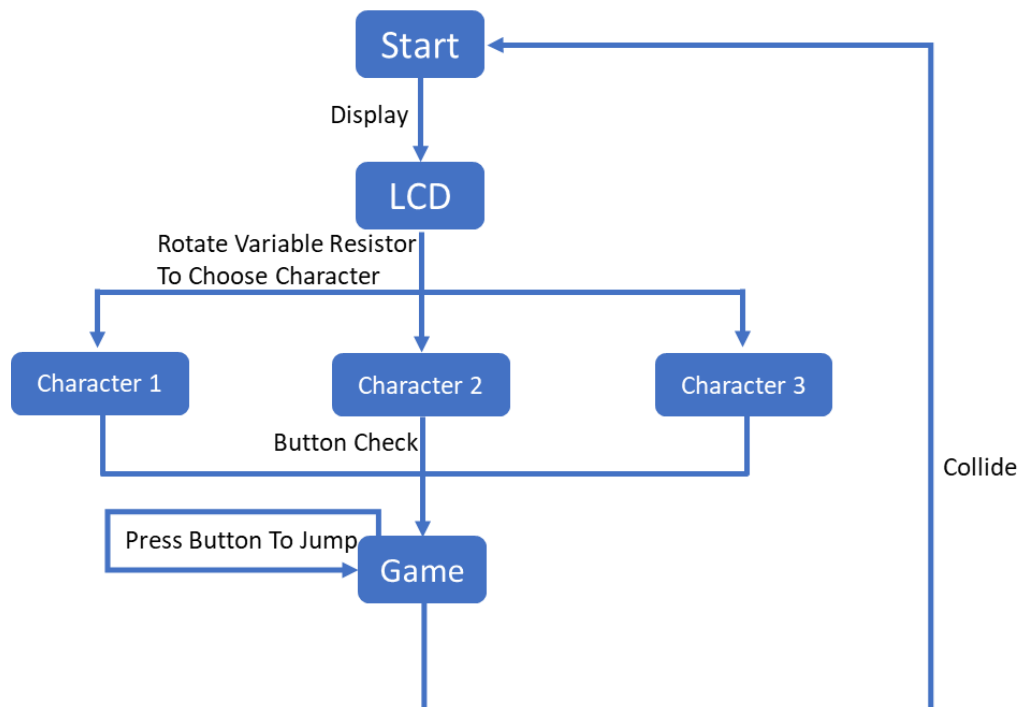
電路圖



架構圖



流程圖



- 系統開發工具、材料及技術

使用 `button interrupt`，當按了 `button` 以後開始遊戲

使用 `timer` 去做遊戲的計時和 `delay`

使用 `ADC` 通過旋轉可變電阻選擇角色

使用 `LCD` 螢幕呈現遊戲畫面

- 周邊接口或 **Library** 及 **API** 使用說明

使用 `16x2 LCD`，並使用 `8` 位模式。

- 實際組員之分工項目

洪英豪: 購買材料、撰寫程式、軟硬體系統整合與測試

邱繼揚: 購買材料、撰寫程式、軟硬體系統整合與測試

張凱盛: 購買材料、撰寫程式、軟硬體系統整合與測試

- 遇到的困難及如何解決

1. `LCD` 螢幕原理探索

我們花了不少時間熟悉 `LCD` 螢幕的原理與使用方法。透過網路查詢相關腳位資訊，加上向 `ChatGPT` 諮詢，我們瞭解了如何將預期畫面呈現在 `LCD` 上，為後續開發打下基礎。

2. 角色選擇畫面延遲問題與優化

在遊戲開始前的角色選擇階段，我們發現畫面更新存在嚴重延遲。由於希望每個角色皆呈現動態效果，每個角色需要佔用六個格子（角色尺寸為兩格），並且透過 `ADC` 中斷所獲取的數位值來動態改變螢幕上顯示的角色。然而，由於 `LCD` 螢幕的記憶體僅能儲存八格圖片，每次 `ADC` 中斷更新角色時，都必須重新將不同的角色畫面載入 `LCD` 記憶體。經過多次測試後，我們推測正是這個載入步驟導致了顯著的延遲。最終，我們的解決方案是捨棄角色選擇時的動態畫面，改為一次性將三個角色各兩幅圖片預先載入 `LCD` 記憶體，從而免除頻繁的重載動作，最終大幅改善了效能。