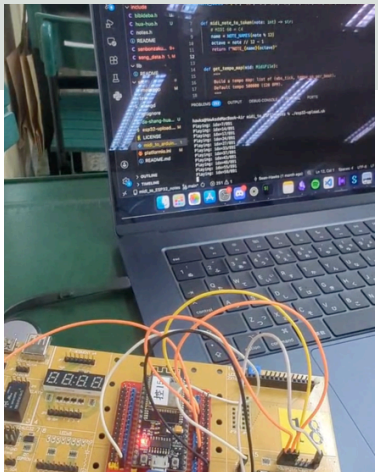

ESP32 MIDI PLAYER

將 MIDI 樂曲轉換為 C++ 陣列的蜂鳴器播放專案

REF

https://github.com/Sean-Hawks/midi_to_ESP32_notes

https://wp.640629.xyz/?p=1360

ABOUT

台北市立大安高級工業職業學校
控制科 -洪軾洋

ENVIRONMENT

- **編譯器：VSCode**
- **作業系統：macOS**
- **開發工具：PlatformIO**
- **硬體：ESP32**

SUMMARY

本實習旨在實作一個簡易的音樂播放器，透過**整合按鈕開關、蜂鳴器（Buzzer）與 LED 指示燈**，並利用 OneButton 函式庫來處理按鍵的互動事件。系統具備播放與暫停切換、長按重置音樂等功能，同時利用 LED 燈提供視覺回饋（播放時閃爍，停止時恆亮）。本專案亦應用了 millis() 函數進行**非阻塞式（Non-blocking）計時**，以確保系統在播放音樂時，按鈕仍能即時響應。

MOTIVATION

為了熟悉**微控制器**（如 ESP32 或 Arduino）的多重硬體控制介面與狀態機邏輯，希望透過**實作音樂播放器**，學習如何處理按鈕去彈跳（Debounce）、**多重按鍵事件**（單擊、長按），以及擺脫傳統 delay() 的限制，達到多工處理的效果。

HOW TO USE

- 硬體接線：
  - 將無源蜂鳴器連接至微控制器指定的 GPIO 腳位。
  - 將按鈕開關的一端連接至指定的 GPIO 腳位，另一端直接接地（GND）。
  - 將 LED 指示燈連接至指定的 GPIO 腳位。
- MIDI 檔案轉換：
  - 在撰寫程式前，需先使用 **Python 腳本 midi_to_arduino_notes.py** 將 MIDI 檔案轉換為 C++ 標頭檔（song_data.h），以產生音符頻率與持續時間陣列。
  - 接著在終端機輸入轉換指令：
  - **python3 midi_to_arduino_notes.py < 您的 MIDI 檔案.mid>**
- 軟體與邏輯撰寫：
  - 引入 OneButton 函式庫與剛生成的樂曲數據標頭檔 song_data.h（以及定義音符頻率的 notes.h）。
  - 初始化 OneButton 物件，並綁定單擊（Click）事件來切換「播放/暫停」狀態，以及長按（Long Press）事件來執行「重置並停止」。
  - 主程式的迴圈（loop()）中使用 millis() 進行非阻塞式計時，**控制蜂鳴器的發聲節奏與 LED 的閃爍頻率。**

FEATURE

只需要透過 midi_to_arduino_notes.py 這個工具，就可以從 **midi 檔案直接轉成音高和節奏的陣列**，所以可以方便的透過這個功能完成換曲子的動作，不需要像是其他同儕一樣一個一個慢慢調音節和節奏，這是這個專案**最有特色**的地方

https://youtube.com/shorts/8LcLHq8pObk?si=IWoKHGMI1EyaQekj

INSIGHTS

這次的實習作業原先的條件只是要讓我們一個一個把音樂跟節奏給手戳出來，不過也是因為這樣，不過因為過程很枯燥，所以大多數人都直接讓 AI 把曲子的音高和節奏一鍵生成。不過理所當然，由於 LLM 的特性，**大型語言模型根本沒辦法有效率的把曲子生成**，所以我決定做一個 **Python 腳本**，從 Musescore 或各種地方抓到 MIDI 檔後，可以很方便的在不到一分鐘內就把一首曲子給抓下來。所以在實習過程中其實還蠻有成就感的。我也順便把所有的原始碼都放在 GitHub 上**開源**了

FUTURE

希望之後可以有辦法把音樂的**資料庫**建起來，達成一個按鈕就可以下一首或下一首的功能，如果有一個**顯示器**做出來按鈕的反饋我覺得也是不錯的想法。