中華民國第 57 屆中小學科學展覽會
作品說明書

高級中等學校組　電腦與資訊學科

052512

MIDI 樂器（電子琴）

學校名稱：天主教振聲學校財團法人桃園市振聲高級中等學校

作者：	指導老師：
職三 柯劍雲	陳國生

關鍵詞：Arduino、MIDI 樂器、陣列鍵盤

摘要

本科展專題將嘗試製作出電子琴，並且對電子琴製作的歷程進行記錄。電子琴將擁有一個控制介面、一個鋼琴鍵盤、一個自製鍵盤、以及一張以 MIDI 為基礎的音效卡。以 Arduino Mega 2560 製作，效能強大。

使用 3.5mm 接口將音源信號接至喇叭之後，您只要照著習慣彈奏鋼琴，便可以彈奏出數十種的樂器（意即不單單只有鋼琴聲可以演奏）。

設備也會裝置一具有 LCD 顯示器的機板和按鈕，使用者可以透過編寫得來的簡易人機介面，隨時對設備進行調整。

該設備可以連接上電腦，進行協同運作，給予使用者更為進階的功能。而相反地，設備應該也要能夠單機運作，並不需要完全依賴外部進行電力供應或是資料輸入。

壹、 研究動機

在電子產品大行其道的今天，有許多的音樂設備也跟著電子化。而從中也延伸出極多的理論，製造了龐大的市場。

直至今日，電子樂器實際上已經圍繞在我們的身旁，不管是電視節目、場地活動的配音，還是我們所熟悉的音樂舞曲。因此，學習、研究，甚至實作一台電子樂器成為了迫切的需求。

貳、 研究目的

一、 找出可以製作優美旋律的方式

1. DAC / PWM? 製作出弦波的方法
2. 樂器數位介面（MIDI）

二、 讓微控制器可以控制音聲、音高的方式

1. Arduino Mega 2560 開發板
2. Music Instrument Shield 音效卡
3. 陣列鍵盤原理
4. 製作環節

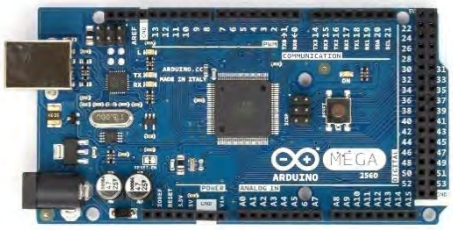
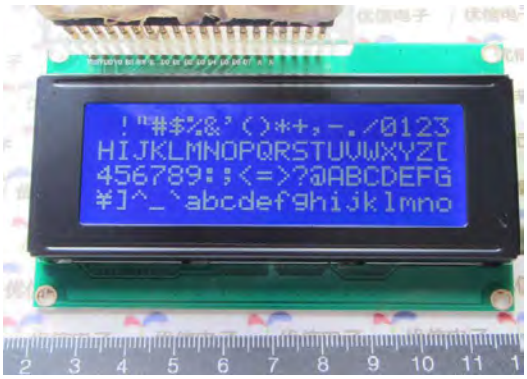
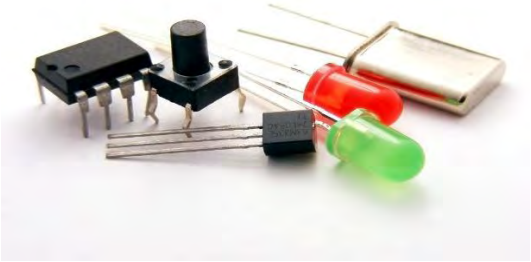
三、 如何讓使用者操縱

1. LCD 顯示器：2004A 的特性
2. 介面特性
3. 製作環節

四、 硬體架構的設計、製作

1. 硬體動作流程圖
2. 硬體電路、硬體圖
3. 不使用函式庫實作陣列鍵盤
4. 連接上電腦

參、 研究材料

材料名稱	圖像	備註
Arduino Mega 2560		本次專案使用之微控制器。
SparkFun Music Instrument Shield		本次專案使用之音效卡。
LCD 2004A		本次專案，讓使用者和機器溝通的橋樑。
陣列鍵盤		本次專案之輸入單元。
數以千計的電子/電工元件		諸如 74HC 系列 IC、LED、開關、萬用電路板、漆包線等。

肆、 研究過程和方法

一、 找出可以製作優美旋律的方式

起初，在整個方案尚未確立、思考的時候，我沒有限制住自己的想像力，可惜的是，各式各樣的點子竟都無法實行。

當真正開始著筆設計鋼琴的基礎架構時，發現這並不簡單。首先，一個樂器會是樂器，表示它有著獨特的音聲，並且可以演奏出旋律(音調)。

首先是音調，其原理非常簡單，將一個波動震盪於 440Hz；這是一個國際公定的音調代號為 A4（四度 Ra），其處於音頻的”中間”。

根據樂器十二平均律定理，只要對這個 440Hz 乘上 $2^{K/12}$，便可以求出升高 K 個音調後的頻率。

但是，事情卻不是如此簡單！因為知道了一個波動震盪的頻率，卻還得知道另一件事情是：波形可以決定聲音聽起來怎麼樣。

假設在喇叭上放一 440Hz 的方波，其確實會是四度 Ra 的音調，但因為波形是方波，峰值始終如一而沒有滑潤感，那是非常難聽的！所以，另外還得設想一個好方法，找出製作樂器產生之弦波的方式。、

（一）DAC / PWM? 製作出弦波的方法

首先，電子學課本，或者微電腦性質的課本上有這類的課程：透過一些對方波電壓峰值的調整，製造出正弦波的方式，例如簡易的 R/2R 電路或是著名的 ID：DAC0804。

它們確實都可以讓我們製造出一些弦波動，然而其中有一嚴重的致命傷，以致於此方法並不能使用，即聲音的性質依舊無法確定。

一個電子琴；使用 DAC0804，程式寫好、硬體布置好，卻永遠只能聽到 8-bit 的假弦波，其實依舊不是很優美。因為其雖然比方波好很多，聽起來卻仍舊不是鋼琴。

（二）樂器數位介面（MIDI）

在不斷地搜尋資料中，有一名詞將我的眼光徹底地吸引住，即為”*MIDI*”。

樂器數位介面（Musical Instrument Digital Interface，簡稱 MIDI）是一個工業標準的電子通訊協定，為電子樂器等演奏裝置（如合成器）定義各種音符或彈奏碼，容許電子樂器、電腦、手機或其它的舞台演出配備彼此連接，調整和同步，得以即時交換演奏資料。

MIDI 使得電腦、合成器、音效卡以及電子樂器(電子鼓、電子琴等)能互相控制、交換訊息。現在電腦的音效卡都是與 MIDI 相容的，並能逼真地模擬樂器的聲音。

引用於 Wikipedia

MIDI 在大概三十年前被提出，提出後隨即火熱於世界。受限於當時網路頻寬的不足，一個數 MB 的音樂檔案並沒有辦法在網路上進行串流。但是一個只有幾十 KB 不到的 MIDI 樂譜檔，由於容量極小，因此傳輸上方便，還可以做出非常多延伸。

只要一張支援 MIDI 的音效卡、以及支援.mid 格式的播放器，便可以直接播放樂譜裡面收錄的資料。

結論

由於音頻製造、音頻放大等等設備難度較高，且製作時間較長，或許使用這些方式並不是非常良好。

而由於現在電子樂器的潮流都是跟隨 MIDI 而走，如果我們也能夠使用 MIDI 設備來完成設備的話，不僅在未來能夠與世界的潮流接軌，製造難度也會大幅度降低。

根據調查 MIDI 動作的方式發現。倘若我們能夠找到方式讓微控制器可以處理 GM1 語言，再配合鋼琴鍵盤，就能以簡單的方式完成電子琴設備。

二、 讓微控制器可以控制音聲、音高的方式

（一）Arduino Mega 2560 開發板

Arduino Mega 2560 是 Arduino.cc 組織開發出的一片開發板，它擁有 54 隻數位輸入/出接腳、以 Atmega2560 微處理器作為其核心，工作於 16MHz 的時脈…。

正因為其強大的能力，以及極多的 GPIO（通用輸入出腳位），此開發板可為開發者提供強大、安穩的後台，來協助開發者來完成想做的事情。

（二）Music Instrument Shield 音效卡

本次專題中，MIDI 占了非常大的表現戲碼，但是近年來個人電腦的發達，讓 MIDI 通常是製作成電腦軟體的。換句話說，如果想用 MIDI 演奏音樂，或許每次都需要準備一台膝上電腦或台式機。這非常的可惜！試想一下，每次想在外演奏音樂都必須把電腦帶著的壓抑感，那是非常不方便的。

Music Instrument Shield 是 Sparkfun 公司研製出的一張音效卡，其使用 VLSI 公司製作的一個特殊用途 IC：1053B 來當作核心；以 31250 鮑率的串列速度傳輸資料，即標準的 MIDI 速率；具備一 3.5mm 音源輸出孔，可以直接驅動 1/8 瓦的喇叭，或是送出音訊給主動式的喇叭。

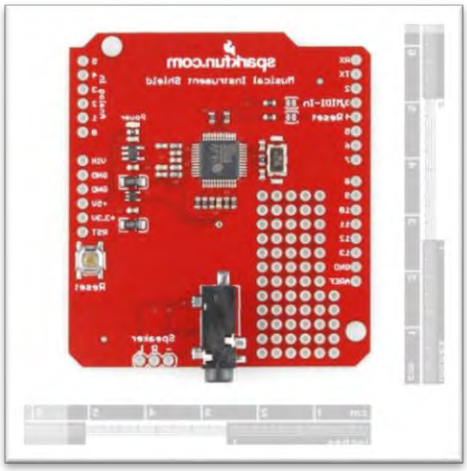


圖 1：Arduino Mega 2560　　圖 2：Music Instrument Shield

（三）陣列鍵盤原理

一台鋼琴通常會有很多很多的按鈕；有白有黑。如果要把 88 個按鈕都單獨接上一條線(全尺寸鋼琴有 88 鍵)，那縱使再強大的開發板，也無法滿足一鍵一線的需求。

陣列鍵盤是一種以高速迴圈不斷偵測二維陣列的技巧，透過將一橫列的按鈕全部接起來，然後再將一直列的按鈕也全部接起來，將其接成如(圖 3)的樣子。

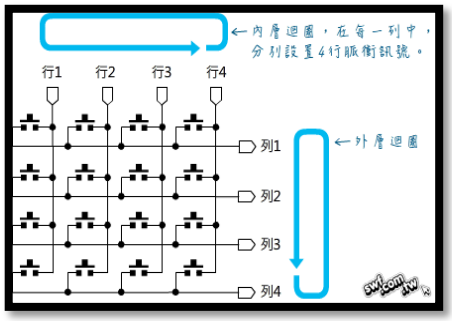
以直行或橫行為基準點，然後控制迴圈，同時只讓某一基準行為真。

如果說，基準行為真時，偵測行卻為假，表示這一行並無按鈕被按下，於是就可以讓迴圈進入下一次偵測；可以觀察(圖 4)來協助理解。

倘若某一次基準行為真時，偵測行居然也為真，表示其兩行交叉之交點上的按鈕被按下了，程式就可以透過變數紀錄下按鈕狀態，然後開始下一次迴圈。如(圖 5)。

如此周而復始，讓處理器不斷進行迴圈偵測，即可將整個鍵盤的狀態掌握在手中，這種陣列鍵盤的技術被稱做掃描（Scan）。

只要透過這個技術，就可以將接線大量減少。其減少的數量可以為：按鈕數量之兩因數相加（例如 90 鍵可以是 9 x 10，那麼接線量可以減少為 9 + 10 = 19 隻）。



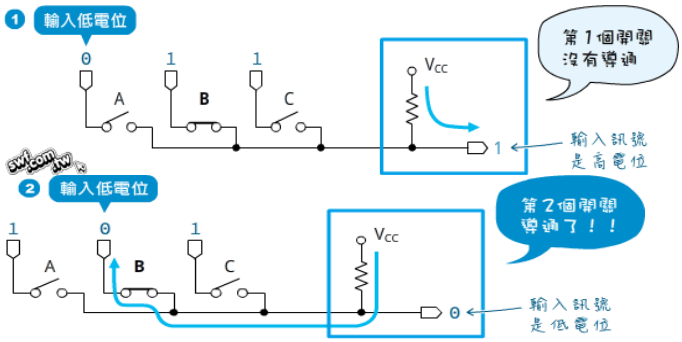


圖 3：接線參考、圖 4(上) 和 圖 5(下)：偵測與否

（四）製作環節

硬體

首先，硬體方面，在草擬鍵盤的設計圖(圖 6)後，我使用以前專案所剩機械鍵盤軸承在麵包板上接出了一個陣列鍵盤，並且將其安置在已裝有音效卡的 Arduino 上。

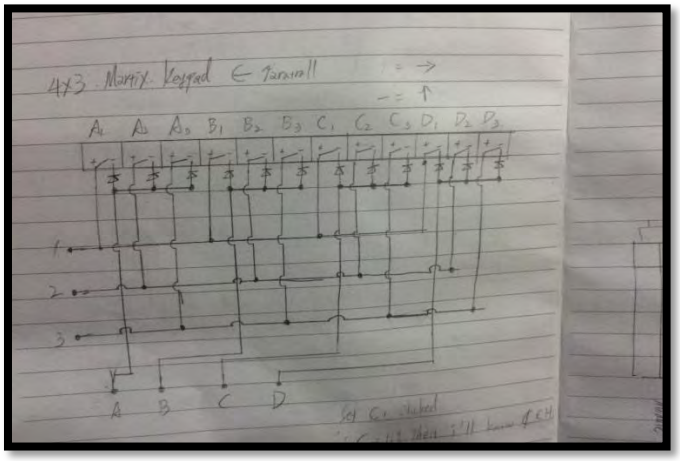


圖 6；鍵盤設計草圖

軟體

在軟體環節中，得做到兩點必須做到的控制，分別是鍵盤控制和 MIDI 控制。

關於軟體編寫，有一點特別重要：對於一個開發者來說，其開發的程式往往不是從”零”開始，對於程式設計師來說，為了縮短軟體的開發週期，常常會使用所謂的函式庫。活用函式庫，不僅可以協助程式的穩定度，更可以讓整個程式編寫更加簡易。

接下來的說明請參照(文 1)。

首先，根據(圖 6)，可以知道這個鍵盤的結構是四列三行，共七線。

在#include 之後，便能夠開始使用 Keypad 函式庫的定義。在將各個值定義後，鍵入 Keypad(鍵盤地圖 行腳位 列腳位 行數量 列數量)函式，便可以完成少線且具完善功能的鍵盤。由於有了函式庫，前後的實作只需數秒鐘即可完成！

接下來的說明請參照(文 2)。

建立一段程式碼。運作中，Arduino 檢測到了鍵盤的電位狀況改變了(按下按鈕或釋放按鈕都會改變鍵盤的電位狀況)，就會開始對鍵盤進行掃描動作。掃描後，我們可以得知這個按鈕是被按下、按住、釋放還是閒置。

只需取按下跟釋放兩個狀態來使用。倘若按下的話，就對音效卡發送出這個鍵的音節，如果釋放按鈕則對音效卡說發聲停止，雖然可能鋼琴不需要發聲停止這一行，但如果以直笛等參數當樂器時，程式就會需要發聲停止的功能。

```
#include <Key.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
//陣列鍵盤宣告
const byte ROWS = 4; //列 橫
const byte COLS = 3; //行 直
char keys[ROWS][COLS] = { //鍵盤輸出定義 如果你需要特殊用途,才定義每個按鈕的名子,因為我們將會使用kcode而不是kchar
  {'1', '2', '2'},
  {'3', '4', '2'},
  {'1', '2', '2'},
  {'1', '2', '2'},

};
byte rowPins[ROWS] = {23, 25, 27, 29}; //列接線定義
byte colPins[COLS]  = {22, 24, 26}; //行接線定義

Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

文 1：陣列鍵盤程式碼

```
void Com1() {
if (kpd.getKeys()){  // 如果鍵盤之電位被改變
  for (int scan=0 ; scan<LIST_MAX; scan++) { //開始對鍵盤掃描
    if (kpd.key[scan].stateChanged ) {
        switch (kpd.key[scan].kstate) {  //回報作用按鈕的狀況
            case PRESSED: //如果此按鈕是 被按下
              noteOn(0, kpd.key[scan].kcode+60,95); //發出按鍵位置的音 +60(一個四度)
            break;
            case RELEASED:
             noteOff(0, kpd.key[scan].kcode+60, 60); //結束按鍵位置的音 +60(一個四度)
            break;}}}}}
```

文 2：鍵盤控制發聲程式碼

三、 如何讓使用者操縱

一台電子設備，如果無法讓使用者依照目前環境所需來控制，以得到自己所需的功能，那便會非常遜色。

如果可以額外模組出一個螢幕與控制鍵盤，以對設備進行隨時控制，便可以將 MIDI 音效卡所擁有的功能全部發揮出來。

（一）LCD 顯示器：2004A 的特性

2004A 是一片液晶文字顯示器。四行，每行二十字。透過 HD44780 相容的解碼晶片，可以讓顯示器上顯示出英文、數字和希臘字母。

我們可以透過 Liquid Crystal 這個 Arduino 內置的函式庫，並使用正確的接線，來讓 LCD 顯示器列出字元，顯示資訊。

（二）介面特性

有一麻煩的問題：Liquid Crystal 這個函式庫只提供了字元的顯示，卻沒有提供”頁”的功能。因此，為了讓 LCD 顯示出介面訊息，我們得編寫出具介面功能的程式。

在普通的操作中，螢幕上會出現標題與選項，我們可以透過小鍵盤來控制指標，而指標選擇到我們所需的功能選項時，可以按下確定鍵進入其選項之分支。

然而，適當控制指標的程式其實非常複雜。舉例如：假設指標現在正在第四行，也就是面板的最下方。然而下方其實還有選項。那麼，程式該做的便不是移動指標，而是將所有選項往上拉。

在某些狀況下，我們需要輸入數字來調整設備，例如調整基礎音色、設備當前樂器等設置。在這些狀況下使用者應可鍵入小鍵盤上的數字及確認鍵對設備進行設定。

因此，我們可以透過上述的線索，列出圖 7 的流程圖。

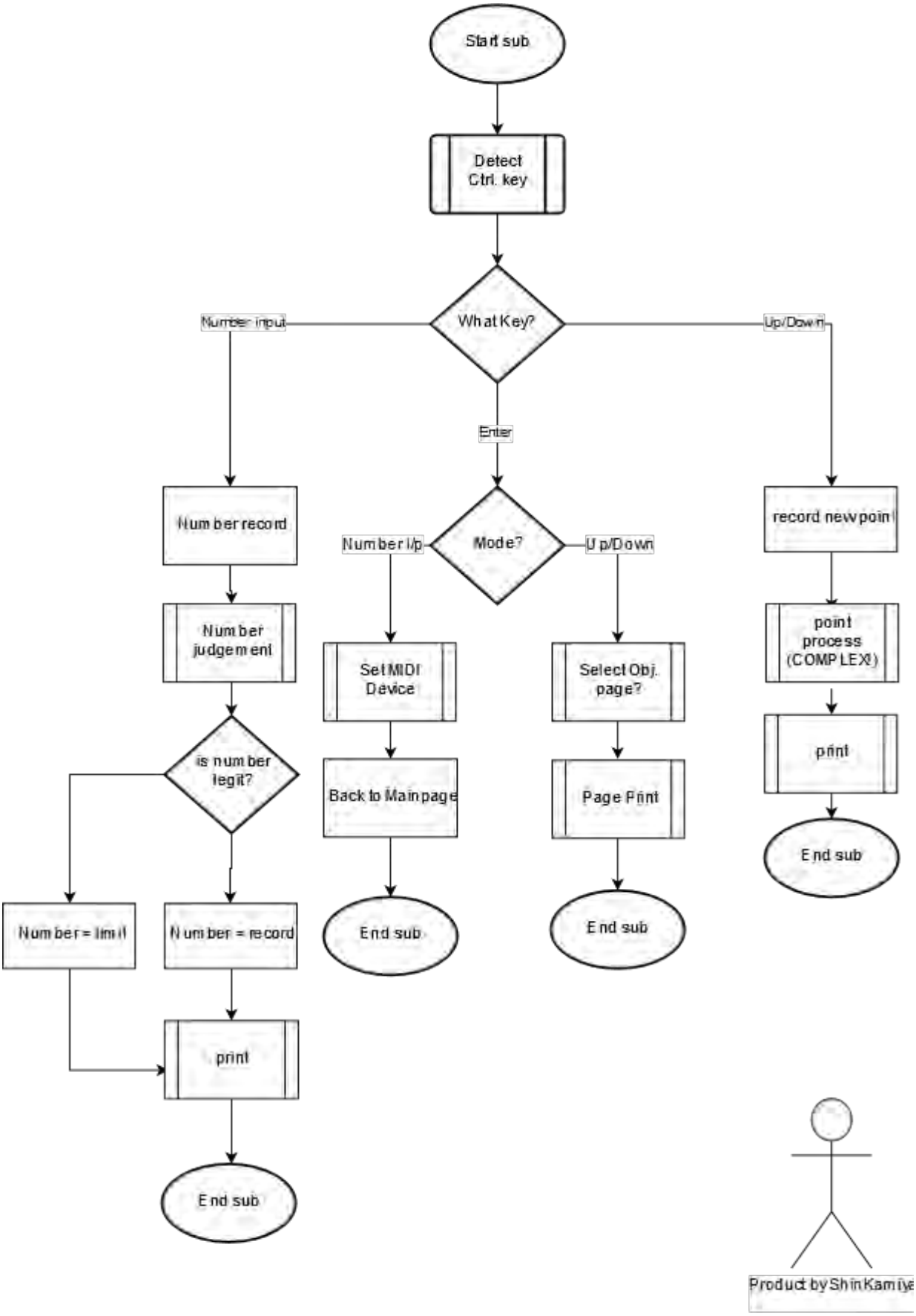


圖 7、LCD 顯示器流程圖

（三）製作環節

硬體

在這個實作測試中，我們需要準備一個陣列鍵盤，以及 LCD 顯示器。

由於陣列鍵盤的製作已經在前面的測試中製作過，在這個測試裡面只需沿用即可。而 LCD 顯示器的接線圖，可以根據表 2 的線索，接成如圖 10 的樣子。

軟體

由於操控介面的軟體較為複雜，軟體的部分煩請直接瀏覽專題之原始碼，而不在報告上多加著筆。

腳位編號	名稱	說明
1	Vss	接地 (0V)
2	Vdd	電源 (+5V)
3	Vo 或稱 Vee	對比(0-5V), 可接一顆 1k 電阻，或利可變電阻調整適當的對比
4	RS	Register Select: 1: D0 – D7 當作資料解釋 0: D0 – D7 當作指令解釋
5	R/W	Read/Write mode: 1: 從 LCD 讀取資料 0: 寫資料到 LCD 因為很少從 LCD 這端讀取資料，可將此腳位接地以節省 I/O 腳位。
6	E	Enable
7	D0	Bit 0 LSB
8	D1	Bit 1
9	D2	Bit 2
10	D3	Bit 3
11	D4	Bit 4
12	D5	Bit 5
13	D6	Bit 6
14	D7	Bit 7 MSB
15	A+	背光(串接 330R 電阻到電源)
16	K-	背光(GND)

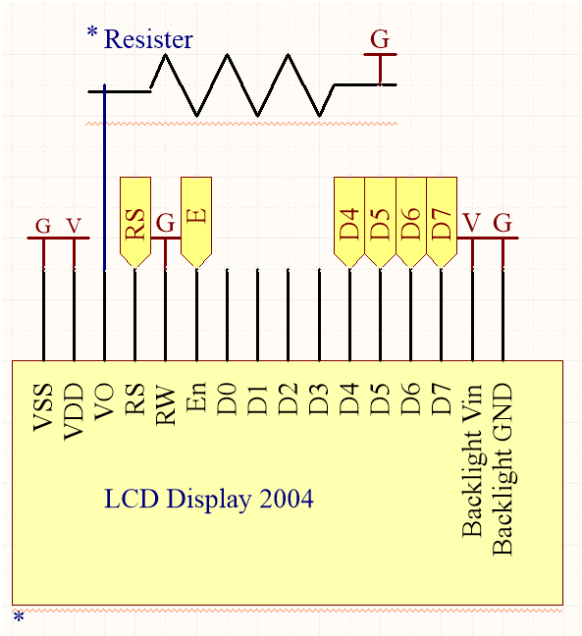


表 2、接線定義 和 圖 8、電路圖

當實作完成之後，其操作表現，您可以參閱：https://goo.gl/lXjNkV

四、硬體架構的設計、製作

（一）硬體動作流程圖：

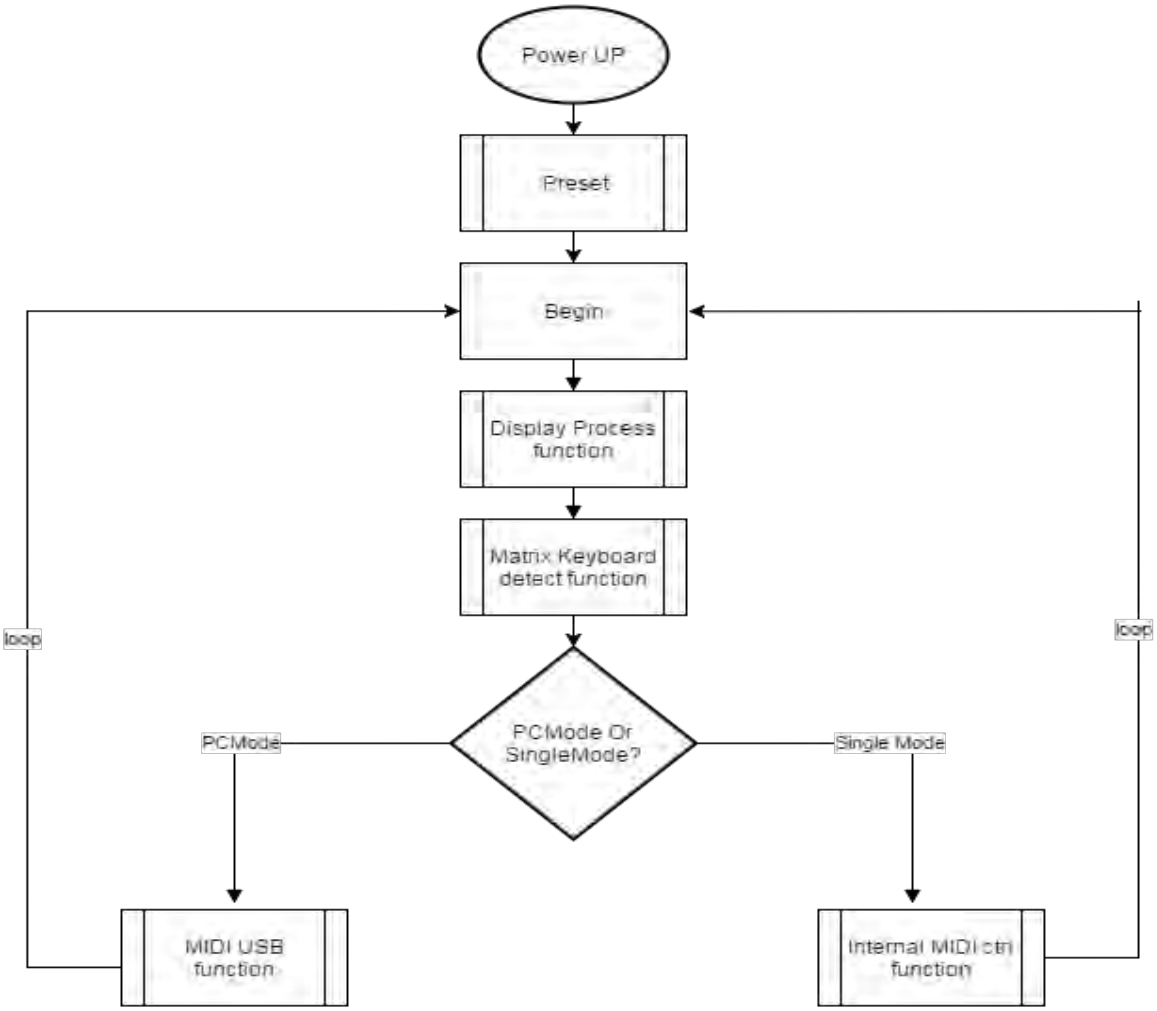


圖 9、簡易硬體電路流程圖

（二）硬體電路、硬體圖

首先，在硬體電路上，有四個主要區塊，分別為：微控制器區塊、鋼琴陣列鍵盤區塊、使用者設定區塊以及電源區塊。

第一個區塊：微控制器區塊即 Arduino Mega，主要用途為處理並收發數位資訊。請參閱圖 12 的硬體電路圖。

這個區塊，最重要的是接線的用途。每個接點有不同的行為，而每個接點都可以對應到表 3 的接線定義。

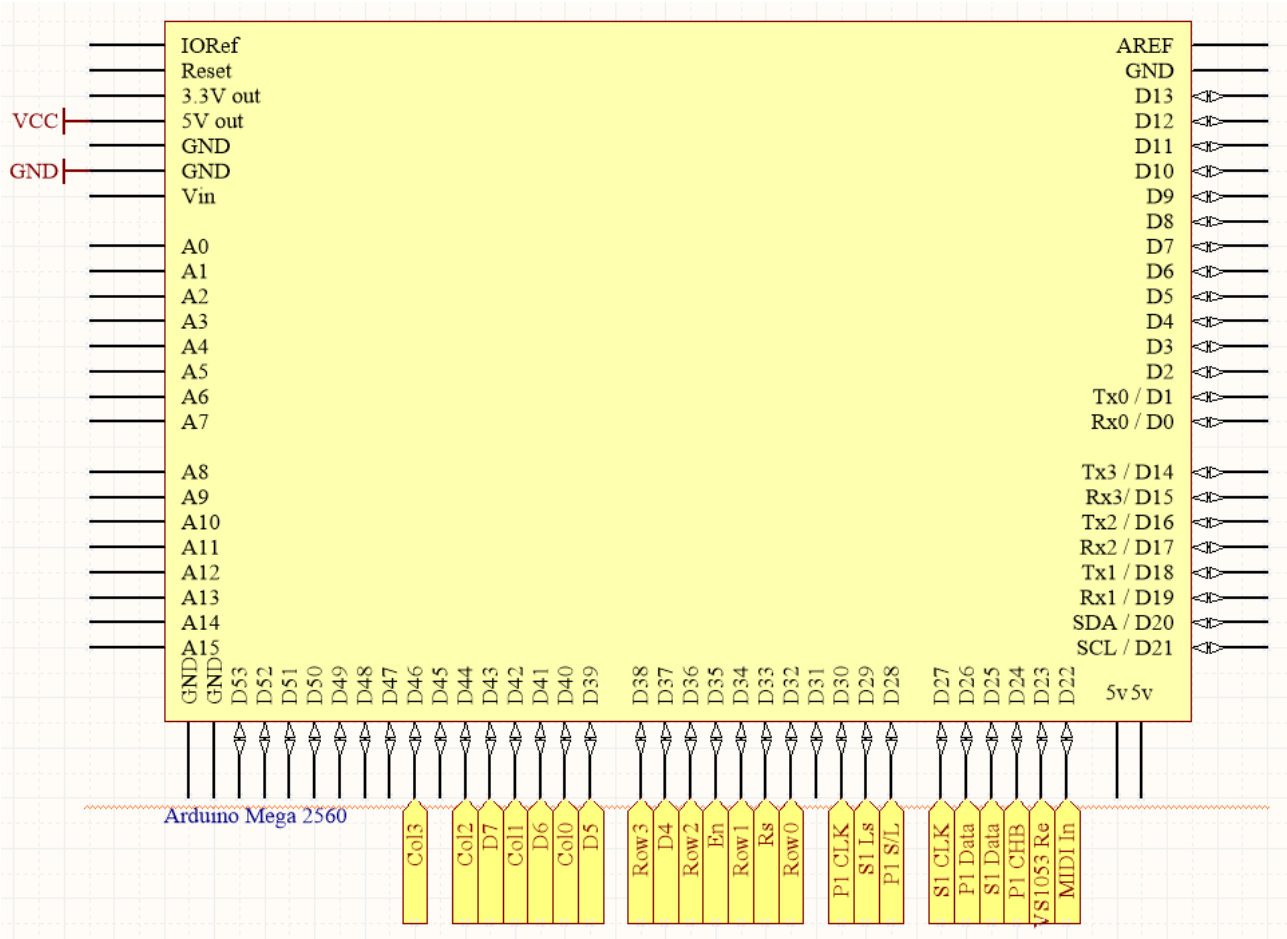


圖 10、Arduino Mega 接線圖

Arduino Pin	接線名子	用途	備註
D22	MIDI in	以 31250bps 傳送 GM1 語言給音效卡	D22 另為 SerialTx
D23	MIDI Reset	重設音效卡	低態，時間 100ms
D24	P1 CLKHIB	對 74165 IC 控制其行為	低態時可輸出資料
D25	S1 Serial Data	輸入串列資料給 74595 IC	

表 3、接線定義

Arduino Pin	接線名子	用途	備註
D26	P1 Serial Data	接收 74165 IC 給予的串列資料	
D27	S1 CLK	給予 74595 的時脈	正緣動作
D28	P1 SH/LD	對 74165 IC 控制其行為	LD 低態，SH 時輸出資料，LD 時儲存資料
D29	S1 Latch	對 74595 IC 控制其行為	低態時不可輸入資料
D30	P1 CLK	給予 74595 的時脈	負緣動作
D32	Kpd Row0	陣列鍵盤列 0	掃描電位進入
D33	LCD Reg	顯示器之 RS 腳位	
D34	Kpd Row1	陣列鍵盤列 1	
D35	LCD Enable	顯示器之 En 腳位	
D36	Kpd Row2	陣列鍵盤列 2	
D37	LCD Data4	顯示器之 D4 腳位	D0,D1,D2,D3 空接
D38	Kpd Row3	陣列鍵盤列 3	
D39	LCD Data5	顯示器之 D5 腳位	
D40	Kpd Col0	陣列鍵盤行 0	掃描電位接收
D41	LCD Data6	顯示器之 D6 腳位	
D42	Kpd Col1	陣列鍵盤行 1	
D43	LCD Data7	顯示器之 D7 腳位	
D44	Kpd Col2	陣列鍵盤行 2	
D46	Kpd Col3	陣列鍵盤行 3	

表 3 續、接線定義

第二個區塊：鋼琴陣列鍵盤在使用者操作時，可以反應出聲音信息回饋給使用者。

由於此區塊並沒有使用函式庫 Keypad，而是使用暫存器 IC 與純理論重新製作。其硬體、軟體規模較大，因此，會在後面為這個區塊單獨講解、分析。

第三個區塊：使用者設定區塊，可以讓使用者在設備單機運作時，對設備進行設定，以輔助單機運作。其接線方式如圖 11。

透過左方的 4x4 陣列鍵盤，我們可以控制右方的 LCD 面板。由於左方的陣列鍵盤尺寸並不大，因此我們直接使用 Keypad 即可。所以程式碼實際與上面對陣列鍵盤的時作為大同小異。

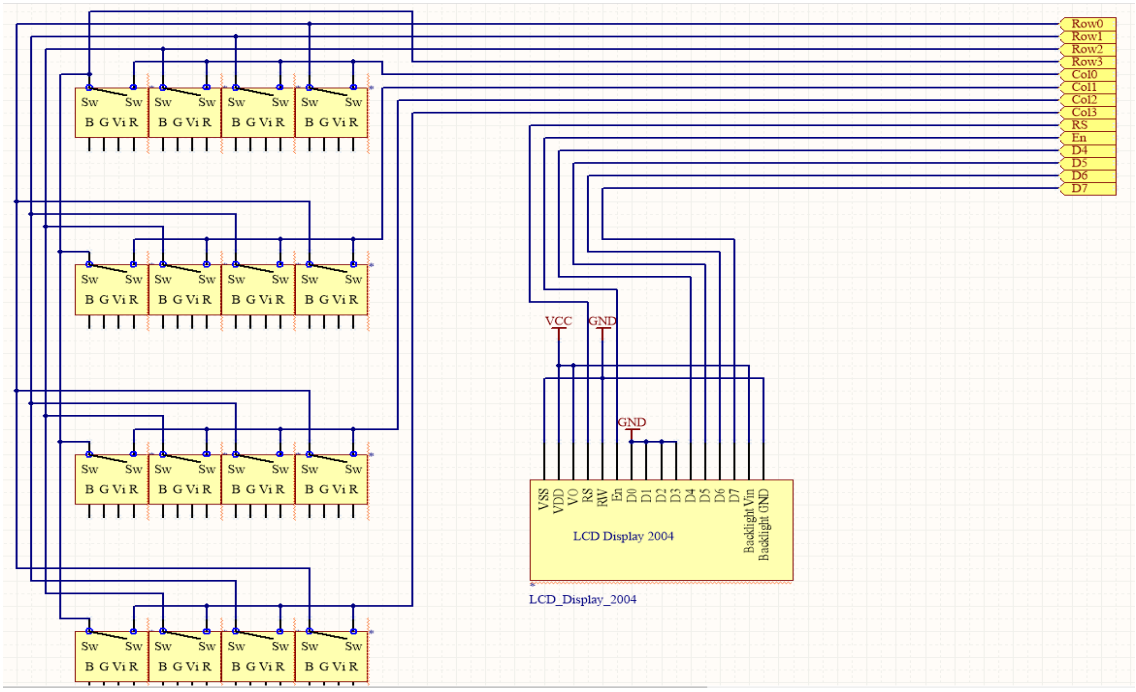


圖 11、使用者設定區塊之接線圖

第四個區塊：電源區塊可以讓設備不一定需要接上電源才能運作，從而實行單機運作的功能。其接線方式如圖 12。

主要做法為：透過一塊只有電路，無外殼的行動電源，直接將其上的電力端接至電路的電力供應區，並將外部供電的線路接回電源的充電端。

在實作中，電源充電端實際上會與電源供應區短路，產生假負載效應而無法動作，因此加一顆二極體讓電路只以單向行走。

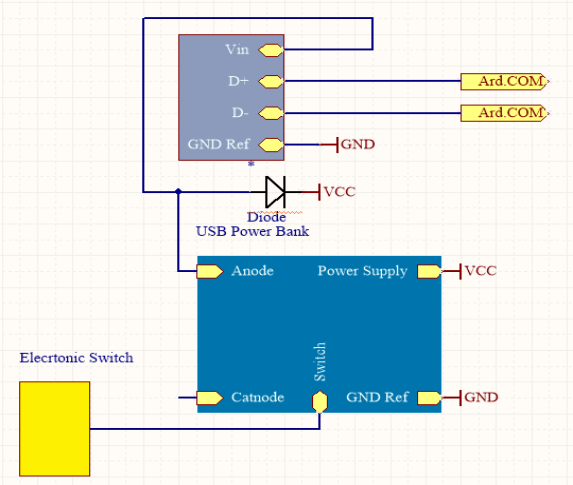


圖 12、電源區塊

（三）不使用函式庫實作陣列鍵盤

在本次專題中，鋼琴控制器使用的是 Roland(羅蘭)公司出品的 F-30 系列電子琴的陣列鍵盤，機板型號為 F-Sk-8p7。其為薄膜鍵盤，並在每個按鈕都裝置有開關二極體。

而在鍵盤上裝有二極體的原因是因為陣列鍵盤具有一嚴重的缺陷：鬼鍵（Ghost Key）

鬼鍵是甚麼？鬼鍵的定義是：當陣列鍵盤同時按下多個按鈕時，會有某些按鈕遭到意外導通。

今假設有一 2x2 陣列鍵盤，長相、定義如圖 13。我們給予三個狀況：第一個狀況如圖 14，我們按下 Q 鍵，鍵盤透過 C1 掃描，與 R1 線路導通，而 R2 與 C1 則是斷路。再來透過 C2 掃描時，其與 R1、R2 皆為斷路，因此我們就能夠斷定 Q 鍵被按下。

第二個狀況如圖 15，我們同時按下 Q 和 S 鍵，C1 掃描與 R1 導通，與 R2 斷路。第二個時間單位 C2 掃描，與 R1 斷路，與 R2 導通，因而斷定 Q 和 S 鍵被按下。

而第三個狀況卻出了問題。基於圖 16，我們同時按下 Q、W、A 鍵，第一個時間單位 C1 掃描時，與 R1、R2 都是導通。掃描到 C2 時，因為 W 鍵的關係，C2 與 R1 是導通的，但是與 R2 的關係卻因為其他三鍵的短路線，因而與 C2 導通了。因此電腦將會得到：四個按鈕都是被按下的這個結論。而 S 鍵實際上並沒有按下。此時，S 鍵所發生的狀況就是所謂的鬼鍵。

可以發現前兩個狀況都原先都是正常的，但當我們假設第三個狀況時，結果卻會不如預期。這在一些多鍵操控上將會是極為嚴重的問題，鋼琴就是一種。而解決方法有很多，如電阻量偵測、製造 ASIC 檢測器…但其中最直接、簡易的方法還是在電位回傳的地方加上一顆二極體，如 1N4148 或蕭特基類型的高速導通二極體。

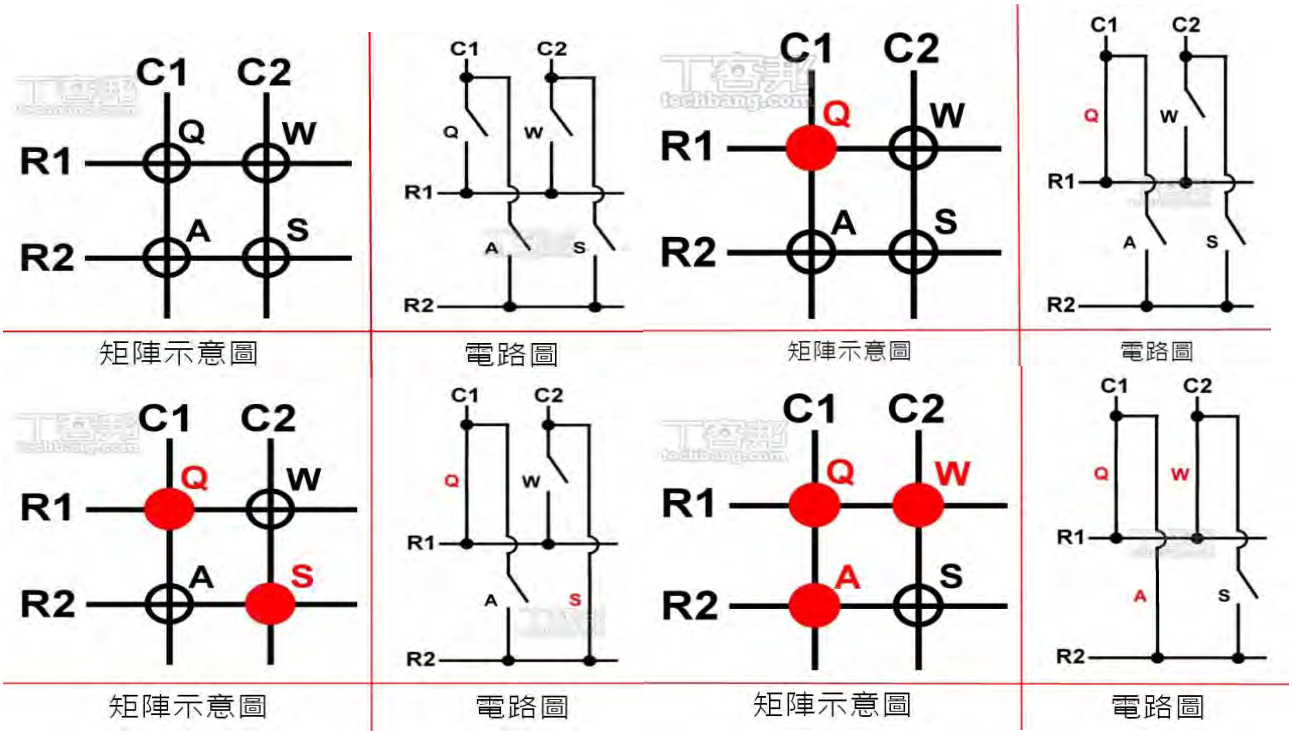


圖 13(左上)、圖 14(右上)、圖 15(左下)、圖 16(右下)

當我們加上二極體後，電路將變成如圖 17，這時，可以回頭重新檢查 Q、W、A 同時按下的狀況中，會不會依舊讓 S 被觸發。

當 C1 開始掃描，如圖 18，C1 將與 R1、R2 短路，而 Q、W 確實是有按下的，所以第一個時間單位所得到的資料正確。

第二個時間單位，C2 開始掃描，R1 透過 A 鍵與 C2 導通，而重點在於：R2 原本應該會因為 A 和 W 鍵的短路線而也造成與 C2 的導通。然而在這個電路上，二極體卻有效地阻隔這些倒灌的電流，所以 R2 並沒有和 C2 接通，電腦從而能正確判斷出 S 鍵並沒有被按下這個結論。

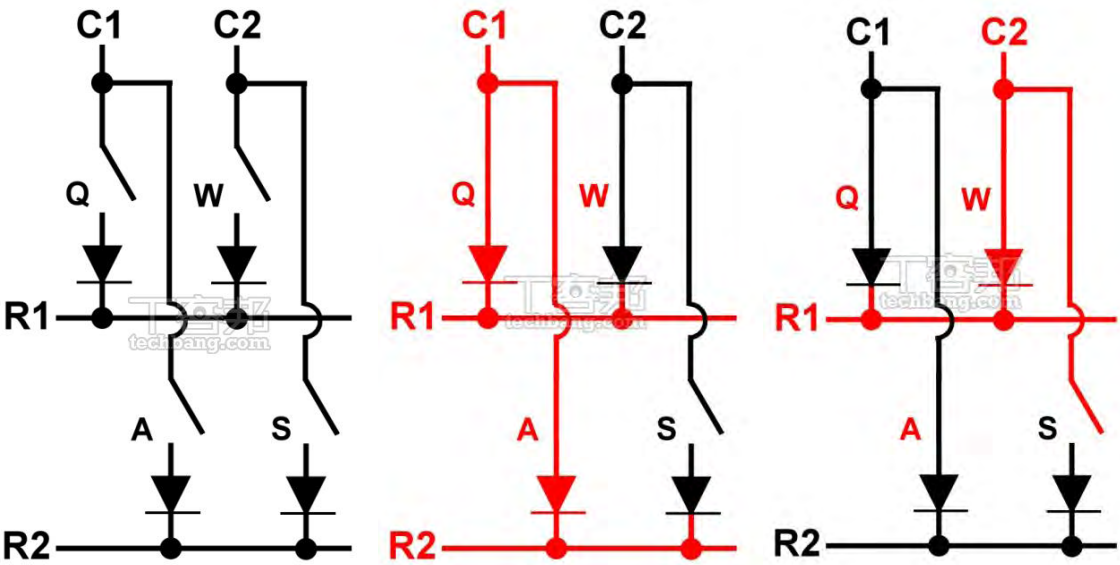


圖 17(左)、圖 18(中)、圖 19(右)

在前面，我們曾經製作過一 4x4 陣列進盤，將 16 條線壓縮至只剩 8 條線。但當按鈕數量變多時，接線量也還是會跟著急遽上升

由於此鍵盤共有 76 鍵，為 10x8 之規格；18 條線依舊是個較多的數目。那麼，有沒有甚麼方法可以再度減少接線數目？

我們可以透過暫存器 74595(資料串入並出)和 74165(資料並入串出)來完成此工作。但倘若這樣做，Keypad 將會因為不支援而無法使用，故須自行根據基本理論重新製作陣列鍵盤。

透過控制 74595 的 Latch、Data、CLK，並且利用 Arduino 的 ShiftOut 函式，便可以將三隻接腳變成八隻輸出腳位。而透過控制 74165 的 SH/LD、CLKHIB、CLK，我們就可以透過 Data 腳位接收串列資訊，重新處理後得到 8 個邏輯。也就是將 4 隻接腳轉換成 8 隻輸入腳位。

我們透過兩顆 74595 串接，接上鍵盤 10 線的掃描點，然後使用一顆 74165，接上鍵盤的結果回傳點，完成一輪之後，將所有結果列表為陣列資訊，便可以將原先需要 18 線的工作，變成只需 7 線即可完成。

而實際運用這兩種的方式 IC，請參考下方文 3 的副程式和文 4 的程式碼。

```
void r595ShiftOut(int Shift_Val) {
  int scannumber = 1 << Shift_Val; //左移Shift_Val位元 製造掃描碼
  byte Transhigh = highByte(scannumber); //擷取掃描碼的高位
  byte Translow = lowByte(scannumber); //擷取掃描碼的低位 透過傳輸此兩碼可以將資料堆疊給串接的74595
  digitalWrite(SILatchpin, LOW); //將Latch電位下拉 使74595進入資料輸入狀態
  shiftOut(SIDatapin, SIClockpin, MSBFIRST, Transhigh); //傳高位元給高位74595
  shiftOut(SIDatapin, SIClockpin, MSBFIRST, Translow); //傳低位元給低位74595
  digitalWrite(SILatchpin, HIGH); }//將74595的Latch電位拉回，使74595進入資料輸出狀態
void r165Load() {
  digitalWrite(PIClkHBpin , HIGH); //讓74165不允許資料輸出
  digitalWrite(PISH_LDpin , LOW); //讓74165之D型正反器接受資料
  delayMicroseconds(1);
  digitalWrite(PISH_LDpin , HIGH); } //重新讓74165進入準備輸出狀態
void r165Read() {
  digitalWrite(PIClkHBpin, LOW); //讓74165進入資料輸出狀態
  digitalWrite(PISH_LDpin , HIGH); //確保74165不可再度接受資料輸入
  for (int i = 1 ; i <= 8 ; i ++) { //給予8個時脈，每給予一個時脈就接收一個邏輯（正緣觸發）
    digitalWrite(PIClockpin, LOW);
    if (digitalRead(PIDatapin) == HIGH) {
      Result[i] = 1 ; }//結果列表成陣列
    else {
      Result[i] = 0 ; }
    digitalWrite(PIClockpin, HIGH);  }}
```

文 3、使用之副程式

```
void MIDICtrler_detect() {
  for (int Scan = 1 ; Scan < 11 ; Scan++) {    //給予掃描碼資訊
    r595ShiftOut(Scan - 1); //先輸出掃描電位
    r165Load(); //暫存鍵盤當下所收到的結果電位
    r165Read(); //讀取結果電位
    for (int delect = 1 ; delect <= 8 ; delect++) { //檢測8次
      if (Result[delect] == 1 && delect + (8 * (Scan - 1)) <= 76) { //如果暫存列表為1
        KEYBOARD_map[delect + (8 * (Scan - 1))] = 1 ;} //那麼鍵盤地圖的此位+[8*(掃描碼-1)]有按
      else if (Result[delect] == 0 && delect + (8 * (Scan - 1)) <= 76) {//如果暫存列表為0
        KEYBOARD_map[delect + (8 * (Scan - 1))] = 0 ;}}}}//那麼鍵盤地圖的此位+[8*(掃描碼-1)]為釋放
//以下MIDI音效卡驅動
void MIDIblock(){
  for (int MAPscan = 1 ; MAPscan <=76 ; MAPscan++){ //鍵盤圖檢測碼
    if (KEYBOARD_map[MAPscan] == 1 && KEYBOARD_oldmap[MAPscan] == 0) { //如果鍵盤碼這個位置為按下狀態
      noteOn(0,MAPscan+MIDI_tune,127); //與音效卡溝通
      KEYBOARD_oldmap[MAPscan] = KEYBOARD_map[MAPscan];}  //記錄此按鈕已經做過動作


     if (KEYBOARD_map[MAPscan] ==0 && KEYBOARD_oldmap[MAPscan] == 1) { //如果鍵盤碼這個位置為釋放狀態
      noteOff(0,MAPscan+MIDI_tune,127); //與音效卡溝通
      KEYBOARD_oldmap[MAPscan] = KEYBOARD_map[MAPscan];}}} //記錄此按鈕已經做過動作
```

文 4、主要操控程式碼

在硬體電路上又分兩部分，分別是重新實現 Roland F-30 陣列鍵盤電路以及暫存器電路。

起初拿到鋼琴時；由於鋼琴之電路並非自行製作，而是購買的材料。故需進行重新定義。 經過不斷的檢測、試驗，最終整理得出表 4 的定義表格。

第一機板			第二機板		
腳位	定義	備註	腳位	定義	備註
1	無線路	應是測試孔?	1	Row 1	
2	無線路	應是測試孔?	2	Row 2	
3	1 Col_1	擇一使用	3	Row 3	
4	1 Col_2		4	Row 4	
5	2 Col_1	擇一使用	5	Row 5	
6	2 Col_2		6	Row 6	
7	3 Col_1	擇一使用	7	Row 7	
8	3 Col_2		8	Row 8	
9	4 Col_1	擇一使用	9	5 Col_1	擇一使用
10	4 Col_2		10	5 Col_2	
11	Row 1		11	6 Col_1	擇一使用
12	Row 2		12	6 Col_2	
13	Row 3		13	7 Col_1	擇一使用
14	Row 4		14	7 Col_2	
15	Row 5		15	8 Col_1	擇一使用
16	Row 6		16	8 Col_2	
17	Row 7		17	9 Col_1	擇一使用
18	Row 8		18	9 Col_2	
備註： 不可將 Col 與 Row 對調使用，因為此陣列鍵盤含有二極體。 Row 編號相同者短接			19	10 Col_1	擇一使用
			20	10 Col_2	
			21	接地	被接入鋼琴
			22	接地	被接入鋼琴

表 4、Roland F-Sk-8p7 定義

透過陣列鍵盤基礎以及兩顆 IC 之動作方式推算，我們可以知道掃描碼從 74595 輸出，得到結果碼再從 74165 送回 Arduino。結論 74595 應該與各個 Col 短接，再將鍵盤上同編號的 Row 接上 74165，如圖 22。透過這個想法將電路分為兩塊：上半部的：74595 接鋼琴，及下半部的：鋼琴接回 74165。

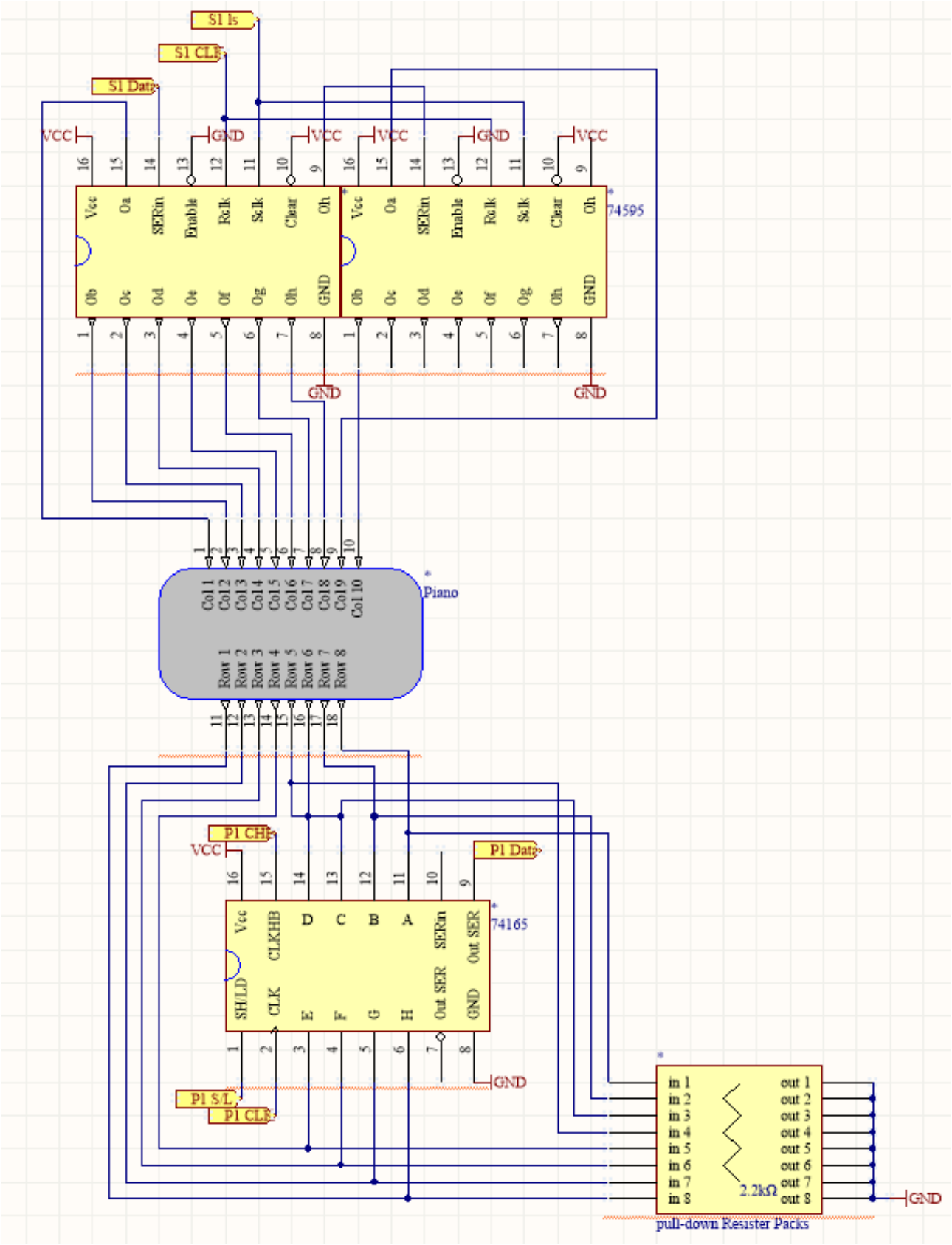

S1 ls
S1 CLK
S1 Data
VCC
GND
74595
Piano
P1 CLK
P1 Data
74165
P1 S/L
P1 CLK
2.2kΩ
pull-down Resister Packs

圖 20、鋼琴區塊接線圖

（四）連接上電腦

如果設備只能單機運作，僅此一功能，那就非常遜色了。倘若設備能夠連接上電腦，與 MIDI 的程序兼容，我們就可以使用其他設計師所製作的軟體，從而極大地擴展設備的可用性。

在最初的討論之中，我們知道 MIDI 是基於 GM1 語言來操控，而 GM1 語言的函式規格為：

TalkMIDI(cmd , Data1 , Data2) //如果 cmd And 0xF0 小等於 0xB0 則 Data2 不傳輸

cmd 為控制代碼，例如 0x90 為發聲。其中，這三筆資料全部都是 Byte 形式。

而關於資料的傳輸方式，根據 MMA 協會（MIDI Manufacturers Association）的訂定，是使用 MIDI PS/2 接口進行控制；屬於 COM Port 傳輸，鮑率為 31250bps。

巧合的是，由於 Arduino 也是使用 COM Port 與電腦溝通、交換訊息，所以實際上的實作是非常簡單的。

而製作的方法需稍稍修改程式碼。在程式中，每操控一次音效卡，便將操控音效卡的 GM1 複製一次，且將這些資料送至與電腦溝通的 COM Port，便可以完成；如文 5。

而在電腦端，我們還是需要一些軟體來協助這個工作。

由於 Arduino 連結上電腦時所使用的頻道，其規格並不是 MIDI 所制定的，因此一般的 MIDI 程序並不能識別這個頻道是 MIDI 設備。

關於這個問題，我們可以使用 nerds.de 的免費程序 LoopBe1 創建一個虛擬的 MIDI 頻道，然後使用 Angus Gratton 設計師提供的 hairless-midiserial 程式將 Arduino 的 MIDI 資訊灌入這個虛擬 MIDI 頻道來完成。

而關於其詳細的配置方式，請參閱 https://goo.gl/8bGy5T

當設定完成之後，在設備上開啟 PC 模式，將鋼琴上彈奏的資訊同時給予電腦，讓 LoopBe1 的 MIDI 頻道也接收到資料，接著我們將 MIDI 程序指定到這個頻道，其將會接收到鋼琴上所彈奏的資訊，便可以對程序進行操控。

```
//以下MIDI音效卡驅動
void MIDIblock(){
  for (int MAPscan = 1 ; MAPscan <=76 ; MAPscan++){ //鍵盤圖檢測碼
    if (KEYBOARD_map[MAPscan] == 1 && KEYBOARD_oldmap[MAPscan] == 0) { //如果鍵盤碼這個位置為按下狀態
      noteOn(0,MAPscan+MIDI_tune,127); //與音效卡溝通
      KEYBOARD_oldmap[MAPscan] = KEYBOARD_map[MAPscan]; //記錄此按鈕已經做過動作
      //以下新增於MIDIblock
      if (CPC == true){ //如果PC模式是開啟的
        CPCOutput(0x90 , MAPscan+MIDI_tune,127 );}} //對電腦也傳輸一筆鋼琴當前的GM1語言資料

     if (KEYBOARD_map[MAPscan] ==0 && KEYBOARD_oldmap[MAPscan] == 1) { //如果鍵盤碼這個位置為釋放狀態
      noteOff(0,MAPscan+MIDI_tune,127); //與音效卡溝通
      KEYBOARD_oldmap[MAPscan] = KEYBOARD_map[MAPscan]; //記錄此按鈕已經做過動作
            //以下新增
            if (CPC == true){ //如果PC模式是開啟的
                CPCOutput(0x80 , MAPscan+MIDI_tune,127 );}}}} //對電腦也傳輸一筆鋼琴當前的GM1語言資料

    void CPCOutput(byte cmd,byte data1 , byte data2){
  Serial.write(cmd);
  Serial.write(data1);
  if ( (cmd & 0xF0) <= 0xB0)
   Serial.write(data2);
    }
```

文 5：修改後程式碼

伍、 結果與討論

一、 找出可以製作優美旋律的方式

結果：最終，我們決定使用 MIDI 協定做為設備的主要核心。

討論：雖然說使用音頻產生電路與音頻放大電路也同樣可以實作這項專題，但這種方式下，變化過少，並且難以擴充新的功能。但如果以 MIDI 為基礎來進行設備製作的話，不僅擴充新功能容易，並且可以與全世界的 MIDI 程序連結。

二、 讓微控制器可以控制音聲、音高的方式

結果：請參閱：https://goo.gl/03WjFO

透過 Arduino 微控制器與支援 MIDI GM1 語言的 SparkFun Music Instrument Shield 擴充版，並且完全學習陣列鍵盤後，我們成功實行了音效卡與鍵盤配合來演奏音樂的功能。

討論：SparkFun Music Instrument Shield 使用的是鮑率 31250bps 的 COM Port 傳輸資訊，透過給予 GM1 語言，我們便可以輕易地控制其發聲。

而後，我們又使用了 Keypad 函式庫，可以簡易的部署陣列鍵盤。最終，透過兩者的結合，完成了第一個實驗：發聲實驗。

三、 如何讓使用者操縱

結果：請參閱 https://goo.gl/ZH53O2

源碼：請參閱 https://goo.gl/oLl3ND

透過操控鍵盤，我們可以自由選擇選項，並且進入選項中的頁面，對設備進行修改。

討論：透過摸索 HD44780 解碼晶片相容的 LCD 顯示板，我們使用 Liquid Crystal 函式庫，配合流程圖的規劃，成功寫出了一隻極為龐大的介面控制程式。

四、 硬體架構的設計、製作

結果：請參閱 https://goo.gl/32Pk1p 與 https://goo.gl/7378NY

透過將電路分成四個區塊觀察、測試，並且規劃之後，我們將之組合、組裝，最終成功製作出電子琴設備。

討論：首先，設備擁有四個區塊，分別是微控制器區塊、鋼琴陣列鍵盤區塊、使用者設定區塊及電源區塊，各自有大顯神通的地方。

在規劃微控制器時，我們將其腳位列表，使其可以與第二、三、四區塊接合。而鋼琴陣列鍵盤的區塊，由於二極體的加入，我們有必要理解鬼鍵的重要性。由於鋼琴之機板並非我們自行製作，而又有必要長時間觀察，列表出其機板的各腳位定義。

理解此陣列鍵盤後，又使用了 74595 和 74165 兩種暫存器 IC 來從零編寫陣列鍵盤，原因是出自於 Keypad 函式庫並不能使用 IC 實作，以及在鍵盤大小擴大後，此函式庫並不方便。

關於使用者區塊的規劃，則可以從先前的實驗中修改得來，算是方便許多。在電源區塊時，我們可以直接將電源供應的接點接上板子，來提供設備運作的電力。但在替此電源充電時，卻因為充電口直接和電源供應接通，造成了假負載效應。針對這個問題，可以加上一顆二極體，使電路變成單向來解決。

陸、 結論與應用

這次科展的專題製作下，最美中不足的地方就是：硬體的核心，即音效卡事實上並非一般學生能夠自行設計、製造的。倘若日後能夠自行設計 MIDI 相容的音效卡，那麼對於電子琴的製作將能夠錦上添花。

透過這次的製作能夠學習到：如何控制 MIDI 設備，如何實現大型陣列鍵盤，以及如何寫出簡易的人機介面。

成品照片您可以參閱圖 21、22。

在實際應用上，本設備由於使用 3.5mm 喇叭接口，並且自備電池，因此，在外演出實際上不成問題。您甚至可以透過 5.1mm 轉接口來匹配一般廣場所使用的廣播器，並且接上之後，直接使用此類喇叭進行演奏。

您也可以配接調音器，將發出的聲音進行一定的變化，如迴音效果、延遲、加重低音區塊...等，替您的演奏添加變化。

透過隨附的控制板，可以隨時在演奏時更改設備的設定值，如調整音量、更改演奏樂器、調整基本音調。

透過機板上的 USB 線，您除了可以對設備進行 ISP（In System Programing）、替電池充電等，透過已經準備好的軟體，您甚至可以直接將設備接駁至電腦，直接當成 MIDI Controller 來使用。透過這個方法，您可以使用網路上絕大部分的 MIDI 軟體，例如著名的 Synthesia。

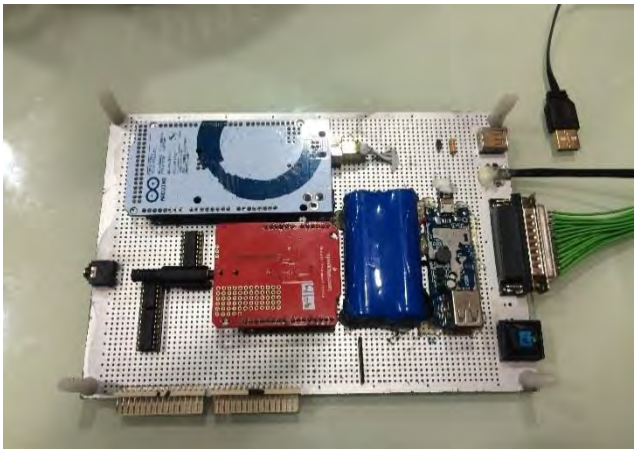


圖 21、主機板

圖 22、控制板

柒、 引用

使用之函式庫來源、參考過的程式碼：

SparkFun Music Instrument Shield 範例：https://www.sparkfun.com/tutorials/302

74595 IC 的使用方法與串接參考於：https://goo.gl/QHR0o

74165 IC 的時序圖參考（第四頁）：https://goo.gl/itqumh

函式庫 Keypad：http://playground.arduino.cc/Code/Keypad

函式庫 LiquidCrystal：https://www.arduino.cc/en/Reference/LiquidCrystal

GM1 語言參考於 https://www.midi.org/specifications/item/gm-level-1-sound-set

連接 PC 的需求程式

Com 轉 MIDI 的 Hariless-Serial 程式：http://projectgus.github.io/hairless-midiserial/

虛擬 MIDI 裝置 Loopbe1：http://www.nerds.de/en/loopbe1.html

使用之製作軟體

流程圖：https://www.draw.io/

電路圖：基於 altium designer 15

程式語言：Arduino IDE

資料參考

MIDI：https://zh.wikipedia.org/wiki/MIDI

陣列鍵盤基礎：http://swf.com.tw/?p=917

鬼鍵基礎：https://goo.gl/QW8nb

製作紀錄

Google 雲端：https://goo.gl/ZwViUr

【評語】052512

1. 本作品以製作 MIDI 電子琴為主，主題明確，作品完整度高。
2. 除了實作 MIDI 電子琴外，建議應設定明確的資訊科學探究主題，設定研究方法，並進行實驗分析，方能凸顯作品的科學方法與適切性。

作品海報

摘要

本科展專題將嘗試製作出電子琴，並且對電子琴製作的歷程進行記錄。電子琴將擁有一個控制介面、一個鋼琴鍵盤、一個自製鍵盤、以及一張以 MIDI 為基礎的音效卡。以 Arduino Mega 2560 製作，效能強大。

使用 3.5mm 接口將音源信號接至喇叭之後，您只要照著習慣彈奏鋼琴，便可以彈奏出數十種樂器。

設備裝置一具有 LCD 顯示器的機板和按鈕，使用者可以透過簡易人機介面，隨時對設備進行調整。該設備可以連接上電腦，進行協同運作，給予使用者更為進階的功能。而相反地，設備應該也要能夠單機運作，並不需要完全依賴外部進行電力供應或是資料輸入。

關鍵字詞：音樂、MIDI、Arduino、LCD Display、陣列鍵盤。

研究動機

在電子產品大行其道的今天，有許多的音樂設備也跟著電子化。而從中也延伸出極多的理論、產品，製造了龐大的市場。

直至今日，電子樂器實際上已經圍繞在我們的身旁，不管是電視節目、場地活動的配音，還是我們所熟悉的音樂舞曲...。因此，學習、研究，甚至實作一台電子樂器成為了迫切的需求。

故透過打造一台電子琴，可以藉此訓練對於電子類實作的技能，並且能夠藉機學習鋼琴之類的樂理。

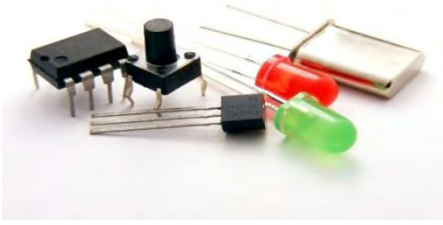
研究材料

Arduino Mega 2560

Music Instrument Shield

LCD Display

Matrix Keyboard (Piano series)



Electronic Parts

研究過程與方法

樂器數位介面（MIDI）

樂器數位介面（Musical Instrument Digital Interface，簡稱 MIDI）是一個工業標準的電子通訊協定，為電子樂器等演奏裝置（如合成器）定義各種音符或彈奏碼，容許電子樂器、電腦、手機或其它的舞台演出配備彼此連接，調整和同步，得以即時交換演奏資料。

MIDI 使得電腦、合成器、音效卡以及電子樂器(電子鼓、電子琴等)能互相控制、交換訊息。現在電腦的音效卡都是與 MIDI 相容的，並能逼真地模擬樂器的聲音。

引用於　Wikipedia

MIDI 在大概三十年前被提出，提出後隨即火熱於世界。受限於當時網路頻寬的不足，一個數 MB 的音樂檔案並沒有辦法在網路上進行串流。但是一個只有幾十 KB 不到的 MIDI 樂譜檔，由於容量極小，因此傳輸上方便，還可以做出非常多延伸。

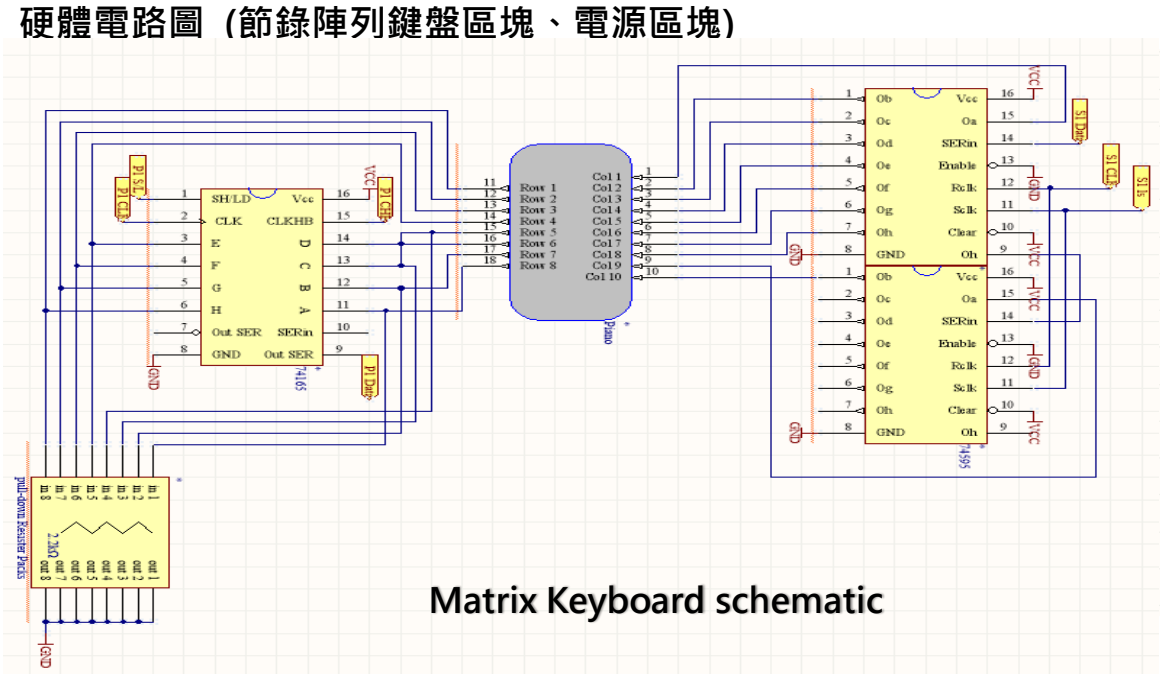
只要一張支援 MIDI 的音效卡、以及支援.mid 格式的播放器，便可以直接播放樂譜裡面收錄的資料。

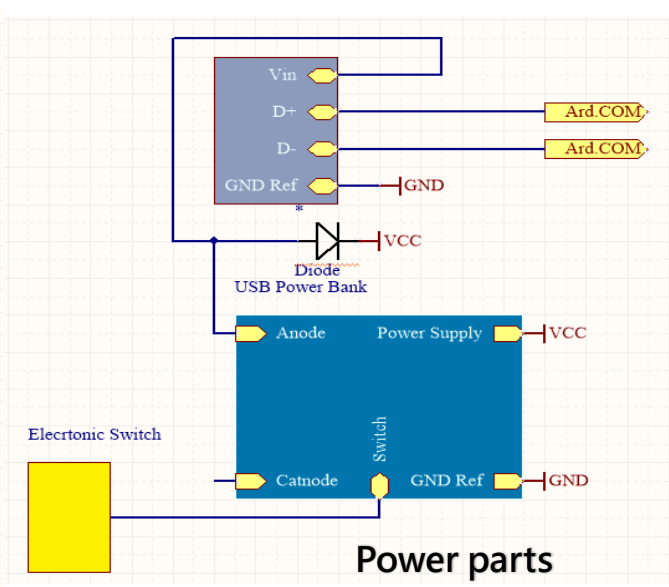
軟、硬體架構

硬體架構分為四大部分，分別為”微控制器”區塊 、 ” 鋼琴陣列鍵盤”區塊 、 ” 使用者控制”區塊 及 ” 電源”區塊。

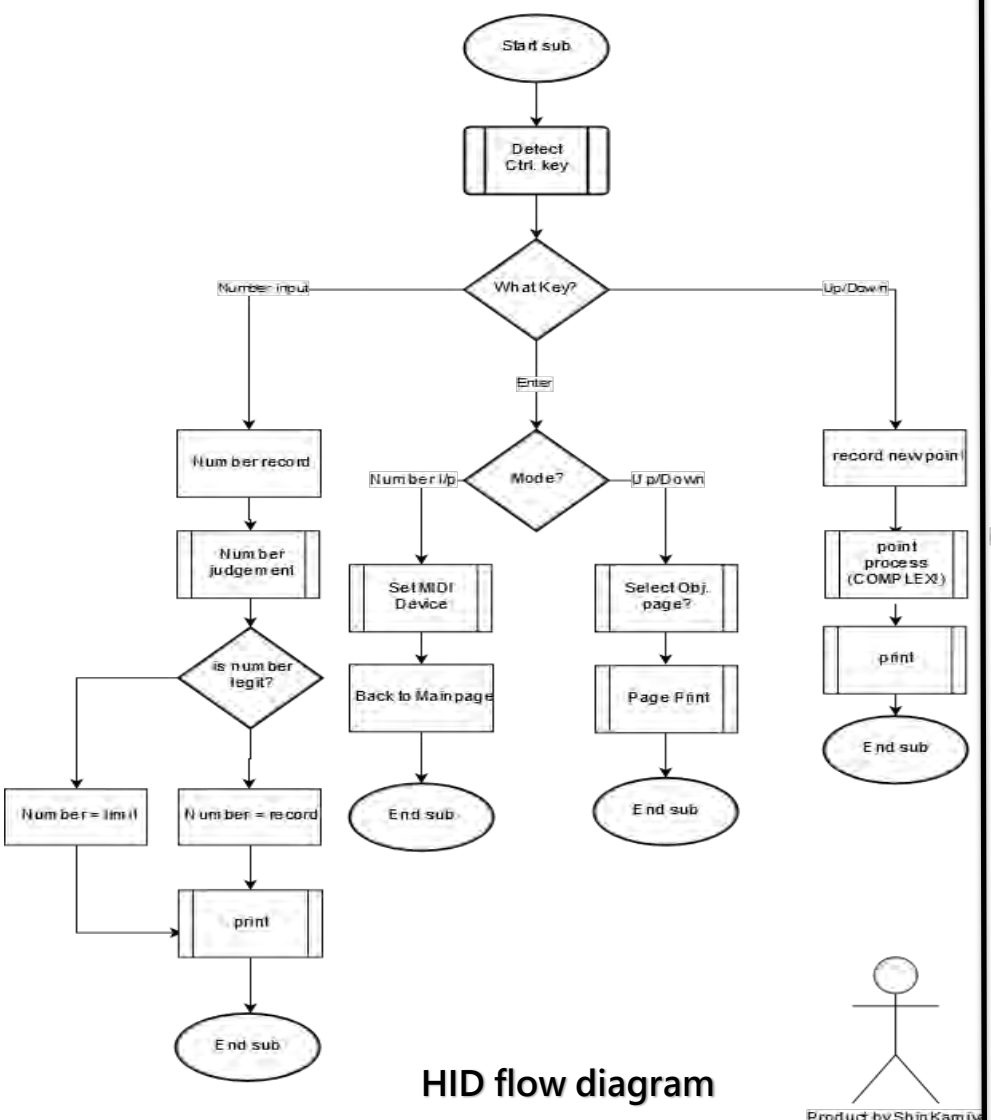
軟體則分為”控制介面”區塊 、 “暫存器讀寫”副程式 、 “陣列鍵盤掃描”區塊 、 “音效卡驅動” 。

硬體電路圖 (節錄陣列鍵盤區塊、電源區塊)

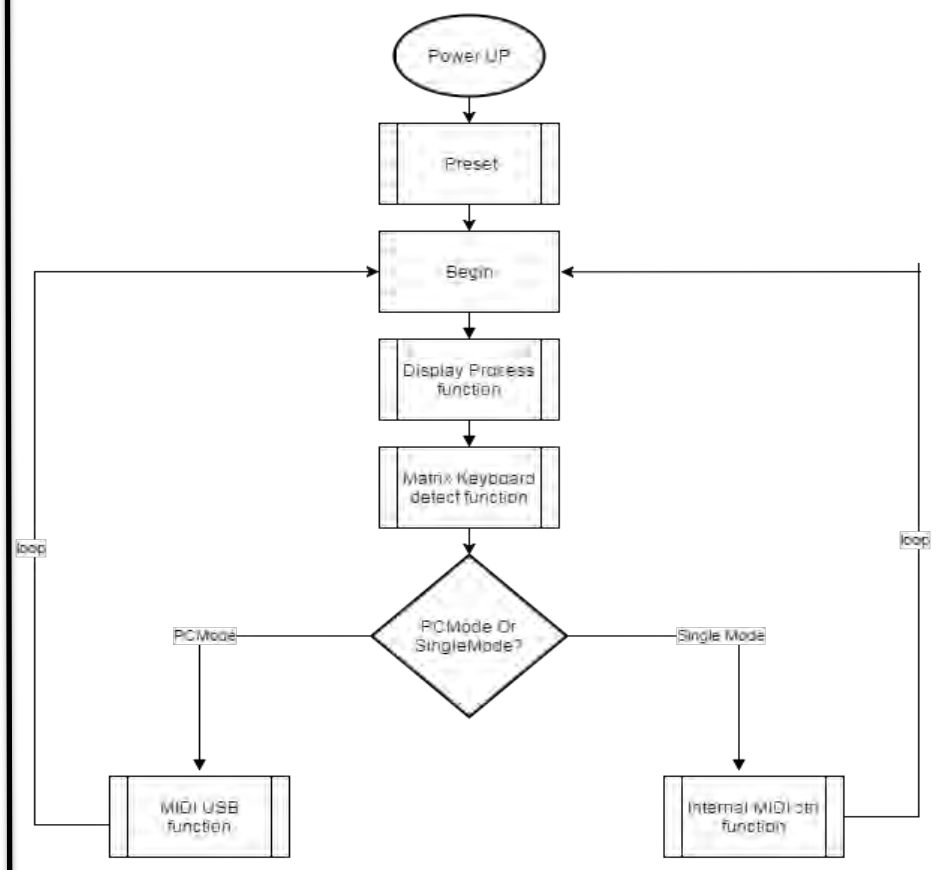




流程圖(控制介面)



HID flow diagram



Program flow diagram

Geranal MIDI Level 1 (節錄音色選取表)

MIDI 可以透過指令碼 $C(.)_{(16)}$來啟動音色選取，此時我們可以鍵入 0 到 $FF_{(16)}$中的樂器音色編號，來改變電子琴的本質音色。

在本電子琴，您可以透過控制面板，依序進入 Setting -> Com1 Settings -> Instrument Set 後，再鍵入十進制數值來選取音色，其樂器編號列表如下：

鋼琴類	打擊類	風琴	吉他類	貝斯類	弦樂類
0：大鋼琴	8：鋼片琴	16：擊桿風琴	24：弦吉他（尼龍線）	32：木貝斯	40：小提琴
1：高音大鋼琴	9：鍾琴	17：打擊風琴	25：弦吉他（鋼線）	33：指撥電貝斯	41：中提琴
2：電氣型鋼琴	10：音樂盒	18：搖滾風琴	26：電吉他（爵士）	34：彈片電貝斯	42：大提琴
3：酒吧鋼琴	11：顫音琴	19：管風琴	27：電吉他（清音）	35：無琴格貝斯	43：低音提琴
4：羅茲電氣鋼琴	12：馬林布拉木琴	20：簧風琴	28：電吉他（弱音）	36：擊弦貝斯	44：輪指演奏小提琴
5：合唱鋼琴	13：木琴	21：手風琴	29：電吉他（驅動失真）	37：擊弦貝斯	45：撥弦演奏小提琴
6：撥弦鋼琴	14：管鐘	22：口風琴	30：電吉他（經典失真）	38：合成貝斯	46：豎琴
7：擊弦鋼琴	15：揚琴	23：探戈手風琴	31：吉他諧波（泛音）	39：合成貝斯	47：定音鼓
合音(奏)類	銅管樂類	簧管樂類	笛類	合成類(主音)	合成類(柔音)
48：弦樂合奏	56：小號	64：薩克斯（高音）	72：短笛	80：方波	88：合成柔音（新時代）
49：弦樂合奏	57：長號	65：薩克斯（中音）	73：長笛	81：鋸齒波	89：合成柔音（暖音）
50：合成弦樂	58：大號	66：薩克斯（次中音）	74：豎笛	82：合成蒸氣風琴	90：合成柔音（複合）
51：合成弦樂	59：弱音小號	67：薩克斯（上低音）	75：排笛	83：合成吹管	91：合成柔音（合唱）
52：合唱聲（啊）	60：圓號	68：雙簧管	76：吹瓶口	84：合成吉他	92：合成柔音（弓弦）
53：人聲（嘟）	61：銅管組	69：英國管	77：日本尺八	85：合成人聲	93：合成柔音（金屬）
54：合成人聲	62：合成銅管	70：大管	78：哨子	86：合成五度	94：合成柔音（光環）
55：樂隊打擊樂	63：合成銅管	71：單黃管	79：陶笛	87：合成低音、主音	95：合成柔音（掃弦）
合成(效果類)	民族樂器類	打擊類(二)	特殊		
96：合成特效（雨）	104 西塔琴：	112：鈴鐺	120：磨擦吉他弦		
97：合成特效（音軌）	105：五弦琴	113：拉丁打鈴	121：呼吸聲		
98：合成特效（水晶）	106：三味線	114：鋼鼓	122：海浪聲		
99：合成特效（氣息）	107：古箏	115：木塊敲擊聲	123：鳥鳴		
100：合成特效（高亢）	108：卡林巴	116：太鼓	124：電話鈴聲		
101:合成特效(哥布林）	109：風笛	117：嗵鼓	125：直升機聲		
102：合成特效（回音）	110：維爾奧琴	118：合成鼓	126：鼓掌		
103：合成特效（科幻）	111：嗩吶	119：鈸（波型反轉）	127：槍擊聲		

單元測試結果短片（QR Code 呈現）

初期驅動音效卡測試結果

控制介面實驗結果

控制介面實驗源碼

實際操縱硬體

與電腦連結傳遞

討論與應用

硬體架構的設計、製作討論

首先，設備擁有四個區塊，分別是微控制器區塊、鋼琴陣列鍵盤區塊、使用者設定區塊及電源區塊，各自有大顯神通的地方。在規劃微控制器時，我們將其腳位列表，使其可以與第二、三、四區塊接合。

鋼琴陣列鍵盤的區塊，由於二極體的加入，我們有必要理解鬼鍵這件事。又鋼琴之機板並非我們自行製作，而需要長時間觀察，列表出其機板的各腳位定義。理解此陣列鍵盤後，又使用了 74595 和 74165 兩種暫存器 IC 來從零編寫陣列鍵盤，原因是出自於 Keypad 函式庫並不能使用 IC 實作，以及在鍵盤大小擴大後，此函式庫並不方便。

關於使用者區塊的規劃，則從實驗中漸進式地改進、修訂得來。

在電源區塊時，我們可以直接將電源供應的接點接上板子，來提供設備運作的電力。但在替此電源充電時，卻因為充電口直接和電源供應接通，造成了假負載效應。針對這個問題，可以加上一顆二極體，使電路變成單向來解決。

實際應用

本設備由於使用 3.5mm 喇叭接口，並且自備電池，因此，在外演出實際上不成問題。您甚至可以透過 5.1mm 轉接口來匹配一般廣場所使用的廣播器，並且接上之後，直接使用此類喇叭進行演奏。

您也可以配接調音器，將發出的聲音進行一定的變化，如迴音效果、延遲、加重低音區塊...等，替您的演奏添加變化。

透過隨附的控制板，可以隨時在演奏時更改設備的設定值，如調整音量、更改演奏樂器、調整基本音調。

透過機板上的 USB 線，您除了可以對設備進行 ISP(In System Programing)和替電池充電等，透過已經準備好的軟體，您甚至可以直接將設備接駁至電腦，直接當成 MIDI Controller 來使用。透過這個方法，您可以使用網路上絕大部分的 MIDI 軟體，例如著名的 Synthesia。

引用

使用之有版權的組件：

Arduino：https://www.arduino.cc/

SparkFun Music Instrument Shield：https://www.sparkfun.com/products/10587

Roland 公司之零件 F-Sk-8p7：https://www.roland.com/global/products/f-30/

德州儀器製造之 74HC595 與 74HC165 IC

使用之函式庫來源、參考過的程式碼：

SparkFun Music Instrument Shield 範例：https://www.sparkfun.com/tutorials/302

74595 IC 的使用方法與串接參考於：https://goo.gl/QHR0o

74165 IC 的時序圖參考（第四頁）: https://goo.gl/itqumh

函式庫 Keypad：http://playground.arduino.cc/Code/Keypad

函式庫 LiquidCrystal：https://www.arduino.cc/en/Reference/LiquidCrystal

GM1 語言參考於 https://www.midi.org/specifications/item/gm-level-1-sound-set

連接 PC 的需求程式

Com 轉 MIDI 的 Hariless-Serial 程式：http://projectgus.github.io/hairless-midiserial/

虛擬 MIDI 裝置 Loopbe1：http://www.nerds.de/en/loopbe1.html

使用之製作軟體

流程圖：https://www.draw.io/

電路圖：基於 altium designer 15

程式語言：Arduino IDE

資料參考

MIDI：https://zh.wikipedia.org/wiki/MIDI

陣列鍵盤基礎：http://swf.com.tw/?p=917

製作紀錄

Google 雲端：https://goo.gl/ZwViUr

QR Code 形式

製作紀錄

Arduino 官網

流程圖編寫服務