

**資工聖誕莫札特**

**第三組**

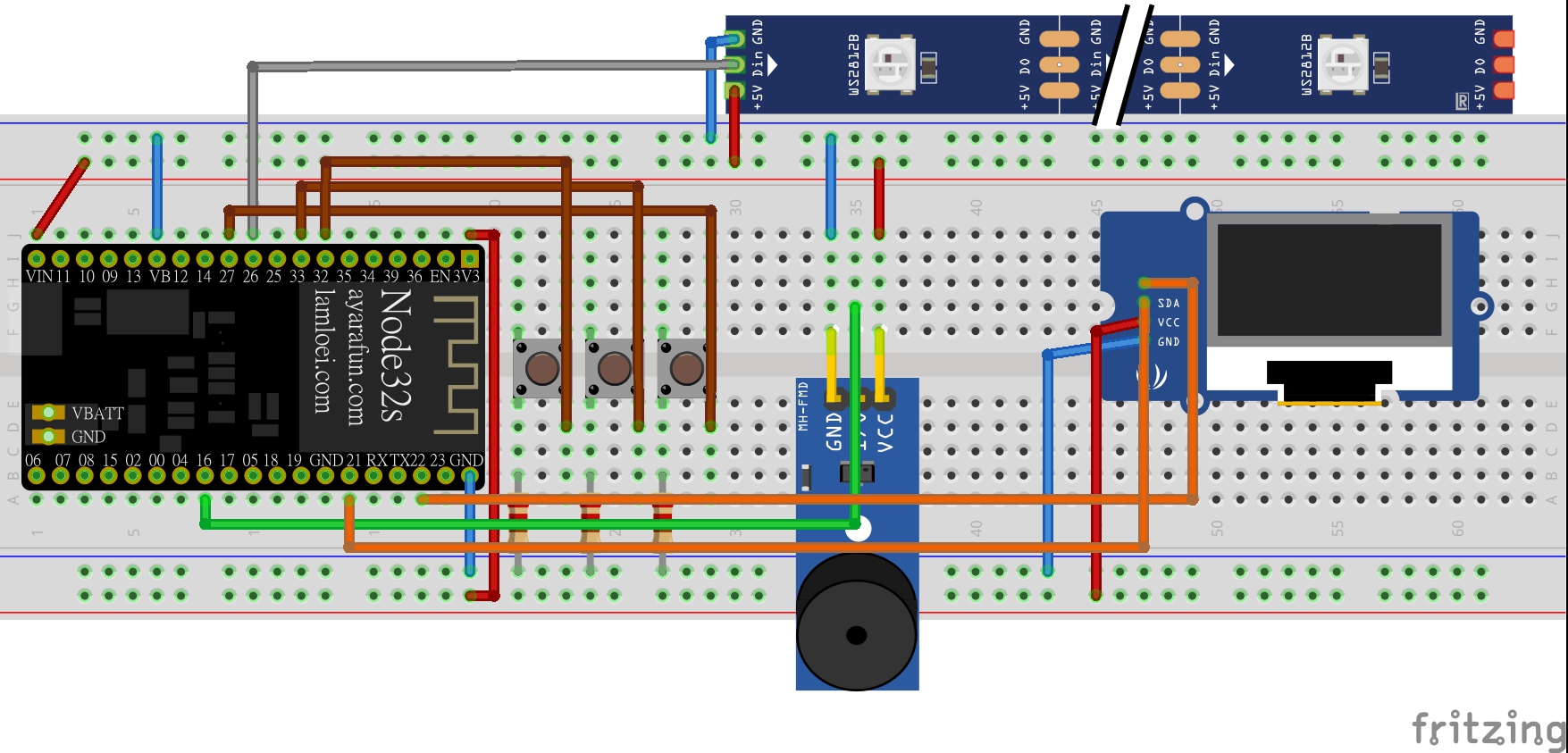
**組員: 1112952 黄柏淞 1112937 范嘉和 (11)**

**1112955 薛博徽 1112969 黃維瀚 (16)**





**電路圖**:



使用一個RGB燈條，一顆蜂鳴器，一個OLED，三顆按鈕

**三種模式:**

1. 速度: 第一顆按鈕可以控制播放聖誕歌的速度
2. 開始/暫停: 開始/暫停播放音樂
3. 切換: 切換歌曲，我們一共有三首歌Jingle Bells, Santa Claus Is Coming To Town, We Wish You a Merry Christmas

**程式碼重點說明:**

### **1. 音樂與控制**

* + 我們使用了陣列來存放音符和持續的時間，並通過 songs 和 durations 的指標陣列整合，配合 songLengths 紀錄每首歌曲的長度，方便管理與調用。所以當我們需要新增或移除歌曲，只需更新陣列就可以了。
  + 使用了speedFactor動態的調整播放速度，並設定合理的範圍，防止速度過快或過慢。
  + 透過按鈕 (BUTTON\_PIN, PAUSE\_BUTTON\_PIN, SPEED\_BUTTON\_PIN) 來切換歌曲、暫停/恢復播放以及調節速度。
  + 使用 isPaused 狀態來避免重複觸發暫停行為。

### **2. 顯示功能（OLED）**

* + 我們使用 U8g2 library提供的繪圖功能來顯示狀態與當前播放的音符。
  + OLED 更新邏輯根據歌曲和音符變化進行刷新，確保顯示內容與播放同步。
  + 我們使用小字體顯示歌曲名稱、大字體顯示如音符。
  + 根據音符的音域（例如高音）顯示額外資訊（如 "High" ）。

### **3. 燈光控制（RGB LED）**

* + 函數 changeLedsByNote 根據音符頻率範圍動態的變更 LED 顏色，設計四個範圍對應不同的顏色組合。
  + 我們使用 Adafruit NeoPixel library管理 LED
  + 函數 turnOffLeds 確保暫停或結束時 LED 全部關閉。

### **4. 結構設計**

* + 使用 delay(300) 來防止按鈕重複觸發，避免干擾。
  + 如 isPaused 用於控制播放狀態，並確保相關邏輯（如暫停顯示、LED 關閉）正確觸發。

### **6. 使用上的優化與防錯**

* + 控制速度的範圍（避免速度過慢 speedFactor < 0.5）。
  + 對於無效音符（note == 0），停止蜂鳴器輸出與 LED 顯示。
  + 按鈕切換與顯示直觀，對應的歌曲名稱與播放狀態（播放中/暫停）清楚的呈現。
  + LED 燈與音符同步，提供即時的視覺回饋。

**分工表:**

|  |  |  |
| --- | --- | --- |
| **組員** | **負責部分** | **負責比例** |
| 范嘉和 | Arduino程式 硬體製作 硬體設計 | 25% |
| 黄柏淞 | Arduino程式 硬體設計 硬體製作 | 25% |
| 薛博徽 | Arduino程式 功能設計 製作簡報 | 25% |
| 黃維瀚 | Arduino程式 硬體設計 書面報告 | 25% |

**心得:**

**范嘉和**

這次的期末專題我們做了一個聖誕音樂盒，在製作時也遇到了不少問題。在試著使用touch的方法替代按鈕時，我們發現會干擾到音樂自動播放的功能，所以最後還是採用了按鈕。最麻煩的還是外觀製作，要做出一個有聖誕氣息的外觀花了很多時間。

**黄柏淞**

這次期末專案主題自選真的稍嫌困難，所以我們沿用了類似鋼琴的音樂盒設計，在設計的過程中還遇到很多問題，像是歌曲要放什麼、接線斷掉要重買等等，希望成品不要太差。

**薛博徽**

我們這次的期末專題是要製作一個能夠播放聖誕音樂的裝置，配合OLED演出小動畫，原本會有鋼琴的元素，不過最後因為腳位數量的限制，部份的腳位不能使用，或是沒辦法正確輸出訊號，所有最後我們只能放棄了這個元件。

**黃維瀚**

這次期末專題我認爲最有難度最燒腦筋的部分就是程式。要設計播放出的音樂的每一個調是否與我們所參照的歌曲是在同一個調上，因爲他很常都會跑音等等，再來就是oled要顯示的部分也花了我們好一段時間。