



BMduino-Shield
MIDI 擴充板

BMV51T001
Arduino Library V1.0.1 說明

版本：V1.02 日期：2024-01-23

www.bestmodulescorp.com

目錄

簡介	3
Arduino Library 函式	3
Arduino Lib 下載及安裝	7
Arduino 範例	8
範例 1：PianoPerformance	8
範例 2：HitPerformance	10
範例 3：PlayMidi	11
附錄一：標準 MIDI 音色表	14

簡介

BMV51T001 是倍創推出的自帶波表合成的 MIDI 音色庫擴充板，使用 UART 通訊方式進行命令收發。本文檔對 BMV51T001 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例演示了琴鍵和打擊功能，以及 MIDI 歌曲的播放功能。

Arduino Library 函式

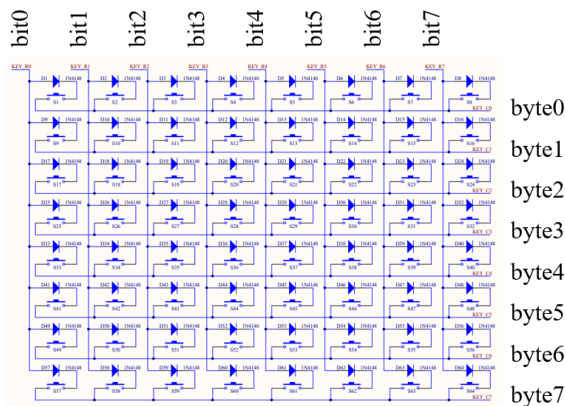
Arduino Lib 名稱：BMV51T001		Lib 版本：V1.0.1
構造函式 & 初始化		
1	BMV51T001 ()	
	描述	構造函式
	參數	—
	返回值	—
	備註	—
2	void begin(void)	
	描述	擴充板初始化
	參數	void
	返回值	void
	備註	—
功能函式		
3	bool setVolume(uint8_t volume)	
	描述	設定音量
	參數	volume：音量，範圍 0~15，0 是最小音量（靜音），預設是音量 5
	返回值	執行情況： true：成功 false：失敗
	備註	—
4	bool setHitSensitivity(uint8_t value)	
	描述	設定打擊靈敏度
	參數	value：打擊靈敏度參數，範圍 0~18 觸發電壓 = $3.3 \times ((100+50 \times \text{value})/4096)$ ，單位：V
	返回值	執行情況： true：成功 false：失敗
	備註	打擊靈敏度設定請見使用手冊 → 感測介面介紹。打擊靈敏度參數預設 0。
5	bool setHitTimeInterval(uint8_t value)	
	描述	設定打擊檢測的間隔時間
	參數	value：間隔時間參數，範圍 0~33 間隔時間 = $35 + 5 \times \text{間隔時間參數}$ ，單位：ms
	返回值	執行情況： true：成功 false：失敗
	備註	打擊靈敏度設定請見使用手冊 → 感測介面介紹。間隔時間參數預設 3。

6	void setHitStrengthLayer(uint8_t layers)	
	描述	設定打擊力度等級
	參數	layers：力度等級・範圍 1~128
	返回值	void
	備註	—
7	bool setKeyboardOut(uint8_t status)	
	描述	設定 8×8-key 鍵盤功能狀態
	參數	status：鍵盤功能狀態 1 (BMV51T001_KEYBOARD_ENABLE)：啟用 0 (BMV51T001_KEYBOARD_DISABLE)：停用
	返回值	執行情況： true：成功 false：失敗
	備註	—
8	bool setHitOut (uint8_t status)	
	描述	設定打擊功能狀態
	參數	status：打擊功能狀態 1 (BMV51T001_HIT_ENABLE)：啟用 0 (BMV51T001_HIT_DISABLE)：停用
	返回值	執行情況： true：成功 false：失敗
	備註	—
9	void scanKeyboard(void)	
	描述	琴鍵功能掃描
	參數	void
	返回值	void
	備註	在 loop 裡面調用
10	void scanHit(void)	
	描述	打擊功能掃描
	參數	void
	返回值	void
	備註	在 loop 裡面調用
11	bool isKeyboard(void)	
	描述	琴鍵是否有動作
	參數	void
	返回值	動作情況 true：有動作 false：沒動作
	備註	—

12	bool isHit(void)	
	描述	打擊是否有動作
	參數	void
	返回值	動作情況 true：有動作 false：沒動作
	備註	—
13	void readKeyboardData(uint8_t keyBuf[])	
	描述	讀取 8×8 矩陣琴鍵資料
	參數	keyBuf[]：8×8 琴鍵狀態，共 8byte，每個 bit 表示一個按鍵 bit=1：按下，bit=0：鬆開
	返回值	—
	備註	8×8 矩陣按鍵用 8 個 byte 儲存，每個 bit 表示一個按鍵 ⁽¹⁾ 。
14	uint16_t readHitADCData(void)	
	描述	讀取打擊檢測 ADC 資料
	參數	void
	返回值	16-bit 打擊的 ADC 資料
	備註	—
15	uint8_t getHitStrengthLayer(uint16_t hitADCData)	
	描述	轉化當前的 ADC 數值為打擊力度等級
	參數	HitADCData：16-bit ADC 資料
	返回值	力度等級：1~128
	備註	—
16	bool reset(void)	
	描述	重置擴充板
	參數	void
	返回值	重置情況 true：成功 false：失敗
	備註	—
17	uint8_t getFWVer(void)	
	描述	獲取版本號
	參數	void
	返回值	版本號
	備註	目前版本為 0x01 版
MIDI 通訊協定函式		
18	void setNoteOn (uint8_t noteNumber, uint8_t velocity, uint8_t channel)	
	描述	發送音符開
	參數	noteNumber：音符，範圍 0~127 velocity：力度，範圍 0~127 channel：通道，範圍 1~16 (打擊樂器音色固定在通道 10)
	返回值	void
	備註	當使用通道 10 時為打擊樂，noteNumber 範圍 24~84，打擊樂種類見附錄 —

19	void setNoteOff (uint8_t noteNumber, uint8_t velocity, uint8_t channel)	
	描述	發送音符關
	參數	noteNumber：音符，範圍 0~127 velocity：音符力度，範圍 0~127 channel：通道，範圍 1~16 (打擊樂器音色固定在通道 10)
	返回值	void
	備註	當使用通道 10 時為打擊樂，noteNumber 範圍 24~84，打擊樂種類見附錄一
20	void setTone (uint8_t toneNumber, uint8_t channel)	
	描述	設定音色
	參數	toneNumber：音色號，範圍 0~127 channel：通道，範圍 1~16
	返回值	void
	備註	音色號見附錄一中的音色表
21	void setChannelVolume(uint8_t channelVolume, uint8_t channel)	
	描述	設定通道音量
	參數	channelVolume：通道音量值，範圍 0~127 (0：最小；127：最大) channel：通道，範圍 1~16
	返回值	void
	備註	區別於板載按鍵上的音量調節，板載按鍵上的音量調節是調節整體音量，這裡僅僅是通道音量
22	void setPitchBend(int16_t pitchValue, uint8_t channel)	
	描述	設定彎音音效
	參數	pitchValue：彎音值，範圍 -8192~8191 (0：沒有彎音效果，-8192：最大向下彎曲，8191：最大向上彎曲) channel：通道，範圍 1~16
	返回值	void
	備註	—

(1) 矩陣按鍵，資料與按鍵的對應關係

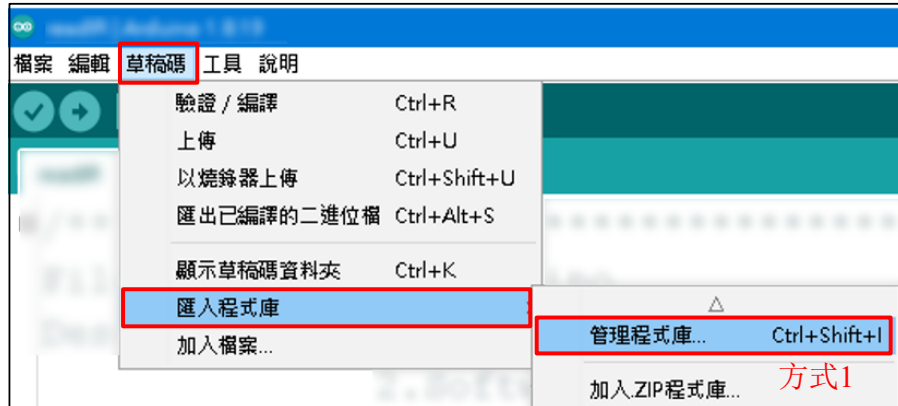


Arduino Lib 下載及安裝

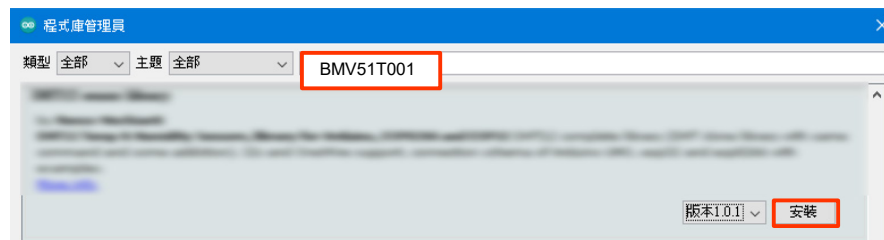
BMV51T001 Library：可參考下面兩種方法安裝 BMV51T001 的 Arduino Library

方式 1：搜索安裝

搜索安裝：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BMV51T001 → 安裝



搜索安裝流程 1

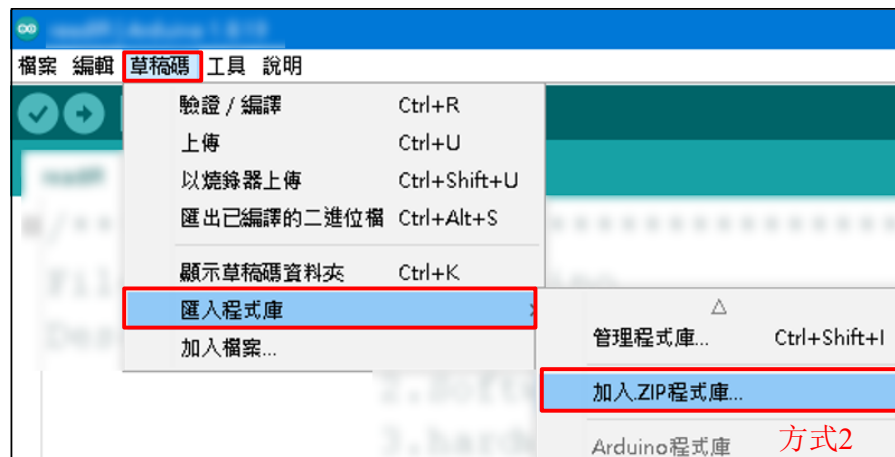


搜索安裝流程 2

方式 2：加入 .ZIP 程式庫，需提前下載 .ZIP 程式庫

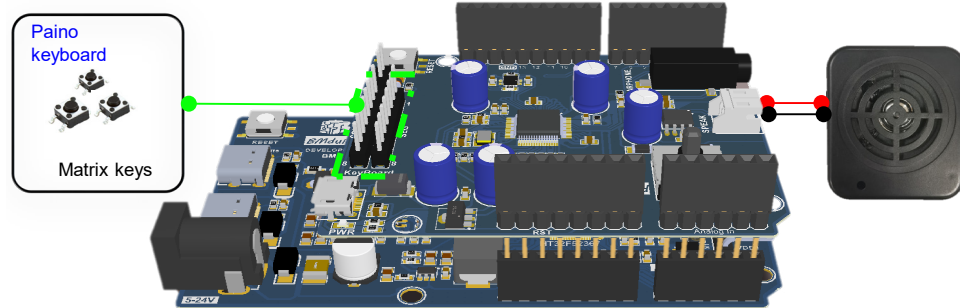
下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bmv51t001.html>) “文件” 目錄下的 Arduino 範例程式 (BMV51T001 Library)。

加入 .ZIP 程式庫：Arduino IDE → 草稿碼 → 匯入程式 → 加入 .ZIP 程式庫 ...



Arduino 範例

範例 1 : PianoPerformance



實物連接示意圖

範例實現功能：通過 MIDI Shield 的琴鍵介面外接 64keys，然後觸發按鍵發聲。

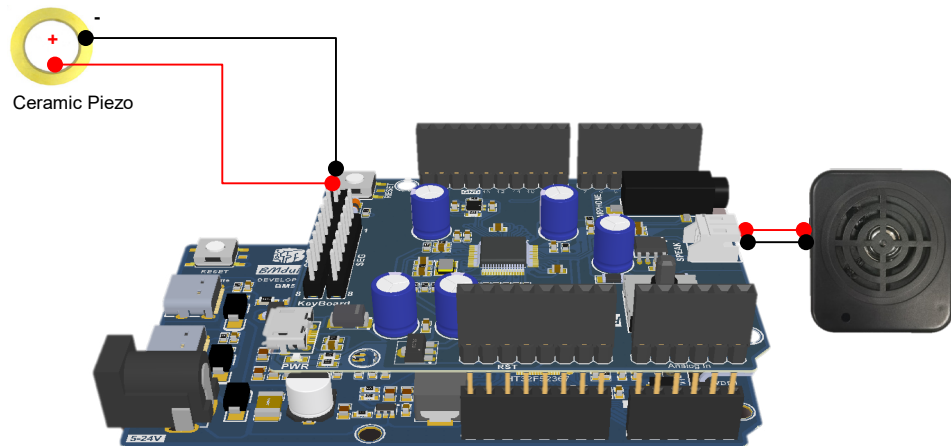
1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMV51T001) → 選擇範例 (PianoPerformance)
2. 範例說明：
 - a. 建立對象 & 聲明功能函式 & 初始化擴充板

```
#include "BMV51T001.h"
BMV51T001 shieldMIDI;           // 建立對象
uint8_t keyBuf[8];              // 存放琴鍵資料，大小必須為 8，因為每次接收是 8byte 為單位
uint8_t lastKeyBuf[8];         // 存放上一次的琴鍵資料
uint8_t temp,i,j;
uint8_t noteNumber ;           // 音高
uint8_t velocity = 0x50;       // 力度，鋼琴沒做力度檢測，預設 0x50
uint8_t channel = 0x01;        // 播放預設在 channel 1
uint8_t keyOutEnableFlag = 0;  // 設定琴鍵輸出標誌
void setup()
{
  shieldMIDI.begin();           // 初始化擴充板
  if(shieldMIDI.setKeyboardOut(BMV51T001_KEYBOARD_ENABLE)) // 啟用琴鍵
                                                                    // 功能
  {
    keyOutEnableFlag = 1;
  }
}
```

b. 在 loop 中執行 PianoPerformance 功能函式

```
void loop()
{
  if(keyOutEnableFlag)
  {
    shieldMIDI.scanKeyboard();           //UART 接收按鍵 掃描
    if(shieldMIDI.isKeyboard()!=0)      // 接收完成
    {
      shieldMIDI.readKeyboardData (keyBuf); // 讀取按鍵狀態
      for (i = 0; i < 8; i++)           //8 組 key 全部判斷一遍
      {
        temp = keyBuf[i]^lastKeyBuf[i]; // 前後兩次按鍵有變化
        if (temp!=0) // 如果有按鍵按下
        {
          for (j = 0; j < 8; j++)       // 每組 key 裡面有 8 個 key
          {
            if ((temp &(1 << j))!=0)   // 該位置有按鍵變化
            {
              noteNumber = 36 +i*8+j; // 由於它是 64 鍵·從 noteNumber
              // 36 大字一組的 do 開始
              if ((keyBuf[i]&(1 << j))!=0) // 按鍵按下
              {
                // 調用 note on 播放音符
                shieldMIDI.setNoteOn (noteNumber, velocity, channel);
              }
              else // 按鍵鬆開
              {
                // 調用 note off 停止音符
                shieldMIDI.setNoteOff (noteNumber, velocity, channel);
              }
            }
          }
          lastKeyBuf[i]= keyBuf [i];
        }
      }
    }
  }
}
```

範例 2 : HitPerformance



實物連接示意圖

範例實現功能：讀取壓電陶瓷片的 ADC 數值，轉成電子鼓發聲力度。

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMV51T001) → 選擇範例 (HitPerformance)
2. 範例說明：
 - a. 建立對象 & 聲明功能函式 & 初始化擴充板

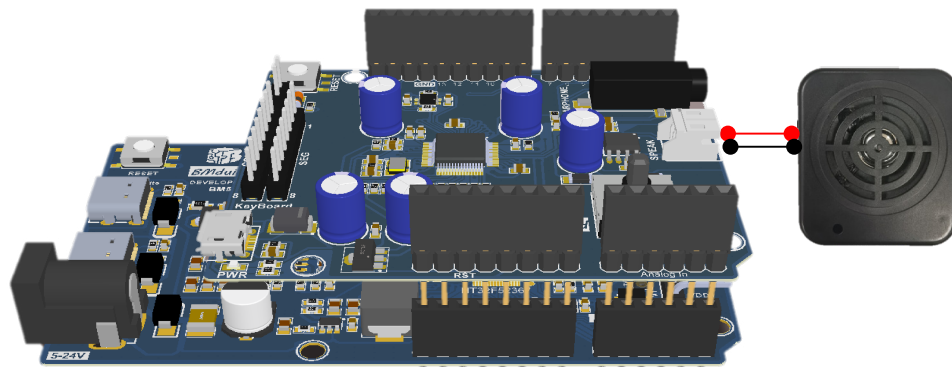
```

#include "BMV51T001.h"
BMV51T001 shieldMIDI;           // 建立對象
uint16_t hitADCData = 0;        // 存放讀取到的打擊資料
uint8_t velocity = 0;           // 力度
uint8_t noteNumber = 0x40;      // 定義為鼓聲
uint8_t channel = 10;           // 鼓聲固定在通道 10
uint8_t hitOutEnableFlag = 0;   // 設定打擊輸出標誌
#define BMV51T001_NO_HIT 0
void setup()
{
    shieldMIDI.begin();          // 初始化擴充板
    shieldMIDI.setHitStrengthLayer(128); // 設定力度等級，將總的力度分為
                                        // 128 級
    if(shieldMIDI.setHitOut(BMV51T001_HIT_ENABLE)) // 啟用打擊檢測功能
    {
        hitOutEnableFlag = 1;    // 啟用打擊功能
    }
}
    
```

b. 在 loop 中執行 HitPerformance 功能函式

```
void loop()
{
  if(hitOutEnableFlag) // 打擊功能已啟用
  {
    shieldMIDI.scanHit();// UART 接收打擊設備
    if(shieldMIDI.isHit() != BMV51T001_NO_HIT) // 接收完成
    {
      hitADCData =shieldMIDI.readHitADCData ();// 讀取打擊檢測資料
      velocity = shieldMIDI. getHitStrengthLayer(hitADCData);// 轉化為
      // 力度等級
      shieldMIDI.setNoteOn (noteNumber, velocity, channel); // 發送音
      // 符開消息：通道 10 為鼓音色專用通道：如果發送通道 10 則表示這是鼓音色
    }
  }
}
```

範例 3：PlayMidi



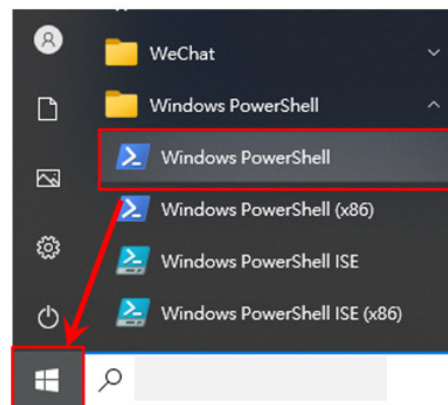
實物連接示意圖

範例實現功能：播放 .mid 歌曲。

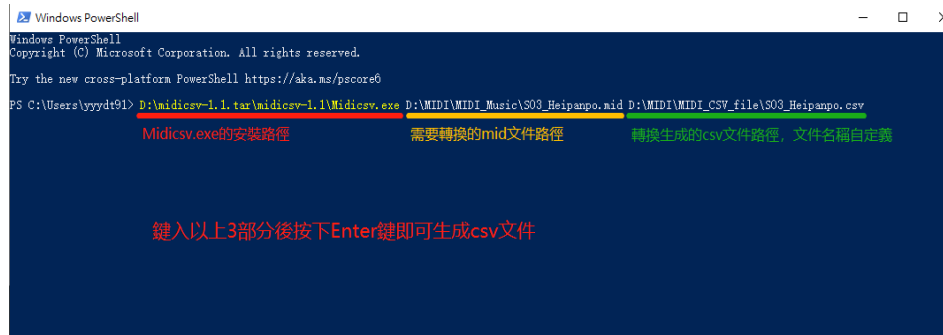
1. 提取 .mid 檔案資料，供 MCU 使用

步驟 1：請下載 Midicsv.exe 軟體。

步驟 2：打開電腦自帶的 Windows PowerShell (支援 WIN11\WIN10\WIN7)，將 MIDI 檔案轉化成 CSV 表格檔案。



如下圖 所示，在 PowerShell 中輸入三個路徑，路徑之間用空格分開① Midicsv.exe 路徑② .mid 檔案路徑③ .csv 檔案儲存路徑。輸入完後按下 enter 鍵即可完成轉換。



注意：上圖中的路徑需要根據用戶實際路徑填入

步驟 3：打開轉化後的 .csv 檔案，建立 .h 檔案。將 .csv 檔案的內容按下圖建立資料，填入 .h 檔案中，左框與右框中的內容按數字編號一一對應

.CSV				
	Track Number	Time	Event	Event Data
CSV File Content	1	0	Header	1 4 120 DIVISION
	2	1	Start_track	2
	3	1	Title_t	"303_Heipanpo"
	4	1	Text_t	"ggg'012"
	5	1	Time_signature	4 2 24 8
	6	1	Key_signature	0 "major"
	7	1	Tempo	666667 TEMPO 1
	8	1	3840 Tempo	666667
	9	1	3840 End_track	
	10	2	0 Start_track	
	11	2	0 Title_t	"303_Heipanpo"
	12	2	0 Control_c	0 7 127
	13	2	7 Note_on_c	0 69 100
	14	2	67 Note_on_c	0 69 0
	15	2	67 Note_on_c	0 69 100
	16	2	127 Note_on_c	0 69 0
	17	2	128 Note_on_c	0 69 100
	18	2	188 Note_on_c	0 69 0

.h	
#include "playMIDI.h"	8
#define TEMPO 666667	1
#define DIVISION 120	2
#define NOTENUM 321	3
midi_struct music 3211	
{	
{ 7, 69, 100, 0, },	
{ 4, 5, 6, 7, },	
{ 67, 69, 0, 0, },	
{ 67, 69, 100, 0, },	
{ 67, 36, 0, 5, },	
}	

上圖中左邊 .csv 製作成右邊 .h 檔案，其中各項內容為：

序號	名稱	描述
1	TEMPO	一拍的時間 (μs 為單位)
2	DIVISION	一拍的 tick 數 (tick 是 midi 中計算時間長短的最小單位)
3	NOTENUM	音符數量，這個需要使用者自己根據 csv 裡面 note_on_c 的行數填入
4	time	當前的音樂進度，以 tick 為單位的時間
5	ctrlnum	音符編號 (音符音高)
6	value	音符力度 (彈奏力度)
7	channel	通道 (音軌)
8	include	使用時必須 include "playMIDI.h" 頭檔

步驟 4：.h 檔案製作完畢後放到範例程式 .ino 檔案的相同路徑下，如下圖



步驟 5：在 .ino 中添加上述製作的 .h 檔案，MCU 即可使用 midi 歌曲資訊

2. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMV51T001) → 選擇範例 (PlayMidi)

3. 範例說明：

a. 建立對象 & 初始化擴充板

```
#include "BMV51T001.h"
#include "playMIDI.h"
#include "heiPanPo.h"
/* 工作原理：按照順序播放音符清單，組成一首 MIDI 曲子，每個音符都有固定的時間，音色，音高，力度，通道在列表裡 "heiPanPo.h" */
playMIDI myMIDIPlayer; // 建立對象
void setup() {
  myMIDIPlayer.begin(); // 初始化擴充板
  myMIDIPlayer.setChannelVolume(127, 0 + 1); // 設定通道 1 音量，127 最大聲。
  myMIDIPlayer.setTone(35, 1 + 1); // 設定通道 2 播放的音色，35 是音色號
  myMIDIPlayer.setChannelVolume(127, 1 + 1); // 設定通道 2 音量，127 最大聲。
  myMIDIPlayer.setChannelVolume(127, 9 + 1); // 設定通道 10 音量，127 最大
  // 聲。開始播放 .h 檔案中陣列名為 music 的 midi 歌曲
  myMIDIPlayer.beginPlayMIDI(music, TEMPO, DIVISION, NOTENUM);
}
```

b. 在 loop 中播放 Midi 曲

```
void loop() {
  myMIDIPlayer.loopPlayMIDI(); // 掃描執行 MIDI 歌曲播放功能
  if(myMIDIPlayer.isPlaying() != MIDI_PLAYER_NO_BUSY) // 判斷是否播放中
  {
    myMIDIPlayer.beginPlayMIDI(music, TEMPO, DIVISION, NOTENUM); // 重複
    // 播放
  }
}
```

附錄一：標準 MIDI 音色表

• 音色表

編號	名稱	編號	名稱
0	Acoustic Grand Piano	64	Soprano Sax
1	Bright Acoustic Piano	65	Alto Sax
2	Electric Grand Piano	66	Tenor Sax
3	Honky-tonk Piano	67	Baritone Sax
4	Electric Piano 1	68	Oboe
5	Electric Piano 2	69	English Horn
6	Harpichord	70	Bassoon
7	Clavichord	71	Clarinet
8	Celesta	72	Piccolo
9	Glockenspiel	73	Flute
10	Music box	74	Recorder
11	Vibraphone	75	Pan Flute
12	Marimba	76	Blown Bottle
13	Xylophone	77	Shakuhachi
14	Tubular Bell	78	Whistle
15	Dulcimer	79	Ocarina
16	Hammond Organ	80	Lead 1(square)
17	Percussive Organ	81	Lead 2(sawtooth)
18	Rock Organ	82	Lead 3(calliope)
19	Church organ	83	Lead 4(chiff)
20	Reed organ	84	Lead 5(charang)
21	Accordion	85	Lead 6(voice)
22	Harmonica	86	Lead 7(fifths)
23	Tango Accordion	87	Lead 8(bass + lead)
24	Acoustic Guitar(nylon)	88	Pad 1(new age)
25	Acoustic Guitar(steel)	89	Pad 2(warm)
26	Electric Guitar(jazz)	90	Pad 3(polysynth)
27	Electric Guitar(clean)	91	Pad 4(choir)
28	Electric Guitar(muted)	92	Pad 5(bowed)
29	Overdriven Guitar	93	Pad 6(metallic)
30	Distortion Guitar	94	Pad 7(halo)
31	Guitar harmonics	95	Pad 8(sweep)
32	Acoustic Bass	96	FX 1(rain)
33	Electric Bass(finger)	97	FX 2(soundtrack)
34	Electric Bass(pick)	98	FX 3(crystal)
35	Fretless Bass	99	FX 4(atmosphere)
36	Slap Bass 1	100	FX 5(brightness)
37	Slap Bass 2	101	FX 6(goblins)

編號	名稱	編號	名稱
38	Synth Bass 1	102	FX 7(echoes)
39	Synth Bass 2	103	FX 8(sci-fi)
40	Violin	104	Sitar
41	Viola	105	Banjo
42	Cello	106	Shamisen
43	Contrabass	107	Koto
44	Tremolo Strings	108	Kalimba
45	Pizzicato Strings	109	Bagpipe
46	Orchestral Harp	110	Fiddle
47	Timpani	111	Shanai
48	String Ensemble 1	112	Tinkle Bell
49	String Ensemble 2	113	Agogo
50	Synth Strings 1	114	Steel Drums
51	Synth Strings 2	115	Woodblock
52	Voice Aahs	116	Taiko Drum
53	Voice Oohs	117	Melodic Tom
54	Synth Voice	118	Synth Drum
55	Orchestra Hit	119	Reverse Cymbal
56	Trumpet	120	Guitar Fret Noise
57	Trombone	121	Breath Noise
58	Tuba	122	Seashore
59	Muted Trumpet	123	Bird Tweet
60	French horn	124	Telephone Ring
61	Brass Section	125	Helicopter
62	Synth Brass 1	126	Applause
63	Synth Brass 2	127	Gunshot

● 打擊樂

編號	名稱	編號	名稱
24	Seq Click H	55	Splash Cymbal
25	Brush Tap	56	Cowbell
26	Brush Swirl	57	Crash Cymbal 2
27	Brush Slap	58	Vibraslap
28	Brush Tap Swirl	59	Ride Cymbal 2
29	Snare Roll	60	Hi Bongo
30	Castanet	61	Low Bongo
31	Snare H Soft	62	Mute Hi Conga
32	Sticks	63	Open Hi Conga
33	Bass Drum Soft	64	Low Conga
34	Open Rim Shot	65	High Timbale
35	Acoustic Bass Drum	66	Low Timbale
36	Bass Drum 1	67	High Agogo
37	Side Stick	68	Low Agogo
38	Acoustic Snare	69	Cabasa
39	Hand Clap	70	Maracas
40	Electric Snare	71	Short Whistle
41	Low Floor Tom	72	Long Whistle
42	Closed Hi-Hat	73	Short Guiro
43	High Floor Tom	74	Long Guiro
44	Pedal Hi-Hat	75	Claves
45	Low Tom	76	Hi Wood Block
46	Open Hi-Hat	77	Low Wood Block
47	Low-Mid Tom	78	Mute Cuica
48	Hi-Mid Tom	79	Open Cuica
49	Crash Cymbal 1	80	Mute Triangle
50	High Tom	81	Open Triangle
51	Ride Cymbal 1	82	Shaker
52	Chinese Cymbal	83	Jingle Bell
53	Ride Bell	84	Bell Tree
54	Tambourine		

Copyright© 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版時倍創已針對所載資訊為合理注意，但不保證資訊準確無誤。文中提到的資訊僅是提供作為參考，且可能被更新取代。倍創不擔保任何明示、默示或法定的，包括但不限於適合商品化、令人滿意的品質、規格、特性、功能與特定用途、不侵害第三人權利等保證責任。倍創就文中提到的資訊及該資訊之應用，不承擔任何法律責任。此外，倍創並不推薦將倍創的產品使用在會因故障或其他原因而可能會對人身安全造成危害的地方。倍創特此聲明，不授權將產品使用於救生、維生或安全關鍵零組件。在救生 / 維生或安全應用中使用倍創產品的風險完全由買方承擔，如因該等使用導致倍創遭受損害、索賠、訴訟或產生費用，買方同意出面進行辯護、賠償並使倍創免受損害。倍創 (及其授權方，如適用) 擁有本文件所提供資訊 (包括但不限於內容、資料、範例、材料、圖形、商標) 的智慧財產權，且該資訊受著作權法和其他智慧財產權法的保護。倍創在此並未明示或暗示授予任何智慧財產權。倍創擁有不事先通知而修改本文件所載資訊的權利。如欲取得最新的資訊，請與我們聯繫。