

VT03 Program Sound Generator Users Manual

FMDemo.rar 是 6502 原文件(Source Code)，midi2vt.rar 是工具軟件，請自行使用 Cakewalk 來製作 midi 文件。

操作簡介：

- 1、用 Cakewalk 製作 midi 文件，注意此 midi 文件最多為 5 軌，分別命名對應 "Square1","Square2","Triangle","Noise","DPCM"。具體可參考附件 midi2vt.rar 中的 midi02.mid。(只能用 Midi Type2)欲轉換之 midi 文件必須要有 5 個音軌，轉換程序會按照由高到低依序對應到 VT03 的 Square1,Square2,Triangle,Noise and DPCM,如果該聲部不用則空一個音軌，轉換 midi 文件是按照 NTSC 的速率匹配，如果是 PAL 制式下需手動調整音樂速度。
- 2、用 midi2vt.exe 轉換製作好的 midi 文件，可生成 bin 文件供即時播放，同時生成 asm 和 h 文件可供編譯使用。
- 3、關於 asm 文檔的添加請參閱 FMDemo.rar 中的 2MusData.asm,壓縮包中 3EffData.asm 同時包括 50 種常用音效可用手柄選擇播放,其代碼數據可供用戶參考使用，播放驅動程序請參閱 5Music.asm，相關說明如下：

程序變量說明

>>> Music_Flag1：音樂播放程序同用戶程序交互標誌 1，主要用於聲畫同步，在每個音樂播放時設置為零。

>>> Music_Flag2：音樂播放程序同用戶程序交互標誌 2，必須嵌入用戶程序控制。

>>> Music_Tone：音調基準值，在整個音樂程序初始化時設置為零

>>> Music_MuteFlag：靜音標誌，由 BIOS 設置

bit 0 音效靜音標誌

0 = Normal

1 = Mute

bit 1-6 Null

bit 7 音樂靜音標誌

0 = Normal

1 = Mute

>>> System_Flag：系統綜合標誌

bit 0 運行狀態

0 = Normal

1 = Demo

bit 7 電視制式

0 = NTSC

1 = PAL

>>> Music_SpeedCount：音樂播放速度遞減計數器

>>> Music_SpeedIndex：音樂速度索引

>>> Music_SpeedValue：音樂速度值，由 4 字節組成，由播放程序依次索引使用

>>> Music_SquareIndexTable：預置方波音色數據索引表

>>> Music_TriangleIndexTable：預置三角波音色數據索引表

>>> Music_NoiseDataTable：預置噪聲音色數據表

>>> Music_DPCMIndexTable：預置 DPCM 音色數據索引表

>>> Music_PlayBuffer ds 5*8：音樂播放數據表

>>> Music_PlayControl1 : 音樂播放控制寄存器 1
bit 0-6 節拍等待計數器
0 = 128
bit 7 本組控制數據處理狀態
0 = Disable
1 = Enable

>>> Music_PlayControl2 : 音樂播放控制寄存器 2
bit 0-3 當前音量設置
bit 4-5 音樂循環計數器

>>> Music_PlayAddress : 當前音樂控制數據地址

>>> Music_TimbreAddress : 當前音色數據地址

>>> **Music_ToneBuffer ds 8*10 : 音色輸出控制**

>>> Music_ToneControl1 : 音色輸出控制寄存器 1
bit 0-6 聲道等待計數器
0 = 64
bit 5 音色輸出狀態
0 = 正常
1 = 索引處理
bit 7 本聲道處理狀態
0 = Disable
1 = Enable

>>> Music_ToneControl2 : 音色輸出控制寄存器 4
bit0-3 循環計數器
bit4-7 索引輸出 Mask 值

>>> Music_ToneControl3 : 音色輸出控制寄存器 2
bit 0-5 輸出間隔值
bit 6,7 索引處理狀態
00 = 索引輸出
01 = 音量索引加輸出
10 = 音符索引加輸出
11 = 音量和音符索引加輸出

>>> Music_ToneControl4 : 音色輸出控制寄存器 3
bit 0-3 索引輸入輸出計數器
bit4-5 索引輸入 Mask 值

>>> Music_ToneAddress : 音色控制數據表地址

>>> Music_ToneOutData0 : 輸出數據，對應\$40x0

>>> Music_ToneOutData1 : 輸出數據，對應\$40x1

>>> Music_ToneOutData2 : 輸出數據，對應\$40x2

>>> Music_ToneOutData3 : 輸出數據，對應\$40x3

程序函數說明

>>> MusicInitial：音樂程序初始化

入口數據：Music_AddrL,H 指向音樂數據索引表

音樂數據索引表格式：

adr 樂曲索引表地址
adr 音效索引表地址
adr Square1 聲部音色索引表地址
adr Square2 聲部音色索引表地址
adr Triangle 聲部音色索引表地址
adr Noise 聲部音色索引表地址
adr DPCM 聲部音色索引表地址

樂曲索引表格式：

adr 第一首樂曲數據地址
adr 第二首樂曲數據地址

.....
adr 第 n 首樂曲數據地址

樂曲數據格式：

byt s1,s2,s3,s4 音樂速度值

adr Square1 聲部音樂數據
0 = 沒有對應數據
其它正常

adr Square2 聲部音樂數據
0 = 沒有對應數據
其它正常

adr Triangle 聲部音樂數據
0 = 沒有對應數據
其它正常

adr Noise 聲部音樂數據
0 = 沒有對應數據
其它正常

adr DPCM 聲部音樂數據
0 = 沒有對應數據
其它正常

音效索引表格式：

adr 第一首音效數據地址
adr 第二首音效數據地址

.....
adr 第 n 首音效數據地址

音效數據格式：

adr Square1 聲部音效數據
0 = 沒有對應數據
其它正常

adr Square2 聲部音效數據
0 = 沒有對應數據
其它正常

adr Triangle 聲部音效數據
0 = 沒有對應數據
其它正常

adr Noise 聲部音效數據
0 = 沒有對應數據
其它正常

>>> MusicPlayEnable : 音樂播放

>>> MusicPlayDisable : 音樂靜音

>>> MusicEffectEnable : 音效播放

>>> MusicEffectDisable : 音效靜音

>>> MusicPlay : 音樂播放處理

入口參數:

REG A = 音樂編號

>>> MusicEffect : 音效播放處理

入口參數:

REG A = 音效編號

播放控制數據格式 : MPCC....Music Play Control Code

>>> MPCC_Wait : 等待節拍

Square/Triangle/Noise/DPCM 聲部

db %0cccccc

c = 等待的節拍數(0-127)

>>> MPCC_Timbre : 設置音色

Square/Triangle 聲部

db %1000iiii

i = 預置音色索引

Noise/DPCM 聲部

無效

>>> MPCC_Volume : 設置音量

Square/Noise 聲部

db %1001vvvv

v = 音量

Triangle/DPCM 聲部

無效

>>> MPCC_Inside : 按預置數據播放

Square/Triangle 聲部

db %1010tttt , %ccccllll

t = 音符

\$0 1

\$1 1#

\$2 2

\$3 2#

\$4 3

\$5 4

\$6 4#

\$7 5

\$8 5#
\$9 6
\$A 6#
\$B 7
\$C 空音符
\$D 空音符
\$E 空音符
\$F 空音符
1 = 音高
0 -...
1 -..
2 -.
3 =
4 +.
5 +..
6 null
7 null
c = 節拍長度
0-31
Noise & DPCM 聲部
%1010iiii
i = 預置音色索引(0-15)

>>> MPCC_Loop : 在指定地址到當前地址循環播放
Square/Triangle/Noise/DPCM 聲部
db %1011cccc
dw Address
c = 循環次數
Address 為指定的循環地址 (不可以嵌套)

>>> MPCC_ToneSet : 設置音調基準值
Square/Triangle/Noise/DPCM 聲部, 一般在 Square 聲部中使用
db %1100tone
tone = 新的音調基準值

>>> MPCC_FlagSet :設置 Music_Flag1 為 data
Square/Triangle/Noise/DPCM 聲部, 一般在 Square 聲部中使用
db %11110000,data
音樂播放開始時 Music_Flag 被清零

>>> MPCC_FlagInc Music_Flag1 加一
Square/Triangle/Noise/DPCM 聲部, 一般在 Square 聲部中使用
db %11110001

>>> MPCC_FlagDec Music_Flag1 加一
Square/Triangle/Noise/DPCM 聲部, 一般在 Square 聲部中使用
db %11110010

>>> MPCC_FlagBne 如果 Music_Flag1 不等於 data 跳轉
Square/Triangle/Noise/DPCM 聲部, 一般在 Square 聲部中使用
db %11110011,data
dw Address
Address = 轉移地址

>>> MPCC_FlagBeq 如果 Music_Flag1 等於 data 跳轉
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11110100,data
dw Address
Address = 轉移地址

>>> MPCC_FlagBneCall 如果 Music_Flag1 不等於 data 執行外部程序
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11110101
dw Address
Address = 外部程序地址

>>> MPCC_FlagBeqCall 如果 Music_Flag1 等於 data 執行外部程序
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11110110
dw Address
Address = 外部程序地址

>>> MPCC_SpeedSet 重新設置播放速度
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11110111

>>> MPCC_SpeedUp 加快播放速度
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11111000

>>> MPCC_SpeedDown 減慢播放速度
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11111001

>>> MPCC_ToneUp 播放升調
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11111010

>>> MPCC_ToneDown 播放降調
Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11111011

>>> MPCC_User 用戶自定義音色
Square/Triangle/Noise 聲部
db %11111100
dw Address
Address 為用戶定義的音色數據表
DPCM 聲部
db %11111100,a,b,c,d
a 對應於\$4010
b 對應於\$4011
c 對應於\$4012
d 對應於\$4013

>>> MPCC_Call 執行外部程序

Square/Triangle/Noise/DPCM 聲部，一般在 Square 聲部中使用
db %11111101
dw Address
Address = 外部程序地址

>>> MPCC_Jump 跳到指定的地址繼續播放
Square/Triangle/Noise/DPCM 聲部
db %11111110
dw Address
Address = 轉移地址

>>> MPCC_End 播放數據結束
Square/Triangle/Noise/DPCM 聲部
db %11111111

音色控制數據格式:MTCC....Music Tone Control Code

>>> MTCC_wait 等待 NMI
db %0-cccccc
c = 等待的 NMI 數(0-127)

>>> MTCC_Out 輸出 PSG 緩衝器
db %1000abcd[,A][,B][,C][,D]
abcd = 輸出 mask
0 = 無對應數據輸出
1 = 有對應數據輸出

>>> MTCC_Put 設置 PSG 緩衝器
db %1001abcd[,A][,B][,C][,D]
abcd = 設置 mask
0 = 無對應數據
1 = 有對應數據
A = ToneOutData0
B = ToneOutData1
C = ToneOutData2
D = ToneOutData3

>>> MTCC_Or 或 PSG 緩衝器
db %1010abcd[,A][,B][,C][,D]
abcd = 或 mask
0 = 無對應數據
1 = 有對應數據
A = ToneOutData0
B = ToneOutData1
C = ToneOutData2
D = ToneOutData3

>>> MTCC_Output 設置並輸出 PSG 緩衝器
db %1011abcd[,A][,B][,C][,D]
abcd = 設置和輸出 mask
0 = 無對應數據
1 = 有對應數據
A = ToneOutData0
B = ToneOutData1

C = ToneOutData2

D = ToneOutData3

>>> MTCC_Setup 索引輸出設置

db %1100abcd,-cccccc

abcd = 輸出 Mask，對 IndexOut 系列命令有效

cccccc = 輸出間隔值

>>> MTCC_IndexOut 索引設置并輸出 PSG 緩衝器

db %1101--00, %abcdnnnn

db data1,data2...data?

abcd = 設置 mask

nnnn = 輸出數量

先輸出再等待間隔值

>>> MTCC_VolumeIndexOut 音量索引加並輸出 PSG 緩衝器

db %1101--01, %---nnnn

db data1,data2...data?

nnnn = 輸出數量

先輸出再等待間隔值

>>> MTCC_ToneIndexOut 音符索引加並輸出 PSG 緩衝器

db %1101--10, %---nnnn

db data1,data2...data?

nnnn = 輸出數量

先輸出再等待間隔值

>>> MTCC_VolumeToneIndexOut 音量和音符索引加並輸出 PSG 緩衝器

db %1101--11, %---nnnn

db data1,data2...data?

nnnn = 輸出數量

先輸出再等待間隔值

>>> MTCC_Loop 循環

db %1110cccc

dw Address

cccc = 循環次數

Address = 循環地址

>>> MTCC_VolumeInc 音量加一

db %11110000

>>> MTCC_VolumeDec 音量減一

db %11110001

>>> MTCC_VolumeBne 音量不等於 data 跳轉

db %11110010, data

dw Address

Address = 轉移地址

>>> MTCC_VolumeBeq 音量等於 data 跳轉

db %11110011, data

dw Address
Address = 轉移地址

>>> MTCC_VolumeReset 音量恢復為默認值
db %11110100

>>> MTCC_VolumeClear 音量清除
db %11110101

>>> MTCC_Flag2Beq Music_Flag2 等於 data 跳轉
db %11110110,data
dw Address
Address = 轉移地址

>>> MTCC_Flag2Bne Music_Flag2 不等於 data 跳轉
db %11110111,data
dw Address
Address = 轉移地址

>>> MTCC_ToneAdd 音符加 data
db %11111000,data

>>> MTCC_SetpIndexOut 單步索引設置並輸出 PSG 緩衝器，每調用一次輸出一
db %11111001, %abcd---
dw Address
abcd = 設置 mask
Address = 索引地址
先輸出再等待間隔值，索引偏移為 Music_ToneControl4，每索引一次自動增加，
Music_ToneControl4 在從未使用過 IndexOut 系列命令情況下為零

>>> MTCC_Flag1Beq Music_Flag1 等於 data 跳轉
db %11111010, data
dw Address
Address = 轉移地址

>>> MTCC_Flag1Bne Music_Flag1 不等於 data 跳轉
db %11111011, data
dw Address
Address = 轉移地址

>>> MTCC_LoopBeq 如果循環遞減計數器等於 data 跳轉
db %11111100,data
dw Address
Address = 轉移地址

>>> MTCC_Call 執行外部程序
db %11111101
dw Address
Address = 外部程序地址
在不使用 IndexOut 系列命令的前提下，外部程序可以使用的寄存器有：
Music_ToneControl2 高四位

Music_ToneControl3 全部
Music_ToneControl4 全部
Music_Flag1 全部
Music_Flag2 全部

>>> MTCC_Jump 無條件轉移
db %11111110
dw Address
Address = 轉移地址

>>> MTCC_End 音色控制數據表結束
db %11111111
