

SD Music+ user's manual



<http://www.apple2.net>

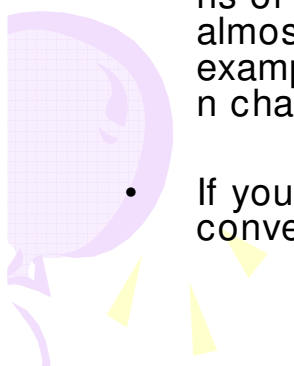


Prologue

- Our APPLE II has a speaker and can make some simple sound and CPU must handle every ticks for tone, Due to these difficulties, new sound card appeared name of Mocking board I, II, A, C but, so long time ago and discontinued.

During long time, It was my the only sound card for APPLE II and I can't find another sound card. Of course I can have a Passport card but it isn't sound card. It is just MIDI interface. After research both FM chip , AY3- 8910 and 6522 functions, I could design new sound card it is the SD Music+.

It isn't just new sound card with new chip sets. If we need new program or new game(?), it would be useless that's why It is compatible as our classic mocking board but, much better musical instruments.

- The SD Music+ integrated with new technologies to drive FM chips(YM2413).
 - You can use this card on APPLE II, II+, //e even IIGS but, as we know, Mocking board does n't work well with APPLE IIGS and have same compatible issue.
 - It can produce 9 channels of sound and 8 channels are used for mocking board channel and the last one is used for noise emulation.
 - But mocking board have limitations and not very good to drive YM2413, For drive all functions of Ym2413 sound chip, It provide 'Native mode' with this mode, you can control YM2413 almost of all functions and there are no limitations as mocking board compatible mode. For example you can use 5 percussion sounds and can add one user music instruments and can change to another music instruments.
 - If you are good at the programming on APPLE II, You can make good music player or music converter.
- 



Introduction

SD Music+ is 2nd version of SD Music card.

It has enhanced mockingboard channels for great music.

It is translate control signals of Mockingboard chips and make great sound.

It has APPLE II spaker mixer without wire connection on [APPLE II speaker.](#)

Enjoy it on your APPLE II with SD Music+ music supports.

Thank you very much;

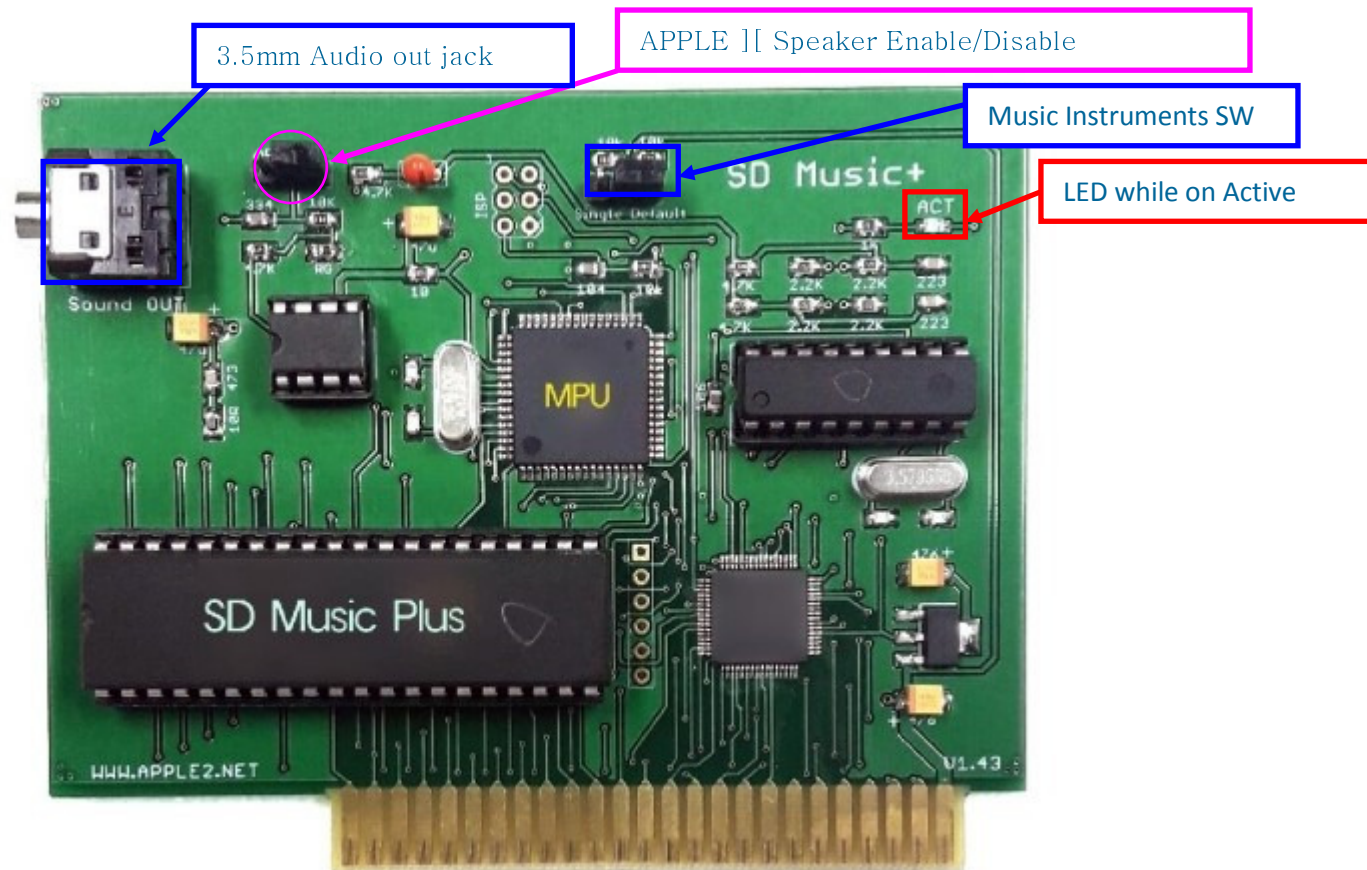


Installation

 **WARNING**

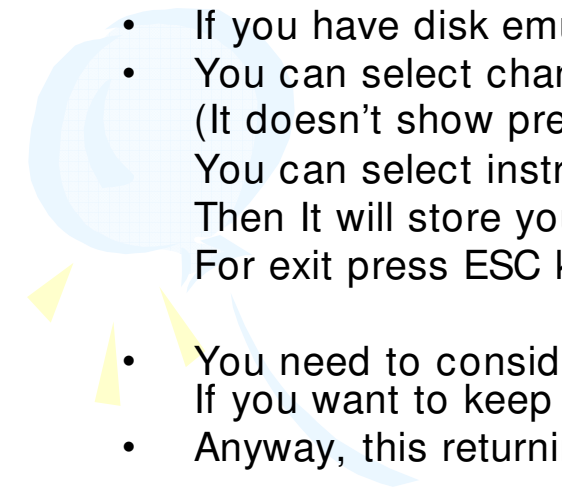
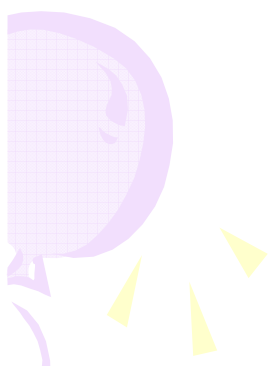
- Before to connect SD Music+, Please ***unplug power cord*** both APPLE II and your Audio amplifier.
- SLOT# 4 is recommended cause some of game just try to detect mockingboard existance on SLOT# 4 but works also another slot if game S/W could select slot# of mockingboard.

SD Music+ connector and switches





Music instrument assign Utility

- You can find image file which is a utility for assign music instruments of SD Music+. You need to transfer this utility disk to your real APPLE II via ADT or SD DISK] [Emulator. But, if you are difficult, Please try to keyin the shor BASIC program.
 - If you have disk emulator, boot the Image disk.
 - You can select channel which you want to change music instruments. (It doesn't show present music instruments sorry for it.) You can select instruments by < > keys and set with Return key. Then It will store your setting into the EEPROM of SD Music+. For exit press ESC key
 - You need to consider the switch cap, Cause it will return to defaults every Power ON. If you want to keep your changes, you have to remove the cap.
 - Anyway, this returning to default setting is doing once the power ON.
- 
- 

Music instruments configuration

Simply key- in channel number to change music instruments

```
SD MUSIC INSTRUMENT CONFIGURATION UTIL
WWW.APPLE2.NET
-----
INSTRUMENT LIST
1) Violin                2) Acoustic Guitar
3) Piano                 4) Flute
5) Clarinet              6) Oboe
7) Trumpet               8) Church Organ
9) French Horn          10) Synth Voice
11) Harpsichord          12) Vibraphone
13) Synth Bass           14) Acoustic Bass
15) Electric Guitar
-----
ESC:EXIT
-----
CHANNEL NUMBER (1~8)?*
```

Then use arrow key(<- >)to select music instruments then Press Enter.
That's all.. If you want exit, just press ESC..

Recommended to remove switch block to avoid resetting again while the Power ON APPLE II.

```
CHANNEL NUMBER (1~8)?1
SELECT MUSIC INSTRUMENT(<->,ENTER:SET)
1) Violin
```

Sound channels

- Mocking board II,A,C are have total 6 channels and each channel assigned to SD Music+ as below.

<No Switch block for maximum instruments>

<i>Mocking board channel</i>	<i>SD Music+ Channel</i>	<i>Remarks</i>
1	1(7)	7th is chorus
2	2(8)	8th is chorus
3	3	
4	4	
5	5	
6	6	
1 st Noise	9	Noise
2 nd Noise	9	Noise

7th channel and 8th channel are used just for chorus and playing with major channel.

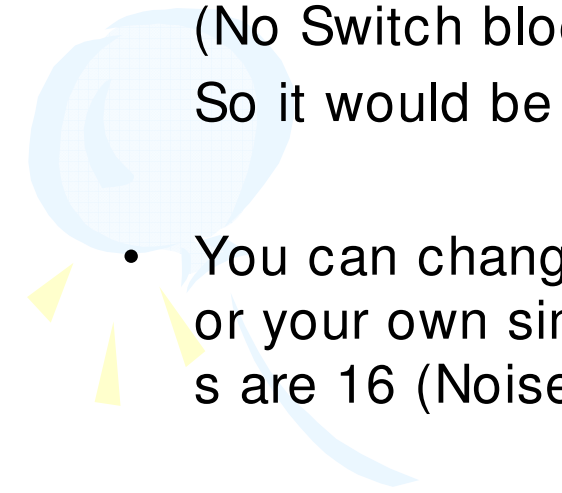
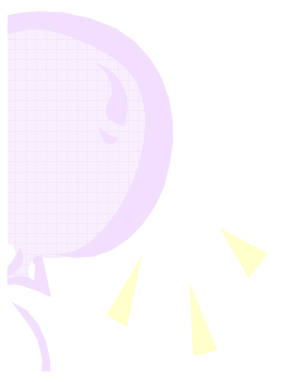


Channels assign

- Maximum 8 music instruments are assigned for each 6 mocking board channels. Each music instrument assigned in 1 to 6 and two more music instruments are used as chorus on 1 and 2

(No Switch block)

So it would be 1(7),2(8),3,4,5,6

- You can change this music instruments array with setting utility or your own simple BASIC code anyways total music instruments are 16 (Noise included)
- 
- 

Simple BASIC source code for Instruments assign

You can use this source code for your SD Music+.

You need to insert SD music+ slot in # 4

Simple put channel number then Music instruments number.

If channel number is '0' then will end of program.

Channel number is valid 1 to 8 and Music instruments number is valid 0~ 15

Please refer music instruments table.

10 HOME

20 INPUT "CHANNEL?";C

30 IF C=0 THEN END

40 INPUT "MUSIC INSTRUMENT?";M

50 S=C*16+M

60 POKE 50306,255:POKE 50307,255

70 POKE 50304,129: POKE 50305,S

80 POKE 50304,255

90 GOTO 20

Reference code on Monitor
(n: is slot number)

Cn82:FF 'Set PortB'

Cn83:FF 'Set PortA'

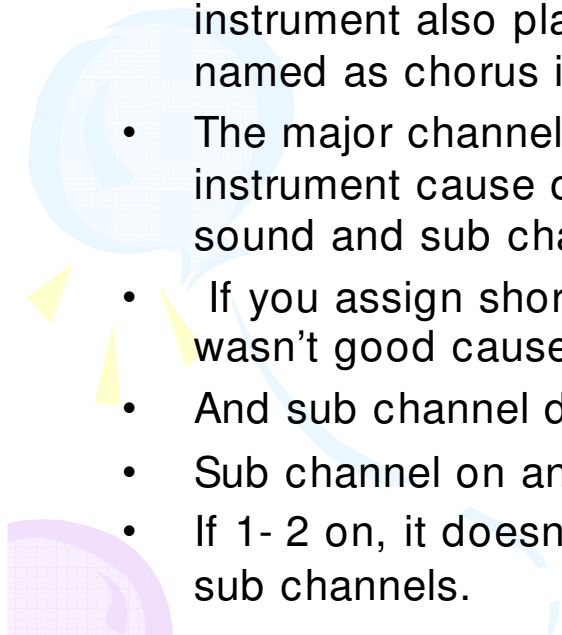

Cn80:81 'Mode for change of instrument'

Cn81: ## 'Channel/Instrument number'

Cn80:FF 'End of configuration'



Tip for music instruments

- Two type of Music instrument are using on SD Music+. One is concurrent sounding instrument and the other is short period sounding instruments and SD Music+ have major 1(7),2(8),3,4,5,6 channels and 7,8 channels are behind of 1st and 2nd channel. When 1st channel is playing then 7 channel of music instrument also playing together and 8 channel also playing with channel 2. It named as chorus instruments.
 - The major channels(1,2,3,4,5,6) are good playing with concurrent music instrument cause of Some of music using changing volume while the playing sound and sub channel is recommended to assign short period instruments.
 - If you assign short period(hit) music instruments to major channel, the sound wasn't good cause of again controlled by volume while the playing decay time.
 - And sub channel doesn't control by mocking board volume.
 - Sub channel on and off is depend on the switch block configuration.
 - If 1- 2 on, it doesn't have sub channel, 2- 3 on is one sub channel. Open is 2 sub channels.
- 
- 

Return to default instruments



※ If switch block was inserted on SD Music+, It will return to default music instruments set every power on the APPLE II. If you want to keep your own configured instruments array, Please remove the switch block.

Mode selection is following by switch block position as below.

- **1-2-3** : One music instrument used(Clarinet). Little similar as the Mockingboard sound. (2 noise channel, no sub instrument)
- **1 2-3** : Set to SD Music+ instruments to default. When you turn on the APPLE II, It will return to SD default. (Factory default) (**2 noise channel, one sub channel**)
- **1 2 3** : **No** switch cap. Using previous stored instruments setting. It doesn't change any music instrument. If you want to use user defined sets, you need set as this mode. (**One noise channel, two sub channels** get activate)

Music instruments channel status

Switch block on 1-2 (Left)

Channel #	Music instrument#	Description
1	5	Clarinet
2	5	Clarinet
3	5	Clarinet
4	5	Clarinet
5	5	Clarinet
6	5	Clarinet
8(1 st noise)	0	Noise
9(2 nd noise)	0	Noise

- Mocking board similar mode
- No sub chorus channel
- 2 noise channel on 8,9
- Channel# 7 is not used

Switch block on 2-3 (Right)

Channel #	Music instrument#	Description
1 (7)	4 (2)	Flute / Guitar
2	9	Horn
3	6	Oboe
4	5	Clarinet
5	7	Trumpet
6	8	Organ
8(1 st noise)	0	Noise
9(2 nd noise)	0	Noise

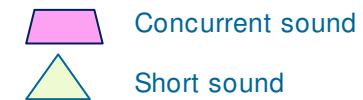
- One sub chorus channel
- 2 noise channel on 8,9

No switch cap

















Channel #	Music instrument#	Description
1(7)	No change	No change
2(8)	"	"
3	"	"
4	"	"
5	"	"
6	"	"
9 (1 st & 2 nd)	0	Noise

- Using previous setting
- One noise channel on 9
- 2 sub chorus channel activate

List of Music instruments



Tone Data

	Instrument		Instrument
0	<i>User define(Noise)</i> 	8	Organ 
1	Violin 	9	Horn 
2	Guitar 	10	Synthesizer 
3	Piano 	11	Harpsichord 
4	Flute 	12	Vibraphone 
5	Clarinet 	13	Synthesizer Bass 
6	Oboe 	14	Acoustic Bass 
7	Trumpet 	15	Electric Guitar 

Control as mocking board

To control mocking board, you need to know how to use 6522. Mocking board have tow 6522 and connection to AY3- 8910 as interface device. There are three major functions, Timer ,Int empts and Port IO.

- **Each 6522 has one timer and two IO ports**

- **Timer**

- You can use timer interrupts for accurate sound processing, 6522 provide one t imer interrupt and 6502 can have proper timing to send next note of your song. You need to learn how to handle this interrupts control and for set it Please refe r 6522 manual.

- **Port A and Port B for AY8910**

- **Set AY3- 8910 to get register number.**

- Set ay3- 8910 register number, Register number must already have on \$Cn01 e arlier, then this function just set act to sound chip to aware the data is Register number.

- **Store Data**

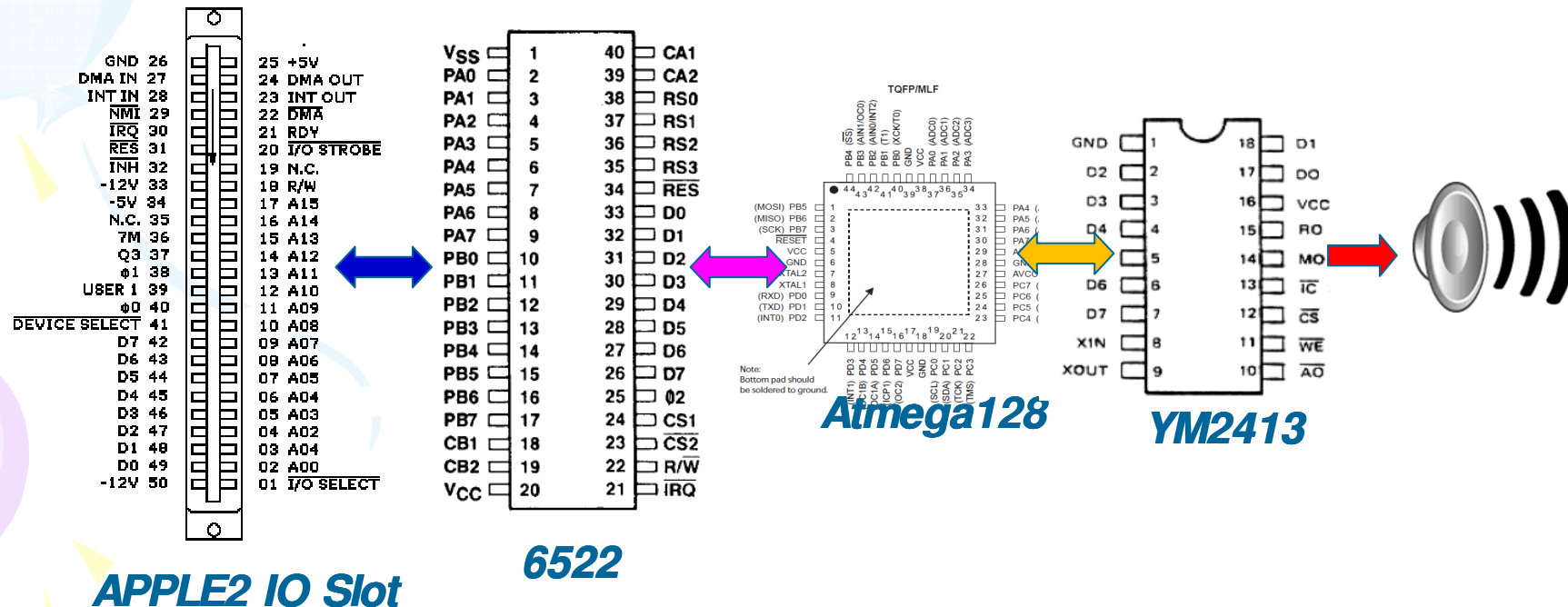
- Data should store in \$Cn01 and send data to previous register number.

- **Reset the sound chip**

- Reset all AY3- 8910 Registers. You can use it to reset the chip.

IO Address and diagram

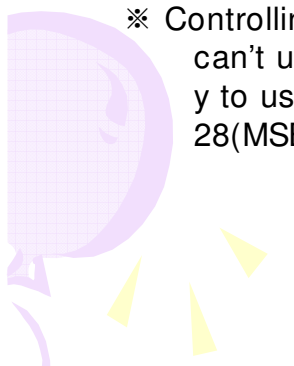
- **Mocking board- II,A,C** have two **AY3-8910**.
 - 1st 6522 occupied **\$Cn0x** and 2nd 6522 is **\$Cn8x** ※ n is slot number
- **Mocking board-I** has one AY3-8910
 - Connected on **\$Cn0x**
- **SD Music+** has one 6522
 - **\$Cn0x and \$Cn8x are mirrored**





Direct control of YM2413

- **SD Music+** also occupied same address as Mocking board. It occupied **$\$Cn0x$ to $\$Cn8x$ but only has one timer and IRQ interrupt.**
 - Mocking board compatible mode are following mocking board control.
- If you want to make new music player, I recommended to use this 'native mode' to control all of functions of YM2413. It is similar way as Mocking board but transparent way to control YM2413 chip.
 - For '**Native mode**', you can control as 2nd AY3- 8910 but, register number is start from **128-**. For example, Register # 128 is Register # 0 of YM2413, Register # 129 is heading to Register # 1 of YM2413 and so on.
So, with this say, you can control YM2413.
 - YM2413 Reset is same as Mocking board.
 - Register are write only.



※ Controlling SD Music+ is can be done same way of Mocking board. it's easy to use with old software. But, it can't use all of functions of YM2413 and sound may limited under mocking board control. So I suggest to try to use 'Native mode' for full control. It is logically separated and located 2nd AY3- 8910 Register number 128(MSB).

Sample Mockingboard routine

- **INIT ; Init of 6522 (once required)**

```
LDA #$FF
STA DDRA; $C403
LDA #$07
STA DDRB; $C402
RTS
```

- **RESET; Reset AY3- 8910**

```
LDA #$00
STA ORB ;$C400
LDA #$04
STA ORB ;$C400
RTS
```

- **Store data for Register**

- LDA # 11(Data)
- STA ORA; \$C401
- RTS

- **Use data for Register number**

```
LDA #$07
STA ORB ;$C400
LDA #$04
STA ORB ;$C400
RTS
```

- **Use data for Register data**

```
LDA #$06
STA ORB ;$C400
LDA #$04
STA ORB ;$C400
RTS
```

1st 6522 base address \$Cn00

2nd 6522 base Address \$Cn80
(n is slot number)

6522 Register table

Table 1. R6522 Register Addressing

Register Number	RS Coding				Register Desig.	Register/Description	
	RS3	RS2	RS1	RS0		Write (R/W = L)	Read (R/W = H)
0	0	0	0	0	ORB/IRB	Output Register B	Input Register B
1	0	0	0	1	ORA/IRA	Output Register A	Input Register A
2	0	0	1	0	DDRB	Data Direction Register B	
3	0	0	1	1	DDRA	Data Direction Register A	
4	0	1	0	0	T1C-L	T1 Low-Order Latches	T1 Low-Order Counter
5	0	1	0	1	T1C-H	T1 High-Order Counter	
6	0	1	1	0	T1L-L	T1 Low-Order Latches	
7	0	1	1	1	T1L-H	T1 High-Order Latches	
8	1	0	0	0	T2C-L	T2 Low-Order Latches	T2 Low-Order Counter
9	1	0	0	1	T2C-H	T2 High-Order Counter	
10	1	0	1	0	SR	Shift Register	
11	1	0	1	1	ACR	Auxiliary Control Register	
12	1	1	0	0	PCR	Peripheral Control Register	
13	1	1	0	1	IFR	Interrupt Flag Register	
14	1	1	1	0	IER	Interrupt Enable Register	
15	1	1	1	1	ORA/IRA	Output Register A*	Input Register A*

NOTE: *Same as Register 1 except no handshake.

AY3-8910 Register table

Fig. 3 PSG REGISTER ARRAY

REGISTER		BIT								
		B7	B6	B5	B4	B3	B2	B1	B0	
R0	Channel A Tone Period	8-BIT Fine Tune A								
R1						4-BIT Coarse Tune A				
R2	Channel B Tone Period	8-BIT Fine Tune B								
R3						4-BIT Coarse Tune B				
R4	Channel C Tone Period	8-BIT Fine Tune C								
R5						4-BIT Coarse Tune C				
R6	Noise Period					5-BIT Period Control				
R7	Enable	IN/OUT		Noise			Tone			
		IOB	IOA	C	B	A	C	B	A	
R10	Channel A Amplitude					M	L3	L2	L1	L0
R11	Channel B Amplitude					M	L3	L2	L1	L0
R12	Channel C Amplitude					M	L3	L2	L1	L0
R13	Envelope Period	8-BIT Fine Tune E								
R14		8-BIT Coarse Tune E								
R15	Envelope Shape/Cycle					CONT.	ATT.	ALT.	HOLD	
R16	I/O Port A Data Store	8-BIT PARALLEL I/O on Port A								
R17	I/O Port B Data Store	8-BIT PARALLEL I/O Port B								

YM2413 Register map

YAMAHA

YM2413

■ REGISTER MAP

Address	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
00	A	V	E	K	MULTI					User Tone Register
01	M	I	C	S						
02	KSL		T		L _{MS}					
03			DC	DM	F B					
04	A R			D R						
05										
06	S L			R R						
07										
0E			R	BD	SD	TOM	T-CT	HH	Rhythm Control	
0F	T E S T								OPLL Test Data	
10	F-Num. 0 ~ 7								F-Number LSB 8 bits	
18										
20			S	K	BLOCK			F	F-Number MSB, Octave set	
24			U	E	0 ~ 2			N	Key ON/OFF Register	
28			S	E				u	Sustain ON/OFF Register	
30			ON	Y				m		
34			OFF	ON				8		
38	INST.				VOL				Instruments Selection and Volume Register	

Register of YM2413 for User define and percussive instruments

■ REGISTER MAP											
Address	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀			
00	A	V	E	K	MULTI					User Tone Register	
01	M	I	G	S							
02	KSL		T L M								
03			DC	DM	F B						
04	A R			D R							
05											
06	S L			R R							
07											
0E			R	BD	SD	TOM	T-CT	HH	Rhythm Control		

- \$0E: Percussion instrument ON/OFF
- \$36~ \$38; Sound level of percussion instrument.

Addr	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
\$36				BD				
\$37	HH			SD				
\$38	TOM			T-CYM				



Other information

- SD MUSIC+ used Ym2413(OPLL) for sound generation
- AVR Atmega128 used as main controller.
- Mocking board compatible 6 channel to mapped music instruments.
- SD Music+ has no speech capability.
- Limited envelope support.
- Monotone sound.
- Components may be changed without notice.

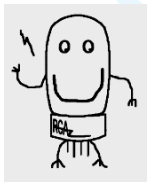


Compare to PSG(AY3- 8910)

- I'm writing this who doesn't understand why SD Music+ doesn't play for nice cybernetic song on Mocking board and it need to explain the reason why It doesn't and if it is bugs?
- Original mocking board used AY3-8910 PSG(Programable Sound Generator) and SD Music+ used Ym2413(FM;Frequency modulation). between two chips there are really big generation gabs.
- PSG is used frequency toggle based sound otherwise, FM chip using analog(sine wave) modified signal to emulate Music instruments.
- PSG can change the frequency very quickly with very wide range but, FM chip can do and the sound is not good to hear(it's depend on the music instrument setup but mostly as this).
- It can compare as tone generator with Piano.
- Piano can make sound only with it be hit, but PSG make sound as 'buu...' if added envelope little similar some type of ping.. but, in my experiment with Ultima5 and 4, they used volume control to emulate the envelope of music instruments.
- To make sound let's consider this. You want play from 100Hz to 5000Hz.
- PSG change frequency then sound will be generated.. it's simply connected from 100hz to 5000Hz.
- But, if you play it on the piano what will be happend? you need to hit from low key to high key quickly. Does it sound similar as PSG? Nope, very different..
- The Music instruments sound chip with PSG having similar reason.
- Due to this differences, SD Music+ can't play very fast changed notes sound, in fact it isn't exist with Music instruments songs. mostly cyber music and channel hopping ways.
- So, This is reason why I named as Music card not as Sound card. cause it is good for Music better than generation sound.
- However, I hope you to enjoy with SD Music+ for your favorite songs.
- I put some technical information of AY3-8910 and YM2413 for who want to control it directly not like as Mocking board but for full function of YM2413.

Other information

- SD MUSIC+ used a **[YM2413\(OPLL\)](#)** for sound generation
- AVR Atmega128 used for main controller.
- Mocking board compatible with channel mapped music instruments.
- SD Music+ doesn't support speech.
- Limited envelope support.
- No stereo sound.
- Components values may changed without notice.
- Warranty is 3 months and not included shipping costs. After this period may charged costs for repair service.
- Disassemble or modification may expire the warranty.
- If you have any question or concerns, please email me **quick09@gmail.com**
- Thank you very much



Jan Kim

Copyright© 2018 by SD Kim All rights reserved.