Musical Memory Mapper

Documentation

Version 2016-05-01

by GDX

Thanks to Ericb59

Index

Introduction

The Musical Memory Mapper (MMM) cartridge is a standard 1024KB memory extension for MSX1, MSX2, MSX2 + and MSX Turbo R.

However, it has other features. A SN76489AN sound chip used in various game consoles of the 80s has been added. The purpose is, among others, to run on MSX the games of other consoles that have this sound chip as main difference with the MSX computers. Softwares are provided to run games of the Sega SG-1000 and Colecovision, with identical sound as the original!

The Musical Memory Mapper has also a write protection function and the ability to manage its memory regardless of other Memory Mapper to make easier launch of MSX ROM files.

Registers description

The cartridge has registers accessible by two different ways except those of SN76489AN.

Registers accessible via I/O ports

Memory Mapper's registers ports

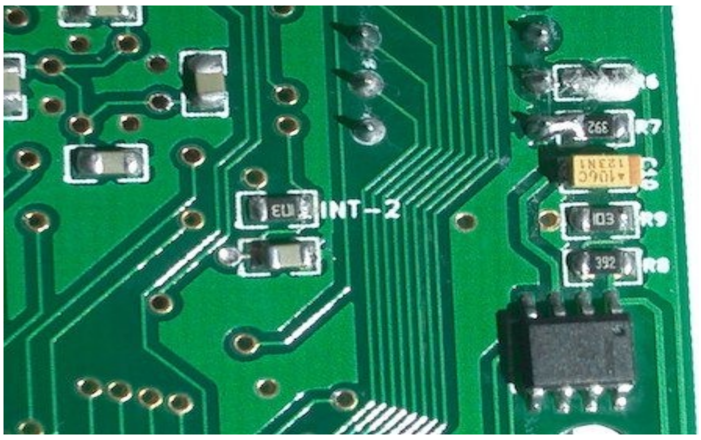
These registers are the standard Memory Mapper registers, and work at the same time and in the same way as all Memory Mapper on MSX.

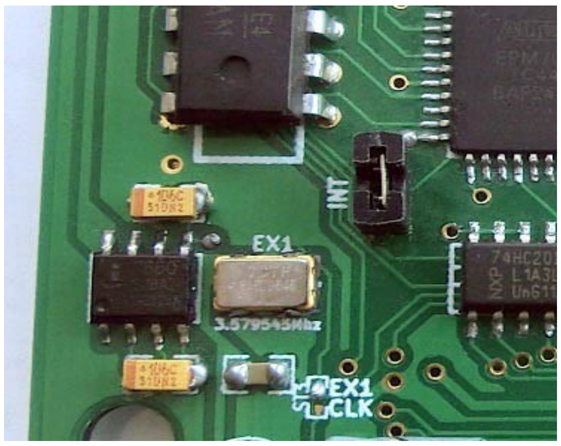
Read (*) / Write

Bank	Port	7	6	5	4	3	2	1	0
0000h~3FFFh	FCh	-	-	Page (0~31)					
4000h~7FFFh	FDh	-	-	Page (0~31)					
8000h~BFFFh	FEh	-	-	Page (0~31)					
C000h~FFFFh	FFh	-	-	Page (0~31)					

Bits 6 and 7 are ignored. They can be set or reset.

* The registers are readable only until the MMM v.1.2 (see the circuit board). This allows to use the MMM as if it were an internal Memory Mapper. If your MSX has an internal Memory Mapper, this can falsify the read value of registers in the configuration of slots. If you are faced with this rare case, please desoldering resistance "INT-2" of 10K (see left photo below). On the PCB v1.3you just need to remove the jumper "INT" (see the right picture below).





Note: MMM initializes Memory Mapper pages on each memory bank of 0000h~3FFFh, 4000h~7FFFh, 8000h~BFFFh and C000h~FFFFh in the order 3, 2, 1, 0 at MSX startup to provide a full compatibility even on MSX1. A standard Memory Mapper does not initialize its pages. This is the BIOS or the MSX-DOS2 that does it during the system initialization. The MSX1 Bios does not take into account the Memory Mapper. This may cause a crash at launch a program if a standard Memory Mapper was chosen as the main RAM.

Control Register Port of MMM

Only bit 7 of this register is accessible via I/O port.

Write only

Port	7	6	5	4	3	2	1	0
3Ch	BA	-	-	-	-	-	-	-

BA : Enables access to registers via access to an address.

Write port to SN76489AN registers

The SN76489AN has four channels sound (three programmable tone generators and one noise generator which can also generate a periodic pulse type waveform).

The SN76489A has 8 internal registers which are used to control the 3 tone generators and the noise source. During all data transfers to the SN76489A, the first byte contains a 3 bits field which determines the channel and the control/attenuation.

The volume of each channel can be controlled separately on 16 levels (high until silence) using registers 1, 3, 5 and 7.

Each byte Written to sound chip port determines either the register to be used with its value or the upper part of the value of frequency.

Tone generators registers:

Tone generator frequency is coded on 10 bits thus it is sent on 2 bytes to send one after the other. The output frequency is square with some imperfections.

First byte:

Write only

Port	7	6	5	4	3	2	1	0
3Fh	1	R2	R1	R0	D3	D2	D1	D0

D0~D3: Least significant 4 bits of frequency value.
R0~R2: Tone control register number

000 =	Tone frequency for channel 1
010 =	Tone frequency for channel 2
100 =	Tone frequency for channel 3
110 =	Noise control (channel 4)

Bit 7: Set to indicate that this is an access to the register of sound control

The second byte contains the most significant bits of wave value:

Write only

Port	7	6	5	4	3	2	1	0
3Fh	0	-	D9	D8	D7	D6	D5	D4

D4~D9: Most significant 6 bits of frequency value
Bit 6: Unused
Bit 7: Reset to indicate that this is the most significant bits of frequency value

Use:

To write the frequency value, you need to send the first byte that contains the 4 least significant bits of the value of the wave and destination and the second of the rest of the frequency value.

So if we want a frequency tone at 135 Hz (11 0011 1010 b) on channel 1, we need to do as in the following example.

First, we write the control word with least significant 4 bits of the frequency value:

```
    LD     A,10001010b
    OUT    (3Fh),A
```

Then, most significant 6 bits of the frequency value:

```
    LD     A,00110011b
    OUT    (3Fh),A
```

The frequency can be calculated by the following:

$$f = 3579545 / 32n$$

f is the output frequency and n is the binary 10-bit value.

Notes Frequency Conversion Table (Hz ↔ hexadecimal):

	Hz	Hex	Hz	Hex	Hz	Hex	Hz	Hex	Hz	Hex
A	110.00	3F8	220.00	1FC	440.00	0FE	880.00	07F	1760.0	03F
A$^{\#}$/B^{b}	116.54	3BF	233.08	1DF	466.16	0EF	932.33	077	1864.6	03B
B	123.47	389	246.94	1C4	493.88	0E2	987.77	071	1975.5	038
C	130.81	356	261.63	1AB	523.25	0D5	1046.5	06A	2093.0	035
C$^{\#}$/D^{b}	138.59	327	277.18	193	554.36	0C9	1108.7	064	2217.5	032
D	146.83	2F9	293.66	17C	587.33	0BE	1174.7	05F	2349.3	02F
D$^{\#}$/E^{b}	155.56	2CE	311.13	167	622.25	0B3	1244.5	059	2489.0	02C
E	164.81	2A6	329.63	153	659.25	0A9	1318.5	054	2637.0	02A
F	174.61	280	349.23	140	698.46	0A0	1396.9	050	2793.8	028
F$^{\#}$/G^{b}	185.00	25C	370.00	12E	739.99	097	1480.0	04B	2960.0	025
G	196.00	23A	391.99	11D	783.99	08E	1568.0	047	3136.0	023
G$^{\#}$/A^{b}	207.65	21A	415.30	10D	830.61	086	1661.2	043	3322.4	021

Note: The frequency of middle C is 523.25 Hz. The frequency of the same note an octave higher is 1046.5 Hz.

Control Registers of volumes:

To control the volume, a single byte has to be sent.

Write only

Port	7	6	5	4	3	2	1	0
3Fh	1	R2	R1	R0	V3	V2	V1	V0

V0~V3: Volume value

0000	= Maximum volume
0001	= Attenuation of 2dB
0010	= Attenuation of 4dB
0100	= Attenuation of 8dB
1000	= Attenuation of 16dB
1111	= Silence

R0~R2: Register number of volumes control

001	= Tone volume for the channel 1
011	= Tone volume for the channel 2
101	= Tone volume for the channel 3
111	= Noise volume (channel 4)

Bit 7: Reset to indicate that this is an access to volume control register.

Example:

Here is a routine in assembler to mute the sound of the four channels.

```
LD     A,09Fh
OUT    (3Fh),A    ; channel 1
LD     A,0BFh
OUT    (3Fh),A    ; channel 2
LD     A,0DFh
OUT    (3Fh),A    ; channel 3
LD     A,0FFh
OUT    (3Fh),A    ; channel 4
```

The equivalent in Basic.

```
OUT&H3F,&H9F:OUT&H3F,&HBF:OUT&H3F,&HDF:OUT&H3F,&HFF
```

Noise Generator control register:

The noise generator consists of a noise source that is a shift register with an exclusive OR feedback network. The feedback network has provisions to protect the shift register from being locked in the zero state. It also takes a single byte.

Format:

Write only

Port	7	6	5	4	3	2	1	0
3Fh	1	1	1	0	-	FB	NF1	NF0

NF1~NF0: Shift rate control

00 = N/512
01 = N/1024
10 = N/2048
11 = Tone generator of channel 3

You can use the output of the tone generator 3 to do interesting effects. If you sweep the frequency of the sound of the voice generator 3, it will cause a nice sweeping effect on the noise.
In general, this mode is used to reduce tone generator 3 with only the noise output as source.

FB: Feedback control

0 = Periodic noise
1 = White noise

The periodic noise is interesting. Depending on the frequency, it can seem very tonal and smooth.

The volume works in the same manner as for other channels.

Registers accessible via access to an address

The MMM cartridge has registers accessible by reading or writing to assigned address. Access the registers in this way allows you to control the functions of the cartridge independently. For example, owners of two same cartridges may control one by one each sound chip.

Control Register of MMM

The first writing to this register enables the sound chip. This produces a parasitic noise because SN76489AN registers can be initialized only by software. To avoid this, it's necessary to execute a routine to mute the volume of 4 voices immediately after the activation of SN76489AN. (See example at “Control Registers of volumes”)

Read / Write

Address	7	6	5	4	3	2	1	0
803Ch / 403Ch (*)	BA	SN	I/O	-	P3	P2	P1	P0

P0: Enable write protect bank 0000h~3FFFh.
P1: Enable write protect bank 4000h~7FFFh.
P2: Enable write protect bank 8000h~BFFFh.
P3: Enable write protect bank C000h~FFFFh.
Bit 4: Unused.
I/O: The bit 5 is used to disable access to the registers of the cartridge by the I/O ports. This allows to use the cartridge completely independent. This is useful in the case where an other same cartridge is inserted in another cartridge port. When this bit is set to 1, bit 7 also goes to 1 and remains in this state until the bit 5 is reseted. Access to internal registers are permanent at the addresses 803Ch and 80FCh ~ 80FFh but if P3 is set to 1, access to registers will be at the 403Ch and 40FCh~40FFh.
The I/O port 3Fh is not affected by the state of this bit, it will still be possible to enable or disable this port with bit 6.
SN: Enables 3Fh port to control the SN76489AN. Important: Do not enable the SN76489AN when the Z80 is in turbo mode if the MMM is less than the v1.3.
BA: Reset to disable access to records via access to an address.

* These addresses have mirrors each next 100h on the same bank memory of the cartridge slot.

Memory Mapper Registers of MMM

IThese registers are used to manage the Memory Mapper MMM independently of other Memory Mapper via the following addresses.

Read / Write

Bank	Address	7	6	5	4	3	2	1	0
0000h~3FFFh	80FCh (*)	-	-	Page (0~31)					
4000h~7FFFh	80FDh (*)	-	-	Page (0~31)					
8000h~BFFFh	80FEh (*)	-	-	Page (0~31)					
C000h~FFFFh	80FFh (*)	-	-	Page (0~31)					

Bits 6 and 7 are ignored. They will be at 1 at reading.

* These addresses have mirrors each next 100h on same bank memory of the cartridge slot.

Programming

Auto-detection of MMM

This assembler routine will allow you to find the cartridge port (or slot) in which MMM is inserted. This routine is necessary to activate the sound chip or using precisely his memory.

```
; Name: mmm_srch
;
; Input:  H = 080h (or 040h) (Significant byte of the memory bank address)
;         B = 16 (Possible number of secondary Slot)
;
; Output: A = Slot Number in the form F000SSPP (0FFh if MMM is not found)
;         B = Remaining iterations number
;
; Modify: All registers
;
; Size: 65 Bytes
;
; Note: The routine can be called multiple times to complete research in all
;       slots until B is set to 0 or until A is set to 0FFh.
;
; Warning: The routine does not replace the original slot after the search.
;

mmm_srch: ld        l,0FFh      ; HL = Access address to Memory Mapper register

mmm_srch_loop:

          push      hl
          ld        a,b         ; The Slot it is extended?
          dec       a
          and       3

          ld        hl,mnrom
          add       a,l
          ld        l,a

          ld        a,b
          dec       a
          or        (hl)

          pop       hl

          jp        m,ext_slt   ; Jump if the slot is extended

          and       %00001111
          cp        4
          jr        nc,nxt_srch ; If not a value of primary slot

ext_slt:  ld        c,a
          push      bc
          push      hl
          call      enaslt      ; Select Slot to scrutinize
          pop       hl
          pop       bc

          di                    ; Start of test

          ld        a,080h
          out       (03Ch),a    ; Enables access to registers by addressing

          ld        a,(hl)      ; The value read in address 080FFh (or 040FFh)
          and       %00011111   ;
          inc       a           ; must be
                                ;
          out       (0FFh),a    ; equal to the value
          or        %11000000   ;
          cp        (hl)        ; written to the port 0FFh
```

```
         ld       a,0
         out      (03Ch),a    ; Disables access to the registers by addressing
         out      (0FFh),a    ; Restore the page of system working area

         ei

         ld       a,c         ; Slot Number in the form F000SSPP
         jr       z,mmm_found

nxt_srch: djnz    mmm_srch_loop    ; Continues the search

         or       0FFh        ; MMM not found -> A = 0FFh R = bit Z à 0
         ret

mmm_found:
         dec      b           ; MMM found
         cp       a           ; Bit Z to 1
         ret
```

Example of calling the routine mmm_srch:

```
         org      0100h       ; Routine for MSX-DOS

ramad1   equ      0F342h      ; Main-RAM Slot (04000h~07FFFh)
ramad2   equ      0F343h      ; Main-RAM Slot (08000h~0BFFFh)
ramad3   equ      0F344h      ; Main-RAM Slot (0C000h~0FFFFh)

mnrom    equ      0FCC1h      ; Main-ROM Slot
slttbl   equ      0FCC5h

example:
         ld       h,080h      ; 80h to search the bank 08000h~0BFFFh
         push     hl
         ld       b,16
         call     mmm_srch

         ld       (mmm_slt),a ; Saving the number of slot found

         pop      hl          ; H=80h (or 40h)
         ld       a,(ramad2)
         call     enaslt      ; Restore the page of bank 08000h~0BFFFh
         ret

mmm_slt: db       0ffh
```

Note : In this example the ramad2 system variable is used. The variables ramad0 to ramad3 are available only when at least one disk is installed.
In a diskless environment, it is necessary to find the Main-RAM with his own routine. For this, it is good to know that some MSX have not all their RAM in the same slot. This is the case of the Sony MSX2 HB-500 for example. The following routine you will probably be useful if your program runs in a diskless environment.

Get the current status of a Slot

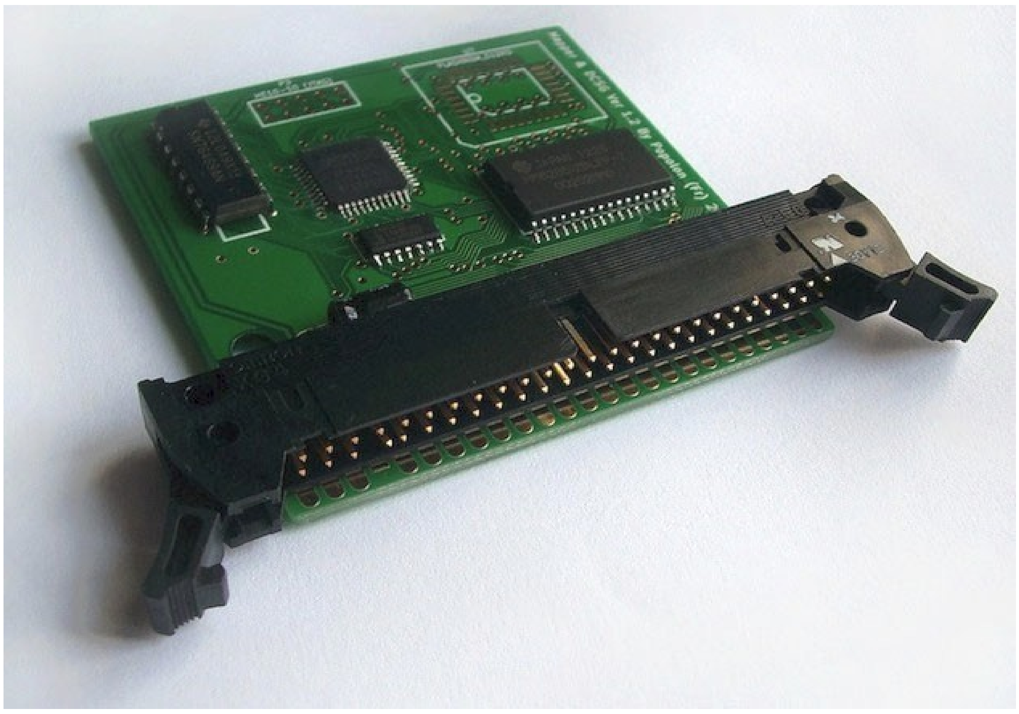
This routine in assembler is used to obtain the number of the selected slot of the corresponding memory bank.

```
; Name:   mmm_gsln (get_slot_number)
;
; Input:  H = 0h, 040h, 080h ou 0C0h (Significant byte of the memory bank)
;
; Output: A = Slot Number in the form F000SSPP
;
; Modify: All registers
;
; Size:   61 bytes
;
; Detail: Returns the slot address number of the memory bank to H
;
; note:   Also works in MSX-DOS
;

mmm_gsln:
         push     hl           ; Address of the bank of memory.
         xor      a
         ld       iy,(mnrom)
         ld       ix,rslreg
         call     calslt       ; A = status of current primary slots
         pop      hl
         push     af
         ld       a,h
         and      %11000000
         rlca
         rlca
         rlca
         ld       b,a          ; Number of shift to be performed to
                               ; retrieve the primary slot number
         pop      af
         inc      b            ; B should not be 0
         rlca
         ld       c,b

gsln_0:  rrca
         djnz     gsln_0

         ld       b,c
         and      %00000011    ; Current primary slot number of
                               ; the memory bank given by H register
         ld       hl,exptbl
         ld       d,0
         ld       e,a
         add      hl,de
         bit      7,(hl)
         ret      z            ; Back if the slot is not extended in
                               ; secondary slot (A = 000000PP).
         ld       d,0
         ld       e,4
         add      hl,de
         ld       c,a
         ld       a,(hl)       ; Secondary slots status in the form
                               ; SS3SS2SS1SS0.
         rlca
gsln_1:  rrca
         djnz     gsln_1       ; Number of rotations...

         and      %00000011    ; Secondary slot Number
         rlca
         rlca
         or       %10000000
         or       c            ; A = Slot Number in the form F000SSPP
         ret
```

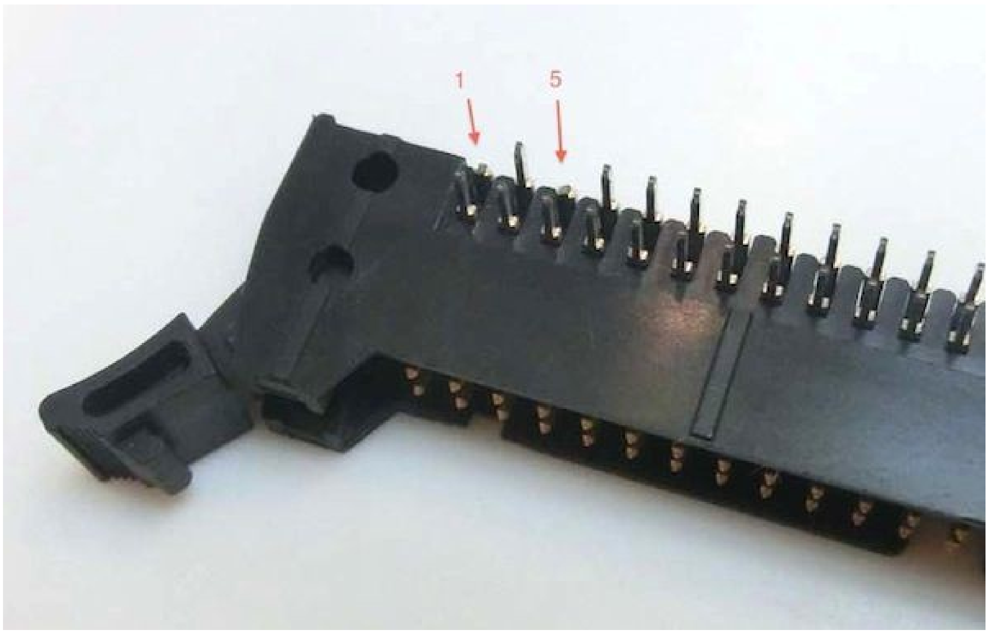
Additional Information

Connected directly the PCB to an expansion BUS

Musical Memory Mapper can connect directly to an MSX extension Bus with a ribbon cable by soldering a connector with the place as you can see in the following image.



Use a narrow SCSI connector for ribbon cable. It will be necessary to cut the pins 1 and 5 because of the hole that keeps the PCB in the cartridge case.

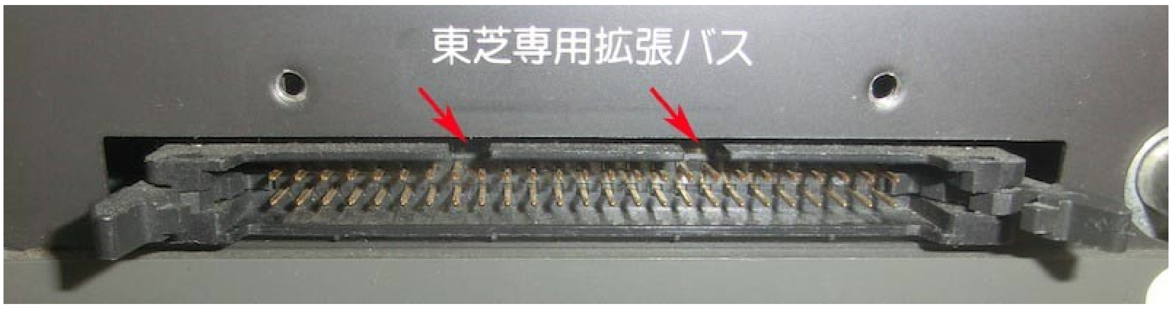


The ribbon cable must not exceed twenty centimeters.

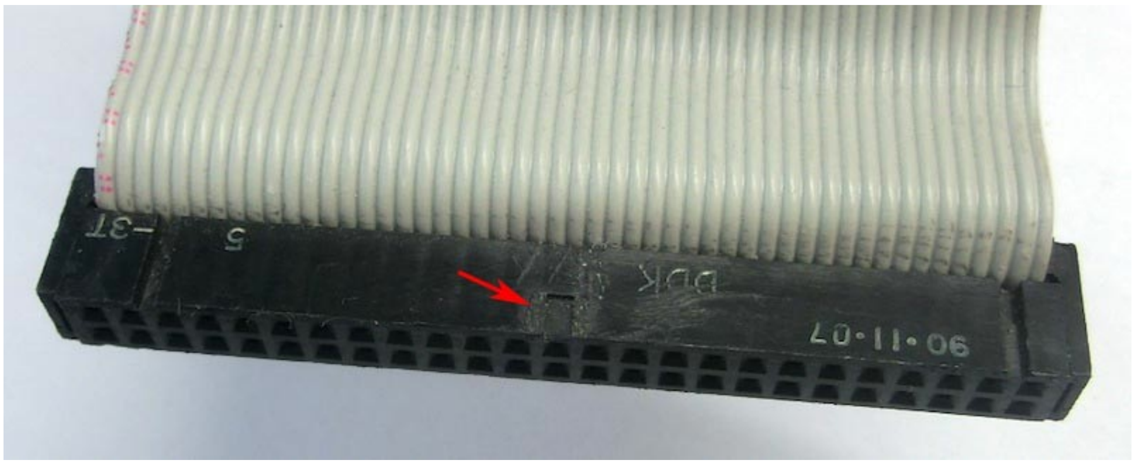
Notes:

- It is also possible to order a ready made version in suitable case but be careful because some expansion Bus have the sound input pin unconnected. In this case, only the Memory Mapper will work. You will have no alternative but to order a cartridge without sound chip unless you are able to add this pin.

- Some MSX has two polarizer on the Bus extension plug instead of one.



The ribbon cables with two polarizer bars are hard to find. The easiest solution would probably be to file down the bar on the cable connector.



Compatibility list

Musical Memory Mapper cartridge was tested with the following MSX.

- Frael Bruc 100 rev.1 (*) and rev.2 (**)
- MSX1 Canon V-20
- MSX1 Casio PV-7 and PV-16 (*)
- MSX1 Casio MX-10 (**)
- MSX1 National CF-2700
- MSX1 Philips NMS-8020
- MSX1 Sony HB-20P
- MSX1 Toshiba HX-10DP
- MSX1 Yamaha CX5MII
- MSX1 Yashica YC-64
- MSX2 Mitsubishi ML-G1 (*We must press the "DEL" key at startup when a memory expansion is inserted in this MSX*)
- MSX2 Philips NMS-8220
- MSX2 Philips NMS-8235 (*Some models of this MSX display 1024KB at startup when it comes to a cartridge Memory Mapper internal type. With the MMM v.1, it will show 128KB because it is an external Mapper type.*)
- MSX2 Philips NMS-8245 (*Some models of this MSX display 1024KB at startup when it comes to a cartridge Memory Mapper internal type. With the MMM v.1, it will show 128KB because it is an external Mapper type.*)
- MSX2 Philips MNS-8250/8255
- MSX2 Philips NMS-8280
- MSX2 Sony HB-F700P
- MSX2 Sony HB-G900
- MSX2 Victor HC-95
- MSX2 Yamaha YIS-503 III КУВТ-2
- MSX2 Panasonic FS-A1
- MSX2+ Panasonic FS-A1WX
- MSX Turbo R FS-A1ST (*Z80 and R800 modes*)
- MSX Turbo R FS-A1GT (*Z80 and R800 modes*)
- OneChipMSX (The Memory Mapper works but not the SN76489)

* MMM has not been tested on this MSX but given that the SOUNDIN pin is not connected on cartridge port / extension bus, the sound of SN76489 doesn't work. The Memory Mapper should work.
** MMM has not been tested on this MSX but given that the cartridge port does not provide + 12V nor -12V the sound of SN76489 should not work unless you have the MMM v.1.3 because it creates the necessary tensions itself. The Memory Mapper should work regardless the version.

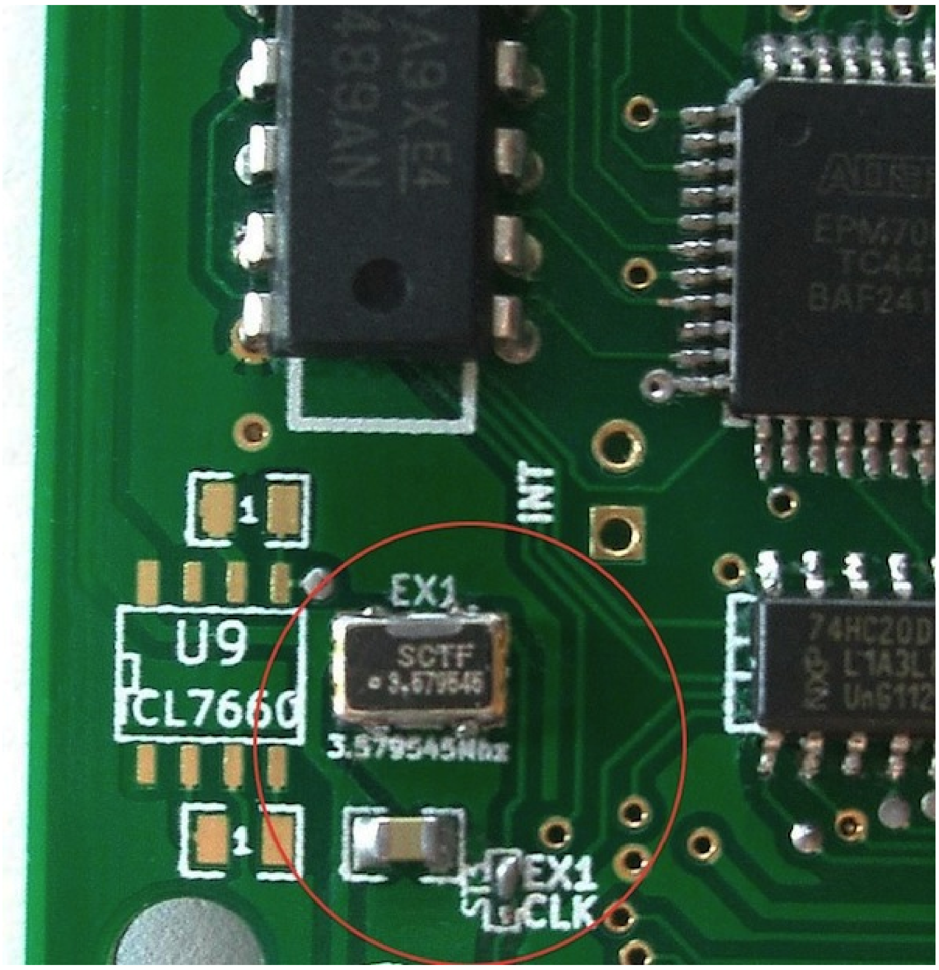
List of tested disk interfaces.

- CF/IDE Interface Sunrise
- CF/IDE Interface by MM (*Cartridge derived from Sunrise*)
- MegaFlashRom SCC+ SD
- MMC/SD Drive v.4.01 by Yeongman Seo (*The MSX-DOS2 does not boot on MSX1 with this interface.*)

The incompatible hardware is listed in red.

Versions history

- V1.0 - A sound filter and amplifier was added on a extra board to get a better sound.
- V1.2 - The filter and amplifier sound was built on the new PCB.
- V1.3 – An oscillator has been added in order to become independent of the clock of MSX. This allows to use even the sound functions on MSX with a Z80 clocked at over 3,579545Mhz.



MMM v1.3 with its oscillator (strap ST3 – EX1 soldered)