

# Documentation



## Musical Memory Mapper

Version 2016-05-01

par GDX

# Index

<b>Introduction</b>	3
<b>Description des registres</b>	4
Registres accessibles par les ports d'Entrée/Sortie	4
Ports des registres du Memory Mapper	4
Port du registre de contrôle de la MMM	5
Port d'écriture aux registres du SN76489AN	5
Registres de fréquence sonore	5
Registres de contrôle des volumes	7
Registre de contrôle du générateur de bruit	8
Registres accessibles par un accès à une adresse	9
Registre de contrôle de la MMM	9
Registres du Memory Mapper de la MMM	9
<b>Programmation</b>	10
Auto-détection de la MMM	10
Obtenir l'état actuel d'un Slot	12
<b>Informations complémentaires</b>	13
Connecter le PCB directement à un BUS d'extension	13
Liste de compatibilité	15
Historique des versions	16

## Introduction

La cartouche Musical Memory Mapper (MMM) est une extension de mémoire standard de 1024Ko pour MSX1, MSX2, MSX2+ et MSX turbo R.

Cependant, elle ne fait pas que ça. Elle possède aussi une puce sonore SN76489AN que l'on trouve dans divers consoles de jeu des années '80. Le but est, entre autres, d'exécuter sur MSX les jeux de consoles n'ayant que cette puce sonore comme principale différence avec le MSX. Des logiciels sont fournis pour faire fonctionner des jeux de la Sega SG-1000 et de la ColecoVision. Ainsi le son est identique à l'original !

La Musical Memory Mapper possède aussi une fonction de protection contre l'écriture et la possibilité de gérer sa mémoire indépendamment du Memory Mapper afin de pouvoir lancer plus facilement les images de ROM MSX.

## Description des registres

La cartouche possède des registres accessibles de deux façons différentes excepté ceux du SN76489AN.

### Registres accessibles par les ports d'Entrée/Sortie

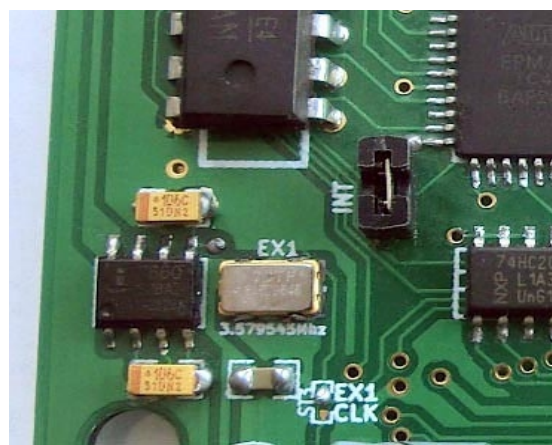
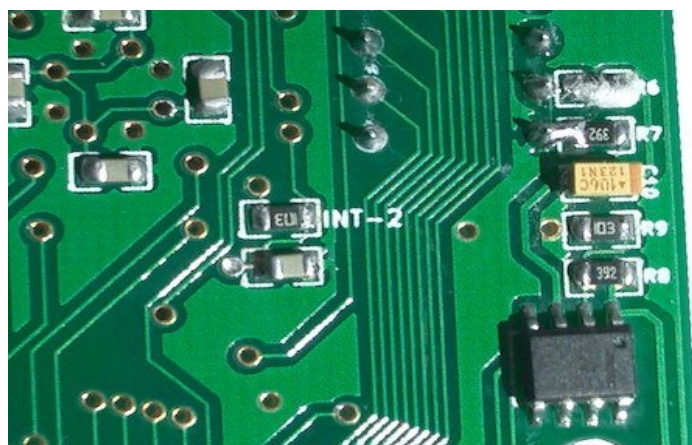
#### Ports des registres du Memory Mapper

Ces registres sont ceux d'un Memory Mapper standard. Ils fonctionnent en même temps et de la même façon que tous Memory Mapper sur MSX.

Ecriture / Lecture (*)									
Plage	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)					

Les bits 6 et 7 sont ignorés. Ils peuvent être à 1 ou à 0.

\* Les registres sont lisibles seulement à partir de la MMM v.1.2 (voir le circuit imprimé). Ce qui permet d'utiliser la MMM comme si il s'agissait d'un Memory Mapper interne. Si votre MSX a un Memory Mapper en interne, cela peut fausser la lecture des registres selon la configuration des Slots. Si vous êtes confronté à ce rare cas, veuillez dessouder la résistance « INT-2 » de 10KΩ sur le PCB v1.2 (voir la photo de gauche ci-dessous). Sur le PCB v1.3, vous devez juste retirer le cavalier « INT » (voir la photo de droite ci-dessous).



Note : Le Memory Mapper de la MMM a la particularité d'initialiser ses pages dans l'ordre 3, 2, 1, 0 sur chaque plage de mémoire de 0000h~3FFFh, 4000h~7FFFh, 8000h~BFFFh et C000h~FFFFh au démarrage du MSX afin d'offrir une compatibilité totale même sur MSX1.

Un Memory Mapper standard n'initialise pas ses pages dans cette ordre, c'est le BIOS ou le MSX-DOS2 qui le fait au démarrage. Le Bios des MSX1 ne prend pas en compte les Memory Mapper. Ce qui peut provoquer un plantage au lancement d'un programme si un Memory Mapper standard est choisi en tant que RAM principale.

## Port du registre de contrôle de la MMM

Seul le bit 7 de ce registre est accessible par le port E/S.

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

**BA :** Active l'accès aux registres par un accès à une adresse.

## Port d'écriture aux registres du SN76489AN

Le SN76489AN dispose de quatre voix sonores (3 générateurs sonores programmables et 1 générateur de bruit qui peut aussi générer une forme d'onde périodique de type pulse).

Cette puce comprend huit registres de contrôle sonore ; 3 pour générer les fréquences sonores, 3 autres pour changer l'amplitude des générateurs sonores, 1 registre de contrôle du générateur de bruit et enfin 1 pour contrôler l'amplitude et la périodicité du générateur de bruit.

Le volume de chaque voix peut être contrôlé séparément sur 16 niveaux (du silence au plus fort) à l'aide des registres 1, 3, 5 et 7.

Chaque octet écrit au port de la puce sonore détermine soit le registre à utiliser avec sa valeur soit, la partie haute de la valeur la fréquence.

## Registres de fréquence sonore

La valeur de la fréquence est codée sur 10 bits ce qui implique de l'envoyer sur 2 octets l'un après l'autre.

Premier octet :

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

**D0~D3 :** 4 bits de poids faible de la valeur de la fréquence.

**R0~R2 :** Numéro de registre de contrôle sonore.

000 = Fréquence sonore pour la voix 1.

010 = Fréquence sonore pour la voix 2.

100 = Fréquence sonore pour la voix 3.

110 = Contrôle du bruit (voix 4).

**Bit 7 :** Mettre à 1 pour indiquer qu'il s'agit d'un accès au registre de contrôle sonore

Le deuxième octet contient la partie haute de la valeur de la fréquence :

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

**D4~D9 :** 6 bits de poids fort de la valeur de la fréquence

**Bit 6 :** Inutilisé.

**Bit 7 :** Mettre à 0 pour indiquer qu'il s'agit de la partie haute de la valeur de la fréquence.

Utilisation :

Pour écrire la valeur de la fréquence vous devez d'abord envoyer le premier octet contenant les 4 bits de poids faible de la valeur de la fréquence et sa destination, puis le second indiquant le reste de la valeur de la fréquence.

Donc, si nous voulons un son de fréquence 135 hz (11 0011 1010 b) sur le canal 1, nous devons faire comme dans l'exemple suivant.

Tout d'abord, nous écrivons le mot de contrôle avec la première partie de la fréquence :

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

Ensuite, la seconde partie de la fréquence :

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

Pour définir la fréquence de l'onde à générer, utilisez cette formule:

$$f = 3579545 / (32n)$$

f est la fréquence de sortie et n correspond à votre valeur binaire de 10 bits.

Tableau de conversion des notes en Hz ↔ Hexadécimal :

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

Remarque : La fréquence du Do du milieu est 523,25 Hz. La fréquence de la même note d'une octave plus haut est 1046,5 Hz.

## Registres de contrôle des volumes

Pour contrôler le volume, un seul octet doit être envoyé.

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

**V0~V3** : Valeur du volume.

0000 = Volume maximum.

0001 = Atténuation de 2dB.

0010 = Atténuation de 4dB.

0100 = Atténuation de 8dB.

1000 = Atténuation de 16dB.

1111 = Silence.

**R0~R2** : Numéro de registre de contrôle sonore.

001 = Volume sonore pour la voix 1.

011 = Volume sonore pour la voix 2.

101 = Volume sonore pour la voix 3.

111 = Volume du bruit (voix 4).

**Bit 7** : Mettre à 1 pour indiquer qu'il s'agit d'un accès au registre de contrôle du volume.

Exemple :

Voici une routine en assembleur pour couper le son des quatre voix.

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

L'équivalent en Basic.

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

## Registre de contrôle du générateur de bruit

Le générateur de bruit a plusieurs modes de fonctionnement. Elle tient aussi sur un seul octet.

Voici le format de la commande :

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

**NF1~NF0** : Décalage de la fréquence (Shift Rate) du générateur de bruit

00 = Niveau de décalage N/512. (comme de la friture grasse)

01 = Niveau de décalage N/1024. (légèrement inférieure)

10 = Niveau de décalage N/2048. (haut grondement)

11 = Sortie de générateur sonore de la voix 3.

Vous pouvez utiliser la sortie du générateur sonore de la voix 3 pour faire des effets intéressants. Si vous balayez la fréquence du générateur sonore de la voix 3, cela provoquera un effet sympa de balayage du bruit.

En général, on utilise ce mode pour atténuer le générateur sonore #3 avec seulement la sortie de bruit comme source.

**FB** : Périodicité (Feedback)

0 = Bruit «périodique»

1 = Bruit «blanc»

Le bruit blanc sonne bien, comme un bruit blanc.

Le bruit périodique est intéressant. Selon la fréquence, il peut sembler très tonale et lisse.

Le volume fonctionne de la même manière que pour les autres canaux.



## Registres accessibles par un accès à une adresse

La cartouche MMM possède des registres accessibles par écriture ou lecture à des adresses attribuées. Accéder aux registres de cette façon permet de contrôler les fonctions de la cartouche indépendamment. Par exemple, les possesseurs de deux mêmes cartouches pourront contrôler une à une chaque puce sonore.

### Registre de contrôle de la MMM

La première écriture à ce registre active la puce sonore. Ceci produit un bruit parasite car les registres du SN76489AN ne peuvent être initialisés que par logiciel. Il est donc nécessaire d'exécuter une routine qui coupera le volume des 4 voix aussitôt après l'activation du SN76489AN. (Voir l'exemple à « Registres de contrôle des volumes »)

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

**P0** : Protection de la plage 0000h~3FFFh contre l'écriture.

**P1** : Protection de la plage 4000h~7FFFh contre l'écriture.

**P2** : Protection de la plage 8000h~BFFFh contre l'écriture.

**P3** : Protection de la plage C000h~FFFFh contre l'écriture.

**Bit 4** : Inutilisé.

**I/O** : Le bit 5 sert à désactiver l'accès aux registres de la cartouche par les ports E/S. Cela permet de rendre la cartouche totalement indépendante. C'est utile dans le cas où une autre même cartouche serait insérée dans un autre port cartouche. Lorsque ce bit est mis à 1, le bit 7 passe aussi à 1 et reste dans cet état tant que le bit 5 n'est pas remis à 0. L'accès aux registres internes restent permanent aux adresses 803Ch et 80FCh~80FFh mais si P3 est à 1, l'accès aux registres se fera aux l'adresses 403Ch et 40FCh~40FFh. Le port d'entrée-sortie 3Fh n'est pas affecté par l'état de ce bit, il sera toujours possible de l'activer ou de le désactiver par le bit 6.

**SN** : Active le port 3Fh afin de contrôler le SN76489AN. **Important : Ne pas activer sur les MSX dont le Z80 est en mode turbo si la MMM est inférieure à la v1.3.**

**BA** : Mettre 0 pour désactiver l'accès aux registres par un accès à une adresse.

\* Ces adresses ont des reflets toutes les 100h sur la même plage mémoire du Slot de la cartouche.

### Registres du Memory Mapper de la MMM

Ces registres permettent de gérer le Memory Mapper de la MMM indépendamment des autres Memory Mapper en y accédant via les adresses suivantes.

Lecture / Ecriture									
Plage	Adresse	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)					

Les bits 6 et 7 sont ignorés. Ils seront à 1 à la lecture.

\* Ces adresses ont des reflets toutes les 100h sur la même plage mémoire du Slot de la cartouche.

# Programmation

## Auto-détection de la MMM

Cette routine en assembleur vous permettra de trouver le port cartouche (ou Slot) dans lequel MMM est insérée. Cette routine est nécessaire afin d'activer la puce sonore ou d'utiliser précisément sa mémoire.

```
; Nom: mmm_srch
;
; Entree: H = 080h (ou 040h) (Octet de poids fort de la plage de memoire)
;         B = 16 (Nombre de Slot secondaires possible)
;
; Sortie: A = Numero du slot sous la forme F000SSPP (0FFh si MMM non presente)
;         B = Nombre d'iterations restantes
;
; modifie: Tous les registres
;
; Taille: 65 octets
;
; Note:   La routine peut etre appelee plusieurs fois jusqu'a ce que B soit
;         mis a 0 ou que A soit a 0FFh pour finaliser la recherche sur
;         la totalite des slots.
;
; Attention: La routine ne replace pas le slot d'origine apres la recherche.
;
```

```
mmm_srch: ld      1,0FFh      ; HL = adresse d'accès au registre du Mapper
```

```
mmm_srch_loop:
```

```
    push    hl
    ld      a,b              ; Le Slot est-t'il etendu ?
    dec     a
    and     3
```

```
    ld      hl,mnrom
    add     a,1
    ld      l,a
```

```
    ld      a,b
    dec     a
    or      (hl)
```

```
    pop     hl
```

```
    jp      m,ext_slt      ; Saute si le slot est etendu.
```

```
    and     %00001111
    cp      4
    jr      nc,nxt_srch ; Si ce n'est pas un numero de slot primaire
```

```
ext_slt: ld      c,a
    push    bc
    push    hl
    call    enaslt      ; Selection du Slot a scruter
    pop     hl
    pop     bc
```

```
    di                      ; Debut du test
```

```
    ld      a,080h
    out     (03Ch),a      ; Active l'accès aux registres par adressage
```

```
    ld      a,(hl)        ; La valeur lue a l'adresse 080FFh (ou 040FFh)
    and     %00011111      ;
    inc     a              ; doit etre
```

```
    out     (0FFh),a      ; egale a la valeur
    or      %11000000      ;
    cp      (hl)          ; ecrite au port 0FFh
```

```

        ld      a,0
        out     (03Ch),a    ; Desactive l'accès aux registres par adressage
        out     (0FFh),a    ; Restitue la page de la zone de travail du système

        ei

        ld      a,c         ; Numéro du slot au format F000SSPP
        jr      z,mmm_found

nxt_srch:djnz   mmm_srch_loop ; Continue la recherche.

        or      0FFh        ; MMM pas trouvée -> A = 0FFh R = bit Z à 0
        ret

mmm_found:
        dec     b           ; MMM trouvée
        cp      a           ; Bit Z à 1
        ret

```

Exemple d'appel de la routine mmm\_srch :

```

        org     0100h       ; Routine pour MSX-DOS

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

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

exemple:
        ld      hl,080h     ; 80h pour recherche dans la plage 08000h~0BFFFh
        push    hl
        ld      b,16
        call    mmm_srch

        ld      (mmm_slt),a ; Sauvegarde du numéro du Slot trouvé

        pop     hl          ; H=80h (ou 40h)
        ld      a,(ramad2)
        call    enaslt      ; Restitue le Slot de la plage 08000h~0BFFFh
        ret

mmm_slt: db      0ffh

```

Note : Dans cet exemple la variable système ramad2 est utilisée. Les variables ramad0 à ramad3 ne sont disponibles que lorsqu'au moins un disque est installé. Dans un environnement sans disque, il est nécessaire de trouver la « Main-RAM » avec sa propre routine. Pour cela, il est bon de savoir que certains MSX 64Ko n'ont pas forcément leur RAM dans le même Slot. C'est le cas du MSX2 Sony HB-500 par exemple. La routine suivante vous sera sans doute utile si votre programme tourne dans un environnement sans disque.

## Obtenir l'état actuel d'un Slot

Cette routine en assembleur sert à obtenir le numéro du Slot sélectionné d'une plage mémoire correspondante.

```
; nom:      mmm_gsln (get_slot_number)
;
; entree:   H = 0h, 040h, 080h ou 0C0h (Octet de poids fort de la plage de memoire)
;
; sortie:   A = numero du slot sous la forme F000SSPP
;
; modifie:  Tous les registres
;
; taille:   61 octets
;
; detail:   Renvoie le numero du slot de l'adresse de la plage de memoire
;           contenue dans H
;
; note:     Fonctionne aussi en Dos
;

mmm_gsln:
    push    hl          ; Adresse de la plage de memoire
    xor     a
    ld      iy,(mnrom)
    ld      ix,rslreg
    call    calslt      ; A = etat des slots primaires actuels
    pop     hl
    push    af
    ld      a,h
    and     %11000000
    rlca
    rlca
    rlca
    ld      b,a        ; Nombre de decalage a effectuer pour
                        ; recuperer le numero du slot primaire
    pop     af
    inc     b          ; B ne doit pas etre a 0
    rlca
    ld      c,b

gsln_0:  rrca
        djnz  gsln_0

        ld      b,c
        and     %00000011 ; Numero du slot primaire actuel de la plage
                        ; de memoire donnee par le registre H
        ld      hl,exptbl
        ld      d,0
        ld      e,a
        add     hl,de
        bit     7,(hl)
        ret     z        ; Rend la main si le slot n'est pas etendu
                        ; en slot secondaire (A = 000000PP)
        ld      d,0
        ld      e,4
        add     hl,de
        ld      c,a
        ld      a,(hl)    ; Etat des slots secondaires sous la forme
                        ; B3B2B1B0.

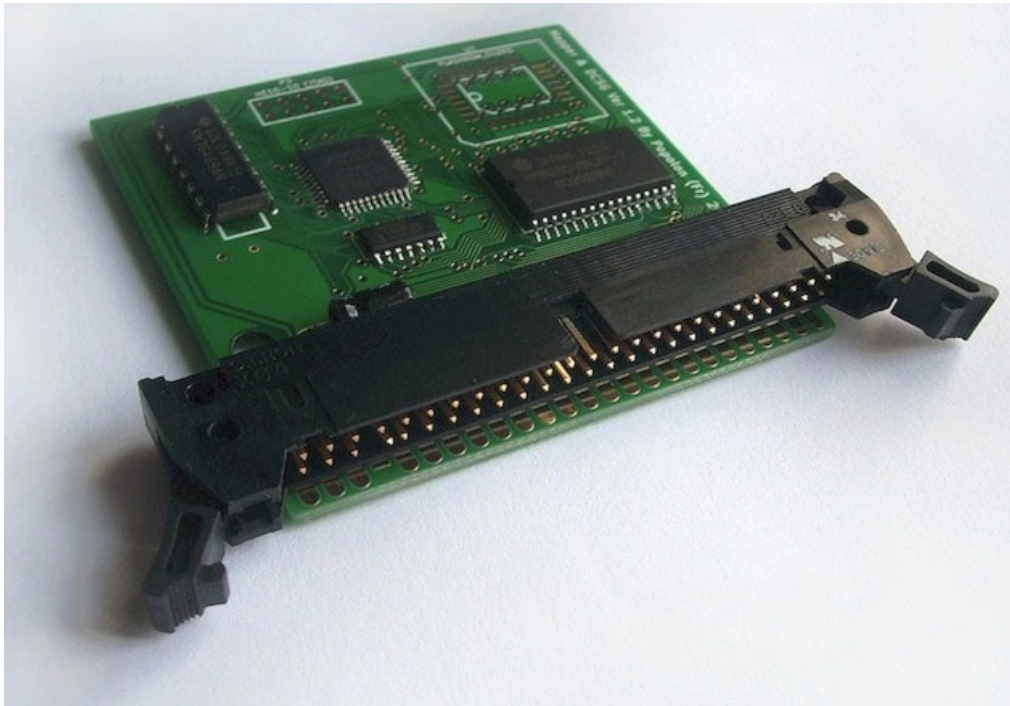
gsln_1:  rlca
        rrca
        djnz  gsln_1      ; Nombre de rotations...

        and     %00000011 ; Numero du slot secondaire
        rlca
        rlca
        or      %10000000
        or      c        ; A = Numero du Slot sous la forme F000SSPP
        ret
```

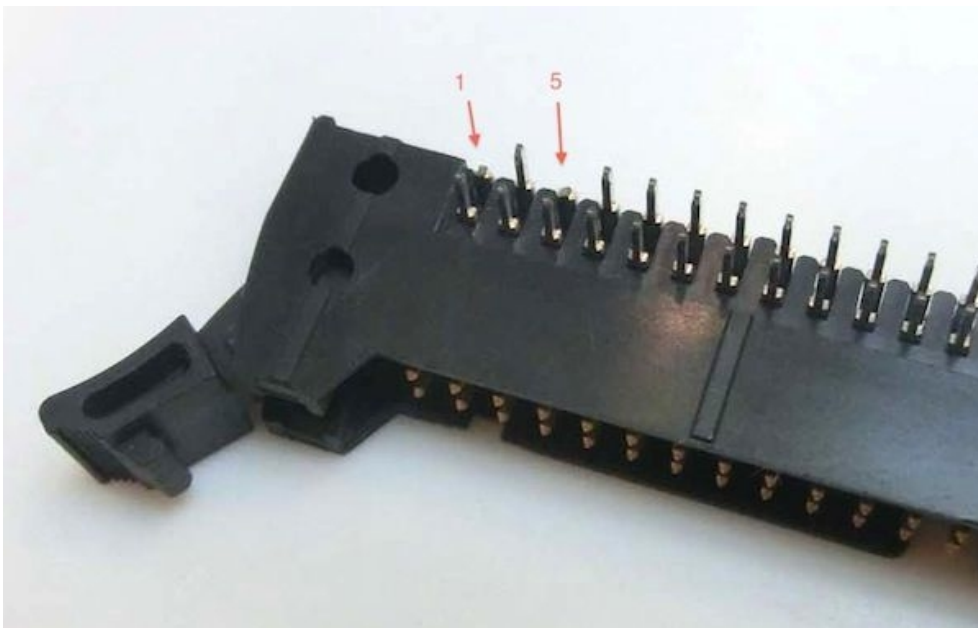
## Informations complémentaires

### **Connecter le PCB directement à un BUS d'extension**

Le Musical Memory Mapper peut se connecter directement sur un Bus d'extension MSX avec un câble en nappe en y soudant un connecteur à l'emplacement prévu comme vous pouvez le voir sur l'image suivante.



Utiliser un connecteur SCSI pour câble en nappe. Il sera nécessaire de couper les broches 1 et 5 à cause du trou de maintien du circuit imprimé dans le boîtier cartouche.



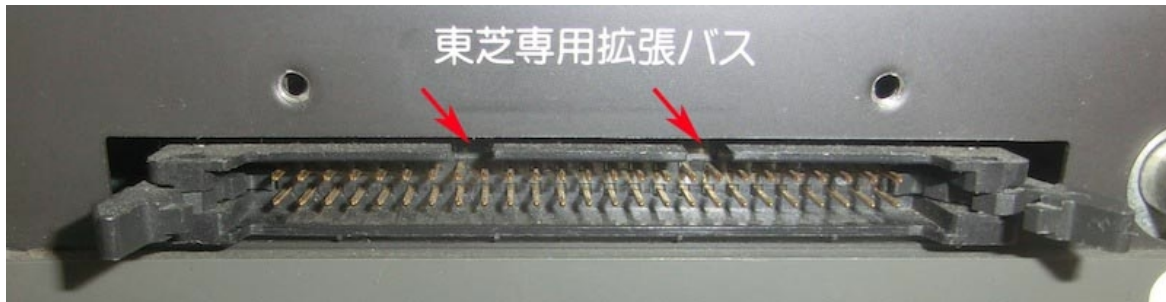
Le câble en nappe doit faire une vingtaine de centimètres maximum.

Remarques :

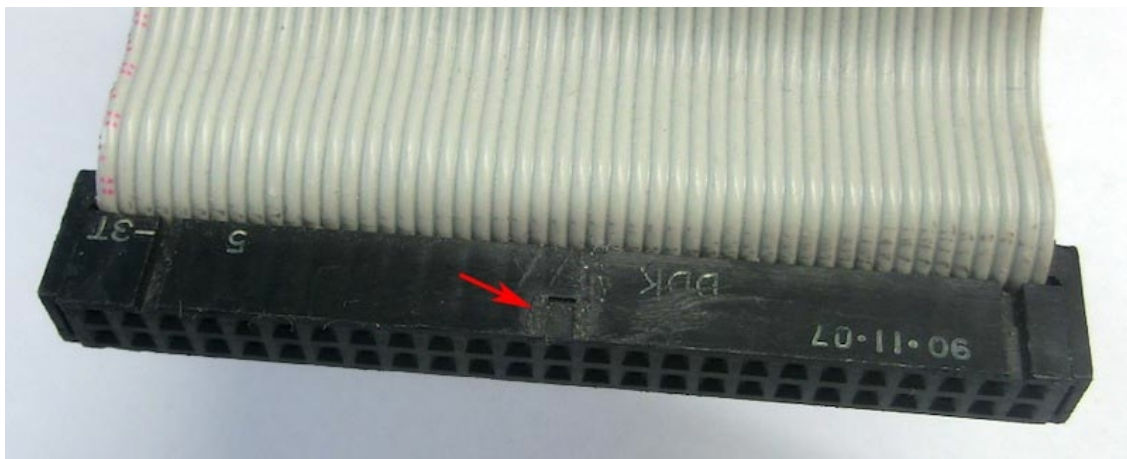
- Il est possible de commander une version dans un boîtier adapté pour le Bus d'extension mais avant de commander, faites attention car certains Bus d'extension ont la broche d'entrée sonore non connectée. Dans ce cas, seul le Memory Mapper fonctionnera. Vous n'aurez pas d'autre solution que de commander une cartouche sans puce sonore sauf si vous êtes capable de rajouter cette broche en vous inspirant du post suivant sur le forum de MSXvillage. Le condensateur et la résistance peuvent varier d'un MSX à l'autre.

<http://www.msxvillage.fr/forum/topic.php?id=913&pt=2#m28403>

- Certains MSX ont deux détrompeurs sur le connecteur du Bus d'extension au lieu d'un.



Les câbles nappe avec deux détrompeurs sont difficiles à trouver. La solution la plus simple sera sans doute de limer le détrompeur sur le connecteur du câble.



## Liste de compatibilité

La cartouche Musical Memory Mapper a été testée sur les MSX suivants.

- Frael Bruc 100 rev.1 (\*) et rev.2 (\*\*)
- MSX1 Canon V-20
- MSX1 Casio PV-7 et 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 (*Il faut presser la touche "DEL" au démarrage lorsqu'une extension de mémoire est insérée dans cet MSX.*)
- MSX2 Philips NMS-8220
- MSX2 Philips NMS-8235 (*Certains modèles de ce MSX affichent 1024KB au démarrage lorsqu'une cartouche de Memory Mapper de type interne est insérée. Avec la MMM v1, ça affichera 128KB puisqu'il s'agit d'un Mapper de type externe.*)
- MSX2 Philips NMS-8245 (*Certains modèles de ce MSX affichent 1024KB au démarrage lorsqu'une cartouche de Memory Mapper de type interne est insérée. Avec la MMM v1, ça affichera 128KB puisqu'il s'agit d'un Mapper de type externe.*)
- 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 KYBT-2
- MSX2 Panasonic FS-A1
- MSX2+ Panasonic FS-A1WX
- MSX Turbo R FS-A1ST (*modes Z80 et R800*)
- MSX Turbo R FS-A1GT (*modes Z80 et R800*)
- OneChipMSX (*Le Memory Mapper fonctionne mais pas le SN76489*)

\* MMM n'a pas été testé sur ce MSX mais étant donné que la broche SOUNDIN n'est pas connectée sur port cartouche / BUS d'extension, le son du SN76489 ne peut fonctionner. Le Memory Mapper devrait fonctionner quant à lui.

\*\* MMM n'a pas été testé sur ce MSX mais étant donné que le port cartouche ne délivre pas de +12V ni de -12V, le son du SN76489 ne devrait pas fonctionner sauf si vous avez la MMM v.1.3 car celle-ci crée les tensions nécessaires elle-même. Le Memory Mapper devrait fonctionner quant à lui quelque soit la version.

Liste des interfaces testées.

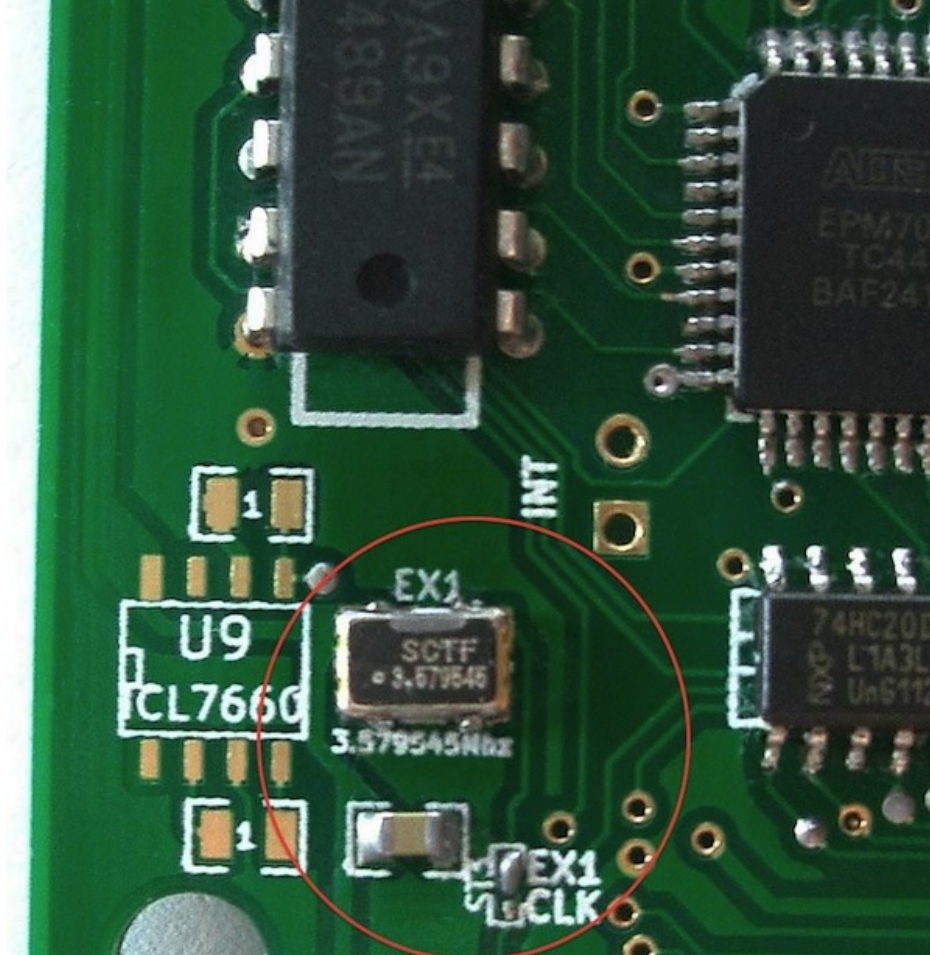
- CF/IDE Interface Sunrise
- CF/IDE Interface by MM (*dérivé de la Sunrise*)
- MegaFlashRom SCC+ SD
- MMC/SD Drive v.4.01 by Yeongman Seo (*Le MSX-DOS2 ne se lance pas sur les MSX1 avec cette interface*)

Le matériel incompatible est listé en rouge.



## Historique des versions

- V1.0 - Un filtre et l'amplificateur sonore a été ajouté sur une petite plaque à trou supplémentaire pour obtenir un meilleur son.
- V1.2 - Le filtre et l'amplificateur sonore a été intégré sur le nouveau PCB.
- V1.3 - Un oscillateur a été ajouté afin de ne plus dépendre de l'horloge du MSX. Cela permet d'utiliser les fonctions sonores même sur les MSX ayant le Z80 cadencé à plus de 3,579545Mhz.



MMM v1.3 avec son oscillateur (pont ST3 – EX1 soudé)