# MIDIbox SID V2

## Dowload Section

http://www.ucapps.de/mios_download.html

# User Manual http://www.ucapps.de/midibox_sid_manual.html

## Chapters

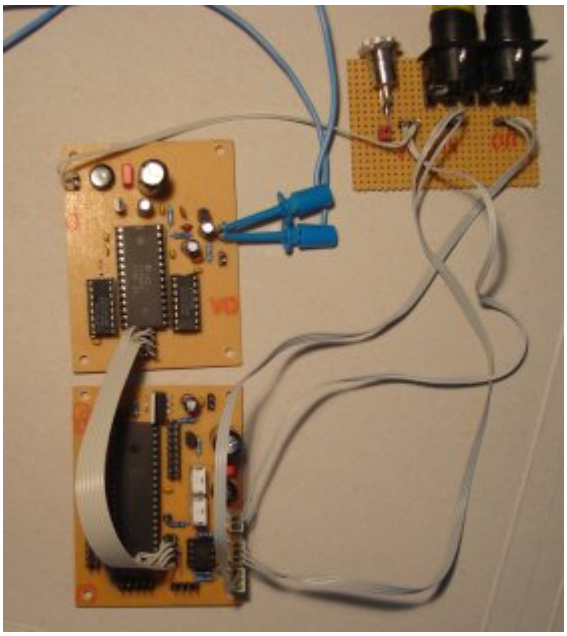

- Hardware Options ..... p 2
- Frontpanel ..... p 9
- Installation ..... p 16
- First Steps ..... p 18
- Lead Engine ..... p 28
- Bassline Engine ..... p 41
- Drum Engine ..... p 49
- Multi Engine ..... p 55
- Ensembles ..... p 60
- SysEx Editor ..... p 64
- CV Options ..... p 67
- SwinSID Review ..... p 69
- V1->>V2 Upgrade ..... p 72
- lien https://github.com/ > MIDIbox SID V2 Parameter Chart
- lien https://github.com/ > MIDIbox SID V2 CC Implementation Chart
- lilien https://github.com/ > MIDfIbox SID V2 SysEx Implementation
- Known Issues and Troubleshooting Tips ..... p 73
- Autres liens utiles ..... p 76


Copyright © 1998-2023, Thorsten Klose. All rights reserved. 

# Options Matérielles

La MIDIbox SID  peut être assemblée progressivement, de la configuration minimale jusqu'à la surface de contrôle complète avec jusqu'à 8 SIDs. Le coût total dépend principalement de la qualité des composants choisis, et de la disponibilité de chipsets SID. Quelques économies peuvent être réalisées en organisant des achats groupés via le Forum MIDIbox!

## Configuration minimale



L'image ci-dessus montre la configuration minimale requise pour faire tourner un SID. Dans cette configuration "low-cost", les paramètres sonores ne sont accessible que via SysEx **(*)**. En ajoutant un BankStick, 128 patches peuvent être stockés et rappellés via Program Change (jusqu'à 7 BankSticks sont pris en charge pour le stockage des patchs).

Le coût total pour cette config est d'environ. 30..50 EUROS (selon disponibilité de chipset SID).

**(*)**  pour l'instant, la MIDIbox SID V2 est en version beta, et l'éditeur SysEx n'est pas encore disponible. Il est prévu de réaliser un éditeur multi-plateforme, et basé sur Java comme c'était le cas pour la MIDIbox SID V1

## Configuration Stéréo

Avec la MIDIbox SID V2, deux chips SID peuvent être pilotés patr un seul module CORE. Le firmware propose un éventail de fonction simplifiant la création d'effets stéréo. Si vous avez le choix entre réaliser deux MIDIbox SID indépendante ou une seule pilotant deux SIDs à partir d'un seul CORE, je vous recommande de plutôt vous lancer dans la config stéréo.

Dans les démos suivantes, certains patches sont joués avec la version stéréo- à l'exception d'un limiter, aucun effet n'a été ajouté:
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_stereo_demo.mp3

La même séquence à été rejouée avec une config Mono - téléchargement direct (par ex. pour une comparaison avec un casque): Stereo Mono

Un autre avantages de la config stéréo: en mode "Multi", 6 voix/instruments polyphoniques peuvent être joués à partir d'un seul module CORE:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_multi.mp3

Dans la démo suivante, un module CORE joue deux basslines - vous trouverez plus d'infos dans cet article du forum.
Attention: >8 MB, peut paraître un peu répétitif!!

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo2.mp3

## Configuration Multi SID

Si vous êtes vraiment un fondu du son SID (comme moi), et si vous rêvez déjà d'orchestrer plusieurs parties jouées par un SID sans avoir à faire d'overdubbing, par exemple pour faire du "fine-tuning" sur plusieurs parties pour obtenir des sons monstrueux, alors intêressez-vous au concept de la MIDIbox SID V2! ;-).

Il defvient ici possible de piloter 3 synthés de plus dpuis la même surface de contrôle. Le premier CORE, qui gère également la surface de contrôle, sera géré comme "master", et les autres COREs comme "slaves". Les échanges de données entre les COREs sont réalisés via une interface CAN, cette méthode est ici appellée MBNet.

MBNet est un canal de communication distinct, fonctionnant plus vite que le MIDI, et permettant des échanges bi-directionnels via un seul conducteur. Le protocole CAN est habituellement utilisé dans les appliquations d'automation industrielles, où une bonne fiabilité est requise. Il ne s'agit donc pas juste d'une "bidouille", mais bien d'une approche très professionnelle...

Ce shéma montre comment les modules CORES doivent être inter-connectés. Chaque module CORE nécessite une ID matérielle MIOS distincte sur le réseau. Cette ID est "brûlée" simultanément dans l'ID d'en-tête du PIC avec le Bootloader. Le module master doit avoir l'ID 00, les modules slaves les ID 01, 02 et 03. Si vous avez acheté un PIC pré-programmé, par ex. chez SmashTV ou Mike, sans avoir spécifié d'IDs matérielles MIOS spécifiques, vous pouvez toujours changer celle-ci grâce à l'application "change_id", disponible sur cette page. Des fichiers .hex (device_id_01.hex .. device_id_03.hex) pré-configurés sont également disponibles, fichiers que vous pouvez utiliser sans avoir à re-compiler l'application;

Comme l'upload d'une application nécessite une connection MIDI bi-directionnelle avec les modules "slaves", vous aurez besoin de relier temporairement le port MIDI Out de votre MIDIbox SID to""""""""""the slave MIDI outs"""""""". Ou sinon, vous pouvez installer les PIC18F4685 destinés aux modules "slaves" dans le module "master" et uploader (et surtout tester!) les applications depuis ce module "master".

Une fois que le MIOS et l'application MIDIbox SID V2 ont été installés, il ne vous sera plus nécessaire de reproduire cette (fastidieuse) procédure: un mécanisme de "clonage" vous permet alors de transférer le firmware du master vers tous les slaves via MBNet. Ceci doit être fait à chaque mise à jour du firmware - maintenez simplement le bouton MENU enfoncé pendant le démarrage, jusqu'à ce que l'écran indique que la procédure de clonage a commencée.

## Sélectionner la puce SID

Il existe deux version du SID: le 6581 (première version, utilisée initialement dans les C64 au boitier marron, et le 8580 (version suivante, utilisée dans les C64 dans des boitiers plus plat et gris). Il existe également un 6582, vendu en remplacement des 8580 défectueux, et qui sonne du façon tout à fait similaire (voire identique?).

Les principales différences entre les deux types résident dans la tension d'alimentation et le choix des condensateurs de filtrages. Le 6581 doit être alimenté en 12V et requiert des condos de 470pF, les 8580 et 6582 fonctionne en 9V, avec des condensateurs de 22nF. Vous devez tenir compte de ceci avant de vous lancer dans la construction d'un module MBHP_SID .

Les deux puces sont également différentes au niveau son. Là, il devient dur de dire si l'une sonne mieux que l'autre, les deux sont particulières et présentent différents avantages...le filtre du 6581 sonne  sombre et sale, là où le filtre des 8580/6582 se révèle plus précis. La Résonance n'a quasi pas d'effet sur le 6581, certaine "mixed waveform" ne fonctionnent pas, et le bruit de fond est plus élevé (voir aussi les exemples audio dans le chapître Lead Engine), du coup les 8580/6582 est plus souvent préférable, plus particulièrement encore si vous voulez faire de bonnes Basslines.

## Alimentation (PSU)

Deux tensions différentes sont requises pour les parties numérique et analogique. Pour la partie numérique, c'est pour les deux types de puce du 5V, pour la partie analogique, les SID 6581 requierentt du 12V, tandis que les 8580/6582 doivent être alimentés en9V.

Pour la config minimale, une simple alimentation AC ou DC délivrant 12V pour les modules CORE/SID avec un 8580, ou 15V pour les modules CORE/SID avec un 6501, fera l'affaire. L'alim doit être capable de fournir au moins 500 mA. Bien que normalement un transfo de 6-9V soit recommandé pour alimenter un module Core (pour éviter que le régulateur 7805 ne chauffe trop), ici le module Core peut être alimenté depuis la même PSU que le SID, car sa consommation reste faible.

A partir du moment où vous installez un écran LCD rétro-éclairé, la consommation augmente de façon conséquente, et cette solution ne permet plus de faire fonctionner l'ensemble. Le 7805 chauffe encore plus, et vous entendrez probablement un buzz important dans les 50Hz/60Hz sur la sortie Audio. Une alimentation séparée pour le module SID est requise ici, ou bien vous pouvez utiliser le circuit "optimized C64 PSU" présenté sur le site.

J'ai expérimenté différents circuit d'alimentation et de régulation, pour finalement en déduire que réutiliser l'alim d'origine des C64 était la meilleure solution. En plus du fait que ceux qui ont canibalisé leur SID sur un vieux C64 dispose souvent déjà de cette alim, celle-ci présente l'avantage de proposer un rail régulé à 5V acceptant des consommation de courant élevées (et les 7805s des modules Cores peuvent êtres retirés), en plus d'une sortie AC utilisable pour alimenter en tension non régulée le module SID:
mbhp_8xsid_c64_psu_optimized.pdf

Soyez vigilant lorsque vous travaillez avec l'alim C64 PSU, elle ne dispose pas de fusible externe et accessible. Le fusible interne peut sauter en cas de court-circuit, et il est quasi impossible d'ouvrir le boitier (et encore plus difficile de refermer le boitier ensuite). Je ne me suis jamais trouvé dans cette situation (même si j'ai parfois mis l'alim en court-circuit par erreur), mais d'autre utilisateurs ont déjà signalé ce problème.

## Sorties Control Voltage (CV/AOUT)

La MIDIbox SID V2 peut piloter jusqu'à 8 sorties analogiques en connectant au Core un module MBHP_AOUT , ou jusqu'à 4 modules MBHP_AOUT_LC .

Ceci ouvre tellement de possibilités qu'un chapître spécial est consacré à ce sujet - voir CV Options.

## Entrées Analog Control

La MIDIbox SID V2 propose 8 entrées analogiques (5 dors et déjà utilisables, 3 réservées pour de futures extensions). Elles peuvent être utilisées pour connecter des pots/faders ou un joystick supplémentaires pour contrôler le son, mais peuvent aussi être pilotées depuis des générateurs externes de modulation. Vous trouverez plus d'infos à la section Frontpanel.

## SID: Entrée Audio

La puce SID dispose d'une entrée audio, qui permet de router des signaux externes vers le filtre. Si aucun signal externe n'est envoyé vers cette entrée (J4 du module SID), utilisez un jumper (ou un jack à coupure) pour mettre cette entrée à la masse, dans le cas contraire, vous aurez du buzz supplémentaire sur la sortie audio!

Alpha, un type inventif, a découvert qu'en renvoyant la Sortie Audio dans l'Entrée Audio via une résistance, il était possible d'obtenirdu SID: Lien vers le site.

Voir aussi: cet sujet sur le forum .

## Surface de Contrôle

...est discutée plus en détails au chapître Frontpanel.



## Encodeurs: Avec Ou Sans Détente?

Il est recommandé d'utiliser des encodeurs avec détente pour les contrôles des menus (datawheel), et sans détente pour les autres paramètres, si une surface de contrôle complète est réalisée.

Les encodeurs avec détente permettront une édition plus précise dans les menus, et ceux sans

détente seront plus appropriés pour réaliser des modulations rapides et plus fines. Notez que la "détente" peut être selon les modèles d'encodeurs facilement retirée (voir cette page du Wiki et ce sujet sur le forum).

## Matrice de Modulation avec transistor de puissance

La Matrice de Modulation consiste en 8x7 (optionnellement 8x8) LEDs, connectées à un 74HC595. Comme le courant en sortie des shift register est limitée à 20 mA, il vous faut utiliser des LEDs basse consommation, qui sont dans la plupart des cas pas plus chères que des LEDs standard.

Sinon vous pouvez rajouter des transistor de puissance derrière les 74HC595. L'option DEFAULT_SRM_USE_SINKDRIVERS doit être activée en éditant votre fichier setup_*.asm, afin d'inverser les sorties. Consultez la page du Wiki de Wilba pour le schéma.

## BankSticks

Les 24LC256 (32k) et 24LC512 (64k) sont pris en charge pour les BankSticks, mais les 24LC512 sont plus appropriés, car ils permettent de sauvegarder 128 patches.

Jusqu'à 7 BankSticks peuvent êtres utilisé pour la sauvegarde des Patches, chacun nécessitant une adresse particulière (CS=0..CS=7).

Le 8ième BankStick (sur CS#7) (pins E0/E1/E2 du BankStick connectés à Vd (5V)) servira à stocker les Ensembles. Ceci est valable aussi bien pour les 24LC256 que 24LC512, dans les deux cas 128 Ensembles peuvent être stockés dans les premiers 32k.

Notez que votre MIDIbox SID peut fonctionner sans BankSticks, mais dans ce cas seul un patch et un seul Ensemble peuvent être sauvés dans l'EEPROM interne, c'est une configuration peu pratique, à éviter.

## Coût Matériel

Des PCBs prêt à l'emploi sont disponibles sur les sites www.mikes-elektronikseite.de ou http://mbhp.coinoptech.com. Ils proposent également des kits complets pour les modules et des composants périphériques (comme des encodeurs rotatifs).

Les prix et numéro de catalogue sont pour Reichelt si rien est indiqué. Une liste de magasins en dehors d'Allemagne est disponible sur le Wiki

**Note:** Le PIC18F4685 est un micro-contrôleur assez récent, rarement disponibles dans les boutiques web classiques.
SmashTV a fait un achat groupé et propose des PICs avec le Bootloader et le MIOS chargés à un prix plus qu'intéressant (**$9.95 !!!** Comparez aux prix sur findchips.com, vous ne trouverez pas de meilleure offre à l'unité!)
Doc s'occupe de la distribution pour l'Europe pour ceux qui voudraient économiser sur les frais de ports - contactez le via le Forum MIDIbox

**Configuration minimale**

| Composants | Description | Prix |
|---|---|---|
| 1 module MBHP_CORE | Le cerveau de la MIDIbox SID **vous devez y installer un PIC18F4685 !!!** | Parts: env. 20 EUR<br>PCB: env. 6 EUR |

| | | |
|---|---|---|
| 1 module MBHP_SID | Module équipé du SID, qui inclus la partie amplification audio et l'interface série pour la connection au module core. | Parts: env. 5 EUR<br>PCB: env. 5 EUR |
| chip SID | Récupéré sur un vieux C64 (acheté par exemple sur EBay), ou à commandé chez des spécialistes (mot clé: Hong Kong) | 0..20 EUR |
| BankStick | au moins 1 x 24LC512-I/P, Part #4915665 chez Farnell, Part #579-24LC512-I/P chez Mouser, Part #24LC512 chez SmashTV. | env. 3 EUR |
| résistance 1k | une résistance 1k est requise pour l'entrée de l'interface CAN (J15:D3 du module core), même si aucun Slave n'est connecté au core! | 0.10 EUR |
| PSU | Une alimentation - soit AC ou DC, env. 12V pour les modules CORE/SID avec 8580 ou 15V pour les modules CORE/SIDavec 6581, env. 500 mA. Bien que normalement un transfo de 6-9V soit recommandé pour alimenter un module Core (pour éviter que le régulateur 7805 ne chauffe trop), ici le module Core peut être alimenté depuis la même PSU que le SID. Voir aussi la page MBHP_SID comment réutiliser l'alim d'origine des C64. | env. 0..6 EUR |

**Stéréo SID**

| | | |
|---|---|---|
| +1 module MBHP_SID | idem | Parts: env. 5 EUR<br>PCB: env. 5 EUR |
| +1 chip SID | Récupéré sur un vieux C64 (acheté par exemple sur EBay), ou à commandé chez des spécialistes (mot clé: Hong Kong) | 0..20 EUR |

**Surface de Contrôle Minimale**

| **Composants** | **Description** | **Prix** |
|---|---|---|
| 2x20 LCD | par ex. Part #LCD 202A LED chez Reichelt | env. 10..15 EUR |
| module DINX2 | Aucun PCBs spécifique n'est dispo pour ce petit circuit, il vous faudra le réaliser sur une platine d'essai. Sinon vous pouvez utiliser un module DINX4. | Parts: env. 5 EUR<br>PCB: env. 5 EUR |
| 1 encodeur | Un encodeur avec détente est recommadé ici, pour permettre une édition précise des paramètres. M-SW-ROT chez Voti ou encodeurs Bourns chez SmashTV's Shop | env. 1..2 EUR |
| 9 boutons | par exemple "DT 6 *colour*" (*colour* = BL, GN, GR, RT, SW) | 6.50 EUR |
| Platine d'essai | pour y monter vos boutons/encodeurs/LCD.<br>par ex. H25PR200 chez Reichelt | 2.25 EUR |

## configuration Multi SID

**Description**

Composants et Modules supplémentaires:

- n x modules core
- n x diodes 1N4148  (pour le bus CAN)
- m x modules SID
- la surface de contrôle minimale (voir plus haut)
- une alim suffisante (env. 800 mA) - L'alim des C64 est à préferer ici car elle dispose d'une sortie +5V séparée pour les modules Core
- 4 boutons supplémentaires
- un module DOUTX1
- 7 LEDs

## Surface de Contrôle Complète (Step C)

**Description**

Composants et Modules supplémentaires:

- n x modules core
- m x modules SID
- la surface de contrôle minimale (voir plus haut)
- une alim suffisante (env. 800 mA) - L'alim des C64 est à préferer ici car elle dispose d'une sortie +5V séparée pour les modules Core
- 3 modules DINX4

- 2 modules DOUTX4
- 14 encodeurs rotatifs de plus (au contraire de l'encodeur pour la Datawheel, ici des encodeurs sans détente sont préférables, même si ceux avec détente fonctionneront aussi. par ex. M-SW-ROT chez Voti ou encodeurs Bourns chez SmashTV's Shop)
- 32 boutons supplémentaires
- 99 LEDs
- env. 2-3 platines d'essai pour monter les boutons/encodeurs/LEDs
- une jolie face avant

Coûts: les encodeurs sont les composants les plus chers (env. 15 EUR), ainsi que la façade frontpanel (fait maison: 25 EUR + et beaucoup de temps, commandé chez un spécialiste: 100..150 EUR!)

## intégration sur un seul PCB:

L'approche modulaire de la MBHP a ses avantages, au niveau de l'intégration de différents circuits ou encore de la facilité à tester les montages. Mais d'un autre côté, la MIDIbox SID V2 a tellement évoluée, que maintenant il faut réunir jusqu'à 20 modules, ce qui s'avèrent parfois difficile à faire rentrer dans la boite. Un PCB unique présenterait ici l'avantage d'offrir plus de robustesse, pour un coût moins élevé.

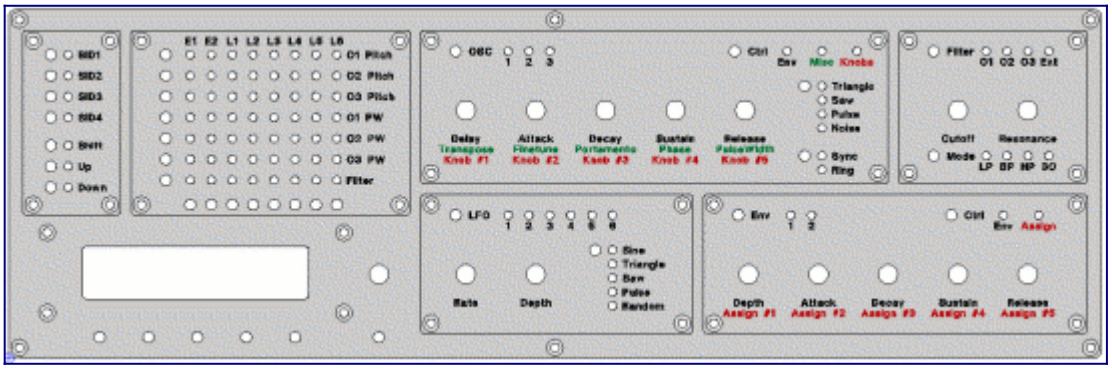
Wilba a créé un PCB spécial pour sa MB-6582, qu'il présente sur cette page du Wiki. Une commande groupée pour ces PCBs a été initiée pour ceux qui sont intéressé par cette solution "tout-en-un" - détails sur le Wiki.

## Ressources

- **Firmware**: voir ce sujet du forum
- **Diagramme de Connection pour les boutons & encodeurs**: mbsid_v2_din_default.pdf
- **Diagramme de Connection pour les LEDs**: mbsid_v2_dout_default.pdf
- **Diagramme de Connection pour l'alimentation**: mbhp_8xsid_c64_psu_optimized.pdf
- **Diagramme de d'interconnection pour le MBNet et les connections MIDI** mbsid_v2_communication.pdf
- **Photos**: midibox_sid_photos/
- **Frontpanel template**: mbsidv2_front.fpd - éditable avec Frontpanel Designer, disponible sur le site de Schaeffer Apparatebau (*)

# Façade

Ce chapître présente la façade de la MIDIbox SID V2. 9a et vous donneras une idée de la configuration minimale nécessaire et un aperçu des différentes options que propose le firmware.



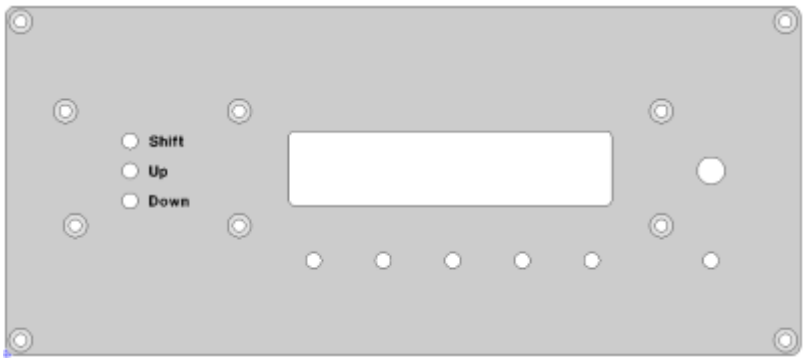
## Façade V1 vs. V2

Comme je l'ai souvent souligné sur le site ou le forum MIDIbox, une des principale contrainte lors de la phase de conception de la MIDIbox SID V2 été de maintenir compatible la nouvelle version avec la façade de la MIDIbox SID V1, afin qu'aucune modification de la partie matérielle ne soit nécessaire pour les utilisateurs de la V1. C'est donc ce qui a été fait - et je suis toujours satisfait de mon premier design, celui en photo ci-dessus. Les seules modifications concerne l'implémentation de fonctions alternatives pour certains boutons et encodeurs. Les sérigraphies ont étés modifiées dans le fichier modèle Frontpanel, qui peut être ouvert et édité à l'aide de "Frontpanel Designer", un logiciel gratuit disponible sur le site Schaeffer Apparatebau. Vous pouvez utiliser ce modèle comme base si vous envisagez de réaliser une SID.

Malgrés tout, la façade originale a été réalisée en 2003, et pourquoi ne pas rajouter des touches dédiées pour certaines des nouvelle fonctions... Ces options sont décrites ci-dessous, libre à vous de toutes les implémenter ou non...

## surface de contrôle minimale

C'est l'option que l'on connaissait sous le nom de "Step A" - elle vous permet déjà d'accéder à tous les paramètres, mais les éléments de contrôle vous permettant d'éditer directement (sans avoir à naviguer dans les menus) manquent. C'est un compromis entre coût matériel et ergonomie.




Cette config requiert un écran LCD 2x20, 9 boutons et un encodeur rotatif, 11 entrées digitales d'un module MBHP_DIN suffisent. Vous pouvez soit réaliser un module DINX2 "à la main" (2 shift register = 16 entrées), sur une platine d'essai (par exemple directement sur la platine où sont montés vos encodeurs/boutons), ou vous pouvez acheter un kit DINX4, sur lequel vous n'aurez besoins de monter que les deux premiers shift registers et les composants qui les accompagnent.

Les 5 boutons de sélection servent à naviguer dans les menus, et à sélectionner le paramètre que vous souhaitez éditer avec l'encodeur rotatif (exemple: voir First Steps Guide). Le bouton en dessous de l'encodeur (bouton MENU) sert à quitter une page du menu. L'encodeur incrémente/décrémente la valeur d'un paramètre ou permet de naviguer dans les différentes pages du menu.

Le bouton SHIFT était anciennement nommé LINK. Sa fonction change par rapport à la V1 - il permet maintenant d'afficher une page donnant accés au choix des canaux audio Left/Right, au choix du moteur sonore, à l'initialisation d'un patch, à la génération aléatoire d'un nouveau patch, ou à l'envoi via SysEx d'un dump du patch en cours.
Par ailleurs, les valeurs sont incrémentées plus lentement (plus précis) tant que ce bouton est maintenu enfoncé.

En particulier pour le choix du moteur sonore, il n'y a pas d'alternative possible, du coup ce bouton s'avère absolument nécessaire.

Deux boutons Up/Down ont été ajoutés à la surface de contrôle minimale, car nous disposons ici de toute façon de deux entrées digitales disponibles. Il ne sont pas absolument requis, mais facilitent la navigation dans les menus. Selon les pages du menu, ils peuvent être utilisés pour incrémenter/décrémenter le numéro d' OSC/LFO/ENV/MOD/WT/Knob/... de n'importe quelle page sans avoir à scroller d'abord jusqu'au premier paramètre d'un menu. A l'intérieur du menu racine ou de l'écran principal, ils peuvent être utilisé pour changer le numéro de patch/ensemble.
Ces deux boutons été utilisés avant pour activer les fonctions CC/Edit - ces deux fonctions sont maintenant accéssibles via les la combinaison de bouton SHIFT+UP (pour la fonction CC) ou SHIFT+DOWN (pour Edit).

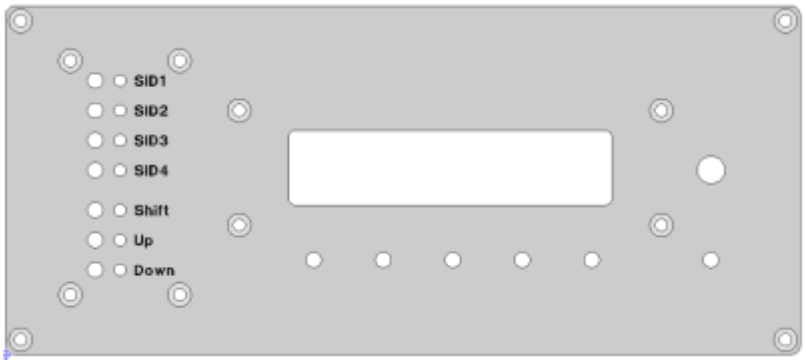
Vous pouvez utiliser un écran LCD 2x16 (pour faire quelques économies). Mais la solution n'est pas pratique, car la page principale ne pourra pas être affichée dans son ensemble. Seule 4 paramètres sur 5 seront visibles simultanément (vous devrez scroller en bout de page pour afficher les derniers paramètres). L'avantage de cette solution est que vous pourrez vous contenter d'un seul shift register ("DINX1"), qui suffira pour l'encodeur (2 pins), les boutons de sélection (4 pins), le bouton MENU (1 pin) et le bouton SHIFT (1 pin).

Pour l'écran LCD, vous avez la possibilité d'utiliser un écran LCD 2x40 , pour afficher jusqu'à 10 paramètres simultanément . Un grand écran s'avère trés utile notament pour l'édition du routing dans la matrice MOD ou des wavetables, mais forcément nécessite une façade plus grande. Cette option requiert 10 boutons de sélections au lieu de 5, et donc 16 pins seront alloués par l'encodeur

et les 14 boutons (-> deux shift registers requis).

## Surface de Contrôle minimale pour Multiple SIDs

C'est l'option anciennement appellée "Step B Surface" - quatre boutons "SID" sont ajoutés, permettant de sélectionner le SID (master/slave) qui doit être édité. Ces boutons ont un comportement "exclusif" tant qu'un seul est pressé - pour passer rapidement d'un SID à l'autre. Il est également possible de sélectionner plusieurs SIDs simultanément en maintenant enfoncé un premier bouton SID puis en sélectionnant les suivant - dans ce cas, l'édition d'un paramètre affectera tous les SIDs sélectionnés.

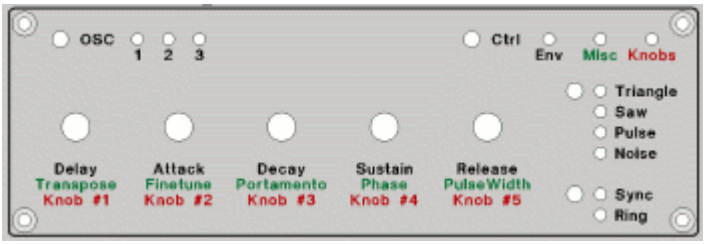



Il est logique d'ajouter des LEDs pour indiquer les SID sélectionnés. Vous aurez besoin alors d'un module DOUT avec un seul shift register (faites le sur une platine d'essai, ou utiliser un module MBHP_DOUTX4). Comme seuls 4 pins se trouvent ainsi alloués, vous pouvez ajouter 3 LEDs supplémentaires pour les boutons SHIFT/UP/DOWN. D'ici peu, la LED à côté du bouton SHIFT clignotera à chaque beat du BPM (pas encore implémenté), la LED "UP" indique si la fonction CC est active, et la LED "DOWN" indique si la fonction EDIT est active.

Le port J5 peut être utilisé comme alternative au module DOUT, en réglant l'option DEFAULT_J5_FUNCTION sur le mode 2 dans le fichier setup_*.asm file - ça peut vous économiser l'assemblage d'un module DOUT, mais vous ne pourrez plus utiliser le port J5 comme entrées analogiques.

## Surface de Contrôle Complète --- section OSC

La surface de contrôle complète est divisée en plusieurs sections; la section OSC contrôle les paramètres spécifiques aux oscillateurs, et vous permet d'accéder aux fonctions "Knob".



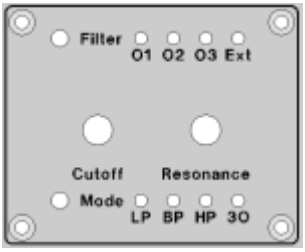

Les 5 encodeurs rotatifs sont mappés sur trois couches (ENV, MISC et KNOBS), sélectionnable via le bouton CTRL. Les boutons OSC vous permettent de sélectionner l'un des trois oscillateurs. Si des paramètres doivent être éditer pour plusieurs oscillateurs simultanément, les flags de sélections correspondant doivent être modifiés via le menu (page la plus à gauche). Il y a un bouton supplémentaire pour la sélection des formes d'ondes, et un autre pour l'activation de la Sync/Ringmodulation . 12 LEDs indiquent l'état des différent paramètres.

Notez qu'un patch consiste en un set de 6 oscillaeurs - 3 pour le canal audio gauche, 3 pour le canal droit. Par défaut, les paramètres affectent les deux canaux simultanément. Si uniquement un des canaux doit être modifié, pressez sur la touche SHIFT et sélectionner l'un des canaux L/R (uniquement valable pour le moteur Lead).

La couche "KNOB" vous donne accés à 5 variations de votre choix, que vous pouvez assigner chacune à deux paramètres au choix du moteur sonore, et utilisables également comme source de modulation. Une course Min/Max peut être spécifiée pour chaque knob séparément. Les assignations et courses sont sauvegardés avec le patch - c'est à dire que chaque Patch peut disposer de sa propre config. Le but de cette fonction est de faciliter le jeu Live, mais peut aussi servir pour réaliser facilement n'importe quel automation (activez la fonction CC afin que vos mouvements soient retranscris via MIDI dans votre logiciel séquenceur).

## Surface de Contrôle Complète --- section FILTRE

La section Filtre vous donne accés aux paramètre CutOff  et Résonnance. Deux boutons vous permettent d'éditer les assignation de canal du filtre, et le type de filtre (Lowpass, Bandpass, Highpass -selections multiple possible). 8 LEDs indiquent l'état des sélections.

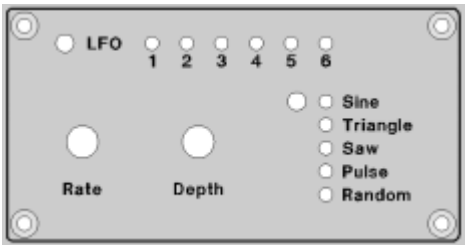



Deux de ces LEDs ne sont pas forcément nécessaire et peuvent être omises pour une nouvelle façade: EXT, qui indique si l'entrée audio externe est routée vers le filtre, et 3O (Oscillator 3 off), qui désactive la sortie audio du troisième oscillator (utile pour des son "sync"). Les deux fonctions ne peuvent être éditées avec les boutons Channel/Filtermode; mais vous disposez deux deux flags dans le menu Filter.

Comme pour la section OSC, la section Filter dispose de deux sets de paramètres - un pour chaque canal audio. Par défaut, les paramètres des deux canaux sont modifiés simultanément. Si un canal en particulier doit être modifié, pressez la touche SHIFT et sélectionnez un des canaux L/R (valable pour tous les moteurs sonores).

## Surface de Contrôle Complète - LFO section

La section LFO vous permet d'éditer les paramètres Depht et Rate des LFOs. Vous pouvez sélectionner le LFO (1-6 pour le mode Lead Engine, 1-2 pour le mode Multi Engine) et la forme d'onde.

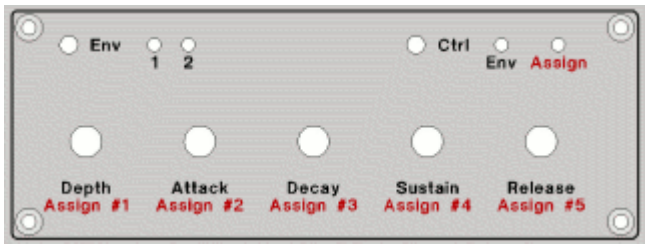



Notez que la MIDIbox SID propose plus de 5 waveforms - les waveforms supplémentaire sont indiqués par des combinaisons de LEDs particulières (pour toutes les formes d'ondes "positive", la

LED Random s'allume en combinaison avec une autre . Par exemple pour une Waveform "Sine Positive",LEDs Random + Sine allumées)

## Surface de Contrôle Complète - section ENV

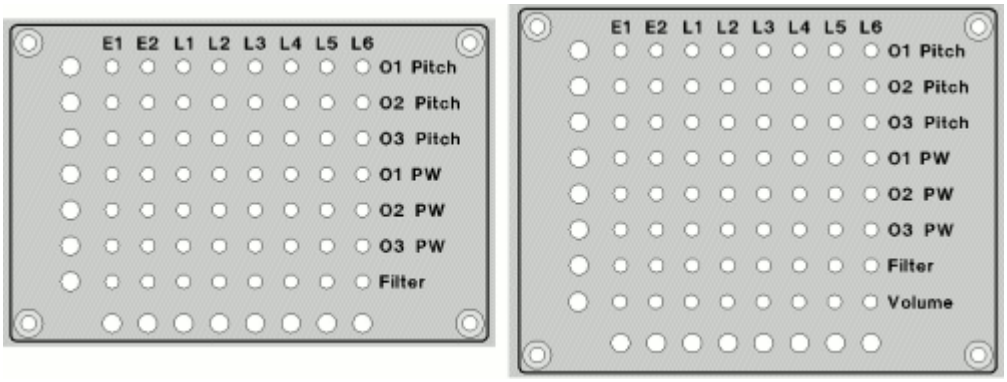
La section ENV permet l'édition directe des paramètres Depth/Attack/Decay/Release des deux enveloppes (Lead engine uniquement, Multi engine: une enveloppe).




Le bouton Ctrl permet de basculer entre l'édition des enveloppes et du layer "assign". Le layer "Assign" peut être adapté selon vos besoin dans le firmware. Par défaut, il sert à piloter les 5 paramètres qui s'affichent à l'écran. C'est une fonction vraiment essentielle et efficace, notament pour accélerer l'édition des wavetables.

## Surface de Contrôle Complète - Matrice de Modulation

Voici probablement la fonctionnalité la plus hallucinante de la MIDIbox SID: La matrice de Modulation, qui permet d'activer trés rapidement les connections source->cible des modulation. La matrice fonctionne comme une patchbay.




Vous disposez de 8 lignes de sources de modulation prédéfinies (ENV1..2, LFO1..LFO6), et 7 (optionellement 8) colonnes de cibles de modulation prédéfinies (OSC Pitch/Pulsewidth, Filter, Volume).

Grâce à une conception trés flexible de la matrice, les sources de modulations peuvent être changées et modifiées via le menu MOD, les marquages "E1..2/L1..6" correspondent donc juste à une config de départ. Lors de la sélection d'un bouton de source, le nom du paramètre réellement concerné ainsi que l'opérateur s'afficheront sur l'écran LCD.

Le mappage des cibles de modulation ne peut être modifié, mais il reste possible d'ajouter des cibles supplémentaires pour des modulations plus complexes (par ex. le contrôle desparamètres depth/rate du LFO ou des sorties CV).

La page Matrice de Modulation propose un set de paramètres supplémentaires pour les canaux gauche et droit. Par défaut, la connection d'un chemin de modulation affecte les deux canaux . Si un seul des canaux doit être affecté, appuyez sur la touche SHIFT et sélectionner l'un des canaux L/R (seulement valable pour le moteur sonore Lead).

Avec la MIDIbox SID V2, le Volume est disponible comme huitième colonnes dans les cibles de modulation. Même si la puce SID ne propose qu'une résolution de 4bit pour ce paramètre, vous pourrez obtenir des effets de trémolo intéressant. Si vous avez prévu une nouvelle façade, vous pouvez donc y ajouter cette huitième colonne de LEDs. Ceux qui utilisent l'ancienne façade peuvent éditer cette assignation via le menu à l'écran.

Notez que la Matrice de modulation n'estr disponible qu'avec le moteur sonore Lead. Le moteur Multi affichera à la placele numéro de l'instrument en cours de jeu. Pour les moteurs Bassline et Drum, il est prévu d'utiliser la matrice pour l'affichage des pas du séquenceur, et/ou comme matrice de sélection des patterns.

Alternativement, la matrice de LEDs peut être utlisée comme "LED-mètre" pour les valeurs de sortie des chemins de modulations. Ce mode peut être sélectionné en pressant deux boutons de source simultanément (par ex. O1 et O2 pitch). Il peut également être activé par défaut dés l'allumage en éditant le flag DEFAULT_LEDMATRIX_MODE dans le fichier setup_*.asm du code source.

## Contrôles Analogiques

Avec la MIDIbox SID V2, les éléments de contrôle analogiques, comme des potars, faders et joysticks sont nativement supportés. Toutes les références que vous pourrez trouver à ce sujet et concernant des extensions de la V1 sont obsolètes, notament parce qu'un pilote spécifique est utilisé à la place de MIOS_AIN pour une optimisation des performances.

Cette fonctionnalité doit être explicitement activée dans le fichier setup_*.asm, en réglant sur 1 le flag DEFAULT_J5_FUNCTION - à partir du moment où ce flag est activé dans le firmware, toutes les entrées analogiques non utilisées doivent être mise à la masse, afin d'éviter des fluctuations de valeurs aléatoires! C'est également requis pour les modules slaves, pour le transfert du firmware maître via ECAN (fonction clonage).

Les 8 entrées analogiques sont échantillonnées à une fréquence de. Le Multiplexage (-> module MBHP_AIN ) n'est pas pris en charge! Donc: 8 entrées au maximum.

Le firmware actuellement n'utilise que les 5 premières entrées du port J5, les 3 restantes sont réservées pour de futures mise à jour.

Les valeurs obtenues aprés conversion sont renvoyées vers le gestionnaire "knob". Cette approche générique permet de bénéficier pleinement du concept de "knobs" : les changement de valeurs peuvent être renvoyés vers deux paramètres simultanément, une course Min/Max peut être spécifiée, et les valeurs samplées sont également disponibles comme source de modulation!

Cette fonctionnalité se comporte différemment sur les master et les slaves:

- **Master:** les entrées analogiques sont renvoyées vers le(s) SID(s) *selectionnés*. Ceci présente l'avantage que chaque SID est pilotable à partir d'un unique set de potars et/ou joysticks. Si cette approche ne vous convient pas, une petit patch pour le firmware (AIN_NotifyChange: supprimez la partie "AIN_NotifyChange_Master") vous permettra de réaliser un set de contrôles dédiés pour le master uniquement.
- **Slave:** les entrées analogiques sont seulement gérées en interne, indépendamment de(s)

SID(s) sélectionné(s).

Notez que cette fonction peut aussi être utilisée pour contrôler le SID à partir d'autres sources analogiques, comme par exemple un step séquenceur ou des LFOs analogiques.

## Boutons/LEDs/Encoders additionnels

Certains utilisateurs ont rajouté à la façade de leur SID des boutons, LEDs et encodeurs supplémentaires. Une petite recherche dans le forumvous permettra de trouver quelques codes et explications. Parmis les modification les plus aboutie vous trouverez l'étonnanteMB-6582 de Wilba's MB-6582. Pleins d'infos sont disponibles dans le Wiki.

Si vous souhaitez simplement réassigner des boutons et/ou des encodeurs à des pins DIN différents, ou des LEDs à différents pins, par exemple pour optimiser l'implémentation de vos circuits accueuillant les éléments de contrôle, vous pourrez le faire sans avoir à apprendre le langage assembleur du PIC :

- **Boutons:** vous trouverez une table dans le fichier setup_*.asm dans laquelle sont assignées les différents boutons de fonctions aux entrées digitales. Cherchez CS_MENU_DIN_TABLE
- **Encodeurs Rotatifs:** vous trouverez une table dans le fichier setup_*.asm dans laquelle sont assignés les différents encodeurs aux entrées digitales. Cherchez MIOS_ENC_PIN_TABLE
- **LEDs:** vous trouverez une table dans le fichier setup_*.asm dans laquelle sont assignées les différentes fonctions LEDs aux sorties digitales. Cherchez CS_MENU_DOUT_TABLE

Pour les experts: de nouveaux boutons de fonctions peuvent être implémentés dans le fichier cs_menu_button.inc, de nouvelles LED dans le fichier cs_menu_leds.inc, de nouveaux encodeurs dans les fichiers cs_menu_enc.inc et cs_menu_enc_table.inc (difficile et peu prudent!)

Voici une liste des différents éléments de contrôle additionnels disponibles dans l'application d'origine, et que vous pouvez activer/désactiver dans le fichier setup_*.asm:

- **M_Vol (Volume Matrix)**: sélectionne la 8ième ligne de la matrice de modulation. Il peut être utilisé en conjonction avec une rangée de LEDs, qui doivent être connectées selon le schéma mbsid_v2_dout_default.pdf
- **Play**: équivalent à "SID1..4 + MENU" (voir liste des combinaisons spéciales)
- **SID_LR**: bascule entre les canaux Left/Right ou les deux canaux audio SID
- **M_Mode**: bascule le mode de la Matrice de LEDs entre affichage normal du routing de la matrice ou le mode Vu-mètre; vous pouvez faire la même chose en pressant simultanément deux boutons MOD SOURCE (voir liste des combinaisons spéciales)
- **Fil_ExtIn**: active/désactive l'envoi vers le filtre du signal présent à l'entrée audio (éditable également via le menu Filtre)
- **Sync**: appelle le menu ENS->CLK (pour modifier le BPM et le mode MIDI Clock)

Les LEDs suivantes sont disponible dans l'application d'origine:

- **Play LED:**: pour être utilisée en conjonction avec le bouton **Play**
- **Meter/Matrix LEDs:**: pour être utilisée en conjonction avec le bouton **M_Mode**
- **SID L/R LEDs:**: pour être utilisée en conjonction avec le bouton **SID_LR**
- **LFO Positive LED:**: extension pour les LEDs "LFO waveform" - s'allume lorsque la forme d'onde du LFO n'est pas bipolaire (normalement, la 5ième LEDs "LFO waveform" s'allume

selon un motif spécial)

## Combinaisons Spéciales de Boutons

Voici la liste des différentes combinaisons de boutons disponibles:

- **MENU lorsque s'affiche le numéro de la version:** transfère le firmware actuel dans tous les modules Slaves (Fonction Clonage). Ceci doit être fait à chaque fois qu'un nouveau firmware a été chargé dans le module Master. Pressez et maintenez enfoncé ce bouton durant l'a mise en route, jusqu'à l'affichage à l'écran d'un message vous indiquant le lancement du clonage. Si le clonage ne se lance pas en suivant cette méthode, pressez plusieurs fois sur ce bouton pendant que s'affiche le numéro de la version à l'écran.
- **SHIFT:** un menu avec des fonctions spéciales s'afffiche. Ces fonctions peuvent être activées/désactivées comme d'habitude avec les boutons SELECT.
- **SHIFT + mouvement d'encodeur:** pour ralentir les encodeurs (édition précise des valeurs)
- **SHIFT + UP:** mode CC on/off. En mode CC, un "knob" (encodeurs dans le menu OSC->Knob layer, ou potars/joysticks analogiques) envoie un message MIDI sur le port MIDI Out, par exemple vers un séquenceur externe. Les valeurs peuvent ainsi être enregistrées en MIDI l'automation.
- **SHIFT + DOWN:** mode EDIT on/off. Dans ce mode, tous les changements de paramètres seront automatiquement enregistrés dans le patch en cours d'éditon si vous changez de patch. Les LEDs clignotent lentement pour vous avertir si des valeurs ont changées, que le précedent patch va être écrasé.
- **SID1..4 + MENU:** joue une note jusqu'à que cette combinaison de bouton soit de nouveaux pressée.
- **DATAWHEEL + MENU:** dans l'écran principal: change d'ensemble (DATAWHEEL sans le bouton MENU: changement de patch)
- **PAGE UP/DOWN + MENU:** dans l'écran principal: change d'ensemble (PAGE UP/DOWN sans le bouton MENU: changement de patch)
- **Boutons MOD TARGET E1..L6:** appelle le menu MOD et sélectionne le chemin de modulation correspondant (1-8) *(moteur Lead uniquement)*
- **SHIFT+MOD TARGET**: bascule vers le menu Envelope/LFO (E1/E2/L1..L6) *(moteur Lead uniquement)*
- **Deux boutons MOD SOURCE simultanément:** bascule le mode de la Matrice de LEDs entre affichage normal du routing de la matrice ou le mode Vu-mètre.

# Installation

## Initial Installation of the Master Core

- **1)** Chargez le bootloader MIOS dans le microcontrôleur avec votre programmateur de PIC (par exemple le MBHP_BURNER). Le fichier .hex est disponible dans le répertoire pic18f4685/burner de l'archive MIOS update.
  Si vous avez acheté votre PIC chez SmashTV ou chez Doc, le PIC est déjà "pré-

programmé", dans ce cas, vous n'avez plus qu'à uploader le code restant via MIDI (pas de programmateur de PIC requis)

- **2)** Installez la dernière version du MIOS à l'aide de MIOS Studio. Le fichier .hex est disponible dans le répertoire MIOS update package (dossier pic18f4685/midi)
- **3)** Installez la dernière version de la midibox_sid_v2 avec MIOS Studio.
- **4)** Si des BankSticks vierges sont connectés au module Core maître, le firmware lancera automatiquement le formatage de ces derniers. L'opération prends quelques seçondes. Aprés le formatage, vous pouvez charger la banque de presets avec votre éditeur SysEx favori (par ex. MIDI-Ox), cette banque est disponible dans le répertoire presets/ de l'archive midibox_sid_v2 .

## Installation initiale des Cores Esclaves

- **1)** Chargez le MIOS Bootloader dans le microcontrôleur si ce n'est déjà fait. Si le logiciel de votre programmateur vous le permet, changez le PIC ID header, utilisez:
  - *0000000000000001* pour le premier module Slave
  - *0000000000000002* pour le second module Slave
  - *0000000000000003* pour le troisième module Slave

  Pas de panique si vous ne trouvez pas cette option dans votre logiciel, vous pourrez changez l'ID plus tard, une fois le MIOS installé.
- **2)** Installez la dernière version du MIOS avec MIOS Studio
- **3)** si l'ID matérielle MIOS n'a pas été adaptée à l'installation du Bootloader, vous devez le faire maintenant. Téléchargez l'application "change_id" disponible sur la page MIOS Download, et uploadez les fichiers .hex suivant avec MIOS Studio:
  - *device_id_01.hex* pour le premier module Slave
  - *device_id_02.hex* pour le second module Slave
  - *device_id_03.hex* pour le troisième module Slave
- **4)** Installez la dernière version de la midibox_sid_v2 avec MIOS Studio.

**IMPORTANT:** une connection MIDI bi-directionnelle est requise pour uploader le code vers les modules Slaves. Normalement le port MIDI Out de votre MIDIbox SID V2 est uniquement connecté au module Core Master. Il y a deux moyens de relier vos modules Slaves à votre interface MIDI :

- **a)** installez temporairement le PIC dans le socle du module Master. C'est la solution la plus simple, notament car vous réalisez ainsi l'installation sur un module déjà testé (si vous avez déjà uploader avec succés le code pour le PIC Master).
- **b)** modifiez le câblage MIDI à l'interieur de votre MIDIbox. Tout dépend de votre implémentation matérielle. Par exemple, sur le circuit MB-6582 vous disposez de jumper vous permettant de modifier le routing des ports MIDI Out.

## Mettre le Firmware à Jour avec une Nouvelle Version

C'est simple: uploadez simplement le dernier firmware midibox_sid_v2 dans le module Master avec MIOS Studio.

Une fois que le Firmware est chargé, et que s'affiche le numéro de version, appuyez et maintenez enfoncé le bouton MENU de votre Surface de Contrôle pour "clôner" le nouveau firmware via l'interface CAN.

# Premières étapes

http://www.ucapps.de/midibox_sid_manual_fs.html

Le but de ce chapitre est de vous donner un oubli grossier sur la gestion de l'interface utilisateur, et les possibilités de modifications rationnelles. Les pages de menu qui sont présentées ici sont décrites en détail dans les chapitres Lead/Multi/Ensemble de ce manuel.

## Mise en correspondance de BankStick

Chaque fois que vous connectez un nouveau BankStick au SID MIDIbox, il sera automatiquement formaté avec 63 (24LC256) ou 127 (24LC512) patchs initiaux. Le premier patch est toujours celui qui est conservé dans la mémoire EEPROM interne. Cette procédure prend quelques secondes et sera informée sur l'écran LCD. En outre, le SID jouera des sons de boulonnement si long est en cours pour notifier que le MBSID n'est pas en mesure de faire quoi que ce soit d'autre cette fois.

```
* Formatting 64k *
* Patch A128     *
```

Le BankStick à la septième place (E0/E1/E2 broche de BankStick connecté à Vd (5V)) sera utilisé pour stocker des ensembles. Il ne s'agit pas d'utiliser un 24LC256 ou 24LC512 ici, dans les deux cas, 128 ensembles seront stockés dans la gamme inférieure de 32k.

```
* Formatting 32k *
* Ensemble 128   *
```

Notez que votre SID MIDIbox fonctionnera également sans BankSticks, mais dans ce cas, un seul patch et un seul ensemble peuvent être stockés dans la mémoire EEPROM interne, qui n'est pas une configuration vraiment utile et donc non recommandée.

## Octeur principal et menus racine

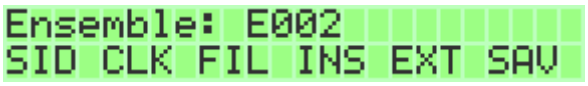
Voici l'écran principal qui est visible après mise sous tension:

```
E002|A001 Ld  Chn. 1
1-**|Lead Patch
```

Il affiche:

- l'ensemble sélectionné (ici: E002)
- les SID sélectionnés et disponibles (ici: 1-z: SID1 sélectionné, SID2 (premier esclave) non sélectionné, SID3 et SID4 (deuxième et troisième esclave): pas de connexion MBNet (c'est au moins le cas si votre MIDIbox SID n'est rempli qu'avec un seul esclave).
- le dispositif transdermique sélectionné de SID1 (ici: A001 - le dispositif transdermique interne)
- le moteur du patch sélectionné (ici: Ld et moteur principal)
- le canal MIDI attribué du premier instrument SID sélectionné (ici: Chn. 1)
- le nom du patch (ici: Lead Patch)

Dans l'écran principal, appuyez sur LE **BOUTON DE SÉLECTION DE LEFTMOST** pour entrer dans le menu racine de l'ensemble.




Ici, vous pouvez changer certains paramètres globaux, comme les affectations de patch/banque, les canaux MIDI, etc...

Retournez à la page principale en appuyant sur le bouton MENU.

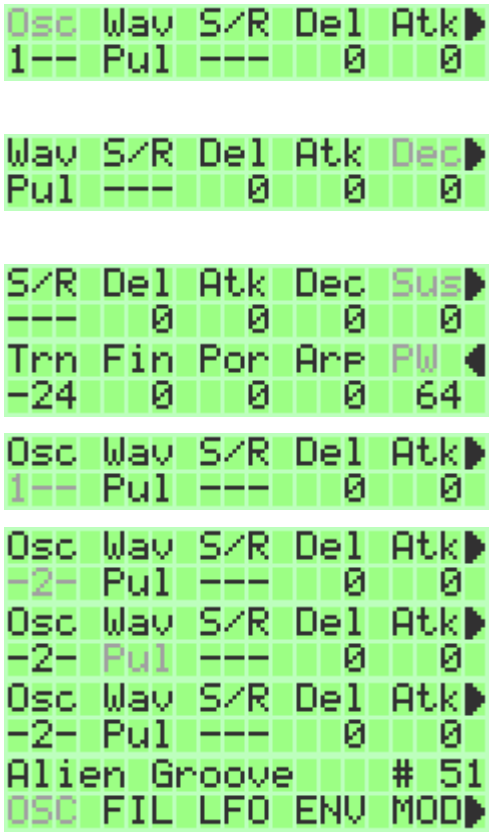
Maintenant, à l'écran principal, appuyez sur l'un des **BUTTES DE DROIT** SÉLECTIONNEMENT (-2, 3 x 4, x 4 ou 5) pour entrer dans le menu racine du moteur de synthèse.




À partir de cette page de menu, vous avez accès à des sous-menus qui contiennent tous les paramètres sonores disponibles.

## Menu navigation

La navigation du menu s'inspire de la fonction de contrôle VST de Logic Control et de la gestion de menu de ma calculatrice HP48 ;-)




Un curseur clignotant (lent) notifie l'élément de menu sélectionné. Ici : OSC
Si l'élément, qui doit être modifié, n'est pas visible à l'écran, vous pouvez parcourir la liste des paramètres avec le codeur rotatif *(boutons Inc/Dec - dans les éléments suivants, il ne sera plus mentionné que le codeur peut être remplacé par deux boutons)*
les paramètres sont décalés à pas à gauche par le codeur...
...jusqu'à ce que la liste des points à l'ordre du jour soit atteinte
Une fois que vous avez trouvé le paramètre, il suffit d'appuyer sur le bouton de sélection approprié pour entrer dans le mode d'édition.

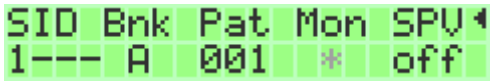
Ajuster le codeur pour changer la valeur.

Vous pouvez appuyer sur un autre bouton de sélection à tout moment pour éditer une autre valeur.
Et vous devez appuyer sur le bouton de menu pour quitter le mode d'édition.
Appuyez à nouveau sur le bouton menu pour passer au menu supérieur suivant.

## Mode stéréo et canal MIDI

Le micrologiciel n'est pas en mesure de déterminer si un ou deux modules MBHP-SID sont connectés au cœur. Si un seul SID est connecté, il est recommandé de sélectionner le mode Mono dans l'ensemble - cela est particulièrement nécessaire si vous utilisez des dispositifs de correction prédéfinis qui sont faits pour l'option stéréo.

Donc, à partir de la page principale, appuyez sur le bouton de sélection la plus à gauche pour entrer dans le menu de l'ensemble. Sélectionnez SID pour entrer dans le menu SID, et sélectionnez le drapeau Mon:




Les utilisateurs de l'option SID stéréo ne doivent pas sélectionner ce drapeau bien sûr pour un son stéréo gras. :-)

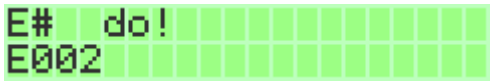
Retournez maintenant au menu de l'ensemble (bouton MENU), et entrez la page INS pour sélectionner la chaîne MIDI. Dans ma propre configuration, SID1 est joué sur la chaîne MIDI 9:




Pour les sons de tête, seul la chaîne MIDI du premier instrument est pertinente. Pour les sons multiples, les canaux des 6 instruments doivent être adaptés (nous essaierons plus tard)

Vous pouvez faire la même chose pour les SID restants (si les esclaves sont connectés). Attribuer différents canaux pour des sons indépendants, ou le même canal pour des sons superposés.

Pour s'assurer que cette configuration est à nouveau disponible après la mise sous tension, l'ensemble doit être stocké dans la MROME interne (E001) ou dans BankStick (E002-E128):




L'utilisation de BankStick est la méthode la plus préférée, car l'EEPROM interne sera écrasée sur chaque mise à jour du micrologiciel. En d'autres termes, E001 n'est conçu que comme solution de secours pour les personnes qui n'ont pas de BankStick spécial prête pour les ensembles.

## L'amusement commence...

Jouons quelques notes sur un clavier (ou d'un séquenceur externe) pour un premier contrôle sonore :

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc1.mp3

Le patch de plomb intial semble pauvre et sans vie, principalement parce qu'il ne joue que sur un seul oscillateur. Faisons-le plus gras en utilisant trois oscillateurs. À partir de la page principale, appuyez sur l'un des boutons de sélection les plus à droite pour entrer dans le menu racine du patch. De là, allez dans le menu OSC et sélectionnez les trois oscillateurs:



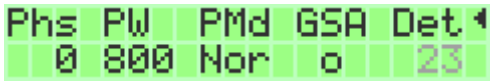

Maintenant, des changements de paramètres seront appliqués sur les trois oscillateurs. Changez la forme d'onde en Pulse (peut également être fait avec le bouton Wave dédié de la surface de contrôle):




Jouez quelques notes - il ne sonne toujours pas gras. À distance - ça semble un peu étrange, parce que tous les oscillateurs sont en synchronisme:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc2.mp3

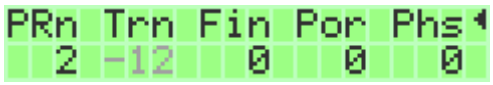
Utilisons une "arme secrète" : le paramètre de détunème. :)




Jouez quelques notes :

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc3.mp3

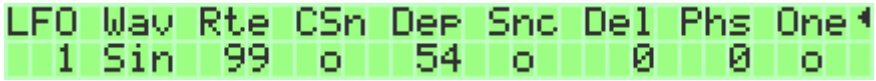
En particulier pour cette mélodie, il est logique de jouer deux oscillateurs avec une octave inférieure pour plus de basse. Donc, changez la valeur de transposition en -12 dans le menu OSC. Faites-le pour OSC2 et OSC3 séparément (remarque: vous pouvez rapidement passer d'un oscillateur à l'autre avec le bouton Page vers le haut/détour):

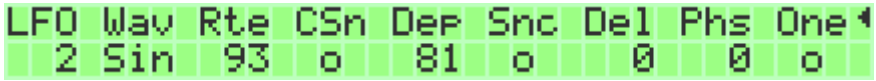


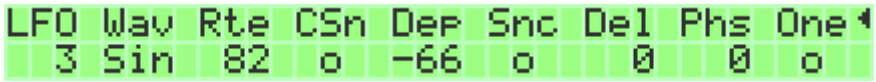

Jouez des notes - toujours ennuyeuses, hm ?

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc4.mp3

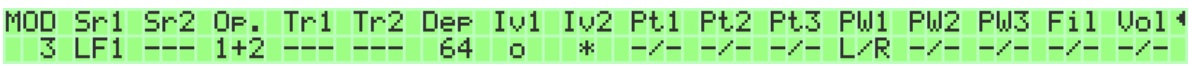
Mais il y a des mesures supplémentaires pour la grossité ultimative: l'utilisation de LFO dérégété pour moduler le pulsewitch. Donc - changer dans le menu LFO et entrer différentes valeurs de profondeur/débit pour LFO1, 2 et 3:

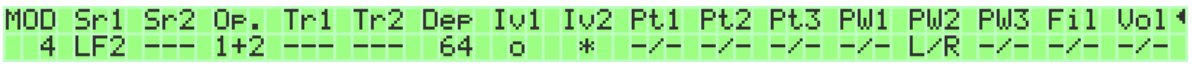


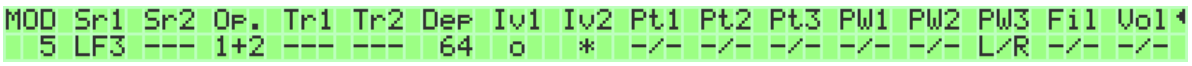




Changez maintenant dans le menu MOD pour configurer les affectations source/cible:




Il convient de noter que :

- nous utilisons les chemins MOD 3, 4 et 5, puisqu'ils sont déjà préparés pour LFO1, LFO2 et LFO3 (missions Src1 en défaut)
- Nous avons défini le drapeau d'inversion In2 pour ces MOD, de sorte que le canal audio de droite sera modulé inverse sur le canal de gauche - cela garantit un son encore plus gras. :-)
- nous utilisons les connexions de modulation directe pour PW1 (Pulsewth of OSC1), PW2 et PW3, affectées à la voie gauche/droite. Chaque oscillateur obtient son propre LFO
- ces connexions de chemin de modulation peuvent également être commandées à partir de la matrice de modulation de la gouverne. Par example, appuyez sur le bouton de la quatrième ligne et le bouton de la troisième colonne à la même longueur d'onde afin d'activer/désactiver la connexion MOD3-PW1.

Le son qui en résulte:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc5.mp3

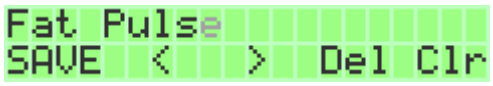
Et ici le mélange final, où j'ai ajouté un retard stéréo à la piste SID - quelques tambours:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_osc5_fx.mp3

Ok, stockons ce nouveau patch avant qu'il ne se perde. Changez la page SAV et sélectionnez un créneau de dispositif de correction libre, par ex. A002 :




Appuyez sur le troisième bouton de sélection (en dessous de "do'") et donnez au dispositif un nom plus ou moins significatif:




Enfin, appuyez sur SAVE pour stocker le dispositif transdermique.

Maintenant, vous pouvez aussi sélectionner le même nouveau patch pour les autres cœurs SID. Retournez à la page principale, sélectionnez SID2, changez de patch avec le codeur principal, sélectionnez SID3, changez de patch avec le codeur principal, sélectionnez SID4, changez de dispositif transdermique avec le codeur principal. S'assurer que tous les SID sont affectés au même canal MIDI. Abaissez le volume de votre ampli Jouez une seule note. Ouah. Respirez profondément et jouez une autre note, cette fois peut-être un peu plus longtemps. Jouissez (désolé, pas d'échantillon disponible, car le son des 24 oscillateurs SID est à massif pour le format MP3 ;-)

## Expériences de filtres

Chaque SID fournit un filtre multi-états analogique de 12db, que nous voulons étudier maintenant. Changez dans le menu du filtre et activez le filtre pour les trois oscillateurs:



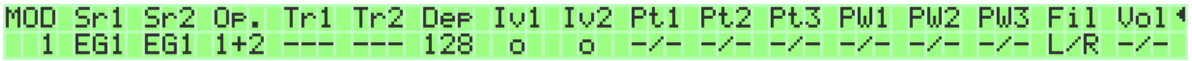

commuter le type de filtre en L (Lowpass), et changer de résonance et CutOff (à partir de ce menu, ou avec les codeurs/boutons rotatifs dédiés de la surface de commande):




Avec le patch modulé PW précédemment créé, il devrait sonner comme suit:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_fil1.mp3

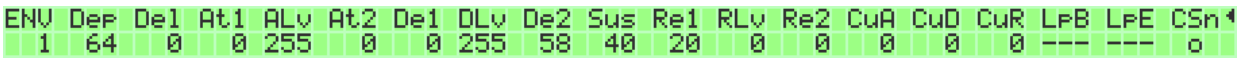
Très statique, non ? Donc, modulons la fréquence de coupure du filtre avec une enveloppe. Changez dans le menu MOD et configurez le chemin MOD1 comme indiqué ci-dessous:




- ENV1 est utilisé deux fois comme source de modulation - l'opérateur est 1x2, ce qui signifie: la modulation ENV sera doublée
- la profondeur est fixée à 128 pour la modulation maximale.
- pas d'inversion

- L'objectif de fil est attribué pour le canal gauche/droite

Changez le menu ENV et changez les paramètres en jouant certaines notes:



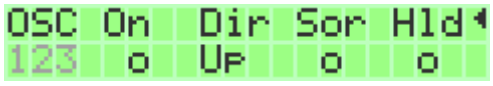

La fréquence de coupure doit être réglée au minimum (0), de sorte que l'enveloppe puisse contrôler l'ensemble de la plage de 12 bits. Exemple de son pour différents types de filtres (Lowpass, Bandpass, Low/Band Pass, Highpass:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_fil2.mp3

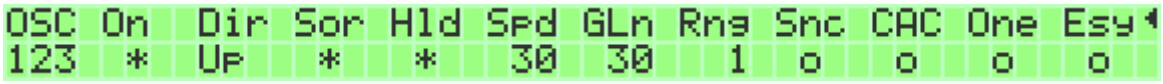
La distorsion du filtre qui se produit lorsque les trois oscillateurs jouent est l'une des principales caractéristiques de la puce SID qui le rend si unique. Aucune émulation ne peut jamais reproduire ce son original.

## Arpéggiateur

Le moteur de plomb fournit 6 arpégigateurs indépendants, mais pour les premières expériences, je reçois à les utiliser avec des réglages identiques pour éviter les effets chaotiques. Donc, changez dans le menu arp et sélectionnez tous les oscillateurs:



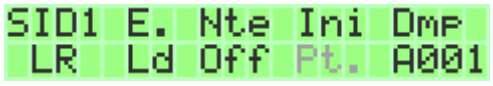

Maintenant, allumez arp (cliquez sur le bouton ci-dessous "On") et essayez tous les paramètres. Voici un example de montage:




## Modulation

Maintenant quelques mots sur l'une des principales caractéristiques de MIDIbox SID V2: la matrice de modulation basée sur l'opérateur. Nous avons déjà fait des expériences de modulation en utilisant des LFO et des Enveloppes, mais c'est juste des choses classiques qui sont disponibles dans la plupart des synthétiseurs. Le concept de modulation de MBSID V2 permet d'appliquer une opération sur deux sources de modulation avant que la forme d'onde résultante ne soit transmise aux cibles de modulation. Cette approche puissante est essentiellement inspirée des synthétiseurs Waldorf (concept de modèle).

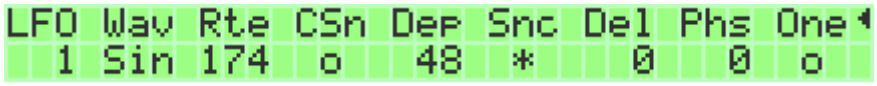
Nous allons essayer cela sur un seul oscillateur. Tout d'abord, le dispositif transdermique doit être initialisé pour obtenir un démarrage défini. Appuyez sur la touche "Salut" (anciennement LIEN) et appuyez sur le bouton de sélection sous "Ini" pour obtenir le patch par défaut:



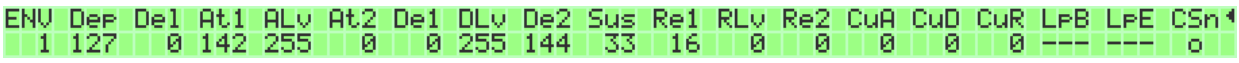

Maintenant, activez le filtre pour tous les oscillateurs, fixez la coupure à 400 et la résonance à 15:



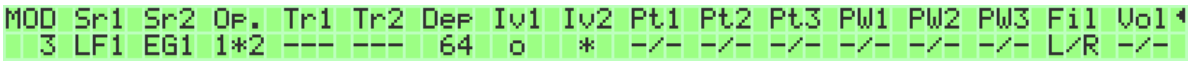

LFO1 doit être mis à des valeurs suivantes (veiller à ce que le drapeau de synchronisation soit fixé, de sorte que le LFO redémarre sur une nouvelle note):




Configurer ENV1 comme indiqué ci-dessous:




Et enfin la configuration de la matrice MOD - définir les valeurs étape par étape afin d'entendre les changements:




C'est le résultat lorsque LFO1 est joué seul (Op-Sr1):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_mod1_1.mp3

Ici, ENV1 est joué seul (Op-Sr2):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_mod1_2.mp3

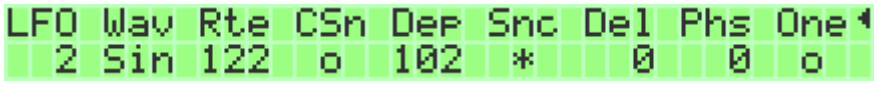
Et ici, LFO1 est multiplié par ENV1 (Op-1-2), c'est-à-dire: l'enveloppe contrôle la profondeur de LFO1:

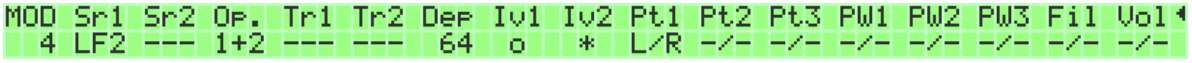
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_mod1_3.mp3

Enfin, le drapeau d'inversion pour le canal audio de droite est défini - il n'a jamais été aussi facile de créer de beaux effets stéréo :-)

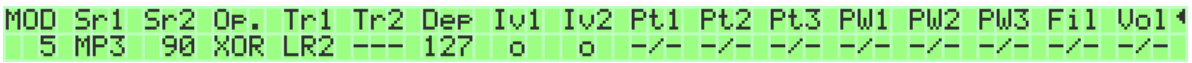
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_mod1_4.mp3

Maintenant, essayons une expérience inhabituelle: modulons le pas de l'oscillateur avec LFO2:




Et le débit LFO2 avec la même forme d'onde, qui contrôle déjà le filtre (MP3-output of modulation path 3), combiné via XOR avec la valeur constante 90:




Sons en résultats:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_mod1_5.mp3

*Avertissement: essayez des valeurs constantes différentes - vous remarquerez des changements spectaculaires De telles combinaisons binaires peuvent vous occuper pendant des heures. :-)*

Il existe des centaines d'autres possibilités d'utilisation de la matrice de modulation, notamment avec les opérateurs binaires. Mais les illustrer est très long, je vais donc fournir un tas d'exemples dans une banque spéciale prédéfinie.

## Sequenceur onduleur

Le séquenceur Wavetable vous permet d'échelonner une liste de valeurs afin de contrôler les paramètres sonores. C'est une méthode de modulation alternative qui peut être utilisée pour générer des formes d'onde personnalisées, mais aussi pour jouer des notes (acharnement

amélioré). 4 pistes (lebit) ou 6 (multi) séquenceurs sont disponibles, qui peuvent être cadencées indépendamment les unes des autres.

Plus de détails sur le séquenceur WT peuvent être trouvés dans le chapitre du moteur principal. Ici, nous ne voulons qu'obtenir de la pratique, alors commençons sans les connaissances de fond théoriques.

Initialisation du dispositif transdermique pour obtenir un démarrage défini. Appuyez sur la touche "Salut" (anciennement LIEN) et appuyez sur le bouton de sélection sous "Ini" pour obtenir le patch par défaut:

```
SID1 E. Nte Ini Dmp
 LR  Ld Off Pt. A001
```

Sélectionner tous les oscillateurs, entrer la forme d'onde de pouls et changer le paramètre de détunème:

```
Phs PW  PMd GSA Det
  0 800 Nor  o   23
```

commuter le type de filtre en L (Lowpass), et changer de résonance et CutOff (à partir de ce menu, ou avec les codeurs/boutons rotatifs dédiés de la surface de commande):

```
Chn Cut Res Mod Ext
123 100  15 L-- o
```

Changez la page WTC et entrez la configuration suivante:

```
 WT Par SID Spd MOD Beg End Lop Val WTE
  1   4 L/R  30  o   00  05  00 Hex Go!
```

Cela jouera la première piste de la vitesse 30 à la vitesse 30. Il sera rebouclé (sans fin) redémarrant à '00. L'affectation au paramètre 4 signifie Fréquence CutOff. Le nom du paramètre sera affiché à l'écran pendant que vous le changez. N'hésitez pas à essayer différents paramètres plus tard.

Changez maintenant la page WTE et éditez la séquence pour la piste 1, qui commence à -00:

```
Pos $00 $20 $40 $60 Val WTC
 00  18<+00 +00 +00 Hex Go!
```

Saisissez la séquence suivante à la première colonne de la piste, en commençant par la position 0 (utiliser le bouton Page en haut/en bas pour changer la position):

$00 = 18
$01 = 60
$02 = 30
$03 = 60
$04 = 40
$05 = 30

Séquence résultante:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_wt1.mp3

Maintenant, créons un bel effet stéréo: changez WT1, de sorte qu'il ne contrôle que le canal audio gauche:

```
 WT Par SID Spd MOD Beg End Lop Val WTE
  1   4 L/- 30  o   00  05  00 Hex Go!
```

Et entrez les valeurs suivantes dans la configuration WT2, de sorte qu'il contrôle le canal audio droit et joue le même table d'ondes avec un offset:

```
 WT Par SID Spd MOD Beg End Lop Val WTE
  2   4 -/R  30  o   03  05  00 Hex Go!
```

Séquence résultante:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_wt2.mp3

Maintenant, nous ajoutons une piste WT qui joue des notes:

```
 WT Par SID Spd MOD Beg End Lop Val WTE
  3 252 L/R  30  o   40  43  40 Hex Go!
```

Dans l'Adité de l'éditeur de la voûte (WTE), entrez la séquence suivante à la troisième colonne:

```
Pos $00 $03 $40 $60 Val WTC
 00  18< 60 F-0<+00 Hex Go!
```

$40 = F-0
$41 = +++
$42 = F-1
$43 = +++

*(Note: vous pouvez également entrer Ky0/Ky1 au lieu de F-0/F-1, de sorte que les deux notes changent avec l'accord joué)*

Séquence résultante:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_wt3.mp3

## Déclenchement

La matrice de déclenchement permet de réagir sur les événements, comme Note On/Off, LFO overruns, ENV ou l'horloge globale. Pour une brève introduction, il devrait être suffisant pour améliorer le patch précédemment créé par une connexion de déclenchement entre l'échelon L1P (L1P) et l'échelon WT (W1S.W4S):

```
TRG O1L O2L O3L O1R O2R O3R E1A E2A E1R E2R L1  L2  L3  L4  L5  L6  W1R W2R W3R W4R W1S W2S W3S W4S
L1P  o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   *   *   *   *
```

Comme le WT est normalement déclenché par le générateur d'horlogerie global, cette connexion doit être désactivée:

```
TRG O1L O2L O3L O1R O2R O3R E1A E2A E1R E2R L1  L2  L3  L4  L5  L6  W1R W2R W3R W4R W1S W2S W3S W4S
Clk  o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o   o
```

Enfin, la vitesse WT des **trois pistes utilisées WT1..WT3** doit être réglée sur 1, de manière à ce que l'événement L1P passe la table d'ondes sans prédiviseur:

```
 WT Par SID Spd MOD Beg End Lop Val WTE
  3 252 L/R   1  o   40  43  40 Hex Go!
```

Lorsque LFO1 commande en plus le pulsewth des trois oscillateurs, il en résulte un son suivant:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_wt4.mp3

Il est à noter que maintenant la vitesse WT est directement contrôlée par LFO1. Cela signifie que la modulation provoquée par cette LFO est toujours en synchronisme avec la modulation WT.

Nous pouvons le prouver en modulant le débit LFO1 et le pas de tous les oscillateurs avec ENV1 - profitez du résultat. :-)

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_fs_wt5.mp3

# Lead Engine

## Main Screen

This is the main screen which is visible after startup. You can always go back to this screen by pressing the MENU button one or two times (depending on the current hierarchy level).

```
E002|B036 Ld  Chn. 9
1---|C64 Bass
```

The screen shows the activated SID (corresponds to SID1/2/3/4 LED) and Ensemble (E001..E128), the bank and patch number, the MIDI channel, and the patch name of the selected SID.

By moving the datawheel, the patch can be changed within the current bank. The bank can only be changed within the ensemble menu.

.

## Root Menu

Within the main screen, press one of the **RIGHT SELECT BUTTONS** (#2, #3, #4 or #5) to enter the root menu. The leftmost select button will enter the ensemble menu, which is described here.

```
Lead Patch      A001
OSC FIL LFO ENV MOD TRG CFG ARP WTC WTE KNB EXT RND SAV
```

It gives you quick access the submenus, which are described below.

## "Shift" Menu

By pressing the SHIFT (formerly LINK) button, a special menu will be displayed which gives you quick access to following functions:

```
SID1 E. Nte Ini Dmp
 LR  Ld Off Pt. A001
```

- **SIDn:** toggles between LR, L- and -R. Here you can specify, if OSC, FIL and MOD parameter modifications should affect only the left, the right, or both SID channels. The SID number (SID1, SID2, SID3, SID4) has to be selected with the deticated SID buttons.
- **Engine:** currently only Ld for Lead. Later you can select between Lead/Multi/Bassline/Drums

- **Note Off:** disables the notes of all selected SIDs
- **Init:** initializes a patch depending on the selected engine.
- **Dump:** sends a SysEx dump of the currently selected patch.

## OSCillator menu

Within the oscillator menu you can control the parameters of the 2*3 SID oscillators. Note that by default changes will take place for the left and right channel. By selecting a single channel within the "Shift" menu, you can do modifications for the six oscillators separately.




- **OSC (Oscillator):** selects a single oscillator (1--, -2-, --3) or multiple oscillators (e.g. -23 or 123)
- **Wav (Waveform):** selects the waveform:
  - **Tri:** Triangle
  - **Saw:** Sawtooth
  - **T+S:** Triangle + Sawtooth (mixed)
  - **Pul:** Pulse
  - **P+T:** Pulse + Triangle (mixed)
  - **P+S:** Pulse + Sawtooth (mixed)
  - **PST:** Pulse + Sawtooth + Triangle (mixed)
  - **Noi:** Noise

  In following audio example, the different waveforms are played from a 8580:

  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_waveforms_8580.mp3

And here the same sequence played on a 6581 - you will notice, that some mixed waveforms don't work on this older SID. And note especially the background noise, which is produced by a leackage issue in the VCA/mixer section:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_waveforms_6581.mp3

- **S/R:** activates Sync and/or Ringmodulation.
- The **sync** flag synchronises the fundamental frequency of a slave oscillator with the fundamental frequency of a master oscillator, producing a "hard sync" effect. Varying the frequency of the slave oscillator with respect to the master prodcues a wide range of complex harmonic structures from the slave oscillator at the frequency of the master oscillator. In order for sync to occur, the master oscillator should be set to a frequency lower than the frequency of the slave.
  The sync assignments are hardwired within the SID:
  OSC1 can be synched by OSC3
  OSC2 can be synched by OSC1
  OSC3 can be synched by OSC2
  An audio sample where OSC3 syncs OSC1. The frequency of OSC3 is increased by an envelope. OSC3 itself is silent (3Of in FIL menu selected):

  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_sync.mp3

- The **ringmodulation** flag works only when a **triangle** waveform is selected!
  It multiplies the output of one oscillator with the output of another, which results into non-harmonic overtone structures for creating bell/gong sounds and special "metallic" effects.
  The ringmodulation assignments are hardwired within the SID:
  OSC1 can be modulated with OSC3
  OSC2 can be modulated with OSC1
  OSC3 can be modulated with OSC2
  An audio sample where OSC3 is ringmodulated with OSC1. The OSC1 frequency is slowly increased.:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_ring.mp3

And here a sample where the Sync and Ring flag are activated the same time.
Don't cry baby, MBSID V2 will be released soon!

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_syncring.mp3

- 
- **Delay/Attack/Decay/Sustain/Release:** controls the VCA envelope of the SID oscillator.
- **PRn (Pitchrange)** the semitone range within Pitchbender and Finetune can increase/decrease the frequency.
- **Trn (Transpose):** the semitone transpose value (-64..+63). +12 will transpose by one +1 octave, whereas -12 will transpose by -1 octave.
  In following audio sample the sequence begins with transpose=0, after a while it will be decreased to -12 (-1 octave), and later to -24 (-2 octaves):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_transp.mp3

- **Finetune:** allows to slightly decrease/increase the frequency within the given pitchrange. Use this parameter on different oscillators to make the resulting sound more fat! Note that there is an additional Detune parameter at the end of this page which can do this automatically for all 6 oscillators.
  Demonstration of using finetune on three oscillators (see also the detune example below):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_finetune.mp3

- **Por (Portamento)** sets the intensity of the portamento (glide) effect. When changing from one to another note, this effect will continously sweep the frequency until the target frequency has been reached. There are three different portamento modes available; they can be selected with the **PMd** item:

- **Normal:** "normal" portamento, also known as "logarithmic portamento". The sweep time depends on the frequency.
- **CTG:** linear constant time glide - the sweep time is always the same, independent from the frequency.
- **Gls:** glissando: the frequency is sweeped in semitone steps, like if you would strum over the keyboard from one to another note.

Demonstration of the three portamento modes:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_porta.mp3

- **Phs (Oscillator Phase Offset):** this function allows you to synchronize the phases of all three oscillators.

With Phs=0 they are freerunning - this is the preferred option for "analog" sounds. With Phs=1, they will be started at the same moment, whereas with Phs>1 the phase offset between the oscillators will be controlled in 0.4% steps (!). In following audio sample the oscillator phase offset between three pulse waveforms is slowly changed from 0..33% - once this value is reached, the resulting frequency is 3 times higher than the base frequency:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_phaseoffset.mp3

- 
- **PW (Pulsewidth):** this 12bit value (range 000-FFF, displayed in hexadecimal format) controls the duty cycle of a pulse waveform. This parameter has no effect on other waveforms. A 1:1 duty cycle is set with value 0x800.
  In this sample each oscillator got an own LFO for pulsewidth modulation. The modulation values for the right channel are inverted - this results into a nice stereo effect:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_pw.mp3

- **PMd (Portamento Mode):** see description above.
- **GSA (Gate Stays Active):** once activated, the VCA envelope won't be released anymore, resulting into a permanent sound. This allows you to control the sound volume completely with the internal, or an external filter, or with an external VCA. It's an option which allows you to overcome the infamous ADSR bug of the SID.
- **Det (Detune):** detunes all oscillators by increasing/decreasing the finetune parameter. This allows you to create fat sounds with a single encoder turn! (therefore the same value is visible in all OSC pages)

- Left SID channel: OSC1 +detune/4, OSC2 +detune, OSC3 -detune
- Right SID channel: OSC1 -detune/4, OSC2 -detune, OSC3 +detune

Demonstration of using detune in legacy mode on six oscillators (stereo - compare with finetune example above):

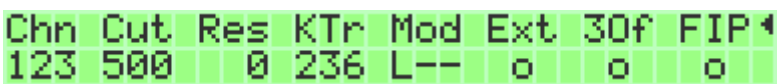
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_osc_detune.mp3

- **DtM (Detune Mode):** switches between normal (legacy) and SuperSaw +/- mode, which has been created by Lis0r based on the research from this paper
  In SuperSaw mode the calculation changes to:
  - Left SID channel: OSC1 no detune, OSC2 +/- detune/3, OSC3 -/+ 2*detune/3
  - Right SID channel: OSC1 no detune, OSC2 -/+ detune/3, OSC3 +/- 2*detune/3

  It especially works much better for higher detune values!
  *TODO: add demo for the amazing SuperSaw mode!*

## FILter menu

Within the filter menu you can control the parameters of the SID filter. Each SID has only one 12db multistate filter, which can be assigned to the three oscillators and/or to the external audio input. Note that by default changes will take place for the left and right channel. By selecting a single channel within the "Shift" menu, you can do modifications for the two filters separately.



- **Chn (Channel):** assigns the filter to oscillator channel 1/2/3
- **Cut (CutOff):** sets the cutoff frequency (12bit value) from 000 to FFF (hexadecimal format).
- **Res (Resonance):** is a peaking effect which emphasizes frequency components at the CutOff frequency of the filter, causing a sharper sound. The intensity of the resonance can be set from 0-15.
- **Mod (Filter Mode):** sets the filter mode - note that mixed notes like Lowpass/Highpass are possible, which results into a notch effect (inverted bandpass).
    - **L (Lowpass):** all frequency components below the Cutoff frequency are passed unaltered, while all frequency components above the Cutoff are attenuated at a rate of appr. 12 dB/Octave
    - **B (Bandpass):** all frequency components above and below the Cutoff frequency are attenuated at a rate of appr. 6 dB/Octave
    - **H (Highpass):** all frequency components above the Cutoff frequency are passed unaltered, while all frequency components below the Cutoff are attenuated at a rate of appr. 12 dB/Octave
- **KTr (Key Tracking):** increases the CutOff depending on the played note based on the keytracking factor (0..255). The usage is especially important on high resonance filter sounds, as it keeps the resonant frequency spectrum "on-track" with the note.
- **Ext (External):** external Audio IN is passed to the filter
- **3Of (3rd oscillator off):** a special SID function which disconnects the oscillator 3 output from the mixer. This allows to use OSC3 for sync/ringmodulation without any undesirable output.
- **FIP (Filter Interpolation):** this option smoothes CutOff changes. It is especially useful when CutOff is changed via CC (-> low resolution). to reduce the steppiness. The option is not useful when the CutOff parameter is modulated rom a fast LFO or envelope (on the other hand: it produces interesting effects in such cases ;-)

Here a quick demonstration of the SID filters. In the first example two 6581 from calendar week 40/84 and 25/84 are used, which sound very different and which are therefore not really qualified for stereo patches. On the other hand they sound warm and analog - but note especially the background noise at the end, a typical issue of the 6581!

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_filter_6581.mp3

In the second example the same sequence is played on two 8580R5 from the same batch:

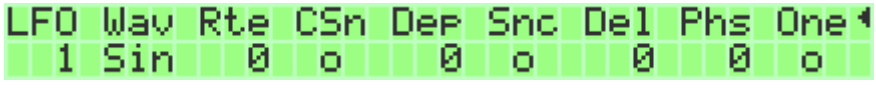
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_filter_8580.mp3

Decide for yourself which one sounds better!

*To be honest:* ***I want them all! A MBSID V2 can be stuffed with 8 SIDs, so what? :-)***

## LFO menu

The lead engine provides six low frequency oscillators which can be assigned to different modulation targets within the modulation matrix.

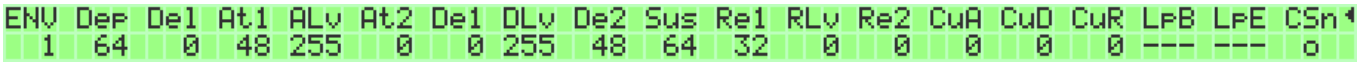



- **LFO:** selects one of 6 LFOs.

- **Wav (Waveform):** selects the shape of the LFO. Following waveforms are provided: Sine, Triangle, Pulse, Ramp (Sawtooth), Random (Sample&Hold), and positive Sine/Triangle/Pulse/Ramp
- **Rte (Rate):** the speed of the LFO can be adjusted from 0.008 Hz (1) to ca. 45 Hz (255).
- **CSn (Clock Synchronisation):** allows you to synchronize the LFO speed to the global clock generator. This is especially useful when a clock is provided from a MIDI clock master. If CSn is active, the rate will select special presets from 16 bars down to 1/32th notes.
- **Dep (Depth):** the intensity of the LFO effect. It can be controlled from -128 to +127. A negative value inverts the LFO waveform.
- **Snc (Sync):** the LFO will be restarted with each new note.
- **Del (Delay):** works only in conjunction with the Sync flag. With a delay>0, the LFO will be held inactive before it will be restarted.
- **Phs (Phase):** works only in conjunction with the Sync flag. Specifies the phase (0-255) at which the LFO will be restarted. With 64, the LFO will start at 25%, with 128, the LFO will start at the middle.
- **One (Oneshot):** only one period of the LFO waveform will be processed, thereafter the LFO is stopped and has to be retriggered from a trigger matrix source (see below).
  Note that Oneshot works only when LFO Sync is enabled as well.

## ENVelope menu

The lead engine provides two additional envelopes which can be assigned to different modulation targets within the modulation matrix.

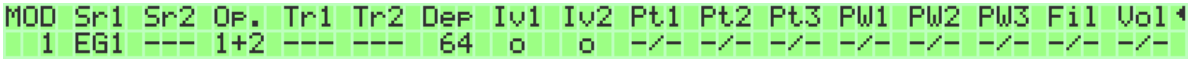



- **ENV:** selects one of 2 ENVs.
- **Dep (Depth):** the intensity of the envelope effect. It can be controlled from -128 to +127. A negative value inverts the ENV waveform.
- **Del (Delay):** delay the attack phase of the envelope.
- **At1/ALv/At2 (Attack Rate/Level):** sets the attack rate and attack level between the two attack phases.
- **De1/DLv/De2 (Decay Rate/Level):** sets the decay rate and decay level between the two decay phases.
- **Sus (Sustain):** sets the sustain level.
- **Re1/RLv/Re2 (Release Rate/Level):** sets the release rate and release level between the two release phases.
- **CuA/CuD/CuR (Curve Attack/Decay/Release):** with curve=0, the rates are linear, whereas with negative values (-128..-1) the curve of the appr. envelope phase can be bended down, and with 1..127 it can be bended up.
  This is a simple way to imitate the exponential charging/decharging curve of a capacitor, so that the envelope behaves like an analog circuit. But note that the relation between the rate and the curve parameter is not linear (than higher the curve intensity, than shorter the waveform), and it needs some practice to achieve good results.
- **LpB/LpE (Loop Begin/End):** allows to cycle between two phases. Once the phase specified with LpE has been finished, the envelope generator will jump back to the phase specified with LpB. Not all combinations make sense.

- **CSn (Clock Synchronisation):** allows to synchronise the envelope rates to the global clock generator. This especially makes sense when the generator is clocked from an external MIDI master.

## MODulation matrix menu

The modulation matrix provides 8 independent pathes. Within each path, two sources are combined by an operator. A depth parameter is applied before the result is distributed to multiple targets. This powerful approach is basically inspired from Waldorf synthesizers (Modifier concept).




- **MOD (modulation path):** selects one of the eight modulation paths, which should be displayed.
- **Sr1 and Sr2 (Source #1 and #2):** the predefinitions are matching with the MIDIbox SID1 assignments, but you can change them as desired.
  Following sources are available:
  - **---:** disable source
  - **EG1, EG2:** Envelope 1 and 2
  - **LFO..6:** LFO 1-6
  - **MP1..8:** the modulation result of path 1-8 is available as source as well. This allows you to chain multiple operations on different sources!
  - **MdW:** last received modulation wheel position
  - **Key:** last received note value
  - **K#1..5:** position of Knob 1-5 (see Knob menu)
  - **L#V:** last received Velocity
  - **L#P:** last received Pitchbender position
  - **L#A:** last received Aftertouch value
  - **WT1..4:** value provided from wavtable 1-4 (see wavetable concept below)
  - **0..127:** a constant value - especially useful in conjunction with the binary or Min/Max operators.

- **Op (Operation):** the operation which is applied on the two sources:
  - **---:** disable operation (no modulation)
  - **Sr1:** Source 1 only
  - **Sr2:** Source 2 only
  - **1+2:** Source 1 added to Source 2
  - **1-2:** Source 2 substracted from Source 1
  - **1*2:** Source 1 multiplied with Source 2
  - **XOR:** binary XOR operation between Source 1 and 2
  - **OR:** binary OR operation between Source 1 and 2
  - **AND:** binary AND operation between Source 1 and 2
  - **MIN:** forwards either Source 1 or 2, the minimum value will be taken
  - **MAX:** forwards either Source 1 or 2, the maximum value will be taken
  - **1<2:** generates a positive pulse when Source 1 is less than Source 2
  - **1>2:** generates a positive pulse when Source 1 is greater than Source 2
  - **1=2:** generates a positive pulse when Source 1 is equal to Source 2

    - **S&H:** Sample & Hold - samples the value of Source 1 whenever Source 2 changes from a negative to a positive value. The direct integration into the MOD matrix is an unusual, but very powerful solution, as in difference to common synths it not only allows you to capture values periodically (e.g. by using a LFO), but with any pattern. E.g., you could trigger the capturing from the wavetable sequencer, or with a certain modwheel or velocity level (requires a second MOD path which is doing a comparison 1>2)

- **Tr1 and Tr2 (Target 1 and 2):** following modulation targets are available:
    - **Pt1..6:** Oscillator pitch. OSC1-3 are the oscillators of the left channel SID, OSC4-6 the right channel SID
    - **PW1..6:** Pulsewidth. OSC1-3 are the oscillators of the left channel SID, OSC4-6 the right channel SID
    - **Fl1 and Fl2:** CutOff frequency of left/right SID filter
    - **Vl1 and Vl2:** Volume of left/right SID
    - **LD..6:** LFO depths
    - **LR1..6:** LFO rates
    - **Ex1..8:** the external control voltage (CV) channels
    - **WT1..4:** the wavetable positions of WT1..4 (MOD flag must be enabled in WTC menu)

- **Dep (Depth):** allows to control the intensity of the modulation result from -128..127. A negative value will invert the result
- **In1 and In2 (Inversion):** an additional possibility to invert the result, but for Target 1 and 2, resp. for left and right SID channel separately. Inverting one channel mostly leads to very interesting stereo effects!
- **Pt1..Vol**: 2*8 additional modulation targets are available for each path which can be enabled/disabled in this menu, and with the LED modulation matrix of the control surface. They are statically assigned to OSC 1..3 Pitch, Pulsewidth, to Filter CutOff and Volume. The left and right channel can be enabled separately. Remind the inversion function for nice stereo effects!

Demos:

- http://www.ucapps.de/mp3/midibox_sid/mbsidv2_demo1.mp3
  2*3 slightly detuned pulse waveforms.
  The pitch is modulated by a LFO (saw waveform), combined via AND operator with a constant value. The constant value is manually changed with a knob. The modulation result is sometimes crazy, sometimes surprising, sometimes just exactly what we expect from an 8bit synth :)

- http://www.ucapps.de/mp3/midibox_sid/mbsidv2_demo2.mp3
  2*3 filtered mixed waveforms (pulse/saw/triangle).
  The filter is modulated by one LFO, the volume by another LFO. The depth of volume modulation is 0 at the beginning, and changed to +127 at 0:07
  The amount of volume modulation is inverted between left and right channel.
  Rate of LFO which modulates volume is changed at 0:13
  A third modulation path is used at 0:26, where LFO1 and LFO2 rate is controlled by an enveloppe. Who needs a Leslie? :)

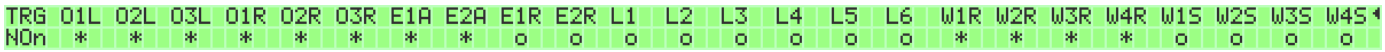
- See random example at the bottom of this page for some randomly selected combinations of modulation sources/targets/operations

## TRiGger matrix menu

Commonly known from modular synthesizers, it's possible to trigger functions of MIDIbox SID from various, predefined sources, which can result into very lively effects.

A trigger you propably already know is the envelope gate, which is activated on a Note On event. Why not restarting the envelope when a LFO period has passed, when a (divided) MIDI clock has been received, or when a second envelope has reached the sustain phase? The trigger matrix allows you to combine such events.

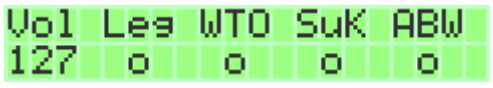
Another nice possibility is to restart LFOs, or to step through a wavetable which modifies sound parameters.




- **TRG (Trigger Source):** selects the trigger source which should be displayed:
  - **NOn:** a Note On event has been received
  - **EOf:** a Note Off event has been received
  - **E1S/E2S:** Envelope Sustain Phase has been reached
  - **L1P..L6P:** LFO period
  - **Clk:** global clock event
  - **Cl6:** triggered on each 6th global clock event
  - **C24:** triggered on each 24th global clock event
  - **MSt:** MIDI Clock Start event has been received
- **O1L..O3L/O1R..O3R:** (Re-)trigger gate of oscillator 1..3 of left/right SID channel (assigned to NOn and NOf by default, whereas NOf has a special purpose here: it clears the gate instead of retriggering it)
- **E1A/E2A:** start attack phase of ENV1/ENV2 (assigned to NOn and MSt by default)
- **E1R/E2R:** start release phase of ENV1/ENV2 (assigned to NOf by default)
- **L1..L6:** restart LFO1..LFO6 (the NOn->L1..L6 connection can also be controlled with the Snc flag in the LFO menu)
- **W1R..W4R:** restart wavetable 1..4 (assigned to NOn by default)
- **W1S..W4S:** increment step of wavetable 1..4 (assigned to Clk by default)

## ConFiGuration menu

Some basic settings for the MBSID Lead engine are collected here.



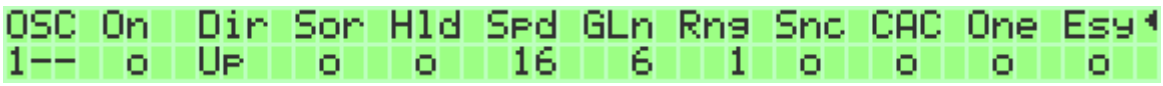

- **Vol (Volume):** adjusts the output volume of the SID. Note that the SID mixer only provides a 4bit volume, but 7bit (0-127) can be modified here (for the V2A feature). Set the volume to the middle value (64) if it should be modulated from the modulation matrix!
- **Leg (Legato):**

    - **Legato off (Mono mode):** the gate (or to be exact: the NOn event of the trigger matrix) will be re-triggered whenever a new note is played on the keyboard.
    - **Legato on:** the gate (or to be exact: the NOn event of the trigger matrix) will only be re-triggered so long no other note is played.
- **WTO (Wavetable Only):** only the wavetable can play notes - useful if it plays arpeggiator or note sequences
- **SuK (SusKey):** if enabled, the portamento effect will only be activated if more than one note is played. Also known as "fingered portamento". Very useful in conjunction with external sequencers (like MIDIbox SEQ to trigger a glide on overlapping notes.
- **ABW (ADSR Bug Workaround):** an option which provides a less usual method to overcome the ADSR bug. Whenever the envelope is retriggered, the ADSR registers will be zeroed for at least 30 mS (time can be increased with the delay parameter). Thereafter the original ADSR values will be written back, and the gate will be activated. This results into a more deterministic envelope, but the latency makes it unsuitable for live playing. So, this feature can only be used in conjunction with a sequencer, which allows to compensate the delay (which allows to play the notes earlier by a given time).

## ARPeggiator menu

In "digitunes" (also known as 8bit tunes), an arpeggiator is mostly used to produce fast "broken chords". The notes of a chord are quickly cycled, so that the listener (nearly) gets the impression that a whole chord is played, even the oscillators itself are only played monophonic.
A more common usage of the arpeggiator, which you propably also know from other synthesizers, is the slow automated stepping through the notes of a chord. In this case the step speed and the gatelength is mostly derived from an incoming MIDI clock to synch the notes with a sequencer.

Both variants are possible with the MIDIbox SID arpeggiator. And a special feature is, that each of the 2*3 oscillator (left/right channel) has its own arpeggiator, so that different rhythmical textures can be generated and played the same time.




- **OSC (Oscillator):** selects a single oscillator (1--, -2-, --3) or multiple oscillators (e.g. -23 or 123). Left/Right/Both SID channel has to be selected in the "shift" menu.
- **On:** enables the arpeggiator effect.
- **Dir (Direction):** Up/Down/Up&Down/Down&Up/UD2/DU2/Random - the direction and order in which the notes are stepped. To simplify the understanding, here the behaviour when a chord of four notes (1234) is played:
    - **Up:** 123412341...
    - **Down:** 432143214...
    - **U&D (Up&Down):** 12343212343...
    - **D&U (Down&Up):** 43212343212...
    - **UD2 (Up&Down #2):** 12344321123443...
    - **DU2 (Down&Up #2):** 43211234432112...
    - **Random:** randomly selected keys

- **Sor (Sorted):** if enabled, the played notes are sorted. Otherwise they are used in the same order like played on the keyboard (the note you played the first time will be played at the first position).
- **Hld (Hold):** the arpeggiator continues to play when the keys are released. It starts with a new chord once at least one new note is played.
- **Spd (Speed):** the arpeggiator speed derived from the global clock. A speed value of 48 will step the arpeggiator on each 8th note, a value of 24 on each 16th note, 12 on each 32th note, 6 for each 64th note, 3 for each 128th note
- **GLn (Gatelength):** the gatelength. Should be less than the speed value to take effect. Should be higher than the speed value for C64 style "broken chords".
- **Rng (Octave Range):** with an octave range > 1 the notes will be stepped multiple times over different octaves
- **Snc (Synch to Note):** the arpeggiator is restarted whenever a new note is played
- **CAC (Constant Arp Cycle):** the arpeggiator speed varies with the number of played notes, so that the overwhole arp cycle time is always the same (works only correctly with up to 4 notes)
- **One (Oneshot):** arpeggiator sequence will be stepped one time, thereafter the arp will stop until a new chord is played.
- **Esy (Easy Chord):** In difference to normal chord entry, in "Easy Chord" mode depressed keys will still be cylced by the arpeggiator until a new key is pressed. This feature is especially useful when slow arpeggios are played, since it avoids unintended "long notes" on chord changes.
  It is not useful if you intend to play a voice like a common lead instrument, which should turn into a fast arpeggio when more than one key is played (in this case, ensure that the "Esy" switch is disabled)

  Example for arp at fast speed:

  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_arp_fast.mp3

Switching through the direction modes (boring C-Dur chord, only for demonstration):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_arp_slow.mp3

Having some fun with upward direction, octave range 3:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_arp_slow2.mp3

> Note: in difference to other arpeggiator implementations of synths I know, the arp will continue to cycle the notes, even when no key is played anymore, and therefore all OSC gates are released. This matches with the behaviour in C64 tunes.
> In addition, if you release some - but not all - keys while a chord is played, the old chord continues to play until at least one new key has been pressed. This is to improve live playing. If you don't like this behaviour, you can disable it in sid_midi.inc (set EXP_ARP_BEHAVIOUR to 0)

## WaveTable Configuration menu

The term "Wavetable" is used by the C64 community for a synthesis method, where the waveform and frequency of a SID voice is modulated so fast, that the resulting output sounds like a new waveform, which is originally not provided by the soundchip itself. The parameter changes are

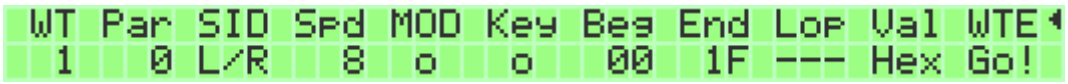
stored in a table - that's the reason for this name - the technique is not related to "wavetable synthesis", which is based on audio samples.

MIDIbox SID provides a very generic wavetable sequencer, which can not only control the waveform and frequency, but any parameter which is also accessible via NRPN (see also the MIDIbox SID V2 parameter chart). In MIDIbox SID V2, four tracks of up to 32 steps are available, which can control 4 different CCs. For common wavetable sounds, it's enough to control the waveform (OSC1 waveform: Parameter #33) and the transpose value (OSC1 transpose: CC#37) of a single voice. It's also possible to control the waveform/transpose value of all oscillators at the same time (CC#32 and CC#36), or to modulate the CC of the other oscillators in parallel to the first one. Another interesting usage of the remaining two tracks is the modulation of LFO or envelope parameters; it should also be mentioned, that the wavetable sequencer can be used to play notes and arpeggiator phrases.

For MIDIbox SID V1 some tutorials are available, which will sooner or later be adapted to the new V2 engine: HowTo #1, HowTo #2, HowTo #3.

Differences to V1:

- 4 instead of 3 wavetables
- independent speed and loop points
- free configurable start and end point within the 128 byte range
- resetable and stepable from the trigger matrix (e.g. via LFO)
- using WT values as modulation source
- controling the WT position from the modulation matrix
- Left/Right/Both SID channels selectable




- **WT1..4:** selects the wavetable
- **Par (Parameter):** changes the parameter assignment. Once you move the encoder, a special page will be displayed which shows the parameter name in long format:

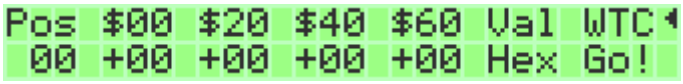



- **SID:** allows to select, if the left, right or both channels should be controlled from the wavetable.
- **Spd (Speed):** the wavetable speed derived from the trigger layer assignment (by default: the global clock). A speed value of 48 will step the WT on each 8th note, a value of 24 on each 16th note, 12 on each 32th note, 4 for each 64th note, 3 for each 128th note.
- **MOD:** if enabled, the wavetable position will be controlled from a modulation target (-> MOD menu) within the begin and end range. The speed and loop parameter has no effect in this case.
- **Key:** if enabled, the wavetable position will be controlled from the key number within the begin and end range. The speed and loop parameter has no effect in this case. This parameter has higher priority than the MOD function. E.g., with Begin=0x00, End=0x7f, KEY=on, Par=200 you are able to define different ENV1 Decay parameters for each note. There are much more usecases of course (e.g. different LFO values, different waveforms, Sync/Ring enable/disable in certain keyranges) - be creative :)
- **Beg (Begin):** the start position within the wavetable (0x00, 0x20, 0x40, 0x60 by default)
- **End:** the end position within the wavetable (0x1f, 0x3f, 0x5f, 0x7f by default)

- **Lop (Loop):** if disabled, the wavetable will only be played once (oneshot mode). Once enabled, the wavetable will be looped between the Loop and End position
- **Val (Value Format):** switches between hexadecimal and decimal display mode
- **WTE:** shortcut to the WTE (wavetable editor) menu

## WaveTable Editor menu

In order to improve the oversight and handling, the wavetable has to be edited in a separate menu page. This has also the advantage, that in this page the wavetable position can be selected with the PAGE UP/DOWN buttons, whereas within the WTC menu these buttons select the wavetable number.




- **Pos (Position):** the position within the wavetable (0-128, whereas mostly 0-31 is relevant for a common wavetable). Turn the encoder full-right to select "All" - in this case, a WT modification will be copied over to all entries of a column.
- **2nd..5th column:** the wavetable entries. At the top you will see the address, at the bottom the relative or absolute value.
- **Val (Value Format):** switches between hexadecimal and decimal display mode
- **WTC:** shortcut to the WTC (wavetable configuration) menu

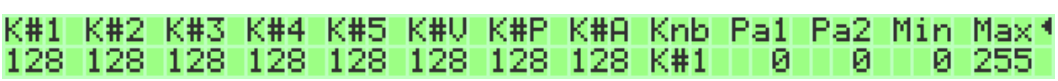
## KNoBs menu

The idea behind the "knobs" function is to have a set of 5 controllers, which can be used to change preselected sound parameters within a min/max range. Each patch has its own set, so that the sound designer can prepare parameter assignments as they are most useful for live control.

Knob values can not only be changed within the KNB menu, but also with the 5 rotary encoders of the OSC assign layer. In addition, they can be controlled via analog pots connected to J5.A0..A4 (DEFAULT_J5_FUNCTION set to 1), and via MIDI, giving access to the parameters on a generic way (e.g. for automation). Knob1 is assigned to CC#1 (Modulation Wheel), and Knob2-5 to CC#16-19

Three additional "knob" functions are available for Velocity (K#V), Pitchbender (K#P) and Aftertouch (K#A).

Knob values are also accessible as modulation source (-> see MOD menu) regardless of the knob assignments.




- **K#1..K#5:** the actual knob values, range: 0..255. They can be changed from this menu, via MIDI or via control surface, if the "assign" control layer of the OSC section is activated.
- **K#V, K#P, K#A:** the additional knob functions for Velocity/Pitchbender/Aftertouch.
  By default, K#P is assigned to parameter #80 (Voice 1/2/3 Pitch Bender), which controls the pitch within the given pitch range. This assignment can be replaced by another one to change the purpose of the pitchwheel of your keyboard (beside of the modulation wheel you

can use it as second controller). Or a second assignment can be made; e.g. transposing a single oscillator which has the sync flag set - you must try it out! :)

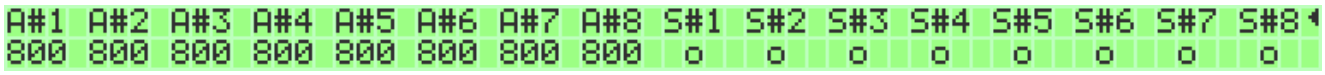
- **Knb (Knob):** selects a knob for which the Min/Max/Parameter assignment should be changed.
- **Pa1/Pa2 (Parameter Assignment 1 and 2):** one knob can control up to two parameters. All available parameters are listed in the MIDIbox SID V2 parameter chart, and the full name is displayed on screen when the assignment is changed with a rotary encoder.
  **Note:** a parameter is automatically scaled between 0..255 independent from the original resolution (1bit to 12bit)
- **Min/Max:** allows to specify the range within the parameter should be changed with a knob. If Min is greater than Max, the resulting value will be inverted.

## EXTernal menu

Up to 8 external control voltages (CV) can be output by MIDIbox SID via a MBHP_AOUT interface, or up to four MBHP_AOUT_LC interfaces. They are available as modulation target, and can be controlled via NRPN, CC (Knobs) and wavetable. This menu allows you to set the offsets of the control voltages (they are stored in the patch)

Note that it is not a requirement that the modulated CV values control external hardware - they could also be used by internal - optional - engine extensions in future.

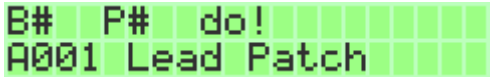
In addition, 8 digital switch outputs are predefined which allow you to enable/disable external effects. Yes, it would be fine if they would be controllable from the trigger matrix as well, but unfortunately there is no place within the trigger matrix anymore! :-( However, maybe a workaround (e.g. optional replacing of WT assignments to Switch assignments) could be feasible if somebody really finds this feature useful.



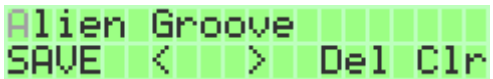

- **A#1..A#8 (analog output 1..8):** 12bit value (internally 16bit), displayed in hexadecimal format (range 000-FFF)
- **S#1..S#8 (digital switches 1..8):** switches on/off an external digital pin

## SAVe menu

Within the save menu a patch can be stored in the given bank/patch slot. The name of the currently stored patch is displayed, so that you know, which one is overwritten once you press the Do! (third selection) button.




Before the patch will be stored, you will be asked for the new name. Confirm with the first select button (below SAVE)



### RaNDom Generator menu

This generator allows to randomize single parts, or the whole patch with slightly constrained values (to improve the results).




Following quickly generated random patches speak for themself:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_rnd.mp3

# Bassline Engine

In this chapter it is assumed that you are already familar with the Lead and Multi Engine, so that only the differences need to be explained in detail.

The bassline engine is optimized - as the name says - for bass sounds and sequences. It provides two instruments, the first is always assigned to the left audio channel, the second to the right audio channel. They can be controlled independent from each other over different MIDI channels and/or keyboard zones assignments in the Ensemble menu. Alternatively, they can be controlled with the same keys by assigning them to the same MIDI channel and keyboard zone - this allows to play layered/stereo sounds or combined sequences.

Following use cases have been considered:

- **Direct MIDI Control:** active when sequencer mode within the SQC menu not enabled. The bassline can be directly played from a MIDI keyboard. Accent is activated with velocity >= 64, slide is activated when more than one key is played the same time (Sus-Key, sometimes also called "fingered" mode).
- **Sequencer Mode:** can be activated within the SQC menu. One of 8 sequences can be selected by playing a note on a MIDI keyboard. Note C selects sequence 1, Note C# selects sequence 2, ... Note G Sequence 8. With Note G#, A, A# and B the current sequence is deselected (not played). The octave doesn't matter (no transpose).
  Bassline L and R (left and right channel) share the same sequence memory. Different sequences can be selected for these basslines by assigning the instruments to different MIDI channels and/or keyboard zones within the Ensemble
  While bassline sequences are selected over the defined MIDI channel and keyboard zone of ensemble instrument #1 and #2, they can be transposed with the settings of instrument #3 and #4. This strict seperation allows several usecases (e.g. no, same or separate transpose for L/R bassline, or transposing from a different MIDI channel which is also used for other MIDI devices, e.g. MIDIbox SEQ)

In order to ensure, that both basslines are in sync, they are always started together. It is possible to start with a single bassline by disabling the second one in the preselection (Seq=-- in SQC menu)
The sequencer continues to play when the MIDI key has been released (hold function). This allows you to tweak sound parameters with both hands - whenever you want to select a new sequence, play a new note. Whenever you want to stop the sequence, play G#/A/A# or B. Optionally the sequence can be preselected before the next measure (16 steps) is reached (S16 flag in SQC menu). Personally I prefer not to use this feature, as switching between different sequences in between a measure results into great variations (preferred configuration for jam sessions! :)

- **Arpeggiator Mode:** is available as well and provides the same parameters like known from the Lead and Multi Engine.
  Note that it isn't possible to combine the sequencer with the arpeggiator - arpeggios have to be played from an external keyboard (or external sequencer...).

The bassline sequencer works like known from Roland TB-303. Up to 16 steps can be played. Each step consists of Note/Gate/Accent/Slide/Octave and an additional, free assignable sound parameter. Each patch contains 8 sequences, which can be selected from a MIDI keyboard, within the SQC menu, or from the modulation matrix buttons of the control surface.

## Sound Demos

Here a MP3 sample from my first spontaneous jam session with the completely overworked MBSID V2 bassline engine - note that this recording is the work of 30 minutes and not more (no optimisations, no MIDI automation). Sequences are played internally, they are selected from a MIDI keyboard. Parameters are tweaked from the Control Surface. The MIDIbox is synchronized via external MIDI clock to Logic Audio. Drums are played from a Spectrasonics Stylus RMX VST.

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo1.mp3

The second jam session, where one SID used a feedbacked audio out (see also this forum posting).
Warning: >8 MB, very repetive stuff!

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo2.mp3

Another jam session, which is also documented on video > YouTube, Vimeo (better audio quality).

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo3.mp3

This package contains the two patches which have been used in this demo + a "HowTo"

During this session, two SSM2044 via AOUT_NG module were used to filter the outputs of two SIDs.

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo4.mp3

Forum Article

The 5th demo, played on a MBSID V2 in MB-6582 case -> YouTube, Vimeo (better audio quality).

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo5.mp3

Another bassline demo - the patches can be downloaded from from this forum thread.

http://www.ucapps.de/mp3/midibox_sid/mbsid_v2_tk2_soundbank_demo.mp3

The second part of this demo created some years later - the bank is part of the release package meanwhile ("presets/bank2_tk2_soundbank.syx")

http://www.ucapps.de/mp3/midibox_sid/mbsid_v2_tk2_soundbank_demo_part2.mp3

## Root Menu

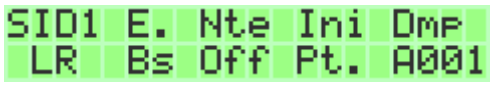
This is the root menu of the bassline engine. In difference to the Lead Engine, you will miss the MOD (Modulation Matrix), TRG (Trigger Matrix) and WTC/WTE (Wavetable) entries. Instead, you will notice a new SEQ and SQC (Sequencer/Configuration menu)




The root menu gives you quick access the submenus, which are described below.

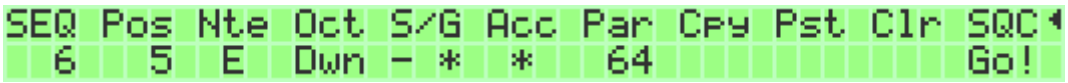
## "Shift" Menu

To initialise a Bassline Patch, press the SHIFT (formerly LINK) button, and press the second select button below the LCD to change to the Bs (Bassline) engine:




## SEQuencer menu

As the TB303-style sequencer is the most important feature of the bassline engine, the appr. page can be entered with the first item of the root menu.




A sequence pattern consists of 16 steps, and 8 patterns are embedded within a patch. The pattern memory is shared between the two basslines.

Following functions are available in the SEQ page:

- **SEQ:** selects 1 of 8 patterns, which should be edited
- **Pos:** the step number. It can also be changed with the Page Up/Down button (preferred usage)
- **Nte:** the note which should be played by the step
- **Oct:** the octave (Down/-/Up). Additional (fixed) transposing is possible within the OSC menu
- **S/G:** slide and gate flag. With activated slide, the note will be stretched to the next step, and the portamento function will be activated when the next step is played.
  The note will only be played if the gate flag is set.
- **Acc:** the accent flag causes following changes:
  - Note will be played with maximum sustain value

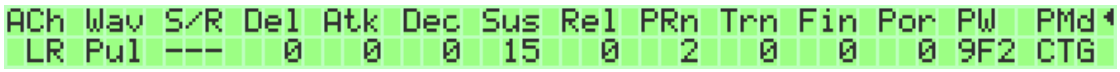
    - LFO and envelope depth will be increased by the "Accent value", which can be specified within the LFO/ENV menu.
    - the envelope generator will select a special Decay value (DcA), which is normaly shorter than the "normal" decay, so that the envelope will reach the sustain phase faster.
- **Par:** allows to control a specific sound parameter in addition to the note (2nd parameter track, see also SQC menu)
- **Cpy:** copies the current pattern into a buffer
- **Pst:** paste: copies the buffer content into the current pattern
- **Clr:** clears the current pattern
- **SQC:** shortcut to the SQC page

The sequencer configuration page contains following items:




- **ACh:** audio channel (Left/Right) - selects the bassline. Changes will be done for both basslines (LR), for the left one only (L-) or for the right one only (-R)
- **On:** enables/disables the sequencer function. *If disabled, the bassline can be directly played via MIDI!*
- **Seq:** selects the initial pattern after a patch change. A pattern is normaly started **and selected** by pressing a note on a keyboard. So, this value is only relevant if you are using an alternative start method (e.g. MIDI start event if MBSID Clock generator in slave mode)
- **Spd:** the sequence speed, derived from the global BPM rate (Ensemble->Clk menu). Since the BPM generator is running at quadrupled clock rate, the frequency should be divided by 4 in order to get back the "normal" BPM rate.
- **Len:** the track length (1..16 steps)
- **S16:** if active, pattern changes (e.g. via MIDI keyboard) won't take place immediately, but once the next bar is played. This allows you to synchronize pattern changes to a 4/4 measure.
- **Par:** specifies the sound parameter which is controlled by the "parameter track" (Par value within the SEQ menu). All available parameters are listed here. Due to the special coding, it is possible to control both, the current, or a specific (L/R) bassline from the parameter track ("Current" means for this case not the LR selection on the control surface, but the bassline which plays the sequence)
- **SEQ:** shortcut to the SEQ page

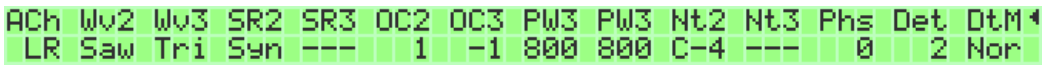
## OSCillator menu




All parameters have the same purpose like described for the Lead engine. Instead of the OSC (oscillator), the **A**udio **Ch**annel has to be selected (LR/L-/-R).

Note that this page only controls a single ("master") oscillator. The remaining two ("slave") oscillators of each SID are configured from the O23 page.

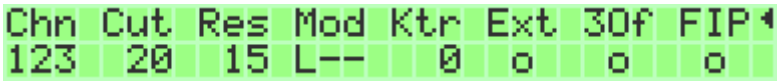
## O23 menu




Oscillator 2 and 3 of the SID are used as "slave oscillators" in bassline mode. This means, that their pitch and gate follow OSC1. This option is nice for fat unisono effects, but also for sync/ringmodulation, or layered waveforms - the resulting sound is less TB303-like, but much more "SIDish".
Yessss, we want this!!! :-)

- **ACh:** audio channel (Left/Right) - selects the bassline. Changes will be done for both basslines (LR), for the left one only (L-) or for the right one only (-R)
- **Wv2 and Wv3:** selects the waveforms of the slave oscillators. If disabled ("---"), the gate of the oscillators won't be triggered. Recommentation: use different waveforms to create SIDish timbres.
- **SR2 and SR3:** activates Sync/Ringmodulation for oscillator 2 and 3.
- **Oc2 and Oc3:** transposes the slave oscillators octave-wise.
- **PW2 and PW3:** sets the width of the pulse waveform (800: 1:1 duty cycle).
- **Nt2 and Nt3:** decouples the pitch of a slave oscillator from the master oscillator. Allows to play a slave oscillator at a specific pitch (e.g. C-3), especially useful for nice sync and ringmod effects.
- **Phs (Oscillator Phase Offset):** this function allows you to synchronize the phases of all three oscillators.
  With Phs=0 they are freerunning - this is the preferred option for "analog" sounds. With Phs=1, they will be started at the same moment, whereas with Phs>1 the phase offset between the oscillators will be controlled in 0.4% steps (!).
- **Det (Detune):** detunes all oscillators by increasing (OSC2)/decreasing (OSC3) the pitch. This allows you to create fat sounds with a single encoder turn!
- **DtM (Detune Mode):** switches between normal (legacy) and SuperSaw +/- mode, which has been created by Lis0r based on the research from this paper

## FILter menu



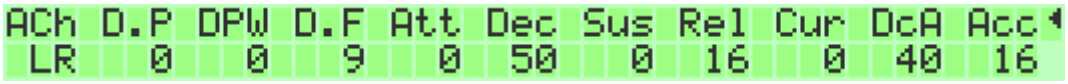

## LFO menu




Each instrument features two LFOs with fixed modulation routing to Pitch, Pulsewidth and Filter. The modulation depth for these targets can be changed individually (**D.P** (Depth Pitch), **DPW** (Depth Pulsewidth), **D.F** (Depth Filter).

The **Acc**ent parameter at the end of the page specifies, how much the depth values will be increased when a note is played "accented" (SEQ: accent flag, MIDI: velocity >= 64).

## ENVelope menu




Each instrument provides one envelope with fixed modulation routing to Pitch, Pulsewidth and Filter. The modulation depth for these targets can be changed individually (**D.P** (Depth Pitch), **DPW** (Depth Pulsewidth), **D.F** (Depth Filter).

In difference to the Lead Engine, only Attack, Decay, Sustain and Release phase is available. The curve parameter affects all ramps.

The **Acc**ent parameter at the end of the page specifies, how much the depth values will be increased when a note is played "accented" (SEQ: accent flag, MIDI: velocity >= 64).

In addition, a special decay value will be selected on accented notes (**DcA**), which should be shorter than the "normal" decay, so that the envelope will reach the sustain phase faster.

## ConFiGuration menu

Some basic settings for the MBSID Bassline engine are collected here. They have the same function like known from the Lead Engine

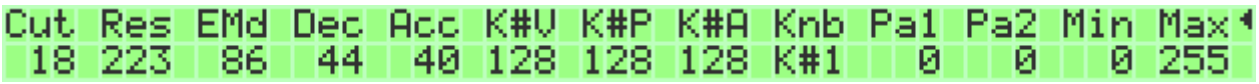
## ARPeggiator menu




The arpeggiator works exactly like for the Lead Engine

Note that the sequencer and arpeggiator functions are handled mutually exclusive - they cannot be used the same time.

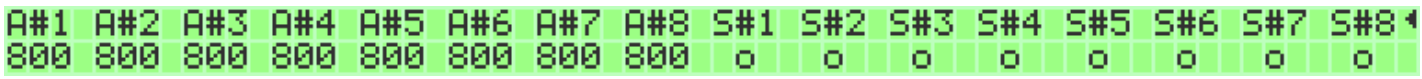
## KNoBs menu




Knob assignments are available as well. The special coding of the parameter table (see MIDIbox SID V2 parameter chart) allows to control sound parameters individually for a specific bassline, for the currently selected, or for both basslines.

The typical TB303 parameters CutOff/Resonance/EnvMod/Decay/Accent are assigned to the 5 knob functions by default. This generic approach allows you to control either both basslines, the currently selected (first, second or both), or a dedicated (first or second) bassline depending on the parameter assignment numbers. It's possible to assign a second parameter to the same knob, a min/max range can be specified, values can be received from MIDI, and they are sent to MIDI if the CC function of the Control Surface is activated for easy recording of parameter changes (MIDI automation). Last but not least, knob values can be controlled from analog inputs. This means, that you could also control them with pots or joysticks.

## EXTernal menu

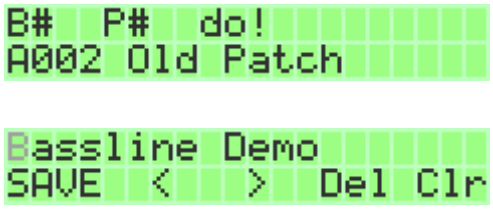



In difference to the Lead Engine, CV outputs are not available as modulation target. The static voltage levels and switch states can be changed here.

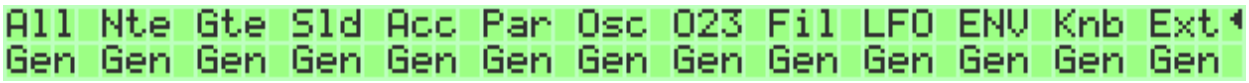
The only way to modulate analog outputs within multi patches is the use of the F2A (Filter-to-Analog) option, which can be activated in the Ensemble menu.

## SAVe menu

...is available as well of course! :)




## RaNDom Generator menu




Note/Gate/Slide/Accent and the sound parameter track can be randomized separately.

## Special Control Surface Handling

The OSC and ENV selection buttons behave different: they select the bassline (LR/L-/-R).

The modulation matrix displays the selected sequence as horizontal (left channel) and vertical (right channel) bar. The sequence can be changed with the appr. modulation matrix button - this is an alternative solution if you don't want to select a sequence with a MIDI keyboard.

Once the sequencer is playing, the LED matrix turns into a step display with flashing LEDs. Since there are only 8 LEDs in a column/row, two steps have to share the same LED. Anyhow, you will like the effect, it looks nice! :) Due to implementation reasons, the special step display works only when basslines are played on the master core.
If you built your MIDIbox SID based on the old Frontpanel, the 8th row will be missing, accordingly the 8th sequence cannot be selected/displayed for the left channel bassline - sorry!

Sequences can be started from the control surface with the Play button. If this button isn't available on your frontpanel, just hold the SID1/SID2/SID3 and/or SID4 button and press menu. The sequence will stop with the Note Off function, which is available within the Shift page.

If all these buttons are not part of your frontpanel, sequences can only be started/stopped via MIDI.

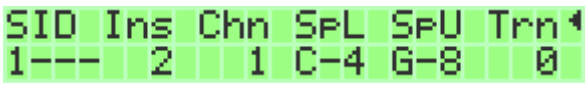
# Recommented Ensemble Configuration

For easiest usage of the sequence selection feature, it is recommended to create a special Ensemble configuration, in which the two basslines are assigned to the same MIDI channel. By creating split points at C-3 and C-4, it is possible to control the transposer with the lower keyboard range, the "left" bassline with a middle keyboard range, and the "right" bassline with the upper keyboard range.

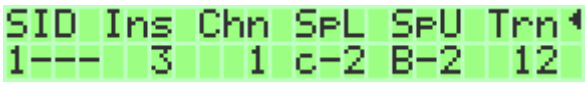
Ensemble->Ins Configuration for Bassline L:



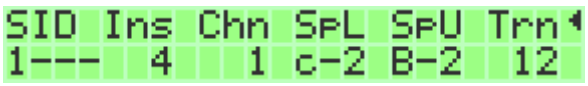

Ensemble->Ins Configuration for Bassline R:




Ensemble->Ins Configuration for Transposer of Bassline L. "Trn +12" is required in this setup to compensate the octave range of the transposer (base note C-3 is outside the selected keyboard zone c-2..B-2):




Ensemble->Ins Configuration for Transposer of Bassline R (here: controlled from same keyboard zone as L bassline):




If more than two basslines should be controlled from the same keyboard (played on the slave SIDs), the ranges need to be reduced to dedicated octaves.

If the basslines should be synchronized to an external MIDI clock master, just select Slave mode within the Ensemble->Clk menu:




Only in slave mode, sequences will be started and stopped via MIDI Start/Stop events - in Master mode such events are ignored.

Note that using the slave mode is currently the only way to keep basslines played on multiple cores in synch. An inter-core synchronisation hasn't been implemented yet (see also this page).

# Drum Engine

The drum engine provides prepared "drum models" with optimized parameter sets, which allow you to create chiptune typical percussive sounds on a fast and easy way.

Each patch can store up to 16 drum instruments (-> drum kit), which can either be played via MIDI, or from an internal sequencer.

Since a SID features only 3 oscillators, and a core can only access two SIDs, a dynamic voice allocation is used, similar to the Multi Engine. This means, that 6 of 16 instruments can be played the same time from a single core. If your MIDIbox SID is stuffed with multiple core/SID modules, you can easily extend the number of voices and instruments of course. This approach is especially useful, if drum sounds should be routed to different Fx busses. Within the Ensemble configuration the drum sections can be assigned to individual keyboard zones and/or MIDI channels.

When drums are triggered via MIDI Notes, the velocity can control one sound parameter of the drum model. This can be configured for each instrument individually. For example the ADSR sustain value (-> volume), the tuning (-> frequency), the wavetable speed, etc... it's even possible, that the velocity of one instrument controls the sound parameter of another instrument in all thinkable combinations. This is especially interesting for ringmodulated sounds.

Alternatively, drums can be played from an internal 8 track/16 steps sequencer. Each track controls two drum instruments. 8 Drum patterns are stored within the patch, and the pattern can be interactively selected from a MIDI keyboard on the same way like known from the Bassline Engine. It's an ideal solution for extensive jam sessions. :-)

## Sound Demos

The sound samples below should demonstrate the wide range of adjustability of the drum models.

- BD1 and BD2
- http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_bd_examples.mp3
- SD1 and SD2
- http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_sd_examples.mp3
- FX1, FX4 and FX5

In the next demo, bassline and drum engine are played together. This is a live recording; the sequences and sound parameters are controlled from a MIDI keyboard. Note: the bass drum is very low-tuned and cannot be regognized with cheap speakers.

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_fx_examples.mp3

Following drum kits and sequences are part of the preset library. They are played by two SIDs, which are controlled by a single core. Note that some drum sounds are not played if only one SID is connected. For such a configuration, you should at least enable mono mode (-> Ensemble) so that all instruments are assigned to the single SID!

- http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_demo1.mp3
- A033: Drum Kit 1
  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_preset_a033.mp3
- A034: Drum Kit 2
  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_drums_preset_a034.mp3

## Root Menu

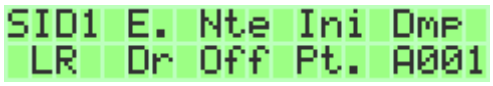
This is the root menu of the Drum Engine:




The root menu gives you quick access the submenus, which are described below.

## "Shift" Menu

To initialise a drum patch, press the SHIFT (formerly LINK) button, and press the second select button below the LCD to change to the Dr (Drum) engine:

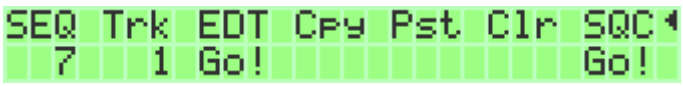



Take note, that the default patch is a very "cheap" one where all sound parameters are set to their default value in order to give you a proper start for editing a new drum kit.

Better drum kits can be found in the preset bank from A033 upwards

## SEQuencer menu

As the TRx0x-style sequencer is the most important feature of the drum engine, the appr. page can be entered with the first item of the root menu.




A sequence pattern consists of 8 tracks with 16 steps, and 8 patterns are embedded within a patch.

Following functions are available in the SEQ page:

- **SEQ:** selects 1 of 8 patterns, which should be edited
- **Trk:** selects 1 of 8 tracks, which should be edited
- **EDT:** enters the step **EDit**or, more infos about this special page below.
- **Cpy:** copies the current pattern into a buffer
- **Pst:** paste: copies the buffer content into the current pattern
- **Clr:** clears the current pattern
- **SQC:** shortcut to the SQC page

The step editor displays the 16 steps of the selected track, and allows you to modify them. The characters at the upper line are standing for:

- **.:** no instrument is played
- ***:** primary instrument will be played
- **A:** primary instrument will be played accented (increased sustain, decreased decay)
- **S:** the secondary instrument will be played (w/o accent)

- Track 1 Controls Instrument #1 (primary: * and A) and #2 (secondary: S)
- Track 2 Controls Instrument #3 (primary: * and A) and #4 (secondary: S)
- Track 3 Controls Instrument #5 (primary: * and A) and #6 (secondary: S)
- Track 4 Controls Instrument #7 (primary: * and A) and #8 (secondary: S)
- Track 5 Controls Instrument #9 (primary: * and A) and #10 (secondary: S)
- Track 6 Controls Instrument #11 (primary: * and A) and #12 (secondary: S)
- Track 7 Controls Instrument #13 (primary: * and A) and #14 (secondary: S)
- Track 8 Controls Instrument #15 (primary: * and A) and #16 (secondary: S)

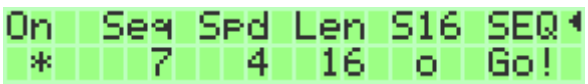
The advantage of this method: it saves memory (8 sequences available instead of 4), and it allows to realize the typical Open/Close HH or High/Low Tom scheme from a single track.




- **G/A/S:** toggles the selected step between Off/Gate On/Accented/Secondary mode
- **Rot:** activates a rotate function, which allows you to scroll the pattern left/right. Press the button again to deactivate this function.
- **Rnd:** overwrites the track with a random pattern
- **Clr:** clears the track
- **DRM:** jumps into the DRM menu page. The instrument number will be set automatically according to the selected track

The step is selected with the main encoder.
Take note that the track can be selected with the Up/Down buttons of the control surface while in this page. Additionally, tracks can be directly selected with the 8 mod target buttons (named with E1/E2/L1/...L6).

The sequencer configuration page contains following items:




- **On:** enables/disables the sequencer function. *If disabled, drums can be directly played via MIDI!*
- **Seq:** selects the initial pattern after a patch change. A pattern is normaly started **and selected** by pressing a note on a keyboard. So, this value is only relevant if you are using an alternative start method (e.g. MIDI start event if MBSID Clock generator in slave mode)
- **Spd:** the sequence speed, derived from the global BPM rate (Ensemble->Clk menu). Since the BPM generator is running at quadrupled clock rate, the frequency should be divided by 4 in order to get back the "normal" BPM rate.
- **Len:** the track length (1..16 steps)

- **S16:** if active, pattern changes (e.g. via MIDI keyboard) won't take place immediately, but once the next bar is played. This allows you to synchronize pattern changes to a 4/4 measure.
- **SEQ:** shortcut to the SEQ page
-

## DRuM menu menu

This is the only page where drum instruments are edited:




- **Ins(trument):** selects the drum instrument (1..16)
- **Mdl (Model):** selects the drum model - the number of available models depends on the firmware version
- **Attack/Decay/Sustain/Release:** controls the VCA envelope of the SID oscillator.
- **Tun(e):** controls the pitch from -2..+2 octaves in 256 steps.
- **GL (Gatelength):** changes the gatelength from -127..0..128 (256 steps)
- **Spd (Speed):** changes the wavetable speed from -127..0..128 (256 steps)
- **PAR:** each drum model provides a special parameter which changes specific parts of the wavetable, for example the pitch of a specific WT step, or the noise frequency. This parameter is adjustable from -127..0..128 (256 steps)
- **VAs:** the voice assignment:
  - **LR (Left/Right):** instrument is either played on the left, or right SID audio channel (6 voices available).
  - **L- (Left):** instrument is only played on the left SID audio channel (3 voices available).
  - **-R (Right):** instrument is only played on the right SID audio channel (3 voices available).
  - **O1..O6:** selects an individual oscillator for the instrument. Take note, that this option also locks the oscillator for instruments which haven't explicitely assigned it. Example: if all Instruments are assigned to LR, but Instrument #3 and #4 to OSC3, only these two instruments will use this oscillator, and all others will share the remaining ones.

  The voice assignment is important, whenever drum sounds are ringmodulated, or routed through the filter (the filter is not switched dynamically, as this would result into unwanted "popping" sounds). Also for routing to external Fx busses it's useful to predefine the SID audio output channel (L- or -R).
- **VeA (Velocity Assignment):** allows to control a single sound parameter with the Note velocity. In difference to the appr. Knob function (-> KNB menu), they are working individually for each instrument. See also the MIDIbox SID V2 parameter chart for available controllers.
- **Rnd (Random):** a randomizer which only changes the currently selected instrument - it's a very useful feature for finding exotic sound configurations!

## FILter menu

Within the filter menu you can control the parameters of the SID filter. Each SID has only one 12db multistate filter, which can be assigned to the three oscillators and/or to the external audio input. Note that by default changes will take place for the left and right channel. By selecting a single channel within the "Shift" menu, you can do modifications for the two filters separately.

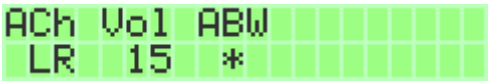
Use the voice assigmment option within the DRM menu (-> VAs) if only specific drum instruments should be filtered.




- **Chn (Channel):** assigns the filter to oscillator channel 1/2/3
- **Cut (CutOff):** sets the cutoff frequency (12bit value) from 000 to FFF (hexadecimal format).
- **Res (Resonance):** is a peaking effect which emphasizes frequency components at the CutOff frequency of the filter, causing a sharper sound. The intensity of the resonance can be set from 0-15.
- **Mod (Filter Mode):** sets the filter mode - note that mixed notes like Lowpass/Highpass are possible, which results into a notch effect (inverted bandpass).
    - **L (Lowpass):** all frequency components below the Cutoff frequency are passed unaltered, while all frequency components above the Cutoff are attenuated at a rate of appr. 12 dB/Octave
    - **B (Bandpass):** all frequency components above and below the Cutoff frequency are attenuated at a rate of appr. 6 dB/Octave
    - **H (Highpass):** all frequency components above the Cutoff frequency are passed unaltered, while all frequency components below the Cutoff are attenuated at a rate of appr. 12 dB/Octave
- **Ext (External):** external Audio IN is passed to the filter
- **3Of (3rd oscillator off):** a special SID function which disconnects the oscillator 3 output from the mixer. This allows to use OSC3 for sync/ringmodulation without any undesirable output.
- **FIP (Filter Interpolation):** this option smoothes CutOff changes. It is especially useful when CutOff is changed via CC (-> low resolution). to reduce the steppiness. The option is not useful when the CutOff parameter is modulated rom a fast LFO or envelope (on the other hand: it produces interesting effects in such cases ;-)

## ConFiGuration menu

Some basic settings for the MBSID Drum engine are collected here.




- **ACh:** audio channel (Left/Right) - selects the left/right or both audio channels. Selection is only relevant for the Vol(ume) parameter.
- **Vol (Volume):** adjusts the output volume of the SID.

- **ABW (ADSR Bug Workaround):** (independent from L/R flag) an option which provides a method to overcome the ADSR bug.
  The drum engine handles this option differently from the other engines. The test bit will be set together with gate for ca. 30 mS in order to reset the oscillator and to start the envelope. The waveform generator will be turned off durig this time in order to disable the sound output. After this delay, the waveform will be enabled again, and the drum sound starts immediately without wacky timings.
  This approach is also known as "hard-sync", and used in most C64 tunes.
  It results into a more deterministic sound output, but the latency makes it unsuitable for live playing. So, this feature can only be used in conjunction with the internal sequencer, or with an external sequencer which allows to compensate the delay.
  Note also, that a part of the ADSR attack phase is skipped (however, thats no real issue for drum sounds...)
  Due to the better results, this option is enabled by default!
  *Hot tip:*: there is a simple trick which allows you to play a bassline and drum sequence in synch during a live session: just add an audio delay unit with t_delay=30 mS (e.g. VST based) to the SID outputs which are playing the basslines in order to compensate the timings.

## KNoBs menu

```
K#1 K#2 K#3 K#4 K#5 K#V K#P K#A Knb Pa1 Pa2 Min Max
128 128 128 128 128 128 128 128 K#1   0   0   0 255
```

Knob assignments are available as well. The special coding of the parameter table (see MIDIbox SID V2 parameter chart) allows to control sound parameters individually for a specific, or the currently selected drum instrument.

## EXTernal menu

```
A#1 A#2 A#3 A#4 A#5 A#6 A#7 A#8 S#1 S#2 S#3 S#4 S#5 S#6 S#7 S#8
800 800 800 800 800 800 800 800   o   o   o   o   o   o   o   o
```

In difference to the Lead Engine, CV outputs are not available as modulation target. The static voltage levels and switch states can be changed here.

## SAVe menu ...is available as well of course! :)

```
B#  P#  do!
A002 Old Patch
```

```
Drum Demo
SAVE  <   >  Del Clr
```

- **RaNDom Generator menu**




Parts of a drum patch can be randomized with a single push on the button.
Please note, that good results can be found faster by using the randomize function within the DRM and SEQ editor page, because it only changes the currently selected item, and doesn't overwrite the parameters/patterns of the remaining instruments/sequences.

# Multi Engine

In this chapter it is assumed that you are already familar with the Lead Engine, so that only the differences need to be explained in detail.

Main difference of the Multi engine is the dynamical voice allocation. Instead of fixed assignments, voices will be selected automatically for six independent instruments.

Advantages:

- Instruments can be played polyphonic. Algorithm: "oldest note will be killed first" (same handling as used in MIDIbox FM).
- But also monophonic assignments are possible, which allows to play up to 6 different instruments the same time (Sidenote: which means, that if your MIDIbox SID is stuffed with 4 cores and 8 SIDs, you can play up to 24 instruments - this section talks about the engine running on a single core, but you are free to use it on multiple cores in parallel).
- MIDI channel assignments and keyboard zones can be configured within an Ensemble
- Each voice has two dedicated LFOs and one envelope with individual depths for Pitch, Pulsewidth and Filter CutOff modulation (makes 12 LFOs and 6 ENVs which are running in parallel!)
- In addition, each voice has an own wavetable (makes 6 wavetables which can run in parallel!!!)
- There are individual velocity and pitchbender assignments for each instrument, but also global velocity/pitchbender/aftertouch and 5 knob assignments which can control the parameters of a specific, or all instruments

But there are also disadvantages, which are mainly based on the construction of the SID chip:

- 6 voices are only available when two SID chips are connected to the MIDIbox SID core (each SID chip contains three oscillators). With a single SID, the 6 instruments have to share three voices, which only makes sense when they are not played the same time.
- in most cases it's better to center the two audio channels (especially in poly mode) - stereo sounds can only be realized by using two instruments and assigning them to specific voices (-> see CFG page), but the Lead Engine provides more sophisticated functions for stereo effects!
- no modulation and trigger matrix available. There are only static modulation assignments (LFO1/LFO2/ENV -> Pitch/Pulsewidth/Filter), but with individual depths!
- external CV outputs can only be modulated via (steppy) wavetables, or with the F2A (Filter-to-Analog) option in the Ensemble menu.
- biggest drawback: each SID chip provides only a single filter for three oscillators! Oscillator assignments to the filter are static and they are not dynamically changed by the voice allocator to avoid loud, unintentional switching sounds.
- and another imperfection of the SID chip should be highlighted again: due to the ADSR bug it can happen, that with certain VCA ADSR settings (especially high attack, decay and/or release rates), a note will be broken while it starts to play. Best working ADSR settings have to be tried out while playing the song. Or A, D and R has to be set to 0 (sustain=15). Alternatively the ABW (ADSR Bug Workaround) function can be activated, but this results into a latency of at least 30 mS.

## Root Menu

This is the root menu of the multi engine. In difference to the Lead Engine, you will miss the MOD (Modulation Matrix) and TRG (Trigger Matrix) entries.



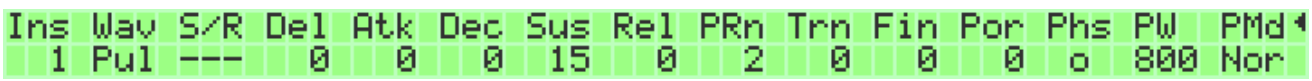

The root menu gives you quick access the submenus, which are described below.

## "Shift" Menu

To initialise a Multi Patch, press the SHIFT (formerly LINK) button, and press the second select button below the LCD to change to the MI (Multi) engine:




## OSCillator menu




All parameters have the same purpose like described for the Lead engine. Instead of the OSC (oscillator), the **Ins**trument has to be selected (1-6).

Note that due to the voice allocation algorithm, there is no fixed relation between instrument and oscillator number by default. Sync and Ringmodulation effects have to be handled with special

care, as the audio pathes for these effects are fixed within the SID chip. A static voice assignment has to be used for these functions (-> CFG menu) to get them properly working.

The variable oscillator phase has only a simple replacement (Phs flag) which restarts the oscillator when the gate is activated. This feature is also known as "hard-sync" and especially useful for percussive sounds.

## FILter menu




As mentioned above, the filter channels are not automatically routed to avoid loud switching sounds while voices are re-assigned. Therefore filtered sounds require a static voice assignment (-> CFG menu) which matches with the enabled filter channel. Alternatively it can be used for a single polyphonic instrument - in this case, all filter channels should be enabled.

## LFO menu




Each instrument features two LFOs with fixed modulation routing to Pitch, Pulsewidth and Filter. The modulation depth for these targets can be changed individually (**D.P** (Depth Pitch), **DPW** (Depth Pulsewidth), **D.F** (Depth Filter).

## ENVelope menu




Each instrument provides one envelope with fixed modulation routing to Pitch, Pulsewidth and Filter. The modulation depth for these targets can be changed individually (**D.P** (Depth Pitch), **DPW** (Depth Pulsewidth), **D.F** (Depth Filter).

In difference to the Lead Engine, only Attack, Decay, Sustain and Release phase is available. The curve parameter affects all ramps.

## ConFiGuration menu




In difference to the Lead Engine, this page has a new **VAs** (Voice Assignments) parameter which provides following selections:

- **LR (Left/Right):** instrument is either played on the left, or right SID audio channel (6 voices available).
- **L- (Left):** instrument is only played on the left SID audio channel (3 voices available).
- **-R (Right):** instrument is only played on the right SID audio channel (3 voices available).

- **O1..O6:** selects an individual oscillator for the instrument. Take note, that this option also locks the oscillator for instruments which haven't explicitely assigned it. Example: if all Instruments are assigned to LR, but Instrument #3 and #4 to OSC3, only these two instruments will use this oscillator, and all others will share the remaining ones.

*If the Mono option is activated in the Ensemble Menu, voices of the instrument are always assigned to both SID audio channels (3 voices only).*

Also new in this menu is the **Poly** flag, which should be selfexplaining! :-) Up to 6 voices can be played the same time. If more than 6 keys are pressed on the keyboard, the oldest note will be killed first. If Poly mode is not activated, the instrument will be played mono (only a single voice is allocated).

**VeA** (Velocity Assignment) and **PbA** (Pitchbender Assignment) allow to control a single sound parameter with the Note velocity and the Pitchbender. In difference to the appr. Knob function (-> KNB menu), they are working individually for each instrument.

The configuration menu has a small inconsistency: the Volume is handled globally for each SID chip and not individually for each separate voice. Sorry for that, but I haven't found a better place.

## ARPeggiator menu

```
Ins On  Dir Sor Hld Spd GLn Rng Snc CAC One Esy
 1  o  Up   o   o  16   6   1   o   o   o   o
```

The arpeggiator works exactly like for the Lead Engine. Polyphonic arpeggios are not supported (the Poly flag in the CFG menu has no effect) - it always plays on a single voice.

## WaveTable Configuration and Editor menu

```
Ins Par Spd Key Beg End Lop Val WTE
 1   0   8  o   00  0F  --- Hex Go!

Pos $00 $10 $20 $30 $40 $50 Val WTC
 00 +00 +00 +00 +00 +00 +00 Hex Go!
```

The wavetable sequencer works similar to the Lead Engine, but instead of 4, up to 6 wavetables can be played in parallel! Each instrument has its individual wavetable. The parameter set is different (see MIDIbox SID V2 parameter chart).

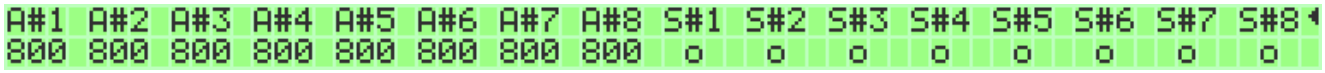
## KNoBs menu

```
K#1 K#2 K#3 K#4 K#5 K#V K#P K#A Knb Pa1 Pa2 Min Max
128 128 128 128 128 128 128 128 K#1   0   0   0 255
```

Knob assignments are available as well. They are working independent from the instrument layer and can control all instruments of a patch. The special coding of the parameter table (see MIDIbox SID V2 parameter chart) allows to control sound parameters individually for a specific instrument, or for all instruments.

Note that each instrument has an additional, dedicated velocity and pitchbender assignment, which has to be configured in the CFG menu.

## EXTernal menu

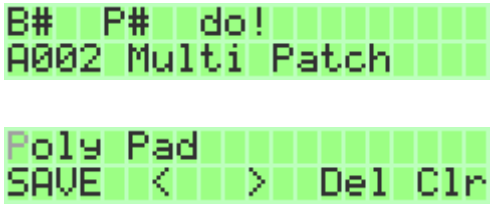



In difference to the Lead Engine, CV outputs are not available as modulation target. The static voltage levels and switch states can be changed here.

The only way to modulate analog outputs within multi patches is the use of the F2A (Filter-to-Analog) option, which can be activated in the Ensemble menu.

## SAVe menu

...is available as well of course! :)




## RaNDom Generator menu




Note, that in Multi mode you can try the random function by controlling a single instrument over a dedicated channel, or by controlling all instruments over a single channel.

# Ensembles

## Introduction

An ensemble contains a collection of four patches, which are assigned to the MBSID master and the three MBSID slaves. In addition, it contains some global settings for each separate unit which are not stored within a patch, like BPM generator settings, AOUT assignments and filter calibration values. MIDI channels and keyboard splitzones can be defined for up to 6 instruments per MBSID unit (relevant for multi patches).

These informations are stored in a separate BankStick, so that you are able to define up to 128 different ensembles. Unfortunately it takes up to 250 mS while switching between the ensembles, as a lot of data is read from different BankSticks (mainly the 400 kbit/s IIC interface limits the speed here). Therefore it is not possible to switch between different patches without a noticable delay!

Ensemble E001 is always stored in internal EEPROM. This is the only available ensemble for the case that no special BankStick is connected to the master core. It is not recommended to store data here, as it will be overwritten with each firmware update!

The "Ensemble BankStick" has to be assigned to CS#7 (Pin 1, 2 and 3 tied to +5V). It doesn't matter if it is a 32k (24LC256) or 64k (24LC512) type, as only a 32k range is allocated.

Please keep also in mind, that due to the high number of definable MIDI channels and splitzones (summa summarum up to 24 different instruments could be used if 4 multi or drum patches are selected) chances are high to loose the oversight. Therefore it makes sense to define some basic channel assignments in a special ensemble (e.g. E002) which matches perfectly with your gear (keyboards, sequencers), and to re-use it for different variations by selecting and copying it to other ensemble slots. Unfortunately it is not possible to name the ensembles, so you have to live with the numbers E001..E128.

Please keep also in mind, that due to special implementation reasons, only the sections of connected slaves will be stored into BankStick. This means: if you store an ensemble while not all slaves are connected to the CAN interface, the sections of the missing slaves won't be overwritten (but this could also be seen as an advantage ;-)

## Root Menu

Within the main screen, press the **LEFTMOST SELECT BUTTON** to enter the ensemble root menu.




It gives you quick access the submenus, which are described below.

The ensemble can be changed by pressing the PAGE UP/DOWN button within this screen (*). Moving the datawheel within this page. Patch and global parameter changes will take place

immediately for all SIDs.
Note that the ensemble can also be changed from the main screen by pressing the MENU button while moving the datawheel or pressing UP/DOWN.

## SID menu




- **SID:** selects the SID (1/2/3/4) which should be configured. Multi-Selections can be done with the four SID buttons (press and hold one SID button, while pressing additional SID buttons) - parameter changes take place on all selected SIDs.
- **Bnk (Bank):** selects the BankStick from which the patch is read. 8 BankSticks are provided: A..H, whereas H is normaly used for ensembles - reading patches from this BankStick is not possible.
- **Pat (Patch):** selects the patch, which can also be changed in the main screen. A 24LC256 (32k) BankStick can store 64 patches, a 24LC512 (64) BankStick 128 patches
- **Mon (Mono/Stereo Switch):** by default it is assumed that two SID chips are connected to a core. By activating the Mono switch, both SIDs play the same sound. This affects the engines the following way:
  - Lead: the sound of the left channel is also played on the right channel. Special settings for the right channel don't have any effect. The OSC detune parameter works slightly different: the first oscillator is not detuned.
  - Bassline: only one bassline available, it plays on both SIDs
  - Drum: only three voices available
  - Multi: only three voices available, static assignments to Voice 4/5/6 are redirected to Voice 1/2/3
- **SPV (Super Poly Voices):** this is currently an experimental feature which allows to play lead instruments with multiple SID cores in a polyphonic manner. The usage is not selfexplaining, and it requires some additional configuration to get it properly working.
  The superpoly handler couples 2, 3 or 4 SID cores (master/slaves). Only lead patches are supported by this function (yet), accordingly it isn't possible to play 8 voices individually, only 4 mono or stereo voices.
  Following configurations are possible:
  - SPV=off: no effect
  - SPV=1: superpoly mode with SID1 only - it doesn't make much sense
  - SPV=2: superpoly mode with SID1 and SID2 (2 voices)
  - SPV=3: superpoly mode with SID1, SID2 and SID3 (3 voices)
  - SPV=4: superpoly mode with SID1, SID2, SID3 and SID4 (4 voices)

  SPV, MIDI Channel, Split Zone and Patch/Bank number have to be set to the same values, this is not done automatically. Just select all SIDs in order to synchronize the settings. This can be done by selecting all (used) SIDs the same time with the SID buttons (press&hold SID1, press the remaining SID buttons, change values).
  Following audio sample demonstrates the possibilities. Note that 4 different tracks have been recorded, one after another (overdubbing). Each instrument (Strings/Bass/Lead Start, Main) is played by 3 stereo SID cores in superpoly mode (SPV=3):

  http://www.ucapps.de/mp3/midibox_sid/mbsidv2_superpoly_experiment.mp3

- **DOR:** Disable automatic Oscillator Reset during patch change. Could also be called DAORDPC flag ;-)
  Background: after patch change, MBSID normaly resets the oscillators to ensure, that a patch always sounds identical w/o unexpected phase cancelation effects between the oscillators. Other peoples prefer such effects, caused by freerunning oscillators. If you belong to those peoples, and know what it means, just activate this option - it will make you happy! ;-)

## CLocK menu

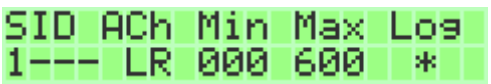
A global clock generator is available which sends clock events to the

- LFOs (when the CSn function is enabled - can be done for each LFO separately)
- ENVs (when the CSn function is enabled - can be done for each envelope separately)
- Arpeggiators
- Wavetables (if MOD flag not enabled)
- Trigger Matrix (1:1, 1:4 and 1:16)




- **SID:** selects the SID (1/2/3/4) which should be configured. Multi-Selections can be done with the four SID buttons.
- **BPM:** selects the tempo (beats-per-minute). This value is only used in master mode. In slave mode, the BPM is controlled from an external MIDI clock master.
- **Mode:**
  - **Mst (Master):** the internal BPM generator is used, which is running at 96ppqn resolution.
  - **Slv (Slave):** the clock generator quadruples the incoming MIDI clock and takes it as reference.
  - **Auto:** the clock generator switches automatically between master and slave mode. Slave mode is activated whenever a MIDI Clock command has been received. Master mode is activated when no MIDI clock has been received for ca. one second.
- **Out:** if enabled, MIDI clock events will be sent by MIDIbox SID depending on the selected tempo.
  This option allows to synchronize external MIDI devices (e.g. arpeggiators of other synths, or audio delays). However, if MBSID should be synchronized together with external sequencers, it's recommended to run it in slave mode, and to synchronize all your MIDI devices to a single MIDI clock master (which could be the DAW running on your computer, or a MIDIbox SEQ).

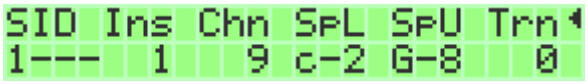
## FILter menu




- **SID:** selects the SID (1/2/3/4) which should be configured. Multi-Selections can be done with the four SID buttons.
- **ACh:** selects the audio channel: L (Left), R (Right) or LR (both)

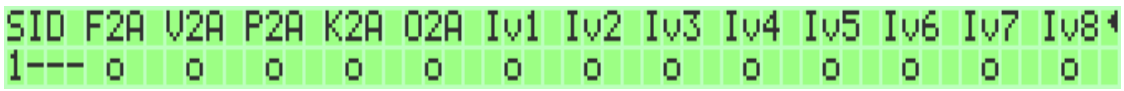
- **Min/Max:** this is a global, patch independent minimum/maximum value which scales the Cutoff between the given range. It allows you to calibrate your SID (especially the 6582 and 8580) for a suitable operation range. In general 000/FFF (full range) is fine for a 6581, 000/600 are recommended for 6582 and 8580.
- **Log:** this function applies a logarithmic curve over filter CutOff values which are sent to the SID. By doing so, it linearizes the CutOff of a 6582/8580. It should especially activated, if the keytracking function is used. It shouldn't be activated for 6581s.

## INStrument menu




- **SID:** selects the SID (1/2/3/4) which should be configured. Multi-Selections can be done with the four SID buttons.
- **Ins:** selects the instrument (1-6). Depending on the patch type, a different number of instruments is really used:
  - **Lead:** only settings of first instrument are used.
  - **Bassline:** the first instrument is used for the left SID channel, the second for the right SID channel. The remaining instruments are not used.
  - **Drum:** only settings of first instrument are used.
  - **Multi:** all 6 instrument settings are used
- **Chn:** specifies the MIDI channel (1-16) of the instrument.
  It's also possible to disable the MIDI channel (turn the encoder full-right) - in this case the instrument won't be played (default for Instrument 2-6).
- **SpL**/**SpU**: defines the lower and upper keyboard limit in between the instrument can be played. By changing the limits (and the transpose value) it's possible to realize keyboard zones, so that different instruments can be played on a single channel independent from each other.
- **Trn:** allows to transpose the instrument semitone-wise.

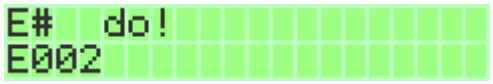
## EXTernal menu




- **SID:** selects the SID (1/2/3/4) which should be configured. Multi-Selections can be done with the four SID buttons.
- **F2A (Filter to Analog):** Cutoff and Resonance are forwarded to external CV outputs as well (assignments have to be done in the setup_*.asm file). Note that the effective Cutoff frequency **behind** the modulation path is taken. This allows you to easily replace the internal filter by an external.
  Please note also, that the filter range can be adjusted with the calibration parameters (min/max) of the ENS->FIL menu.
- **V2A (Volume to Analog):** Volumes are forwarded to external CV outputs as well (assignments have to be done in the setup_*.asm file). Note that the effective volume value **behind** the modulation path is taken - accordingly you are able to realize an external VCA to overcome the ADSR bug.

- **P2A (Pulsewidth to Analog):** Pulsewidth values of up to 6 oscillators are forwarded to external CV outputs (assignments have to be done in the setup_*.asm file) - the output behind the modulation path is taken.
- **K2A (Key to Analog):** Key values of up to 6 oscillators are forwarded to external CV outputs (assignments have to be done in the setup_*.asm file) - together with the gate functions (which have to be configured in setup_*.asm as well) this allows you to play external analog synths with the MBSID engines (e.g. bassline). The slide and accent flag can be forwarded to digital outputs at port CORE:J5 as well, giving you the possibility to control the analog part of a TB303 (or similar analog bassline synth).
- **O2A (Oscillator Frequency to Analog):** Similar to K2A, but the frequency behind the modulation path is taken. Portamento/Slide effect, Finetune and Pitchbender are added as well. This opens new possibilities to play your analog synth(s). E.g., lead engine: enjoy all the stereo modulation possibilities. Bassline: two basslines can be played per core. Multi: polyphonic playing of up to 6 analog synths.
  Disadvantage of this option: it changes the frequency scale to V/octave, accordingly the SID chip won't play enjoyable sound anymore. So, O2A can only be used in isolated manner (temporary mute the audio outputs of the SID module(s) so long O2A is activated, or use the core module without MBHP_SID modules - e.g. as MBSID slave)
- **Iv1..Iv8:** allows you to invert the analog output voltages. This option could save you from adding additional circuitry (e.g. it allows you to control the CutOff CV of a SSM2044 filter without inverting OPAmp stage).

## SAVe menu



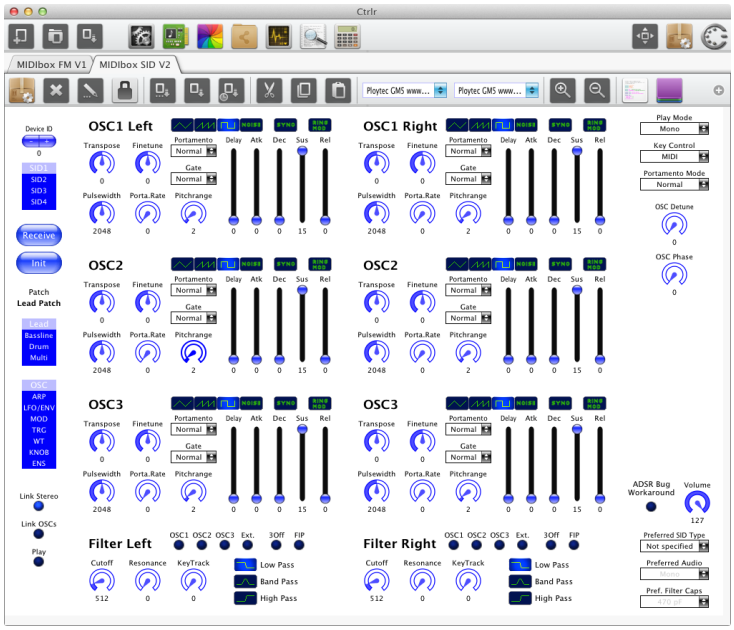

- **E#:** selects the ensemble which should be overwritten
- **do!** stores the ensemble.
  Note that naming is not supported.
  Only the configuration sections of available MBSIDs will be stored.
  The last stored ensemble will be reloaded after power-on.

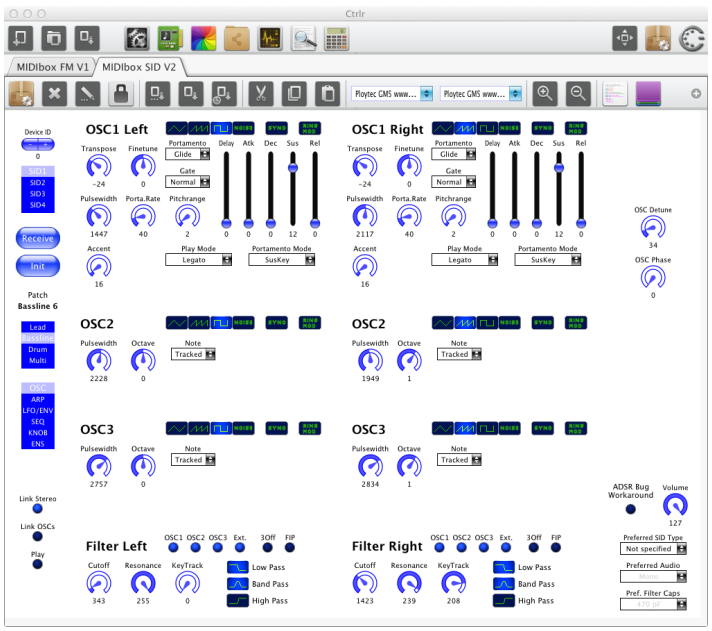
# SysEx Editor

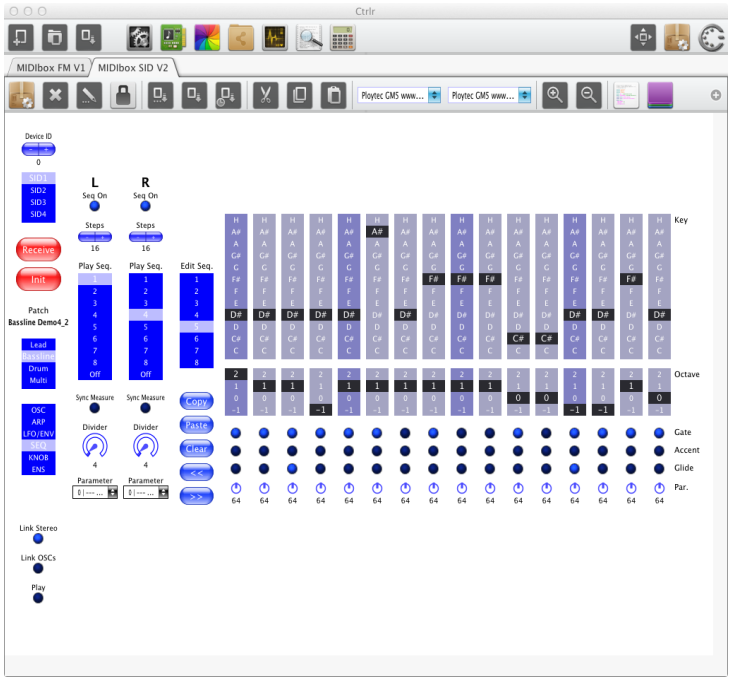
## Introduction

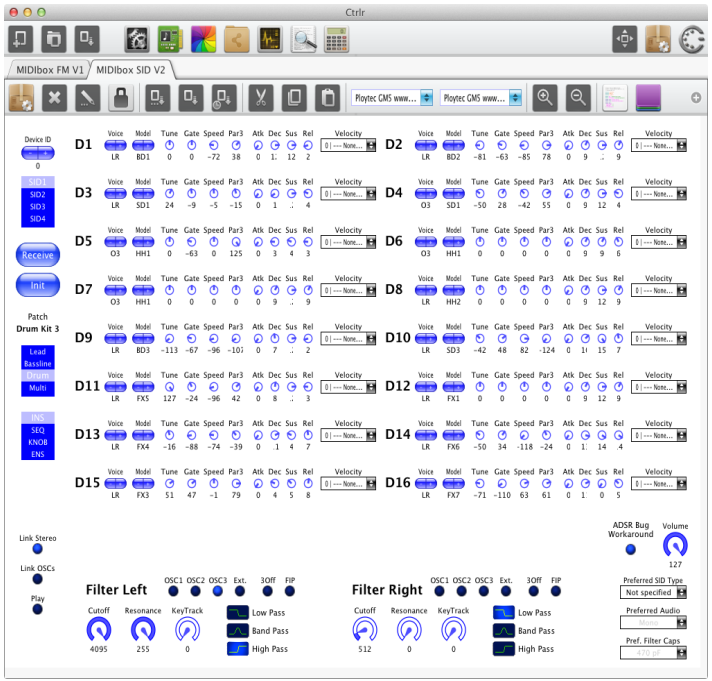
A Ctrlr based SysEx Editor has been created which gives you access over all engines remotely from your computer.

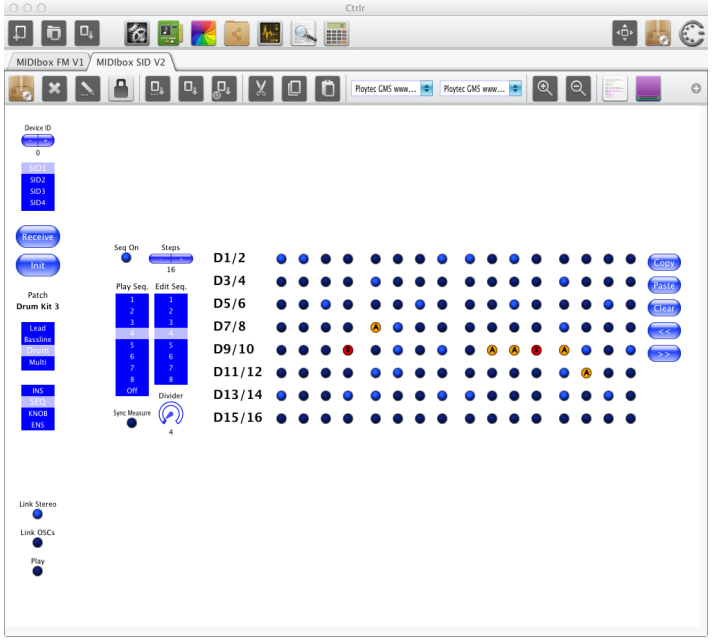
Here some teasing snapshots:

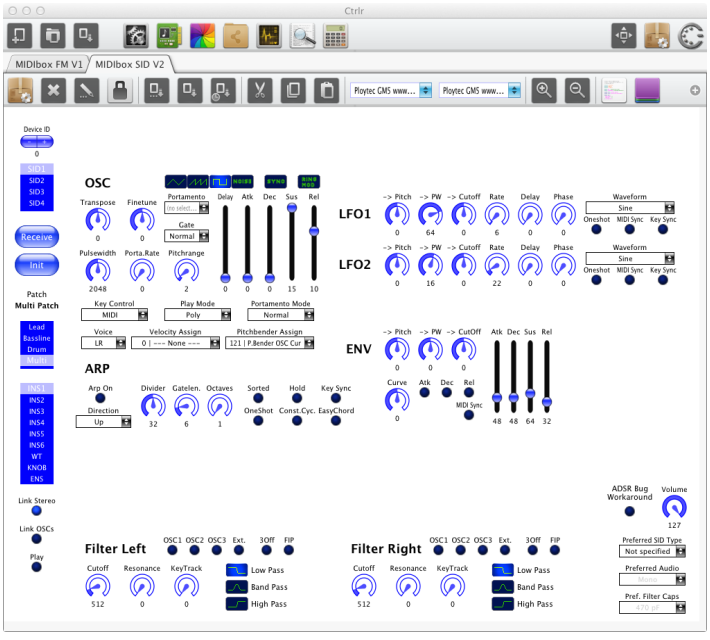












Ctrlr works standalone, but also as AU and VST plug-in under MacOS, Windows and Linux!

The panel is part of the the Device Data Base, so that it can be downloaded directly from Ctrlr (or from this webpage)

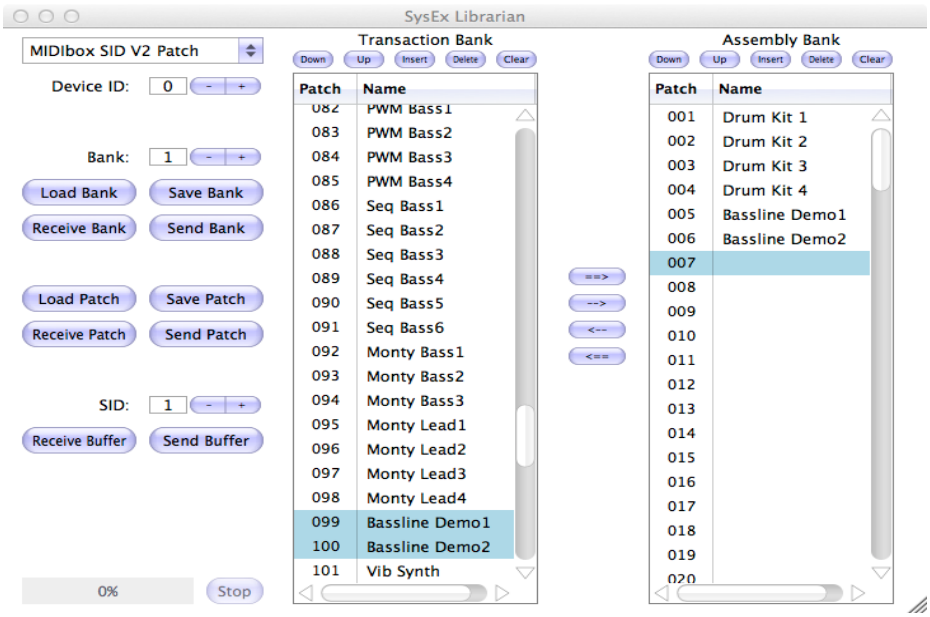
Please follow this forum article for the latest informations.

## Alternative Editors

- a Java based editor has been implemented by Rutger. It's older than the Ctrlr based solution and doesn't work correctly under MacOS 10.7 and higher. It isn't maintained anymore, but since it's complete and nice to use, here a link to the latest version: V0.5 release from 2009-02-18 (470k). Link to related forum article.
- Nils implemented "Patch Turkey", a preliminary version can be downloaded from this forum article. It has a great graphical user interface, but only supports the Lead engine and only works under Windows.

# SysEx Librarian

MIOS Studio has an integrated SysEx Librarian which allows to upload/download/load/save/organize banks of various MIDIboxes, such as MIDIbox SID V2:



# -- CV Options

## Introduction

MIDIbox SID V2 is the worldwide first (and only) SID synth which allows you to control external analog components from the synth engine. Meanwhile the control options are so powerful, that it even allows you to control analog synths from the mighty sound engines, which gives you much more possibilities than common MIDI-CV interfaces! Note that it isn't required to connect MBHP_SID modules to use these features.

## AOUT modules

Three different AOUT (Digital-Analog converter) modules are part of the MIDIbox Hardware Platform, and supported by the firmware:

- MBHP_AOUT: MAX525 based module - expensive!
  AOUT interconnections
- MBHP_AOUT_LC: 74HC595 based module - cheap but a lot of soldering effort.
  AOUT_LC interconnections
- MBHP_AOUT_NG module: TLV5630 based - 8 channels, best price!
  Interconnection Diagram not available yet, but similar to AOUT interconnections

In order to select the right AOUT driver, the setup_*.asm file of the release package has to be edited (search for ENABLE_AOUT); thereafter a new .hex file has to be build as described here.

## x2A

Analog outputs can either be controlled directly from the EXT menu, and the modulation matrix of the Lead Engine, or you can enable one or more x2A function within the Ensemble to forward waveforms, generated by the synth engine, to the analog outputs.

The AOUT channel assignments for x2A functions are static, and need to be selected in the setup_*.asm file.

## Digital Switches (Gates)

In addition to the analog outputs, the firmware allows to control 8 digital outputs (e.g. switches, gates, triggers) as well. The state of the outputs is stored within the patch structure, so that the setup can be recalled. In addition, the outputs are accessible via control surface (-> EXT menu), NRPNs, Knobs, Wavetable sequencer, etc...
The outputs are either available at J5 of the core module (set DEFAULT_J5_FUNCTION to 3), and/or on a DOUT shiftregister (set ENABLE_AOUT to != 0, and set DEFAULT_EXT_SWITCH_DOUT to a value between 1 and 16).

Optionally, internal gate/slide and accent signals can be forwarded to the digital outputs as well, which gives you the possibility to trigger external synths the same way like the 2*3 SID oscillators (per core).

## Using external Filters

You are able to add external filters or other analog effects to the SID channel. The output levels are stored with the patch, and it is even possible to control them via MIDI, and to modulate the outputs from the modulation matrix. This opens a lot of new possibilities for sound experiments with the good old SID chip.

In following demo sample, cutoff and resonance of two CEM3378 filters are controlled from MIDIbox SID V2 (see also this posting for additional infos):

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_cv_out.mp3

During this session, two SSM2044 via AOUT_NG module were used to filter the outputs of two SIDs.

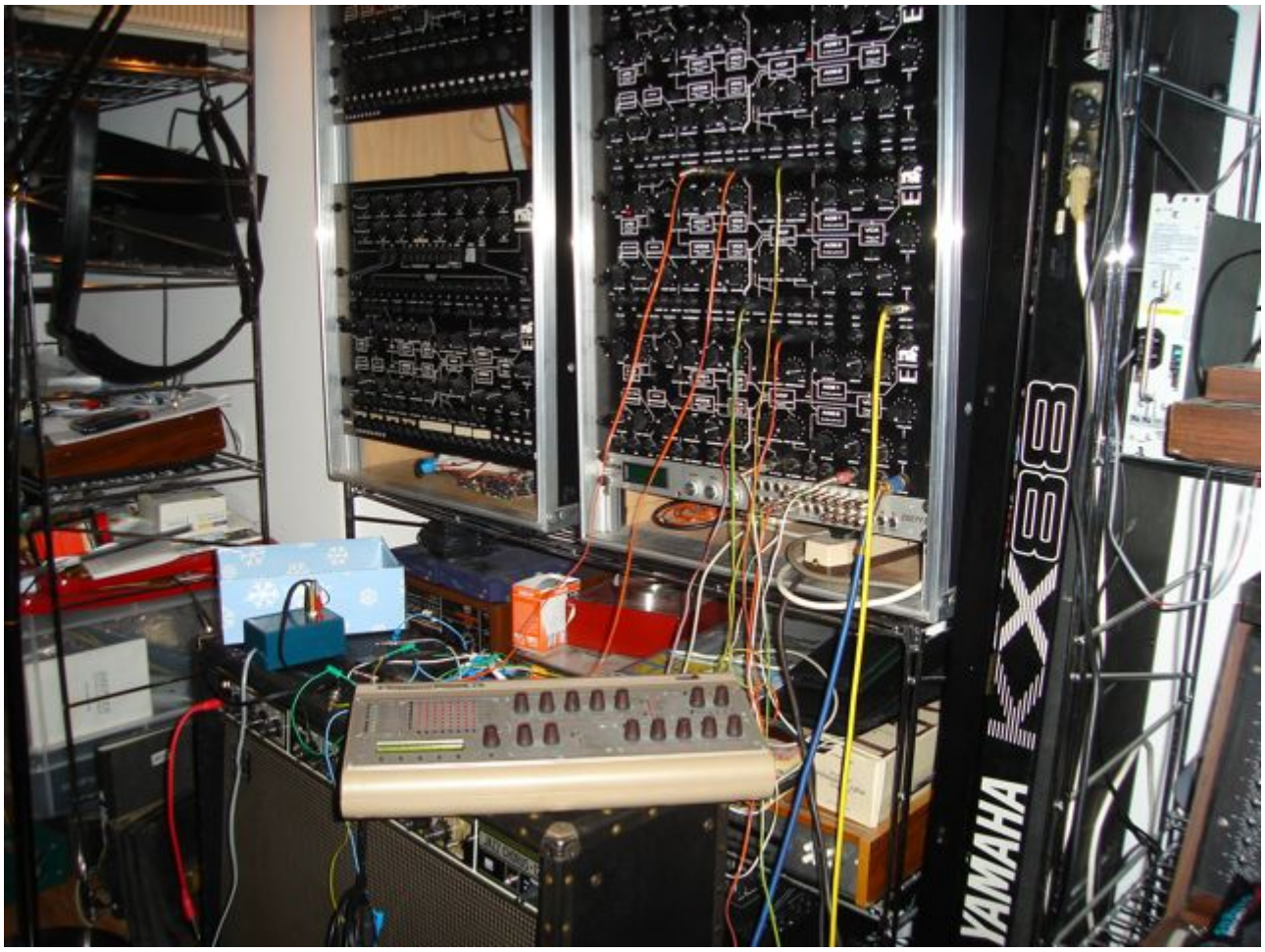
http://www.ucapps.de/mp3/midibox_sid/mbsidv2_bassline_demo4.mp3

Forum Article

An envelope modulated multi patch played with two CEM3378 filters:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_cv_out2.mp3

## Controlling Analog Synths




More informations in this and this posting.

Sound samples, which have been created during a sunday-afternoon session with rsf Kobol Expanders:

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_kobol_modulation.mp3

**Experiments with Stereo Modulation**
*Only for synth freaks with some endurance! LP Filter and Pitch of two Kobols are modulated through the mod matrix with multiple LFOs, combined with different operators. LFO settings are tweaked during the sound is playing. Only a single note is played! The random-like "tunes" are result of the modulation (in fact, the gate was not connected this time) I especially like the brass-like sound at 4:22*

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_kobol_bass_stereo.mp3

**Stereo Bass**
*Just to check the detune function and simple stereo effects*

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_kobol_arp.mp3

**Arpeggiator**
*Nice arpeggiator sequence (just push the button and play a single note)*

http://www.ucapps.de/mp3/midibox_sid/mbsidv2_kobol_bassline_fun.mp3

**Bassline Fun**
*Bassline sequences. Note the killer bass of the lower bassline!*

# SwinSID Review

SwinSID is an Atmel AVR based SID emulation created by Swinkels. Initially it was intended as replacement module for the original SID chip housed in a C64, but it works fine with MIDIbox SID as well.

Meanwhile Swinkels has enhanced the firmware dramatically, so that it can be used as inexpensive Stereo-SID option for MIDIbox SID. Even though a 1:1 emulation cannot be expected from a microcontroller without DSP capabilities (however, PC/Mac based solutions are not capable of emulating the SID accurately as well), it is still an attractive project for people who are not able to find enough SIDs for a fully stuffed MIDIbox SID, for people who are searching for alternative/additional sounds, and/or for people who cannot resist to build every SynthDIY project available on the web. :)

## The Hardware

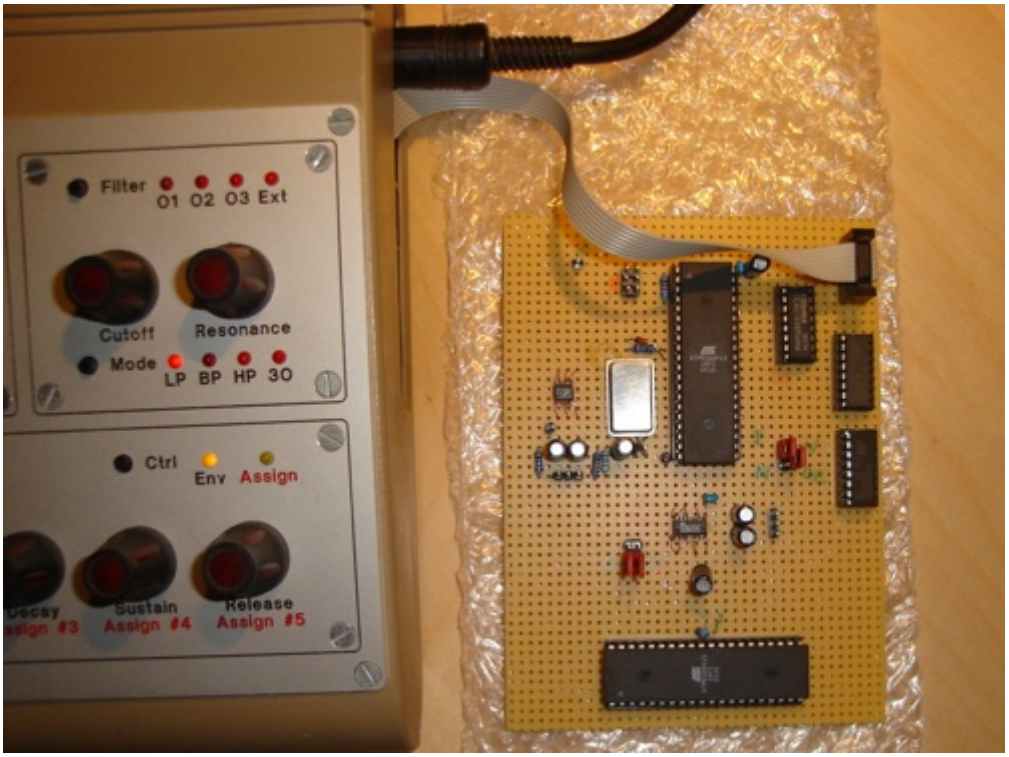
The SwinSID firmware is running on a single ATmega8515 (Mono), or two ATmega8515 (Stereo). Each AVR handles the complete SID emulation independently at a sampling rate of 31.25kHz. In order to achieve such a high rate, the AVRs are overclocked at 24 MHz!

The DAC is connected to the "master" AVR only. The "slave" AVR transfers the sample words of the right audio channel to the master via a simple parallel interface.

Each AVR has a dedicated chip-select input to differ between SID register accesses directed to the left, right or both audio channels, which qualifies this solution perfectly for MIDIbox SID. Like

known from the MBHP_SID module, two 74HC595 are required as serial->parallel interface to MIDIbox Core.

Note that thanks to the interface compatibility, it is possible to control two MBHP_SID modules, and one Stereo-SwinSID module from the same MBHP_CORE in parallel!




A picture of the prototype. Note that at the second AVR has been added later (a stereo option wasn't planned at the beginning) - therefore it's placed with a different orientation.

Virtually all 16bit I2S DACs are compatible to SwinSID, even newer ones which require a second "system clock" (256xsampling frequency) for digital interpolation. Swinkels is using a TDA1543 (unfortunately discontinued, but still available e.g. at Voti), I tested a TDA1387 (poor quality, cannibalized from an old Soundblaster AWE32 soundcard), and a PCM1725 (best quality, available e.g. at Farnell). TDA1387 and PCM1725 require SMD soldering-skills, whereas TDA1543 comes in a DIY-friendly Dual-Inline package!

## Pros & Cons

As a fan of the original SID for more than 20 years I would like to give my personal statements about SwinSID, and provide some demo samples for some first impressions.

After I got the hardware up&running, and listened to the first sounds, I was a bit dissapointed from the sound quality due to the aliasing and distortion effects - something what I haven't experienced with the original SID. Meanwhile I've learned, that such effects are mainly related to the sampling frequency, oversampling factor (to reduce disharmonics on higher oscillator frequencies) and interpolation rate (to reduce disharmonics on lower oscillator frequencies). All synthesizers, which are based on the sample technology, are affected by such issues, and the only counteractive measure is to spend as much compute power as possible.

On the other hand, I realized that Swinkels did his best to bring the emulation to the max. I find his implementation superior to other AVR based synths like the well known "AVR Synth" and "AvrX", when I compare the sampling frequency, DAC resolution, number of oscillators, and especially the quality of low pass filter.

IMHO a better sound quality can only be achieved with less DIY-friendly and much more expensive DSP or FPGA based hardware.

**PROs:**

- **+** Inexpensive and easy to build replacement for SID chip
- **+** Hardware-Compatible to original SID (no software adaptions required)
- **+** Stereo Option with two AVRs
- **+** Possibility to control two MBHP_SID (stuffed with original SID chips) and one Stereo-SwinSID module in parallel from a single MBHP_CORE
- **+** Realtime synthesis with ZERO latency!
- **+** More waveforms
- **+** White noise generator
- **+** All three oscillators provide a sub-oscillator function with selectable waveform; waveform can be reversed.
- **+** FM between oscillators (instead of ringmodulator function)
- **+** Perfectly working ADSR (the ADSR of the original SID has flaws!)
- **+** Fat low-pass filter

**CONs:**

- **-** Aliasing Effects
- **-** Not 100% sound compatible (e.g. Noise waveform, sync function)
- **-** No multi state filter (for Bandpass/Highpass/Notch)
- **-** No ringmodulation

## Demo samples

Following samples have been created with SwinSID, controlled from a MIDIbox SID V2.

http://www.ucapps.de/mp3/swinsid/swinsid_wizardry.mp3

**Wizardry**
Demonstration of the available waveforms (w/o noise waveform) *[1]*
The tune is based on the C64 game, .sid file taken from HVSC, converted with sid2midi, played with the Multi Engine. Song Author: Mike Alsop

http://www.ucapps.de/mp3/swinsid/swinsid_fm_seq.mp3

**FM sounds**
A demonstration of the Frequency Modulation feature. Each oscillator can act as modulator and/or carrier. From time to time the pitch is slowly changed while the tune is playing to demonstrate different FM timbres. *[2]*

http://www.ucapps.de/mp3/swinsid/swinsid_preview1.mp3

**Filter Fun**
A recording made during prototyping phase which demonstrates the self resonating filter *[2], [3]*

http://www.ucapps.de/mp3/swinsid/swinsid_preview2.mp3

**Bassline Fun**
SwinSID controlled from the Bassline Engine. In the background FM modulated oscillators through a reverb effect. *[2], [3]*

http://www.ucapps.de/mp3/swinsid/swinsid_drums.mp3

**New food for the drum machine**
SwinSID allows you to create very nice Drum Kits, which sound different to the original SID, but not less interesting for fans of electronic drums and percussions *[2]*

[1] raw sound output of SwinSID
[2] external effects have been added
[3] StylusRMX based drums have been added

# V1->V2 Upgrade

This chapter lists the requirements for a hardware upgrade from MIDIbox SID V1 to the new MIDIbox SID V2 firmware.

- PIC18F452 needs to be replaced by a PIC18F4685. This is a quite new microcontroller, which is unfortunately hard to find in common webshops yet.
  SmashTV did a batch order and provides the PIC with preburned Bootloader and MIOS for an extremely fair price to the community (**$10.95 !!!** Compare it with stock listings under findchips.com and you won't find cheaper offerings for single parts!)
  Doc is doing the distribution for Europe for those guys who want to save shipping costs - just contact him in the MIDIbox Forum
- If your core module is based on MBHP_CORE_V2, please add R12 (the additional 1k pull-up resistor for IIC modules) like shown in the MBHP_CORE_V3 schematic.
- If your SID module is based on MBHP_SID_V2, please follow these instructions.
  *Hint:* If it was already working with v1.7303, then there is no need for changing the MBHP_CORE->MBHP_SID interconnections.
- If you are planning to use the stereo option (two SID chips controlled from a single core - recommended for a great sound experience!), just connect the second module like the first one. Only the CS# line of the second SID has a different routing (MBHP_SID::J2:SO -> MBHP_CORE::J14), so that individual accesses are possible.
- Disconnect the four data lines D0-D3 of your LCD and let them unconnected (open). The LCD is accessed in 4bit mode to free the ECAN pins (see also mbhp_lcd_4bit.pdf)
- Add a 1k pull-up to the CAN receive input as shown in this schematic.
  This resistor is even required if no slave module is used, otherwise the firmware could crash during the boot phase!
- if you have used 32k BankSticks (24LC256) in the past, it makes sense to replace them by 64k BankSticks (24LC512), so that 128 patches can be stored in each bank! Note, that the 8th BankStick is used to store ensembles, 32k are sufficient here.

Following changes are only required if slaves are connected to the master core:

- Change the MIDI connections to the slaves as shown in this schematic, so that all slaves receive the same MIDI In data stream like the master core (J11:MI inputs of slaves connected to J11:MI of master)
- Add the CAN bus connections as shown in this schematic. Each core requires a diode (e.g. 1N4148)

Thats all - have fun! :-)

## Partlist for a complete MIDIbox SID V1 upgrade

Assumed, that you own a completely stuffed MIDIbox SID V1, following parts are required for the upgrade:

- 4 * (or 2 *, see below) PIC18F4685
- 1 * 1k resistor
- +4 * 1k resistors if your core modules are based on MBHP_CORE_V2 schematic
- 4 * 1N4148 diode
- +3 * 24LC512 (assumed, that 3 * 128 patches are sufficient for your needs, old 32k BankSticks can still be used, they will store 64 patches)

Plus 4 * MBHP_SID modules if you are so crazy like those other guys who always want to bring their MIDIbox to the max! ;-)

If you don't want to spend the money for additional modules, my strong recommentation is to connect 2 MBHP_SID modules to the master and a single slave, and to pass on the two unused core modules, so that only two PIC18F4685 are required.
With such a configuration the upgrade costs you about US $20..$30 (+ shipping)

# Voir aussi (https://github.com/) :

MIDIbox SID V2 Parameter Chart
MIDIbox SID V2 CC Implementation Chart
MIDIbox SID V2 SysEx Implementation

# Known Issues and Troubleshooting Tips

This is a "living document", which will be extended from time to time whenever a "strange behaviour" has been reported by a user which could be interesting for other people, not at least for troubleshooting an apparently non-functional MIDIbox SID.

- *[1]* **PIC not starting, no MIDI output/input, code upload not possible, etc...**
  Please follow this Troubleshooting Guide
- *[2]* **Blank screen**
  LCD won't be initialized so long MIOS hasn't been uploaded via MIOS Bootstrap Loader
- *[3]* **Core reboots after MIDIbox SID application has been uploaded, or it behaves randomly**
  - *[3a]* Use the latest MIOS Studio version.
  - *[3b]* Ensure that following pull-up resistors have been mounted on the core mode (Schematic): R2, R6, R9, R10, R12 --- in general it makes sense to stuff the whole core module if you are unsure.
  - *[3c]* Ensure that in addition the 1k pull-up and 1N4148 diode for the CAN interface has been added as shown in this schematic.
    Resistor and diode are even required if no slave module is used!
  - *[3d]* Ensure that the D0-D3 data lines of your LCD are not connected to the core module (LCD is accessed in 4bit mode to keep the ECAN Rx/Tx pins free)

- *[18]* **LCD freezes immediately or after some seconds, synth still functional**
  A rare number of LCDs don't support 4-bit access mode properly. This has been noticed on two PLED displays and one VFD display so far, "normal" HD44780 or KS0066 LCDs are working stable. Sometimes nibbles are swapped, so that the communication with the CLCD driver of MIOS is disturbed. 4-bit mode is required, as pin RB2 and RB3 are used for the CAN interface. If you notice such an issue, you could activate the alternative "custom" LCD driver, which is already part of the MIDIbox SID V2 firmware. It uses 8-bit mode by using two pins on port E to replace the two pins being used by the CAN. LCD pins D0 and D1 should be connected to PIC pins RB0 and RB1, and LCD pins D2 and D3 connect to PIC pins RE1 and RE2. The driver can be permanently activated via PIC ID Header (LCD type 7, see also Bootloader page), or by activating the DEFAULT_USE_CUSTOM_LCD_DRIVER switch in the setup_*.asm file
- *[4]* **Slave Cores not found by Master**
  Ensure that each slave has its own MIOS Device ID (01, 02, 03). The ID of the master core has to be 00. The ID has to be programmed into the PIC ID Header. If this hasn't been done while burning the Bootloader, you can do this also with the change_id application, which is available at the MIOS Download page.
- *[5]* **Buttons/Encoders not working**
  Please follow this Troubleshooting Guide
- *[6]* **No sound from SID**
  Try the mbsid_interconnection_test and sid_testtone application, which can be found at the MIOS Download page. The testing procedure is explained in the main.asm file.
  Just to highlight it again: the application won't run before you've uploaded MIOS (the operating system).
  The audio output of the MBHP_SID module has to be connected to a mixer or audio amplifier, a direct connection to a loudspeaker or headphones won't work.
- *[7]* **Still no sound from SID**
  Check the MIDI channel. If you are unsure (e.g. no LCD connected to Core Module), try all 16 MIDI channels with your keyboard/sequencer
- *[8]* **Background noise**
  Ensure that the Audio In of the MBHP_SID module is connected to ground when not used.
  A certain digital background noise is common on a 6581 due to a design weakness. It's much less (-90 dB) on a 6582/8580.
  Another known SID bug is the leackage issue in the VCA part (especially 6581) - the oscillators are not completely silent once a note has been played. If you are using a compressor a noise gate in the Fx chain, search for a setting which eliminates this noise
- *[9]* **50 Hz or 60 Hz buzz**
  Consider to build an optimized PSU (Power Supply Unit) like explained in the Hardware Options chapter.
- *[10]* **Some buttons or LEDs are not working**
  Depending on the selected sound engine, some buttons and LEDs have no function. E.g., in Multi mode, OSC and ENV is always 1, and LFO toggles between 1 and 2. The modulation matrix buttons have no function at all, and the matrix LEDs display the number of the selected instrument.
  For a complete CS test, select a patch which uses the Lead engine.
- *[11]* **Modulation matrix doesn't show matrix connections, but flickers strangly**
  It's not a bug, but a feature! ;-)
  Maybe you've selected the meter mode, which displays the MOD output values (e.g. LFO

and ENV output) - press two column buttons of the modulation matrix to switch back to normal display mode.
When the bassline/drum or multi engine is running, the LED matrix displays level meters instead, which allow you to monitor the voice allocation.

- *[12]* **Meter display mode doesn't display MOD/level meter output of slaves.**
  This is not possible due to conceptional reasons - it only displays the MOD outputs of the master SID
- *[13]* **Some mixed oscillator waveforms don't output a sound**
  Thats normal on a 6581, it works on a 6582/8580 - see also the demo samples in the Lead Engine chapter
- *[14]* **Some patches of the preset banks are incomplete**
  Ensure that you've uploaded the .syx files with a delay of 750 mS between each SysEx string. See also the README.txt of the patches/ directory for configuration details
- *[15]* **Knob values of master or slaves are changing randomly**
  Propably you've selected the "analog input option" (DEFAULT_J5_FUNCTION set to 1) in order to connect pots/faders/joysticks to the master, and forgot to clamp all unusued analog inputs to ground (see also auaimbctg.pdf).
  Due to the cloning mechanism, which transfers the firmware of the master to the slaves, this requirement has to be considered for all core modules!
  Clamping unused J5 inputs to ground is not required if DEFAULT_J5_FUNCTION is set to 0 (J5 disabled --- default!) or 2 (J5 used as digital output)
- *[16]* **I haven't found a way to play a 8xSID MBSID V2 as a single polyphonic synth**
  A special "Super-Poly" mode is planned for a post v2.0 release - see also this article.
  Update: here.
  Update (2007-09-30): An experimental solution for 4 stereo lead voices has been implemented, see Ensemble Page.
- *[17]* **I haven't found a way to synchronize the internal clock of all 4 cores**
  This is related to [16], a possibility will be provided later after some evaluations. Currently wavetable and bassline/drum sequencers can only be synched with an external MIDI clock - this works quite well, and is the preferred solution anyhow if you are planning to sync MIDIbox SID with an external sequencer.
- *[18]* **Multi/Drum engine: one or more voices are not used when Multi/Drum instruments are played - why?**
  This is related to the enhanced voice allocation algorithm - check the VAs settings in the CFG/DRM menu page. If one or more instruments are explicit assigned to O1..O6, the appr. voice cannot be allocated by the remaining instruments anymore (there is only one exception: if all voices are assigned to specific oscillators in drum mode, the "exclusive reservation" will be ignored).
  This approach is important for a static filter or sync/ringmodulation routing, and therefore no bug.
  The VAs (Voice Assignment) setting can be found in the CFG (multi engine) or DRM (Drum Engine) menu. It can be easily overlooked when doing modifications on patches from other users.
  *Tip:* if you own a MIDIbox SID with LED matrix, enable the level metering mode to monitor the voice allocation (works for Multi and Drum engine, when the patch is played on the master core SID1)
- *[19]* **An envelope doesn't sweep through the whole value range:** yes, this is true if only the ENV depth is set to maximum level (+127). With the default configuration of the

modulation matrix, this will result into a quarter of the available value range.
E.g. if the filter cutoff is set to 0, ENV depth set to 127, the filter will only sweep between 0%..25%. Same for volume.
In order to sweep through the whole range, the modulation path parameters have to be changed:

- enter the MOD page
- select MOD 1 (if not already done) for EG1
- you will notice, that EG1 is only assigned to Sr1 - assign it also to Sr2, and ensure that the "1+2" operator is selected
  -> this will double the ENV level
- in addition, set the Depth parameter of the modulation path to 127 -> this will double the MP level

As the result, you will get an envelope which sweeps through the whole value range from 0% to 100%

- *[20]* **The AOUT channel is always at maximum level when the V2A option is used:** because the volume parameter in the CFG page sets the offset for this AOUT routing option. Change the volume to 0, so that the AOUT channel offset starts at 0V (similar to CutOff).
  Note that with V2A enabled, the SID volume will be automatically set to maximum and can't be influenced with the Vol parameter anymore.

## En lien :

**MIDIbox SID V2**
http://www.ucapps.de/midibox_sid.html
Page wiki projet **SammichSID**
http://www.midibox.org/dokuwiki/doku.php?id=sammichsid
**MIDIBoxSIDKontrol VST**
http://midibox.org/forums/topic/13740-midiboxsidkontrol-vst/
**Emulator control VST et AU**
http://midibox.org/forums/topic/13759-emulated-midibox-sid-au-and-vst-for-macwindows/
**HardSID dll**
http://www.midibox.org/dokuwiki/doku.php?id=hardsid.dll

__ **Rappel MIDIbox SID V.1 Manuel** :

http://www.midibox.org/dokuwiki/doku.php?id=midibox_sid_v1_manual