

# UN CLAVIER MAITRE MIDI SOPHISTIQUE DANS MAX

*Laurent Pottier*

CIEREC – EA3068

Université Lyon-Saint-Etienne

laurent.pottier@univ-st-etienne.fr

## RÉSUMÉ

Les claviers maîtres MIDI ont été des outils indispensables pour la scène dans les années quatre-vingt/quatre-vingt-dix pour permettre aux musiciens jouant des claviers de piloter à distance des expandeurs.

Le projet présenté décrit la programmation dans Max d'un des plus prestigieux claviers maîtres de cette période : l'ELKA MK88 (ou MK76). Le programme réalisé (« ELK-A1H ») permet, à partir de n'importe quels claviers MIDI (ou d'autres contrôleurs MIDI) et d'un ordinateur, de disposer d'outils sophistiqués permettant de réaliser facilement et rapidement des presets pour commander des synthétiseurs qui peuvent être distants (hardware) ou en local sur l'ordinateur (software) pour la scène. Tous types de paramètres MIDI (notes, contrôleurs, changements de programme, système exclusif) peuvent être accessibles immédiatement en appelant divers presets. Le texte présente l'architecture du programme, l'interface utilisateur, l'utilisation de JavaScript dans Max et le fonctionnement des mémoires.

## 1. INTRODUCTION

Dans le domaine de l'informatique, et en particulier de l'informatique musicale, les développeurs de logiciels, les créateurs et les musiciens peuvent avoir plusieurs comportements : le comportement historiciste et celui de la table rase. Cela dépend de la façon dont ils abordent les problèmes qui sont liés d'une part à l'obsolescence très rapide du matériel (hardware électronique, ordinateurs, versions du système, version des programmes...), et d'autre part au fait qu'en permanence de nouveaux outils et de nouveaux langages apparaissent dont les possibilités, en terme de vitesse d'exécution, de possibilités de programmation, et d'interfaces utilisateurs ne cessent de se renouveler.

La question est donc de savoir si les outils que nous développons doivent avoir une certaine pérennité dans le temps et de savoir si lorsque nous développons de nouveaux outils, de nouvelles créations, de nouvelles interprétations des œuvres, nous devons nous inspirer des outils existants ou repartir de zéro pour être plus en phase avec les technologies d'aujourd'hui, différentes de celles d'hier. Cette question est d'autant plus cruciale dans le domaine de la musique où quand il s'agit de création, certains prônent des attitudes historicistes (se servir des réussites du passé pour aller de l'avant) là où

d'autres parlent de table rase (partir de zéro et tout réinventer).

Dans le cadre de plusieurs projets de recherches récents, nous avons été amené à mettre à jour des technologies, des dispositifs pour la création qui n'étaient plus disponibles pour les rendre utilisables aujourd'hui et de préférence demain, grâce à des technologies plus pérennes [1][2]. Si nos derniers travaux ont porté plutôt sur les aspects DSP, d'autres plus anciens, souvent liés à des productions musicales portaient sur le développement d'interfaces utilisateurs, permettant d'adapter des dispositifs gestuels pour le contrôle de processus de synthèse et de traitement du son [3].

Dans ce texte nous décrivons la réalisation d'un programme dans Max/MSP modélisant un outil numérique hardware qui nous a accompagné pendant de nombreuses années sur scène en tant qu'interprète sur des dispositifs électroniques, et dont nous n'avons pas encore trouvé d'équivalent en terme d'efficacité, de convivialité, d'ergonomie voire de programmabilité. Il s'agit du clavier maître de type MK de la firme italienne ELKA. L'objectif sera dans un second temps d'en réaliser une version en JavaScript, qui soit accessible sur le Web dans tout type de navigateur.

## 2. LES VOCATIONS PREMIERES DE MAX

Lors de sa création en 1986 par Miller Puckette, Max (alors encore appelé « patcher ») servait à contrôler par MIDI l'ordinateur DSP 4X de l'IRCAM pour permettre de visualiser et de modifier rapidement les paramètres de la synthèse et du traitement du son pendant les répétitions et les concerts et ainsi éviter le recours à des ingénieurs qui devaient auparavant, pendant des temps généralement assez long, reprogrammer la 4X quand un paramètre ne convenait pas [4].

L'idée d'avoir un outil avec une interface graphique, programmable, permettant de piloter à distance des équipements MIDI était à l'époque suffisamment séduisante pour que l'IRCAM collabore avec la société Opcode (USA) pour faire de ce logiciel un outil commercial, même s'il n'était pas courant à cette époque d'employer des ordinateurs sur scène. Max n'était pas utilisé uniquement pour contrôler des équipements MIDI à distance, il pouvait aussi servir à élaborer des processus pour composer de la musique algorithmique.

### 3. CLAVIERS MAITRES YAMAHA, ELKA

Dans le domaine des musiques utilisant les sons électroniques en direct, très développés notamment dans les musiques de rock progressif et de jazz rock, avant l'apparition du MIDI, les claviéristes étaient obligés de s'entourer d'une quantité de matériel considérable (Klaus Schulze, Keith Emerson, Rick Wakeman, Chick Corea pour les plus réputés d'entre eux) pour pouvoir d'une part disposer d'une palette de sons suffisamment large et variée et d'autre part pouvoir les contrôler en direct et simultanément.

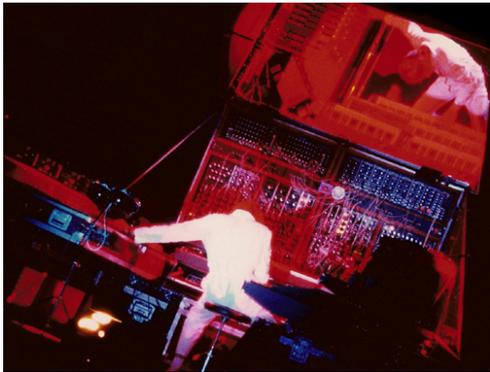


Figure 1. Klaus Schulze en concert (Paris – 1977 © L. Pottier).

Avec le MIDI, sont apparus les expandeurs (FB01 Yamaha en 1986), des synthétiseurs sans claviers, multitimbrés, rackables et pilotables à distance grâce à des claviers maîtres, dont les plus célèbres, au toucher lourd, proviennent de la série KX de Yamaha (Japon) et de la série MK d'ELKA (Italie) et par la suite des séries Fatar (Italie) plus économiques.

Le clavier maître ELKA est un clavier de 88 touches (76 pour le MK76) au toucher lourd construit autour de son fly-case [5]. Il est entièrement programmable avec 6 zones de splits, 18 courbes de vélocité, une entrée MIDI (1 IN) et deux sorties MIDI dédoublées (4 OUT), trois curseurs assignables, deux molettes, une vingtaine de boutons, deux pédales à contrôle continu et quatre pédales switch. Le MK88 dispose de 64 mémoires en interne, extensibles avec 64 mémoires en RAM. Ce qui fait son principal intérêt est le tableau de commande qui affiche toutes les pages disponibles pour la programmation d'une mémoire et qui permet un accès instantané à tous les paramètres.



Figure 2. Panneau de contrôle du clavier ELKA.

Les claviers maîtres Yamaha de la série KX étaient plus répandus, mais réputés pour une certaine austérité dans la programmation, difficilement réalisable sans disposer du manuel de l'utilisateur à proximité.

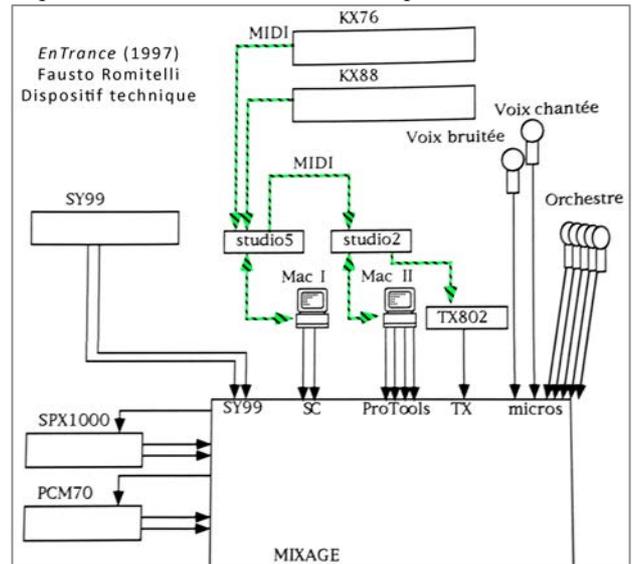


Figure 3. Exemple du dispositif technique d'un concert IRCAM utilisant deux KX Yamaha [6].

### 4. CLAVIERS MAITRE DANS MAX : EXEMPLES

Dans Max, il est assez facile de réaliser sur mesure un programme permettant de piloter des instruments MIDI. Dans les enseignements que je réalise à l'université de Saint-Etienne<sup>1</sup>, c'est d'ailleurs le premier exemple que je donne à programmer en général aux étudiants de licence de musicologie :

Exercice :

Créez un clavier Maître avec Max avec trois zones de split

- une pour les percussions sur do1-mil,
- une pour la basse sur fal – si2,
- une pour un accompagnement polyphonique sur do3-si5.

Sur la partie polyphonique, trois sons sont utilisés, un son percussif (piano, marimba ou guitare), un deuxième son, entretenu, avec une attaque lente (de type « cordes » par exemples) et un troisième son qui n'apparaît que pour des vélocités supérieures à 90 et qui est transposé à la quinte.

Avec ce dispositif, suivant les modes de jeu utilisés (legato, staccato, appuyé ou léger), on dispose déjà d'une palette sonore assez large et variée. De nombreuses variantes peuvent être imaginées sur ce thème.

Le dispositif est très simple à programmer dans Max (cf. Figure 4).

<sup>1</sup> cf. <http://musinf.univ-st-etienne.fr/>

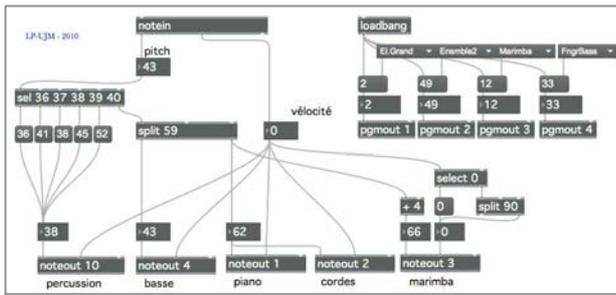


Figure 4 : petit exercice de programmation MIDI dans Max.

L'inconvénient d'un tel programme est qu'il doit être reconstruit à chaque fois qu'un résultat différent veut être obtenu.

L'objectif du programme ELK-A1H était donc de réaliser un outil programmable avec des mémoires. Dans ma pratique de Max, j'ai déjà eu à réaliser à de nombreuses reprises des outils plus ou moins génériques dans cette optique, capable de diriger n'importe quelle information entrante vers n'importe quelle sortie. Des outils ont également été construits par de nombreux développeurs de Max, comme par exemple les bibliothèques Jamona [7]. Le problème est souvent d'avoir à faire à des outils qui sont soit trop spécifiques soit trop généraux. Les outils trop spécifiques sont rapidement limités, les outils trop généraux sont souvent complexes et longs à programmer.

L'objectif ici, envisagé dans un but à la fois pédagogique, mais surtout à destination de situations de concerts, était de proposer un outil intermédiaire. Après avoir réalisé divers outils sur ce sujet dans des productions variées, j'ai finalement choisi de partir d'un outil qui avait fait largement ses preuves et dont jusqu'à présent je n'avais pas rencontré d'équivalent en matière d'ergonomie et d'efficacité, le clavier maître de type MK mis au point dans les années quatre-vingt par la firme ELKA.

J'ai donc voulu reproduire, en suivant la notice de l'instrument, ses différentes caractéristiques, afin de produire un programme permettant de brancher en entrée deux claviers (ou autre contrôleurs) MIDI pour adresser en sortie jusqu'à quatre modules MIDI indépendants. Il est assez facile d'imaginer augmenter le nombres d'entrées et de sorties MIDI de ce dispositif.

## 5. PROJET ELK-A1H<sup>2</sup>

### 5.1. Architecture

Comme le montre le schéma suivant, le clavier Maître se programme sur six zones de split, quatre à partir du clavier du MK et deux provenant de matériel externe grâce à une prise MIDI-IN. Le MK peut adresser chacune de ces zones sur deux sorties MIDI-OUT, sur des canaux MIDI différents.

Les contrôleurs physiques du MK peuvent être adressés chacun à une ou plusieurs zones de split pour contrôler divers paramètres MIDI dont certains sont donnés par défaut (modulation, volume, pitchbend, sustain...) et d'autres peuvent être programmés dans huit registres libres (FREE) et donc piloter tout type de message MIDI, y compris en Système Exclusif [8].

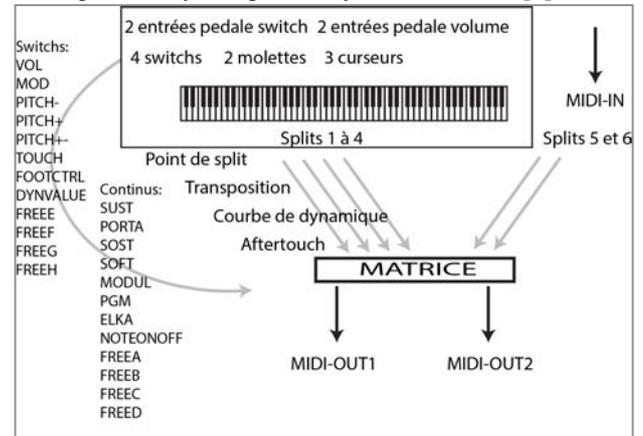


Figure 5. Architecture de transfert des messages MIDI dans un MK.

Cette architecture se retrouve dans le patch Max à l'intérieur du sous-patch MIDI.

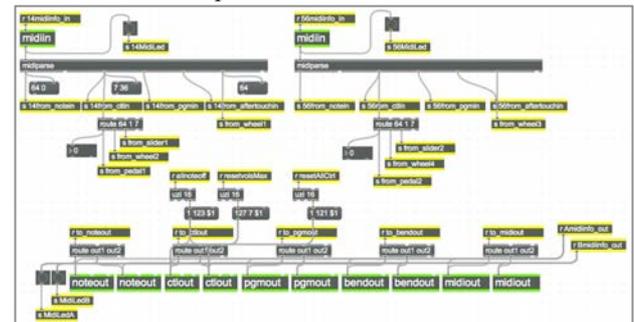


Figure 6. Entrée et sortie des messages MIDI.

### 5.2. Interface

Un des points les plus forts du MK concerne le panneau de contrôle. Si la fenêtre d'affichage des messages est très réduite, un grand tableau de 12 lignes par 20 colonnes, équipé de nombreuses LED permet d'accéder très rapidement à tous les paramètres que l'on peut vouloir programmer pour réaliser un preset (cf. Figure 2). D'un preset à l'autre, toute la configuration des zones de split est totalement remodelée, y compris les définitions des registres libres et l'appel à des changement de programmes vers les sorties MIDI qui peuvent être déclenchés automatiquement lors du changement de preset.

<sup>2</sup> Téléchargeable sur : <http://musinf.univ-st-etienne.fr/recherches/elka1h.html>

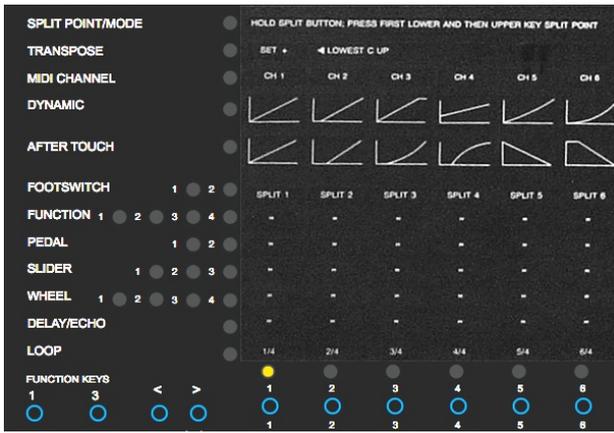


Figure 7. Un extrait de l'interface.

Les dynamiques (réponses des notes à la vélocité en entrée) sont stockées dans 18 tables de transfert.

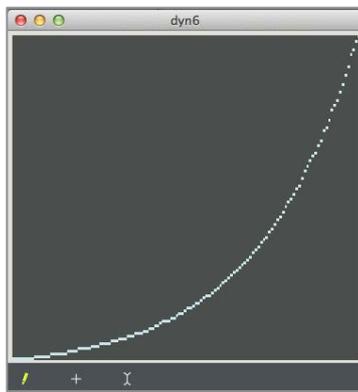


Figure 8. Une des tables de transfert de vélocité.

### 5.3. JavaScript

Pour les affectations des contrôleurs continus, une première tentative avait été réalisée en programmation graphique, (cf. Figure 9), mais la complexité des câblages s'est vite révélée rédhibitoire.

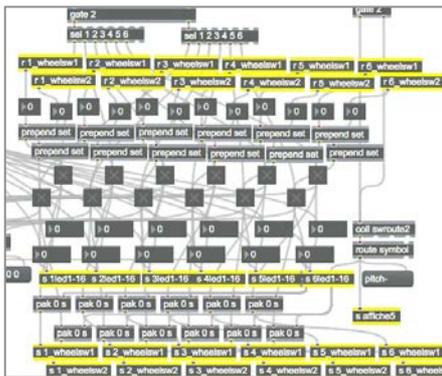


Figure 9. Câblage en Max des affectations de deux molettes (*wheel*) de contrôleurs.

Nous avons donc eu recours à la programmation en JavaScript. Dans une première version, les valeurs reçues des contrôleurs passaient par le script.

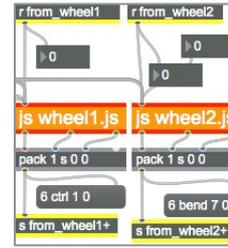


Figure 10. Câblage initial en JavaScript pour deux molettes de contrôleurs.

Mais nous nous sommes vite aperçu que dès que la charge CPU devenait un peu importante, les données n'étaient plus transmises assez rapidement pour un jeu fluide (parfois jusqu'à plus de 300 ms de retard entre l'envoi d'un contrôleur et sa réception en MIDI à la sortie de la fonction JavaScript). Les scripts ont donc seulement été utilisés pour compléter le message envoyé par un contrôleur afin de permettre de l'aiguiller par la suite vers la destination souhaitée.

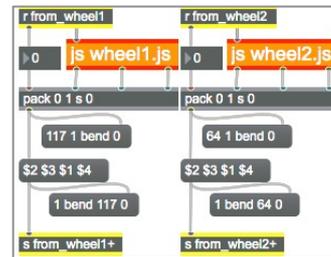


Figure 11. Câblage final en JavaScript pour deux molettes de contrôleurs.

### 5.4. Mémoires

Les mémoires ont été programmées en utilisant l'objet *patchstorage* dans Max. Cela permet de visualiser les données dans un tableau (*storagewindow*) et même de les y éditer directement si besoin et de sauvegarder des mémoires sur disque (64 mémoires possibles dans cette configuration par fichier). Par ailleurs les différents paramètres stockés dans ces mémoires peuvent être accessibles par les programmes JavaScript dès qu'on appelle une mémoire donnée.

Storage Slots [ELKAMK88]					
Name	1	2	3	4	5
namepreset[1]	cinema ...				
zone_split					
1_splitLow	0	0	0	48	48
1_splitHigh	96	47	96	79	79
1_transp	0	12	0	0	0
1_dynamicT	dyn18	dyn1	dyn18	dyn17	dyn17
1_MidiCHout	1	1	1	1	1
1_splitsw	1			1	1
1_Midiout	out1	out1	out1	out1	out1
1_fullsw	1		1	1	1
zone_split[1]					
2_splitLow	0	48	0	48	48
2_splitHigh	96	96	96	79	79
2_transp	0	-12	0	12	12
2_dynamicT	dyn1	dyn18	dyn1	dyn3	dyn3
2_MidiCHout	2	1	2	2	2
2_splitsw	1	0		1	1
2_Midiout	out1	out1	out1	out1	out1
2_fullsw	1	1		1	1

Figure 12. Les paramètres enregistrés dans une mémoire (section « notes »).



jswheel		
eventwheelp1	bend	bend
splitwheelp2	1 0 0 0 0	1000...
eventwheelp2	ctrl	ctrl
typwheelp1	0	2
typwheelp2	1	1
splitwheelp1	1 0 0 0 0	1000...
eventwheelp4	ctrl	ctrl
splitwheelp3	0 0 0 0 1 0	0000...
eventwheelp3	bend	bend
typwheelp4	1	1
typwheelp3	0	0
splitwheelp4	0 0 0 0 1 0	0000...
jswheel[1]		
eventpedalp1	ctrl	ctrl
splitpedalp2	0 0 0 0 0	00000
eventpedalp2		
typpedalp1	64	64
typpedalp2	1	2
splitpedalp1	1 0 0 0 0	1000...
jswheel[2]		
eventsliderp1	ctrl	ctrl
splitsliderp2	0 0 0 0 1 0	0000...
eventsliderp2	ctrl	ctrl
typsliderp1	7	7
typsliderp2	7	7
splitsliderp1	1 0 0 0 0	1000...
midipatch		
patcher		
mp1	1 PROGRAM 1	1 PRO...
mp2	5 PROGRAM 81	5 PRO...

**Figure 9.** Les paramètres enregistrés dans une mémoire (section « contrôleurs »).

## 6. CONCLUSION

Quelques modifications ont été réalisées par rapport à l'original et tout n'a d'ailleurs pas encore complètement été implémenté. Deux molettes supplémentaires ont été ajoutées, ainsi qu'un mode dans lequel tous les événements MIDI d'un split sont retransmis tels quels (tous contrôleurs) permettant d'utiliser comme en entrée MIDI par exemple un mixeur MIDI (ici UC33) pour piloter tous les paramètres d'un synthétiseur logiciel.

Les développements en attente concernent l'aftertouch qui n'est pas encore pris en compte spécifiquement (fonction assez avancée dans le MK) et les modes Delay/Echo et Loop.

Sur ces deux derniers points, le MK permettait de réaliser certains effets MIDI, Delay et Loop, mais dans une version que l'informatique permettrait de rendre beaucoup plus élaborée. La question est donc pour l'instant ouverte de savoir si nous cherchons à reproduire les outils originaux ou à y insérer d'autres outils plus performants parmi certains que nous avons déjà développés sur le sujet.

Dans le cas du projet ANR FEEVER, l'objectif dans un second temps va consister à réaliser une version de ce dispositif uniquement en JavaScript afin de pouvoir l'utiliser directement dans un navigateur Web pour contrôler des programmes de synthèse temps réel également disponibles sur le Web réalisés avec le langage Faust.

## 7. REFERENCES

- [1] Projet ANR ASTREE – 2009-2011  
<http://musinf.univ-st-etienne.fr/recherche2.html>
- [2] Projet ANR FEEVER – 2013-2017  
<http://musinf.univ-st-etienne.fr/recherche2.html>

[3] Pottier, Laurent, "Musiques électroniques : vers de nouveaux modes de jeu", *E-Formes 2, Arts & écritures numériques, au risque du jeu*, dir. M. Maza et A. Saemmer, Saint-Etienne, PUSE, 2011, p. 155-165

[4] Dechelle, François, *A Brief History of MAX*, (approx. 1999)  
<http://jmax.sourceforge.net/history.html>

[5] *Professional ELKA MK88 MK55 Master Control Keyboard, Operating Manual - Manuale d'istruzione*, Castelfidardo (Italie), (approx. 1988), 48 p.  
<http://medias.audiofanzine.com/files/elka-mk55-mk88-manual-472559.pdf>

[6] Pottier, Laurent, « Exemples d'utilisation de la CAO pour la synthèse sonore, "En Trance" de Fausto Romitelli pour soprano, ensemble et dispositif électronique », *Actes des Journées d'Informatique Musicale (JIM 97)*, Lyon : Grame, 1997, p. 23.

[7] <http://www.jamoma.org/>

[8] MIDI 1.0 Detailed Specification V4.2, Los Angeles, The MIDI Manufacturers Association, Sept. 1995.

Schnell, Norbert & Battier, Marc, « Introducing Composed Instruments, Technical and Musicological Implications », *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Irlande, 2002.