

Franck Ernould • Christian Braut

Le guide du MIDI

**De l'équipement à l'exploitation :
fondamentaux et bonnes pratiques**

DUNOD

Direction et conception graphiques de la couverture :
Nicolas Wiel – Elizabeth Riba (graphiste)
Dessin des figures 1.1 à 1.4, 1.7, 1.9, 1.11, 4.3, 4.4, 4.6 à 4.12, 5.1 à 5.12 :
Bernadette Coléno

NOUS NOUS ENGAGEONS EN FAVEUR DE L'ENVIRONNEMENT :



Nos livres sont imprimés sur des papiers certifiés pour réduire notre impact sur l'environnement.



Le format de nos ouvrages est pensé afin d'optimiser l'utilisation du papier.



Depuis plus de 30 ans, nous imprimons 70 % de nos livres en France et 25 % en Europe et nous mettons tout en œuvre pour augmenter cet engagement auprès des imprimeurs français.



Nous limitons l'utilisation du plastique sur nos ouvrages (film sur les couvertures et les livres).

© Dunod, 2024
11 rue Paul Bert 92240 Malakoff
www.dunod.com
ISBN 978-2-10-079743-1

Table des matières

————— CHAPITRE 1 – HISTORIQUE ET SPÉCIFICATIONS MATÉRIELLES DU MIDI —————		1
1.1	L'exécution instrumentale	1
1.1.1	L'instrument acoustique	1
1.1.2	La nouvelle facture instrumentale : l'électronique	2
1.1.3	Vers l'interfaçage	3
1.2	Spécifications matérielles de la liaison MIDI	11
1.2.1	Généralités sur les interfaces informatiques	11
1.2.2	Le MIDI : une liaison série et asynchrone	11
1.2.3	Vitesse et densité de données	13
1.2.4	Les connecteurs MIDI	14
1.2.5	L'isolation opto-électrique	16
1.3	L'interface série/parallèle	16
1.3.1	L'interfaçage	16
1.3.2	Buffer et interruptions	16
————— CHAPITRE 2 – LE TRANSPORT DES DONNÉES MIDI —————		19
2.1	Configurations MIDI physiques	19
2.1.1	Généralités sur l'interfaçage	20
2.1.2	MIDI In et MIDI Out : les liaisons élémentaires	20
2.1.3	Le connecteur MIDI Thru	22
2.1.4	L'USB-MIDI	23
2.1.5	RTP-MIDI : Le MIDI sur les réseaux informatiques	24
2.2	MIDI et systèmes d'exploitation	26
2.2.1	Les débuts : l'Atari ST520/1040	26
2.2.2	Le MIDI sous macOS	26
2.2.3	Le MIDI sous Windows	27
2.2.4	Le MIDI sous Linux	28
2.2.5	Le mode MIDI Class Compliant	28

2.3	Le MIDI sur mobile Android ou iOS	30
2.3.1	iOS, iPadOS	30
2.3.2	Android	30
2.3.3	MIDI et sonneries	31
2.4	Les interfaces MIDI	32
2.4.1	Interfaces dédiées	32
2.4.2	Interfaces mixtes, MIDI et audio	34
2.4.3	Interfaces sans fil	34
2.4.4	Connectique MIDI	35
2.5	Les boîtiers de raccordement MIDI	37
2.5.1	Boîtier MIDI Thru	37
2.5.2	Boîtier MIDI Merger	39
2.5.3	Patch MIDI	40
2.5.4	Convertisseur MIDI vers CV/Gate	41
2.5.5	Extendeur MIDI	41
2.5.6	Moniteur/afficheur de niveau MIDI	43

CHAPITRE 3 – LES CONTRÔLEURS MIDI

3.1	Les contrôleurs-instruments de musique	45
3.1.1	Claviers maîtres	45
3.1.2	Contrôleurs au souffle	46
3.1.3	Contrôleurs de percussions	47
3.1.4	Contrôleurs pour guitares	49
3.1.5	La MIDIfication d'instruments acoustiques	49
3.2	Les surfaces de contrôle « mixage »	52
3.3	Les contrôleurs spécialisés	53
3.3.1	Contrôleur pour plug-ins	53
3.3.2	Contrôleur pour logiciels musicaux	54
3.3.3	Contrôleurs pour DJ : Faderfox, Nakedboards, Traktor, Scratch Live	55
3.4	Autres contrôleurs	56
3.4.1	Contrôleur à tirettes	56
3.4.2	Contrôleur pédalier guitare	56
3.4.3	Contrôleur multidimensionnel	57
3.4.4	Contrôleur vocal	57
3.4.5	Logiciel de contrôle MIDI pour tablette	58
3.5	Autres utilisations	59

————— CHAPITRE 4 – LES MESSAGES MIDI DE BASE —————		60
4.1	Un document évolutif	61
4.2	Le format des octets MIDI : statuts et données	62
4.3	Deux catégories de messages : canal et système	63
4.3.1	La notion de destination : le canal MIDI	63
4.3.2	Distinction messages canaux/messages système	66
4.3.3	Les sept messages canaux	66
4.3.4	Deux catégories de messages canaux : Voice et Mode	67
4.3.5	Trois catégories de messages système	67
4.3.6	La réception des statuts	68
4.3.7	Les statuts des messages MIDI	68
4.3.8	Les familles des statuts	72
4.4	Les messages de voies	73
4.4.1	Note-On/Note-Off	73
4.4.2	Program Change	80
4.4.3	Aftertouch	84
4.4.4	Pitch Bend	86
4.4.5	Control Change	93
————— CHAPITRE 5 – LES MESSAGES DE MODE ET SYSTÈME —————		107
5.1	Les messages All Sound Off, Reset All Controllers, Local On/Off, All Notes Off	107
5.1.1	All Sound Off	107
5.1.2	Reset All Controllers	108
5.1.3	Local Control On/Off	108
5.1.4	All Notes Off	110
5.2	Les messages Omni On, Omni Off, Poly et Mono	111
5.2.1	La multitimbralité	112
5.2.2	Omni Mode Off (+ All Notes Off)	112
5.2.3	Omni Mode On (+ All Notes Off)	112
5.2.4	Mono Mode On (+ All Notes Off)	112
5.2.5	Poly Mode On (+ All Notes Off)	113
5.2.6	Les modes MIDI	113
5.2.7	Les messages de mode	116
5.3	Le Running Status	116
5.4	Les messages système temps réel	118
5.4.1	Les messages de synchronisation	119
5.4.2	Les autres messages en temps réel	122

5.5	Les messages système communs	123
5.5.1	Song Position Pointer (SPP)	124
5.5.2	Song Select	125
5.5.3	Tune Request	125
5.6	Les messages exclusifs	125
5.6.1	Les messages exclusifs « fabricants »	126
5.6.2	Les messages exclusifs universels	127
5.7	Un exemple d'organigramme de réception des données MIDI	143
5.8	La charte d'implantation MIDI (MIDI Implementation Chart)	145
5.8.1	Description	147
5.8.2	Conclusion	153

CHAPITRE 6 – MIDI TIME CODE

6.1	Synchronisation entre séquenceurs	155
6.1.1	Synchronisation MIDI/MIDI	155
6.1.2	Synchronisation MIDI/pré-MIDI	156
6.2	Synchronisation MIDI/audio-vidéo	157
6.2.1	Le Timecode SMPTE	157
6.2.2	MIDI et SMPTE	157
6.3	Le MTC (MIDI Time Code)	158
6.3.1	La traduction SMPTE/MTC : les messages temporels	158

CHAPITRE 7 – MMC, MMS, SDS

7.1	MMC (MIDI Machine Control)	171
7.2	Syntaxe des messages MMC	172
7.2.1	L'environnement	173
7.2.2	Les échanges de données	173
7.2.3	Identificateurs et groupes	173
7.2.4	Commandes et réponses	174
7.2.5	Extensions	175
7.3	MSC (MIDI Show Control)	175
7.3.1	Présentation	175
7.3.2	La syntaxe des messages	176
7.4	Les Dumps MIDI	183
7.4.1	Le Dump : un message exclusif	184
7.4.2	Le Dump Request	184

7.4.3	La syntaxe du Dump	185
7.4.4	Formats de stockage et de transmission des données	185
7.5	SDS (Sample Dump Standard)	186
7.5.1	Data Packet (paquet de données)	186
7.5.2	Dump Request (demande de Dump)	187
7.5.3	ACK (Acknowledge)	187
7.5.4	NAK (Not Acknowledge)	187
7.5.5	Cancel (annulation)	188
7.5.6	Wait (attente)	188
7.5.7	Dump Header (en-tête de Dump)	189
7.5.8	Multiple Loop Points (transmission de points de bouclage)	189
7.5.9	Loop Points Request (demande de transmission de points de bouclage)	190
7.5.10	Mise en œuvre du protocole de transfert	190
<hr style="width: 100%; border: 0.5px solid black; margin: 10px 0;"/> CHAPITRE 8 – LES SMF, LE STANDARD GM, LES FORMATS GS ET XG <hr style="width: 100%; border: 0.5px solid black; margin: 10px 0;"/>		193
8.1	Le Standard MIDI File (SMF)	193
8.2	Le standard General MIDI (GM)	196
8.2.1	Le General MIDI	197
8.2.2	Le General MIDI 2 (GM2)	206
8.3	Le format GS	207
8.3.1	L'assignation des sons	208
8.3.2	L'assignation des kits de batterie	217
8.4	Le format XG	221
8.4.1	Généralités	221
8.4.2	Les catégories	222
8.4.3	La liste des sons	224
8.5	Le format DLS (Downloadable Samples)	239
8.6	Le format XMF (Extensible Music Format)	239
<hr style="width: 100%; border: 0.5px solid black; margin: 10px 0;"/> CHAPITRE 9 – LA MPE ET LE MIDI 2.0 <hr style="width: 100%; border: 0.5px solid black; margin: 10px 0;"/>		241
9.1	La MPE (MIDI Polyphonic Expression)	242
9.1.1	Présentation du concept	242
9.1.2	Implémentation	243
9.1.3	Enregistrement/édition des données MPE	246
9.1.4	Quelques instruments et contrôleurs compatibles MPE	246

9.2	Le MIDI 2.0	249
9.2.1	Les trois B	250
9.2.2	Les trois P : MIDI Capability Inquiry (MIDI-CI)	250
9.2.3	Quelques autres aspects du MIDI 2.0	252
9.3	Les messages MIDI 2.0	254
9.3.1	UMP (Universal MIDI Packet)	254
9.3.2	MIDI 1.0 dans MIDI 2.0	256
9.4	Les pionniers du MIDI 2.0	256
<hr/>		
	INDEX	259

Chapitre 1

Historique et spécifications matérielles du MIDI

Cet ouvrage sort 41 ans après la publication, en 1983, de la spécification MIDI 1.0. Il est rarissime, dans un domaine lié à l'électronique et à l'informatique, de voir un protocole durer aussi longtemps. Ce n'est pas dû au hasard : les fabricants s'étaient consultés en amont, les modes de décision étaient ouverts mais bien contrôlés, et la norme elle-même a été conçue d'emblée pour une adaptation optimale aux besoins non encore exprimés. Sa version 2.0 est rétrocompatible, et relève des mêmes concepts, actualisés.

À l'origine, l'interface MIDI a été conçue dans le but de normaliser la communication entre instruments utilisant des technologies numériques (synthétiseurs, échantillonneurs, etc.). Or, pour cerner avec précision les limites à l'intérieur desquelles le MIDI est en mesure de gérer cette nouvelle facture instrumentale, il est indispensable d'en décrire le fonctionnement. Nous commençons donc cet ouvrage par le rappel des corrélations entre le jeu du musicien et la production de son, sur des instruments traditionnels puis électroniques, d'où découle le « cahier des charges » des premières interfaces musicales.

1.1 L'exécution instrumentale

1.1.1 *L'instrument acoustique*

Sur un instrument acoustique traditionnel, la production d'un son est liée à deux concepts fondamentaux. Le premier est relatif au dispositif physique qui engendre le

phénomène sonore, et le second, à la communication à ce dispositif, par l'interprète, de l'énergie nécessaire pour qu'il entre en vibration (en oscillation) afin de générer un son. Ces deux concepts correspondent respectivement aux notions d'instrument et d'instrumentiste. Le premier est régi par les lois de l'acoustique, tandis que le second est généralement codifié, en Occident, *via* un système de notation musicale écrite (partition).

Les instruments acoustiques, au sens large du terme (qu'il s'agisse de vents, de cordes, de percussions...), ont pour point commun une structure physique en contact direct avec l'instrumentiste, lui permettant de déclencher le mécanisme de production sonore. En informatique musicale, cette partie prend le nom de *contrôleur*. Il s'agit, par exemple, du clavier et des pédales d'un piano, du manche et des cordes d'une guitare, des baguettes et des peaux d'une batterie et, plus généralement, de tout ce qui autorisera le musicien à influencer sur cette production sonore. Les matériaux employés dans la facture d'un instrument acoustique sont directement responsables du son délivré, qu'ils fassent partie intégrante du contrôleur (guitare, percussion...), ou qu'ils y soient interfacés mécaniquement (marteaux, ressorts, étouffoirs d'un piano, soufflerie d'un orgue...). Il est impossible de dissocier la partie contrôleur de la partie génératrice de son.

1.1.2 *La nouvelle facture instrumentale : l'électronique*

À partir des années 1920, l'instrument de musique connaît une évolution décisive. Jusqu'alors, sa partie contrôleur influait physiquement sur la sonorité créée. Le développement successif des techniques électriques, puis électroniques et enfin numériques a autorisé la simulation ou la reproduction de situations acoustiques déterminées, indépendamment de tout contrôleur physique. Ces nouveaux instruments, destinés à créer des formes d'ondes de toutes pièces, se nomment générateurs de son. Ils englobent par exemple le Thérémin, les ondes Martenot, les synthétiseurs, les échantillonneurs, etc.

En sortie de ces générateurs, un signal représentant l'onde sonore sous forme électrique est acheminé vers une chaîne d'amplification, qui aboutit à des haut-parleurs. À l'intérieur de ce générateur, divers procédés sont mis en œuvre afin de déterminer la hauteur, l'amplitude et le timbre du signal sonore à créer. Ces procédés ne nécessitent aucune énergie physique pour entrer en action, par opposition à celle communiquée aux instruments acoustiques par les contrôleurs. Si les synthétiseurs possèdent un clavier, ce dernier, plutôt que de commander par exemple une mécanique de marteaux dans le but de faire vibrer des cordes, se contente d'intimer l'ordre au générateur de son de jouer telle ou telle note, par l'intermédiaire d'un signal électrique à teneur analogique ou numérique (les fameux messages MIDI). Nous verrons par la suite qu'un contrôleur de ce type ne revêt pas nécessairement la forme d'un clavier, bien que ce soit la plus répandue, pour d'évidentes raisons de commodité.

En informatique musicale, nous retiendrons que l'exécution instrumentale passe par trois stades obligatoires, contre deux seulement avec un instrument acoustique traditionnel. Ces trois stades sont les suivants :

1. L'instrumentiste communique physiquement son énergie au contrôleur (appui sur des touches, etc.).
2. Le contrôleur transforme cette énergie en un signal de commande destiné au générateur de son. Cette transformation n'est autre que la codification de l'action mécanique infligée au contrôleur par l'instrumentiste.
3. Le générateur de son, grâce à des composants électroniques et/ou numériques, reproduit une sonorité à réception du signal de commande (par simulation de situations acoustiques déterminées).

Sans anticiper sur le sujet, on imagine aisément l'étendue des possibilités offertes par un instrument de musique électrique, du fait de l'indépendance des étapes 1, 2 et 3. En voici quelques-unes :

- la substitution aux étapes 1 et 2 d'un signal de commande équivalent, en vue d'automatiser la procédure de jeu du générateur de son ;
- le pilotage de plusieurs générateurs de son par le même contrôleur ;
- le pilotage d'un générateur de son par n'importe quel contrôleur, etc.

1.1.3 Vers l'interfaçage

Cette dissociation du contrôleur et du générateur de son implique la mise en œuvre d'un protocole (interface) autorisant ces deux modules à communiquer. Les notions contenues dans ce paragraphe constituent une rapide introduction aux principes d'interfaçage des instruments de musique électroniques. Elles retracent l'histoire des systèmes qui ont marqué la période pré-MIDI. Schématiquement, on distingue trois grandes catégories d'informations nécessaires et suffisantes à toute interface musicale, et en particulier à l'interface MIDI. Ce sont :

- les données destinées à codifier le jeu du musicien ;
- les données de synchronisation ;
- les données représentant les paramètres d'un son.

◆ La codification du jeu du musicien

Les premiers instruments électroniques et synthétiseurs, pratiquement fabriqués à l'unité et à la main au fil des années 1950 et 1960, n'avaient pas été conçus pour s'interconnecter. Il s'agissait d'instruments « solistes », dont la conception modulaire n'excluait pas une certaine complexité, mais qui ne s'ouvraient guère sur l'extérieur. Les paramètres relatifs au son (variations de hauteur, de timbre, d'amplitude) s'effectuent en temps réel, par le biais de sélecteurs, de touches et de potentiomètres ou de curseurs agissant tous directement sur les composants électroniques du synthétiseur. Il est évidemment impossible de mémoriser sur l'appareil l'ensemble de ces réglages, et l'on ne dispose que d'une sonorité à la fois. Le seul moyen de retrouver, approximativement, un son élaboré consistait à noter sur une feuille la position de tous les interrupteurs, potentiomètres, ainsi que les câblages éventuels.

La scission d'un instrument de musique de type synthétiseur en deux modules (le contrôleur d'un côté, le générateur de son de l'autre) implique qu'il existe un système d'interfaçage permettant à ces deux modules de communiquer. Cette interface transforme en signaux codés les gestes que le musicien applique au contrôleur (l'exécution instrumentale) en vue de piloter le générateur de son (la production de notes).

Sur les premiers synthétiseurs analogiques « commerciaux » du début des années 1970 (ARP 2500/2600 et Odyssey, Minimoog, EMS VCS3/Synthi A...), des possibilités d'interconnexion sont prises en compte. L'approche adoptée par les fabricants est de traduire sous forme électrique, par des signaux analogiques, des informations de hauteur (CV, abréviation de *Control Voltage*, pour la tension de commande des oscillateurs, plus ou moins élevée selon la hauteur de la note), de déclenchement du générateur d'enveloppe d'amplitude (gate, une impulsion électrique), de fréquence de LFO, d'horloge interne (*clock*)...

Hélas, chaque constructeur adopte sa propre façon de coder la hauteur par une tension (*Control Voltage*, ou CV) : selon une échelle logarithmique, 1 volt par octave ou, selon une échelle linéaire, 1 volt tous les 1 000 Hz par exemple. Dans le premier système, la tension double à chaque octave. Sachant qu'une octave est divisée en 12 demi-tons, il suffit de multiplier une tension par la racine douzième de deux (soit 1,059 463 094) pour obtenir la tension du demi-ton suivant. Avec le standard volt/octave, chaque progression d'octave correspond à une augmentation de tension de 1 V, soit 83,3 mV par demi-ton.

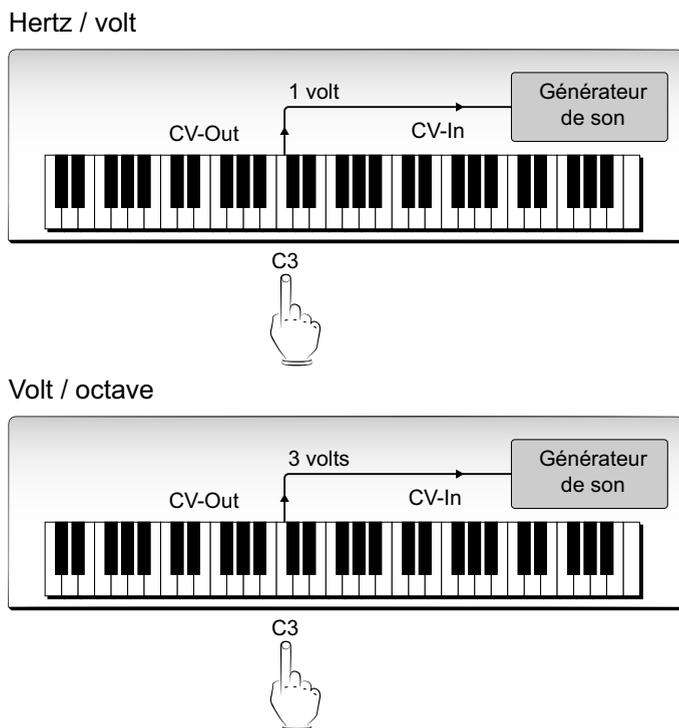


Figure 1.1 – Dès l'enfoncement de la touche Do 3, le clavier envoie au générateur de son soit une tension de 1 V (échelle hertz/volt) soit une tension de 3 V (échelle volt/octave, en bas).

Ce signal de CV indique la hauteur de la note jouée sur le clavier, mais ne dit rien sur sa durée. Cette notion de durée est prise en charge par un signal appelé *Gate* (porte), déclenché par l'appui sur une touche du clavier, et désenclenché lors de son relâchement. Ainsi, il indique au générateur d'enveloppe du générateur de son (section contrôlant l'évolution du volume dans le temps) la période pendant laquelle la note (la forme d'onde produite par l'oscillateur) devra être jouée. Physiquement, le signal Gate est un signal électrique de type tout ou rien : une tension positive (généralement + 5 volts) correspond à une touche enfoncée et une tension nulle, à l'état de repos. L'association des signaux CV et Gate constitue l'interface minimale entre un clavier et un générateur de son.

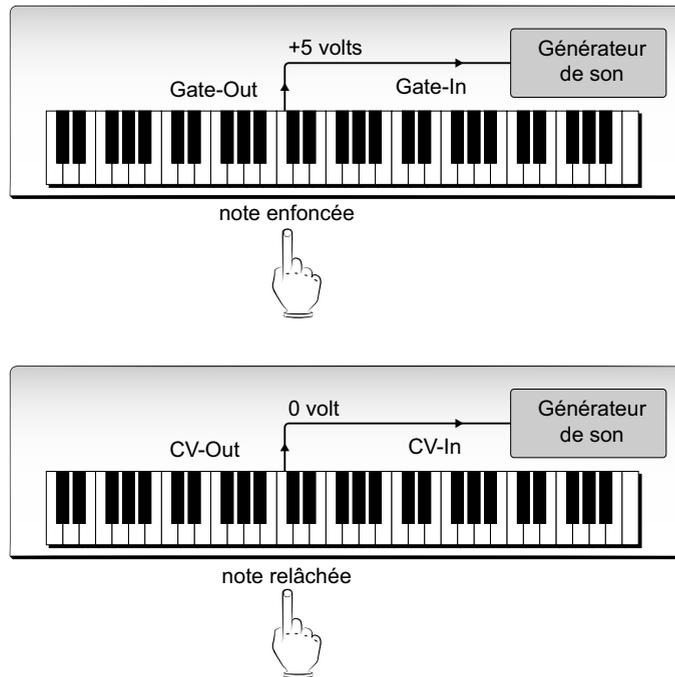


Figure 1.2 – Lors de l'appui sur une touche, la tension Gate passe de 0 à + 5 volts. Elle redescend à 0 volt dès son relâchement.

De cette interface est née l'idée de piloter plusieurs générateurs de son depuis un seul et même clavier. Sur chaque synthétiseur, il suffisait d'implanter des prises CV-Out et Gate-Out (signaux en provenance du clavier), et des prises CV-In et Gate-In (dirigées vers le générateur de son). Ainsi, par le simple raccordement des prises CV-Out et Gate-Out d'un premier synthétiseur vers les prises CV-In et Gate-In d'un second synthétiseur, le clavier du premier synthétiseur pilotait les deux générateurs de son (à condition que les standards CV soient identiques, ou d'utiliser une interface pour traduire le hertz/volt en volt/octave, ou réciproquement). Ce principe de commande en tension d'un générateur de son s'est étendu par l'adjonction d'autres interfaces, comme le *Pitch to Voltage Converter*, grâce auxquelles la hauteur du signal recueilli en sortie d'un microphone est transformée en temps réel en un signal CV.

L'usage montra rapidement qu'utiliser une tension électrique pour un contrôle analogique de hauteur de note présentait un inconvénient : l'instabilité. Avec la dérive thermique des composants électroniques, il s'avérait quasiment impossible de maintenir longtemps un accord juste et régulier de la part d'oscillateurs commandés en tension. Le recours au numérique résolut le problème : c'est le cas du MIDI.

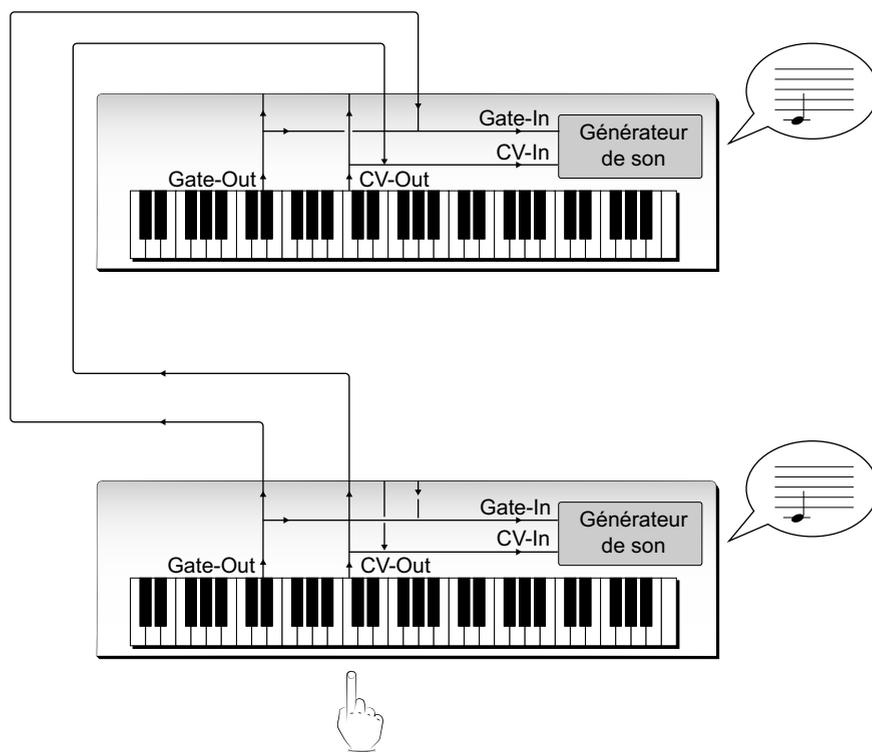


Figure 1.3 – Le clavier de l'instrument A envoie des informations CV/Gate vers le générateur de son de l'instrument B.

◆ Le séquenceur analogique, le signal de synchronisation

Les premiers séquenceurs analogiques monophoniques furent développés sous forme de modules pour synthétiseurs modulaires. L'idée est de remplacer un clavier pour piloter automatiquement un générateur de sons. Tout un courant musical naît, au milieu des années 1970, de ces séries de 8 ou 16 notes, lues en boucle de façon hypnotique et répétitive sur des synthétiseurs modulaires, avec des boucles d'écho magnétique : la musique planante, de Tangerine Dream à Jean Michel Jarre en passant par Klaus Schulze, Ash Ra Tempel ou Richard Pinhas.

Pour jouer une séquence, il faut tout d'abord programmer une suite de hauteurs de notes, donc de tensions CV, ainsi que les signaux de déclenchement de ces notes, les tensions gâtées. Ce type de programmation s'appelle pas par pas (*step by step*). Pour définir le tempo selon lequel lire la suite de notes programmées, le séquenceur intègre une horloge fractionnant la noire, référence du tempo, en un certain nombre de subdivisions/cases mémoire, chacune envoyant ou non les tensions CV et Gate. C'est sur ce principe que fonctionnaient les premières boîtes à rythmes, réunissant dans un même boîtier un séquenceur et un générateur de son spécialisé dans l'imitation d'instruments à percussion.

Les techniques évoluant, les séquenceurs en temps réel (*real time*) ont rapidement supplanté leurs homologues pas à pas. Plutôt que de programmer chaque tension l'une après l'autre dans la mémoire du séquenceur, il devenait beaucoup plus simple d'y raccorder les signaux en sortie du clavier, pour enregistrer directement et en temps réel le jeu du musicien, en mémorisant les tensions CV/Gate correspondant aux notes enfoncées et relâchées. En résumé, le séquenceur temps réel s'assimile à un magnétophone d'un type un peu particulier, se contentant d'enregistrer et de reproduire un certain nombre de signaux de commande destinés à automatiser le jeu d'un ou de plusieurs générateurs de son. Il stocke les messages CV/Gate en provenance du clavier pour les retransmettre ultérieurement aux générateurs de son. Ce type de séquenceur connaîtra, grâce au MIDI, une belle popularité au milieu des années 1980, avant d'être supplanté par les logiciels de séquence, aux possibilités plus nombreuses.

Bien avant le MIDI, il existait donc déjà des machines destinées à piloter des générateurs de son en temps réel, et à un certain tempo (séquenceurs, boîtes à rythmes, mais aussi arpégiateurs, etc.). Pour faire jouer plusieurs de ces machines ensemble, il était nécessaire de les synchroniser très précisément. D'où l'apparition d'une seconde famille de signaux de contrôle : les signaux de synchronisation.

À l'instar des signaux gate, les signaux de synchronisation (encore appelés *triggers* ou signaux d'horloge) se traduisent sous forme d'une alternance de tensions nulles et positives. Cette alternance de tensions intervient au rythme d'un certain nombre d'impulsions par noire, ou PPQN (*Pulse Per Quarter Note*). En synchronisant l'entrée *trigger* d'une machine esclave sur la sortie *trigger* d'une machine maître, les deux unités conservent le même tempo tout au long du morceau. Toujours par manque de coordination, les procédés en vigueur étaient nombreux : 24, 48, 96, 384 PPQN, etc. Une fois de plus, la norme MIDI mit un terme à cette absence de standard.

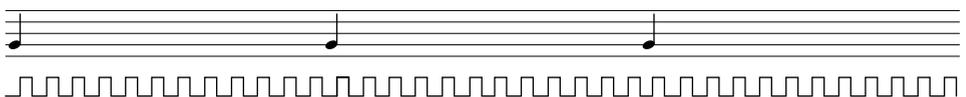


Figure 1.4 - Un signal carré de synchronisation, de résolution 24 PPQN.

◆ **L'analogique géré numériquement**

Vers la fin des années 1970, les ingénieurs introduisent la polyphonie et des mémoires dans les synthétiseurs analogiques, ce qui implique d'en gérer les paramètres de façon numérique – donc d'y inclure un microprocesseur. De là à développer des protocoles d'interface numérique, il n'y a qu'un pas. On transporte toujours les informations de contrôle sous la forme d'une tension électrique, mais celle-ci correspond désormais à des 0 et des 1. Parmi les premières marques à proposer de telles interfaces, citons Fender (avec la « Trident », dont était équipé le Chroma), Roland (avec son DCB, *Digital Control Bus*, présent par exemple sur le Jupiter-8), Oberheim (avec la CSI, *Computer Synthesizer Interface*, offrant la possibilité d'interconnecter un synthétiseur, une boîte à rythmes et un séquenceur de la marque), Sequential Circuits (dont la *SCI Digital Interface* servait à relier un synthétiseur, en l'occurrence le Prophet-10, à un séquenceur polyphonique de la marque)... Hélas, au lieu de se concerter, de collaborer pour établir un protocole universel, les constructeurs travaillent, une fois encore, chacun de leur côté...

◆ **Les prémices du MIDI**

Revenons sur la marque américaine Sequential Circuits (créée en 1974, aujourd'hui disparue), et plus particulièrement sur son fondateur, Dave Smith, décédé en 2022 après avoir recréé la marque Sequential en 2018. Avec des modèles comme le Prophet-5 (premier synthétiseur polyphonique programmable, 1978) ou le Six-Trak (premier synthétiseur multitimbral, 1984), Sequential Circuits a largement contribué au développement du concept d'instrument de musique électronique.

L'interface numérique développée par la marque fonctionnait selon le principe suivant : à chaque cycle d'horloge du séquenceur, les touches du clavier du Prophet-10 étaient scrutées, leur état analysé, puis mémorisé, pour reproduction ultérieure. L'inconvénient majeur de ce procédé résidait dans sa gourmandise en espace mémoire : même s'il ne se passe rien, le débit des informations reste identique.

Conscient de ce problème, Dave Smith étudie, en collaboration avec Chet Wood (Sequential Circuits), Tom Oberheim (Oberheim) et Ikutaro Kakehashi (Roland) une solution plus rationnelle, consistant à ne transmettre une information qu'à détection d'un événement (enfoncement ou relâchement d'une touche, etc.). Ce projet est présenté lors de la Convention de l'AES (*Audio Engineering Society*) de New York, en 1981. Baptisée USI (*Universal Synthesizer Interface*), cette interface série utilise des connecteurs de type jack 6,35 mm, véhicule des signaux électriques logiques de type TTL (passage de 0 à 5 volts) et travaille à une vitesse de 19 200 bauds.

À l'issue de cette présentation, un questionnaire est envoyé à divers constructeurs pour les « sonder » en vue de développer un véritable standard. Trois mois plus tard, lors du NAMM Show d'Anaheim (exposition essentiellement axée instruments de musique) se déroule une réunion entre Sequential Circuits, Roland, Yamaha, Korg,

Kawai, E-mu, Oberheim, Rhodes, Octave Plateau, Passport Designs et Syntauri. Il y est question d'augmenter la vitesse à 31 250 bauds, d'utiliser des opto-isolateurs, et surtout de ne pas se limiter à la transmission d'événements relatifs aux notes. D'autre part, il est admis que pour s'imposer, ce standard ne doit pas revenir trop cher à implémenter sur les instruments.

◆ La naissance du MIDI, les premiers problèmes

Ce mois de janvier 1982 marque donc la naissance « officielle » du MIDI (*Musical Instrument Digital Interface*). C'est courant 1982 que le premier document MIDI est élaboré, entre les États-Unis (Sequential Circuits) et le Japon (Roland). Il revient à Bob Moog lui-même d'évoquer le premier, dans un article du numéro d'octobre 1982 du magazine américain *Keyboard*, la naissance du MIDI. Deux mois plus tard, Sequential commercialise le premier synthétiseur intégrant une interface MIDI : le Prophet 600. Yamaha suit de près, en février 1983, avec le DX7, puis Roland présente le Jupiter-6, le le JX-3P, la boîte à rythmes TR-909, le séquenceur MSQ-700 et les pianos numériques HP-300 et HP-400. Hormis ces trois marques, plus Korg et Kawai, qui ont adhéré immédiatement au principe, les autres fabricants se cantonnent dans une position attentiste.

La première démonstration publique de connexion *via* MIDI d'instruments de modèles et de marques différentes se déroule au Winter NAMM de 1983, avec un Prophet 600 et un Jupiter-6. Encore expérimentales, ces connexions n'allaient pas sans poser de problèmes, en raison d'interprétations parfois très différentes de la norme. Ainsi, lorsque deux constructeurs décidaient, chacun dans leur coin, d'attribuer les fonctions de leur choix à des messages de contrôleurs continus (CC) non définis par la norme, le musicien obtenait des résultats inattendus. Le DX7 Yamaha se servait par exemple du CC n° 6 en tant que *Data Entry Slider*, tandis que le CZ-101 Casio l'utilisait pour l'accord de l'instrument... Pour traduire des informations de *Pitch Bend*, le Prophet 600 exploitait d'une part le CC n° 0, mais de l'autre un message de statut non encore défini. Quant au DX7, il transmettait à intervalles réguliers deux octets spécifiques, en lieu et place de l'*Active Sensing* (qui allait voir le jour par la suite). Il ne permet pas de recevoir les données MIDI en mode omni, ne possède pas de mode Local Off, n'émet que sur le canal MIDI numéro 1, attribue le CC n° 3 à l'aftertouch, etc. En revanche, l'assignation choisie par Yamaha pour certains CC (*Breath Controller*, *Data Increment/Decrement* par exemple) sera reprise ultérieurement dans la norme MIDI publiée un peu plus tard. Sa première version compte seulement 8 pages !

◆ La coordination

Afin d'institutionnaliser le MIDI, une association coordinatrice est créée pendant l'été 1983 : l'IMA (*International MIDI Association*). C'est grâce à elle qu'est publié le premier document MIDI officiel, le 5 août 1983, sous le titre de *MIDI Detailed Specification version 1.0*.

Le rôle de l'IMA consiste à diffuser auprès des utilisateurs musiciens les informations provenant des fabricants. En sens inverse, elle répercute auprès de ces derniers les désirs des utilisateurs, pour que la norme évolue selon leurs besoins. Tout un chacun peut devenir membre de l'IMA, qui édite un journal mensuel disponible sur abonnement, *The IMA Bulletin*. L'association se charge par ailleurs de diffuser en exclusivité une série d'ouvrages sur le MIDI, dont les documents officiels.

Parmi les fabricants, il existe également deux associations créées pour éviter toute dérive anarchique de la norme MIDI : la MMA (*MIDI Manufacturers Association*), qui regroupe des sociétés américaines, canadiennes et européennes, différents magazines spécialisés et quelques consultants, et le JMSC (*Japanese MIDI Standard Committee*), qui regroupe des marques japonaises. Nous y reviendrons au début du chapitre 4.

C'est grâce à cet effort d'universalisation que la norme MIDI s'est imposée. Appliquée sous la même forme par l'ensemble des fabricants, elle assure, depuis plus de 30 ans, une compatibilité totale entre appareils MIDI. Aujourd'hui encore, le MIDI demeure le seul « standard unique » en vigueur dans l'informatique musicale. En quelques années, le MIDI a bouleversé les habitudes des musiciens, apportant une souplesse de travail jusqu'alors inimaginable. La norme a considérablement évolué depuis, sans perdre son objectif de départ. Rien n'a pu la remplacer. Kakehashi et Smith se sont vu décerner, en 2013, un Technical Grammy pour leur création : une référence dans le domaine de la technologie musicale.

En parallèle de la norme MIDI, l'ordinateur personnel (compatible PC IBM) faisait son apparition. Ces deux inventions sont complémentaires, l'ordinateur apportant une interface graphique, des possibilités d'enregistrement, de programmation... C'est l'Atari 520, en 1984, doté à l'origine d'une interface MIDI, qui lance le développement de l'informatique musicale – même si Roland avait dévoilé très tôt sa propre interface MIDI : la MPU-401, rattachée à une carte ISA dans le PC lui-même. Cette référence survit encore aujourd'hui dans les menus de certains logiciels de séquence... En effet, Roland eut l'intelligence d'en publier les spécifications, ce qui favorisa l'apparition de logiciels écrits pour l'interface. En 1985, on pouvait relier son Commodore 64, son Apple II ou son PC sans problème. L'arrivée du Macintosh Plus a convaincu bien des musiciens de s'équiper. On connaît la suite...

Même si les configurations MIDI pouvaient parfois se montrer capricieuses, voire fastidieuses (les « conflits d'extensions » sous le Système 7 des Macintosh de l'époque, provoqués par l'introduction d'extensions MIDI dans le système d'exploitation qui n'en incluait pas d'origine), le passage à des OS plus ambitieux, tels Mac OS X (devenu macOS) et ses API MIDI intégrées, a simplifié le processus.

Aujourd'hui, les instruments virtuels sont omniprésents au sein des logiciels de production audio, et ils utilisent le MIDI en interne. Tous les logiciels de station de travail audio possèdent une partie de séquence MIDI, avec éditeur de partition le cas échéant. Le changement le plus visible est l'essor du MIDI *via* USB : de plus en plus de

contrôleurs possèdent un port Type-B sur leur panneau arrière (qui permet également de les alimenter le cas échéant), remplaçant ou complétant la traditionnelle triplette de prises DIN 5 broches MIDI In/Out/Thru. Le format mini-jack 3,5 mm 3 points a également fait son apparition. Et des solutions MIDI sans fil sont aussi utilisables, proposées par de grandes marques, assurant une remarquable fiabilité.

1.2 Spécifications matérielles de la liaison MIDI

1.2.1 Généralités sur les interfaces informatiques

Au sens informatique du terme, la définition du mot interface est la suivante : « dispositif alliant matériel et logiciel, grâce auquel s'effectuent les échanges d'informations entre deux systèmes. » C'est grâce à des interfaces intégrées que l'unité centrale d'un ordinateur communique avec son clavier, son écran, ses supports de stockage de masse, son imprimante, etc.

Quand on tape le chiffre « 1 » et qu'il s'affiche à l'écran, on fait travailler trois systèmes reliés par deux interfaces : le clavier, l'unité centrale et le moniteur. La première interface traduit l'appui sur une touche du clavier en un code compréhensible par le processeur, qui le transforme en une donnée à afficher, laquelle apparaît, *via* une seconde interface, à l'écran.

En règle générale, une interface se définit à la fois par des caractéristiques matérielles (le « moyen de transport » des données, leur format de représentation) et par des caractéristiques logicielles (le langage utilisé). Le rôle de l'interface MIDI est de véhiculer des informations à teneur musicale entre plusieurs systèmes (claviers, générateurs de son, etc.). Nous allons donc, dans un premier temps, approfondir ses spécifications matérielles, avant d'étudier ses spécifications logicielles.

1.2.2 Le MIDI : une liaison sérielle et asynchrone

L'interface MIDI est sérielle, asynchrone et son débit est de 31 250 bauds (bits par seconde) +/- 1 %.

Les données sont transportées sous forme binaire, autrement dit codées sous forme de 0 et de 1, regroupées par octets (groupes de 8 bits, repérés D0 à D7).

Chaque octet est encadré d'un *bit de start* (0 logique) et d'un *bit de stop* (1 logique). Ces bits de start et de stop permettent à l'appareil récepteur d'identifier le début et la fin d'un octet.

Pour faire parvenir 8 bits de données MIDI, il faut donc transmettre un ensemble de 10 bits – parfois appelé, improprement, « octet MIDI ». Ces octets MIDI (*MIDI bytes*, en anglais) sont regroupés par paquets, selon une syntaxe bien précise, pour constituer ce que l'on appelle un message MIDI.

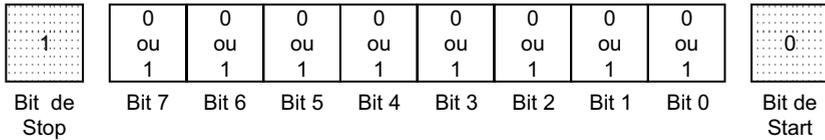


Figure 1.5 – L'octet MIDI est composé de 8 bits utiles, encadrés d'un bit de Start et d'un bit de Stop, permettant de savoir où commence et où finit l'information.

Le concept d'interface *série* signifie que les bits sont acheminés les uns derrière les autres, sur un seul et unique câble. À 31 250 bauds, il faut donc 320 microsecondes pour transmettre un octet MIDI de 10 bits, 10 divisé par 31 250 ; à titre indicatif, pendant cette durée, un son se propage sur 10 cm.

En une seconde, l'interface MIDI transmet donc plus de 3 000 octets MIDI. Il aurait été possible d'envisager une interface *parallèle*, utilisant autant de câbles que de bits à transmettre simultanément, mais la complexité et donc le coût de ce type d'interface ont conduit à son élimination par les concepteurs de la norme MIDI.

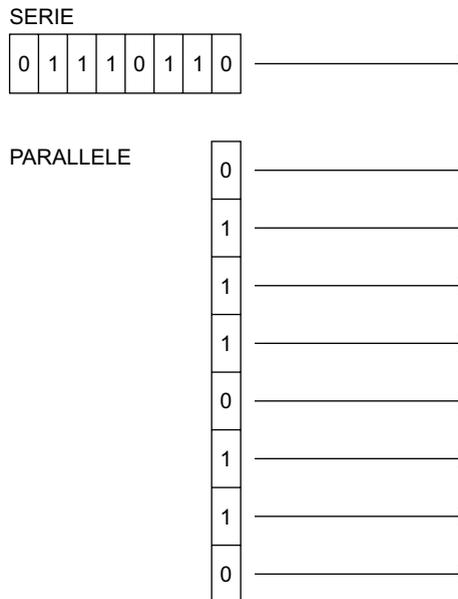


Figure 1.6 – L'interface série véhicule les bits d'un message les uns à la suite des autres sur une même ligne, alors qu'une interface parallèle utilise autant de lignes que de bits à transmettre simultanément. (ex 3;2)

La présence des bits de Start et de Stop est imposée par le caractère asynchrone de la liaison MIDI. Ce sont ces bits qui permettent au récepteur de détecter la présence d'une information, puisqu'il en ignore complètement l'heure d'arrivée. À l'inverse, dans une interface synchrone, l'horloge de l'émetteur et celle du récepteur sont synchronisées.

Plus exactement, le signal d'horloge d'un des appareils est envoyé à l'autre *via* une liaison spécifique, ce qui nécessite un fil supplémentaire. La réalisation d'une liaison synchrone est nettement plus délicate que celle d'une interface asynchrone. De toute façon, elle est inadaptée au MIDI, puisque les données ne sont pas émises à un horaire précis, mais générées à un moment complètement imprévisible, déterminé par le jeu du musicien.

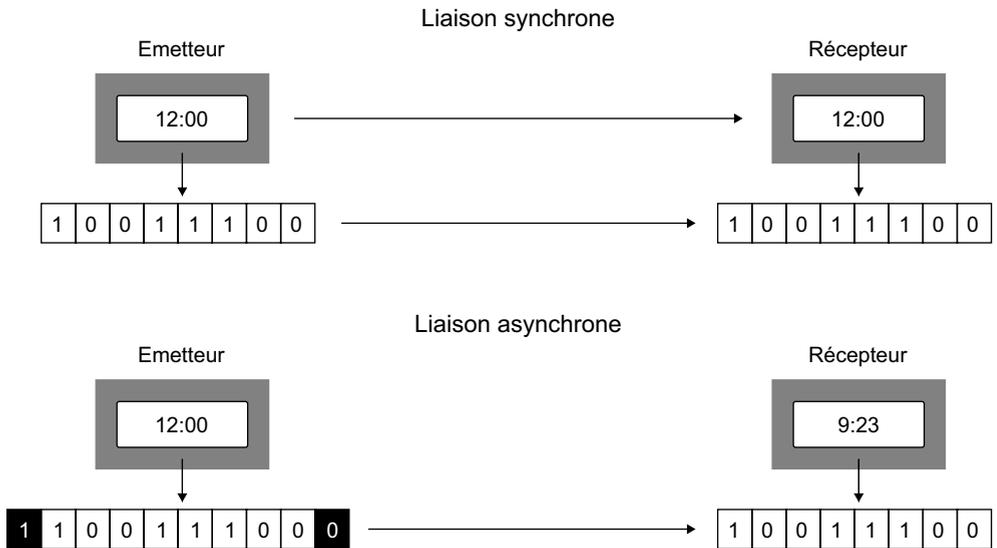


Figure 1.7 – Dans une liaison synchrone, l'heure de l'émetteur est communiquée au récepteur. Dans cet exemple, le récepteur sait qu'à midi pile, l'émetteur lui enverra une donnée. Dans une liaison asynchrone, le récepteur est prévenu de l'arrivée d'une donnée (susceptible d'intervenir n'importe quand) par un tout autre moyen – ici, le bit de start.

1.2.3 Vitesse et densité de données

Nous avons vu que les données des messages MIDI circulent à un débit constant de 31 250 bauds. Ainsi, lorsqu'un appareil récepteur reçoit un bit de Start, il sait que le premier bit de l'octet « utile » arrivera 32 microsecondes plus tard, et ainsi de suite pour les 7 autres bits « utiles », et le bit de Stop. Cependant, rien ne régit l'intervalle de temps séparant deux octets MIDI au sein d'un même message. En théorie, pour optimiser les transmissions, cet intervalle devrait être le plus court possible. Dans la réalité, avec des messages composés de plusieurs dizaines d'octets (de type *System Exclusive*, par exemple), on constate que certains appareils MIDI n'émettent pas les données à plein régime, alors que d'autres saturent en réception si la densité de transmission est utilisée à son potentiel maximal.

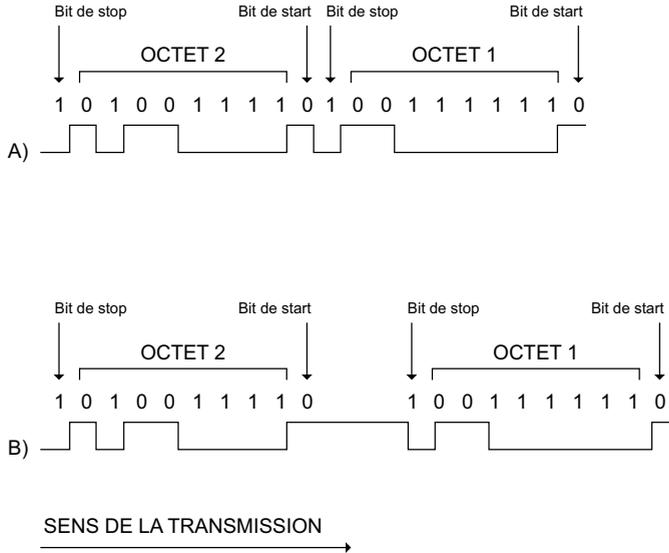


Figure 1.8 – Pour un taux de transfert identique, l'appareil B n'utilise pas toute la bande passante disponible, contrairement à l'appareil A ; l'intervalle entre les octets MIDI est supérieur.

1.2.4 Les connecteurs MIDI

Physiquement, la norme MIDI imposait à ses débuts l'utilisation de connecteurs DIN 5 broches. Certains fabricants ont quand même employé des XLR ou des jacks, mais l'adoption du format DIN, issu de la hi-fi européenne des années 60 donc peu répandu dans les studios et chez les musiciens, présentait l'avantage de distinguer immédiatement les connexions MIDI des connexions audio. Selon d'autres sources, ce type de connecteur était tombé en désuétude quelques années auparavant, et les fabricants japonais en possédaient des stocks dont ils souhaitaient se débarrasser.

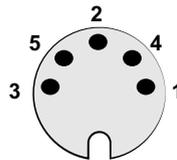


Figure 1.9 – Le brochage d'une prise DIN (ex 3.5).

Fonctionnellement, l'interface MIDI utilise trois types de prises : MIDI In, MIDI Out et MIDI Thru. Leurs rôles respectifs sont détaillés plus loin dans ce chapitre. Les broches 1 et 3 sont inutilisées, la broche 2 est réservée à la masse, et ce uniquement sur les prises MIDI Out et MIDI-Thru (la prise MIDI In étant dépourvue de masse). Les broches 4 et 5 véhiculent une boucle de courant électrique de 5 milliampères. Une tension logique de + 5 volts correspond à un bit à 0, tandis qu'une tension logique de 0 volt représente un bit à 1.

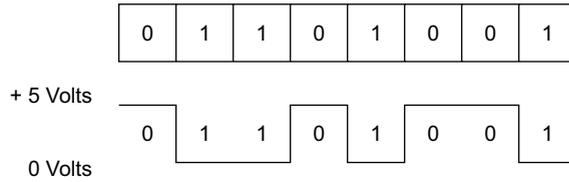


Figure 1.10 – La représentation d'un octet sous forme d'une tension logique, 0 V, + 5 V.

Notons que l'auteur/musicien/spécialiste ès technologie musicale américain, Craig Anderson, a proposé, en 1989, d'ajouter une tension fantôme de 9 à 15 volts sur le connecteur MIDI. L'idée était évidemment de pouvoir alimenter directement des interfaces, par exemple, peu gourmandes. Elle n'a pas connu d'application commerciale : aujourd'hui bien des interfaces MIDI sont alimentées par le port USB de l'ordinateur auquel elles sont reliées.

Bien qu'il existe des câbles vendus comme spécifiquement MIDI, il est tout à fait envisageable d'utiliser des câbles hi-fi traditionnels, à la condition impérative que deux broches ne soient pas inversées entre les deux extrémités du câble ni « strappées » (soudées deux à deux). Les câbles vendus en magasin de musique sont toutefois généralement plus robustes, ainsi que leurs connecteurs. En tout état de cause, la longueur d'un câble MIDI ne doit pas dépasser 15 mètres : au-delà, on s'expose à une possible altération des données au cours de leur transport.

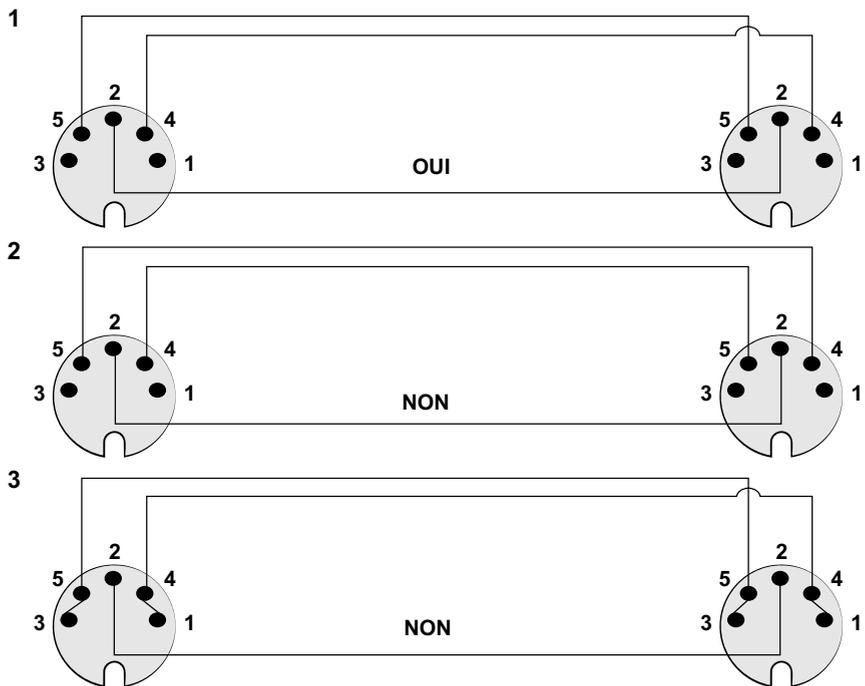


Figure 1.11 – Parmi ces trois cordons MIDI (masse en 2, conducteurs en 5 et 4), seules les connexions du premier sont correctes. En effet, les broches 5 et 4 du deuxième sont inversées, alors que les points du troisième sont « strappés » (5/3 et 4/1 soudées entre elles).

Depuis la fin des années 2000, il est de plus en plus fréquent de voir sur les appareils MIDI des ports USB, doublant ou remplaçant les traditionnels connecteurs DIN. Le débit numérique et le *jitter* (irrégularités d'horloge) de la liaison USB sont parfaitement compatibles avec le transport de données MIDI. Une extension à la norme MIDI, « USB-MIDI », a été élaborée dans ce but.

1.2.5 *L'isolation opto-électrique*

Chaque connecteur MIDI possède une isolation opto-électrique, afin d'éviter les parasites électriques, notamment l'apparition de boucles de masse. Le principe d'un tel dispositif consiste à alimenter une LED (diode électroluminescente) par la boucle de courant. Suivant son état de tension, la LED éclaire ou non un transistor photoélectrique, qui à son tour laisse ou ne laisse pas passer le courant électrique. Grâce à cette isolation, les risques d'interférences entre les informations MIDI et les circuits analogiques et numériques de l'instrument les utilisant sont éliminés.

1.3 **L'interface série/parallèle**

Les microprocesseurs communiquent en parallèle, *via* des bus de données d'une largeur de 8, 16, 32, 64... bits. Il faut donc interfacer le MIDI avec le processeur de façon à transformer, à la réception, une communication série en communication parallèle, et inversement lors de l'émission.

1.3.1 *L'interfaçage*

En MIDI, la conversion série/parallèle s'effectue par l'intermédiaire d'un circuit spécialisé nommé UART (*Universal Asynchronous Receiver Transmitter*) ou ACIA (*Asynchronous Communications Interface Adapter*). À réception d'un bit de Start sur la prise MIDI In, l'interface série se prépare à recevoir 8 bits les uns derrière les autres, afin de les stocker un à un dans sa mémoire interne. Le bit de Stop lui confirmant une transmission correcte de l'octet, elle est alors prête à envoyer d'une seule salve les 8 bits au microprocesseur (sur 8 fils différents, le « bus de données »). Cette transmission effectuée, l'UART bascule en mode d'attente du prochain bit de Start, et ainsi de suite.

1.3.2 *Buffer et interruptions*

Imaginons que l'UART et le microprocesseur soient remplacés par deux téléphones. Si le vôtre (le microprocesseur) ne possédait pas de sonnerie, il vous faudrait décrocher sans arrêt pour savoir si quelqu'un vous appelle (si une donnée lui est envoyée par l'UART). La sonnerie permet de ne décrocher qu'à bon escient : lorsqu'un correspondant désire vous parler. C'est un « système d'interruption », qui vous permet de mener n'importe quelle activité en parallèle, même si vous attendez un appel.

En informatique, le concept est identique. Pour éviter au microprocesseur de devoir guetter en permanence les octets MIDI présents en sortie de l'UART, on utilise un « système d'interruption ». Ainsi, dès que l'UART a réceptionné une donnée MIDI, elle fait savoir au microprocesseur, par l'intermédiaire d'une ligne directe, qu'elle est prête à la lui envoyer. Le microprocesseur interrompt alors ses activités en cours et accorde la priorité à cette opération de réception, en stockant éventuellement l'octet MIDI dans une mémoire tampon (*buffer*), afin de la traiter ultérieurement. Ce buffer s'apparente à un répondeur téléphonique, qui enregistre les appels (les données) tandis que vous êtes occupé à une autre tâche.

L'avantage de l'association interruption/mémoire tampon est triple :

- le microprocesseur peut vaquer à ses occupations sans devoir scruter la prise MIDI In ;
- chaque octet est pris en compte au moment précis de son arrivée ;
- le risque d'une perte irrémédiable de données n'intervient qu'à saturation du buffer.

En effet, tout comme la mémoire interne d'un répondeur téléphonique n'accepte qu'une certaine durée de messages, le buffer, s'il n'est pas vidé assez rapidement, peut saturer à réception d'un volume excessif de données. Les octets suivants sont alors irrémédiablement perdus. L'utilisateur en est averti par des messages de type « MIDI Overflow », « Buffer Overflow », « MIDI Buffer Full », etc. D'où l'intérêt, pour un appareil MIDI, de disposer d'un buffer de taille convenable.

INTERFACE PARALLELE/SERIE

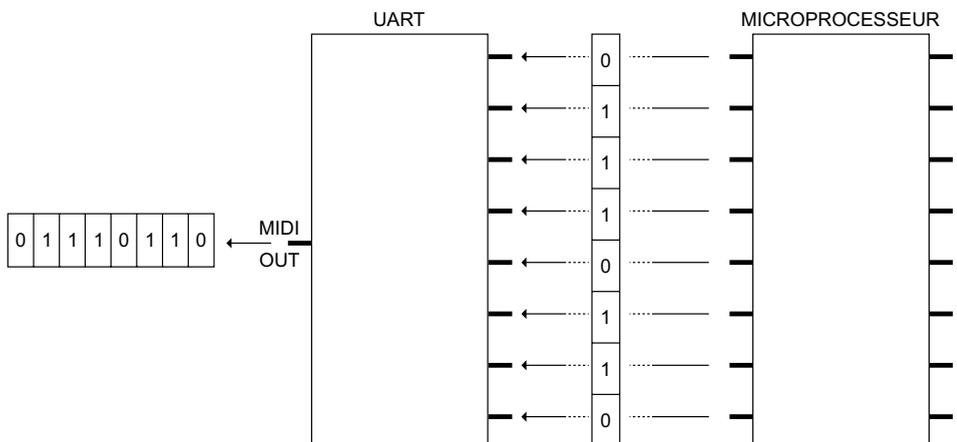


Figure 1.12 – L'interface série/parallèle (haut) achemine les 8 bits d'un octet MIDI sur 8 lignes distinctes, à destination du processeur. L'interface parallèle/série (bas) effectue l'opération inverse.