LE WEB À L'HEURE DU

Benoit Viguier
@b_viguier

M6Web LFT 29/09/2017

Vendredi Confession

#1

Parfois...

... je fais du Javascript...

#2

J’aurais voulu être

...

🎹 Musicien 🎷

Javascript ET Musique

Do you speak « Music »?

2015: 18% des Français jouent d'un instrument de Musique

RÉGION
MIDI-PYRÉNÉES

⚠ Pas ce Midi là ⚠

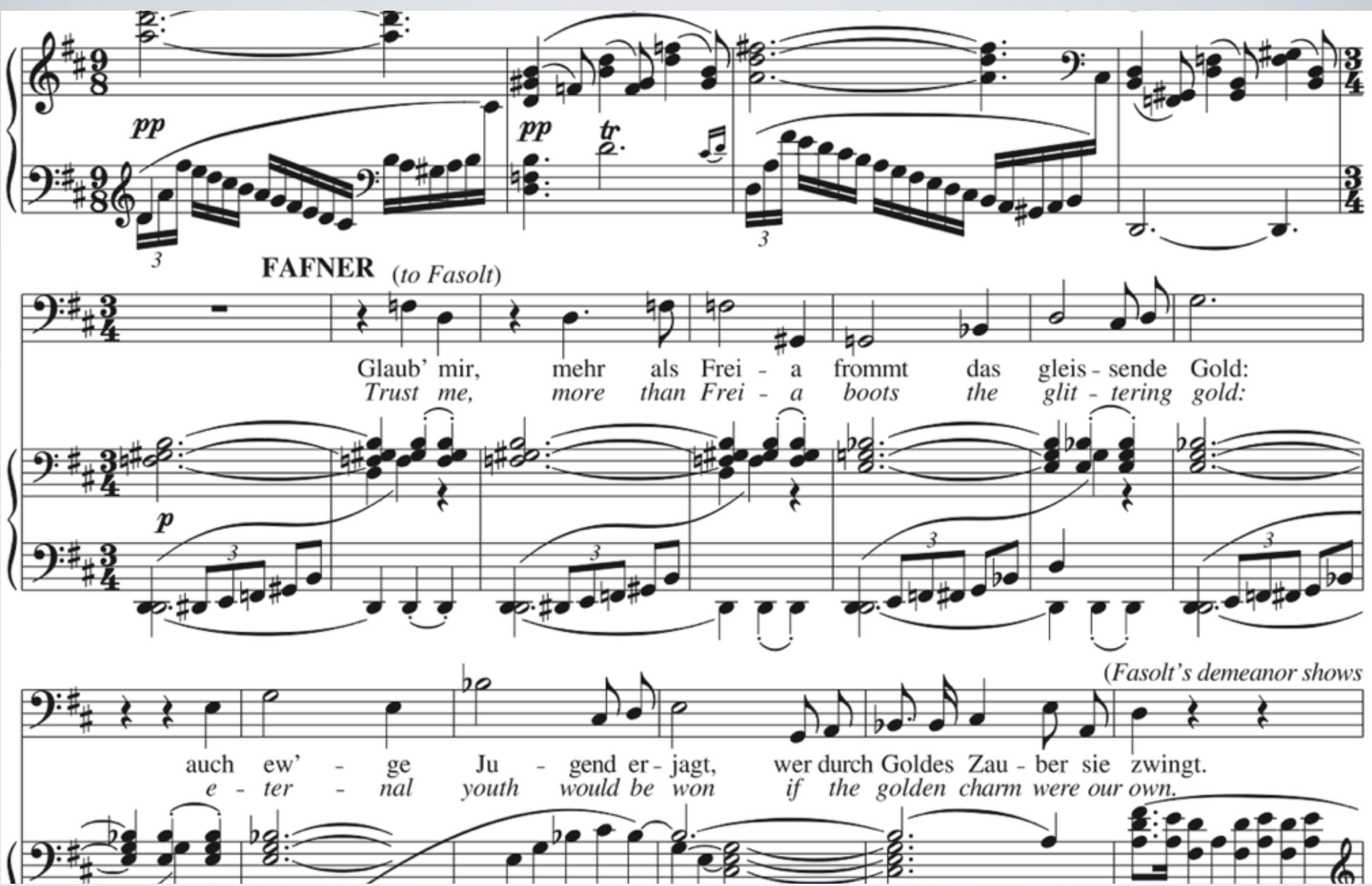
Musical Instrument Digital Interface

« Le MIDI c’est pas le truc avec des sons pourris ? »

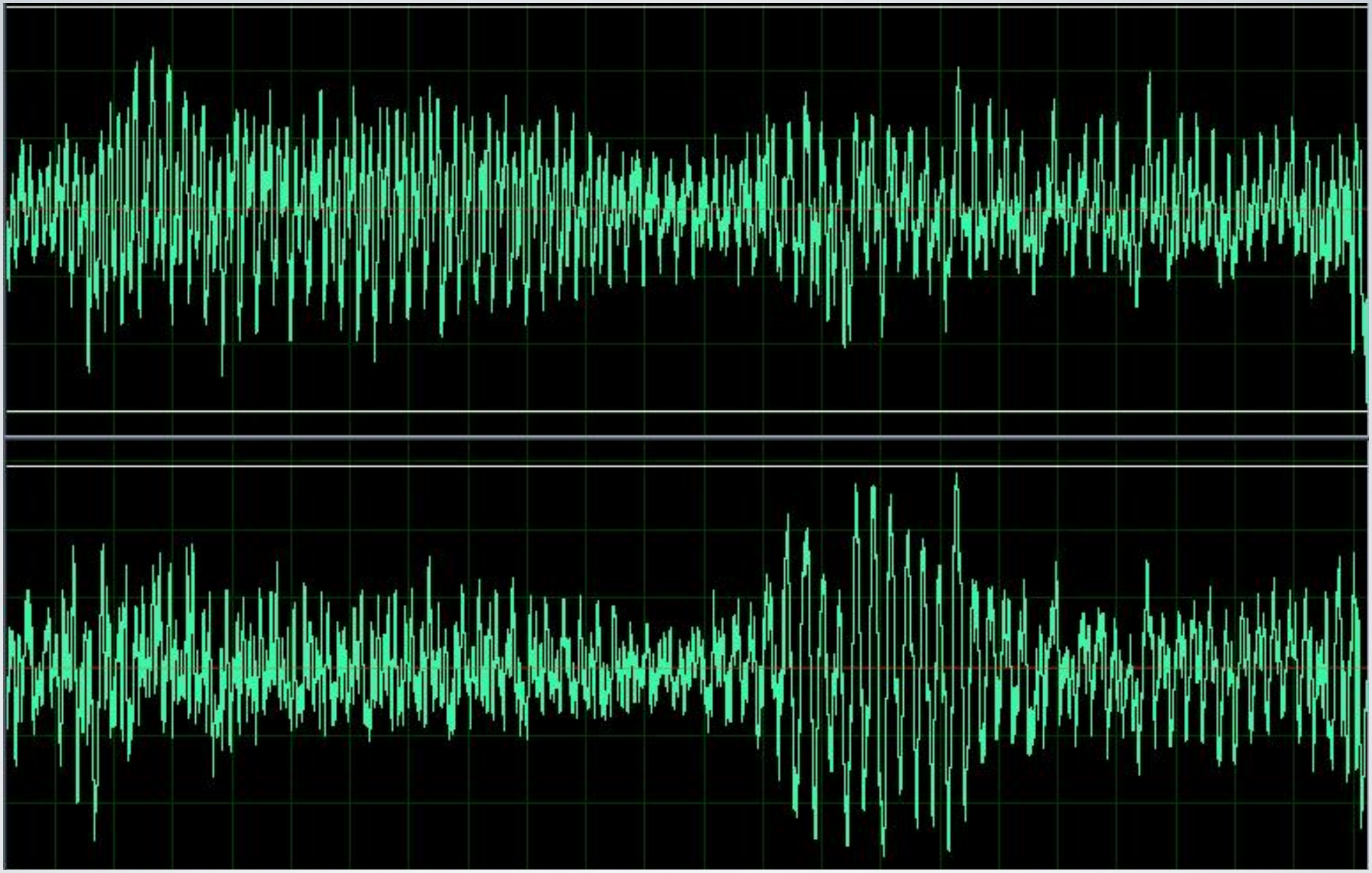
-Guillaume B.

Fichier MIDI

*.mid *.midi *.smf



MIDI = évènements



Son (*.mp3, *.wav ...)

Le MIDI est plus vieux que le WEB !

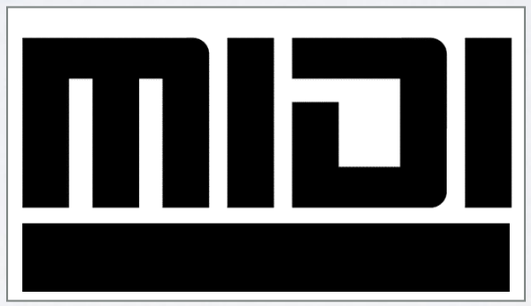
1983 vs 1989

Microsoft
Microsoft
Windows98
Upgrade

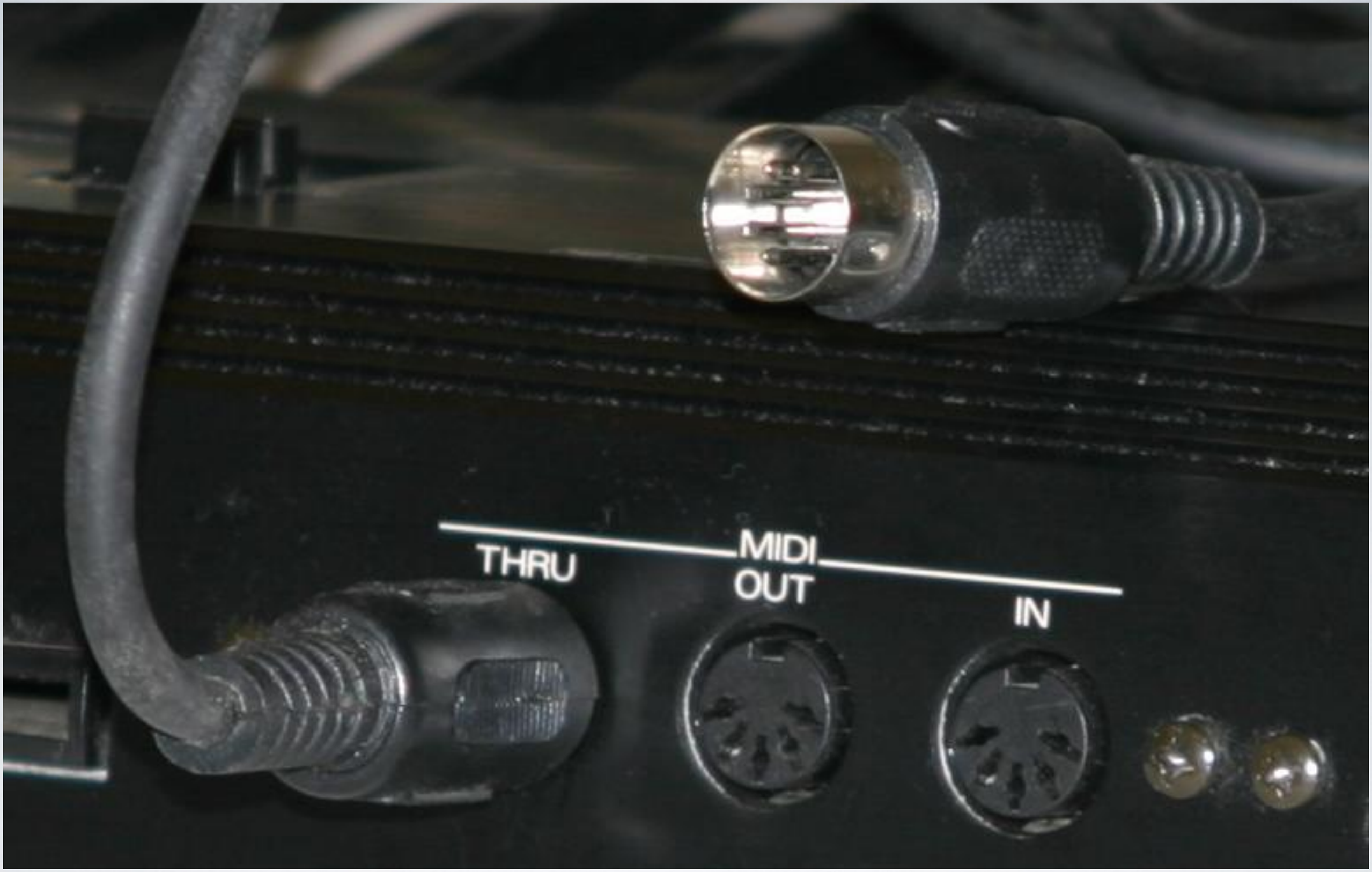
Le MIDI

Vu de l'extérieur

Contrôleur



Module de son



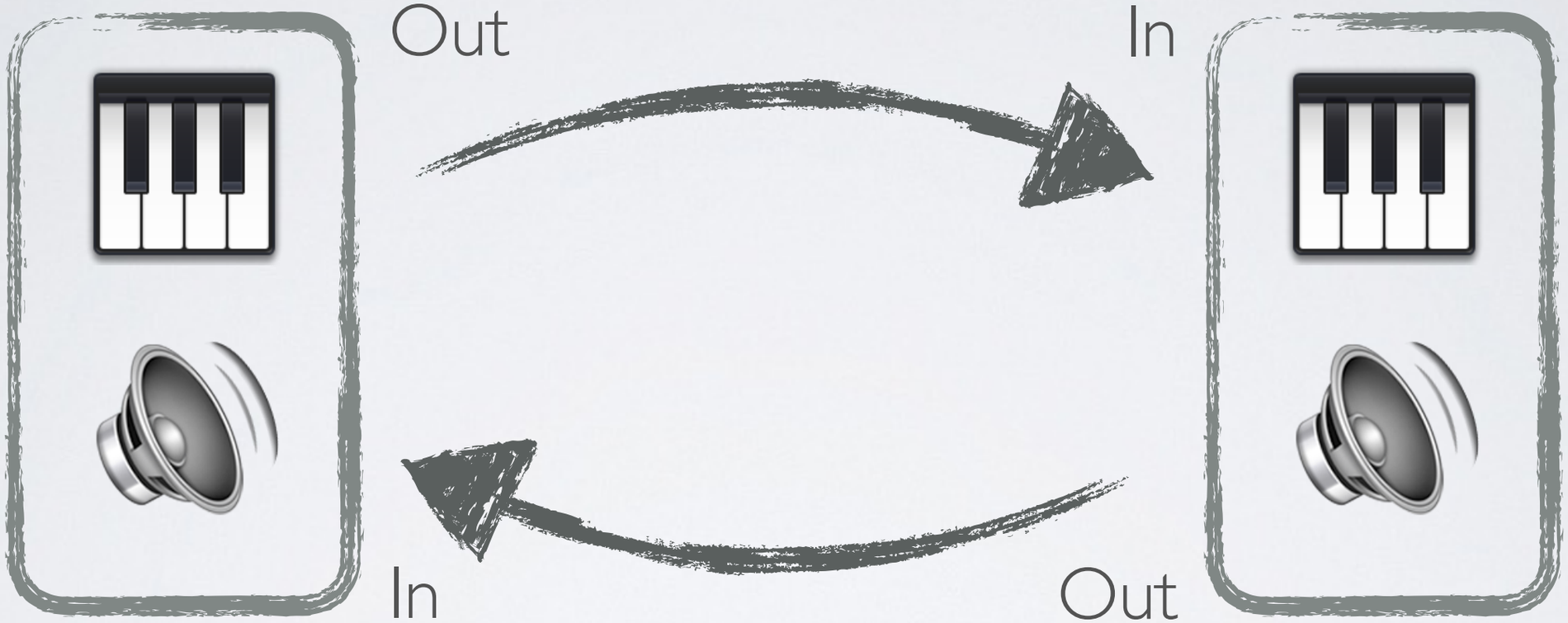
Connectique

Out

In

Out

In


Out
In
In
Out

Out

In

Thru

In

Utilisations

Scène, Composition, sonneries mobiles, apprentissage...

Pas seulement pour la musique

Des shows, des jeux…

MIDI Show Control

Contrôleurs

Exemples

AKAI
professional
LPK25
ARPEGGIATOR
1/4
1/4T
1/8
1/8T
1/16
1/16T
1/32
1/32T
UP
DOWN
EXCL
INCL
ORDER
RAND
ARP OCT 0
ARP OCT 1
ARP OCT 2
ARP OCT 3
PROG 1
PROG 2
PROG 3
PROG 4

Akai LPK25

AX-Synth

TD-50KV

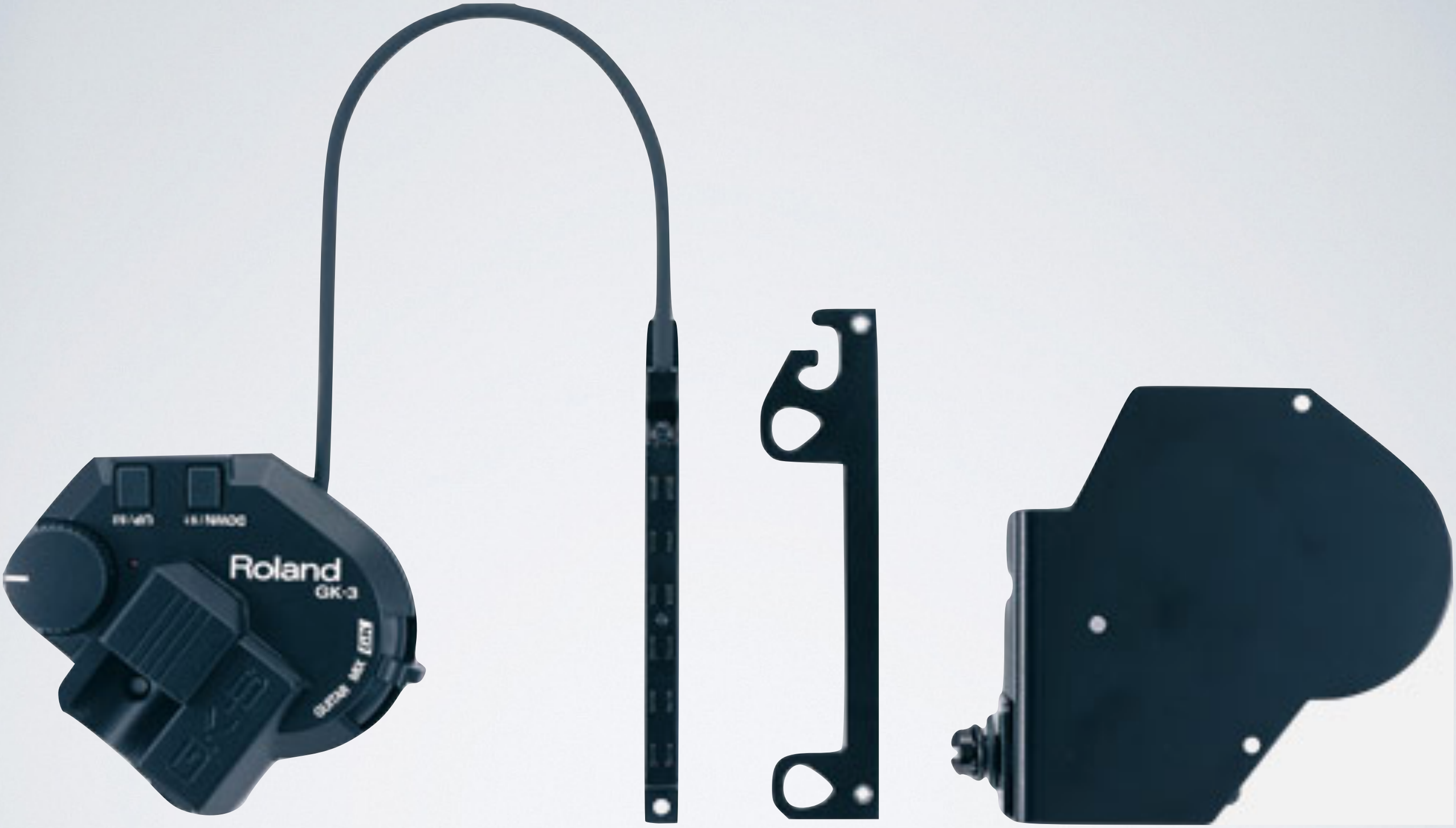
OCTAPAD SPD-30

RockBand Drum Controler

Novation Circuit

PITCH BEND
A VOLUME
MASTER
LOAD
BROWSE
LOAD
A
B
BACK
ENTER
B VOLUME
PITCH BEND
SYNC
DECK A
CUE
Numark
DJ2GO
SYNC
DECK B
CUE

DJ 2 Go



Roland GK-3

DC IN
POWER
COMPUTER
IN MIDI OUT
PHONES
L/MONO OUTPUT R
GUITAR OUT
GK IN
Roland GR-55 GUITAR SYNTHESIZER
OUTPUT LEVEL
EXP PEDAL SW ON/OFF
USB MEMORY
LEAD 01-1
Metal Synth Lead
PAGE
EDIT
SOUND STYLE
V-LINK
LEAD
RHYTHM
OTHER
USER
EZ EDIT
EXIT
ENTER
WRITE
AUDIO PLAYER
USB MEMORY
1
BANK
2
BANK
3
CTL
REC/PLAY/DUB
BANK SELECT
TUNER
PHRASE LOOP
GR
COSM

Roland GR-55

EWI 5000

Module de son

Exemples

Roland
FA
06
D-BEAM
SOUND MODIFY
KEYBOARD
ARP/RHYTHM
SEQUENCER
SAMPLER
0188:Metal Pad

Roland FA-06

Roland
JP-08
LFO
RATE
DELAY TIME
RND
VCO MOD
LFO MOD
ENV MOD
FREQ MOD
VCO-1
VCO-1+VCO-2
VCO-2
PULSE WIDTH MOD
LFO
M
ENV-1
CROSS MOD
VCO-1
32 16 8 4 2 64
SYNC
ON
OFF
VCO-2
TUNE
SOURCE MIX
LOW FREQ
HPF
CUTOFF
VCF
CUTOFF
RES
SLOPE
−12dB
−24dB
LFO MOD
ENV MOD
ENV-1
ENV-2
KEY FOLLOW
VCA
LEVEL
LFO MOD
ENV-1
A D S R
ENV-2
A D S R
KEY FOLLOW
1+2
ENV-2
ENV-1
OFF
PATCH NUMBER 11-88
1 2 3 4 5 6 7 8
PATCH NUMBER
PATCH PRESET
1 2 3 4 5 6 7 8
UPPER/ LOWER
DUAL
MANUAL
SOLO — UNISON — POLY
SEQ
►/■

JP-08

Roland Jupiter-8

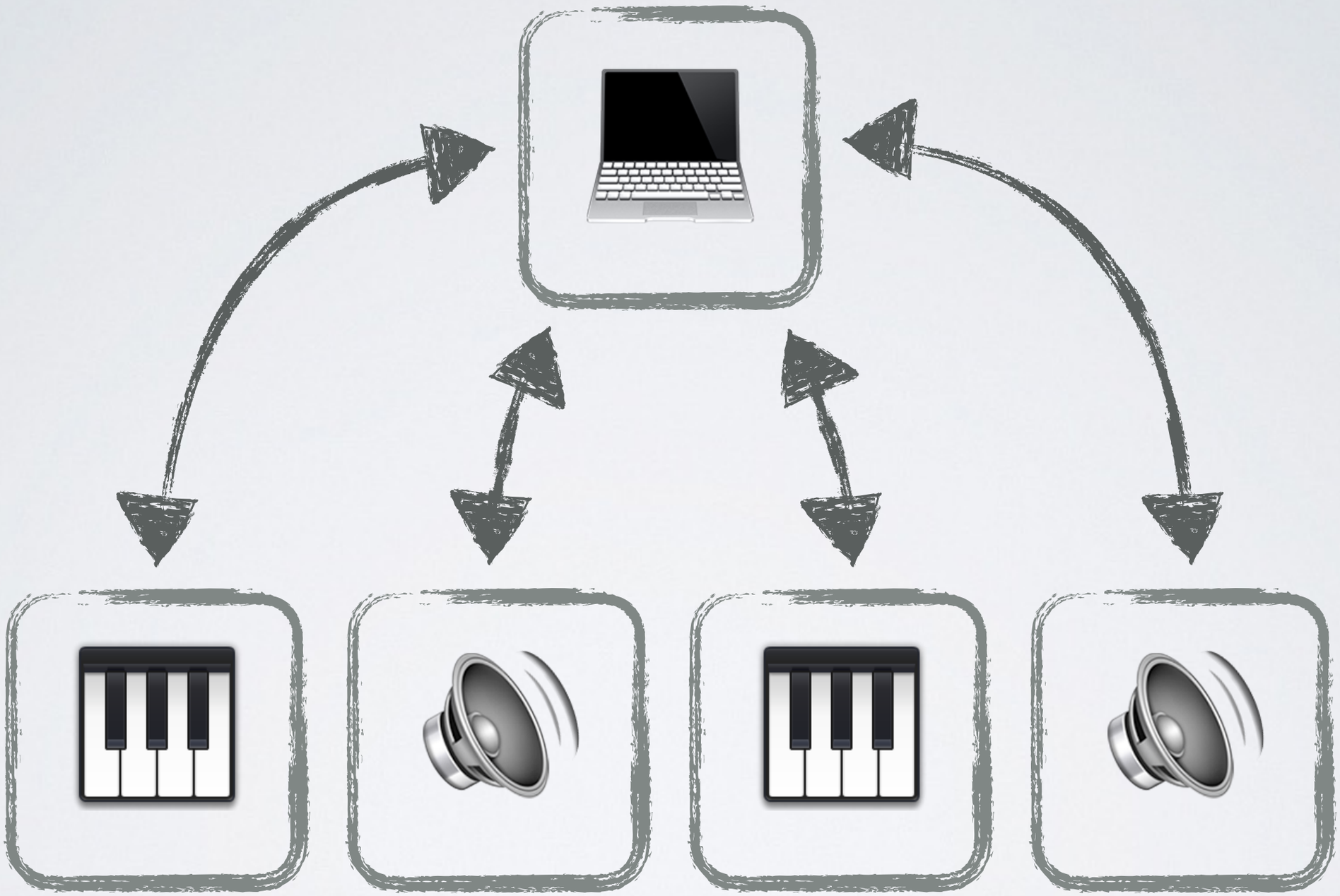
Rack

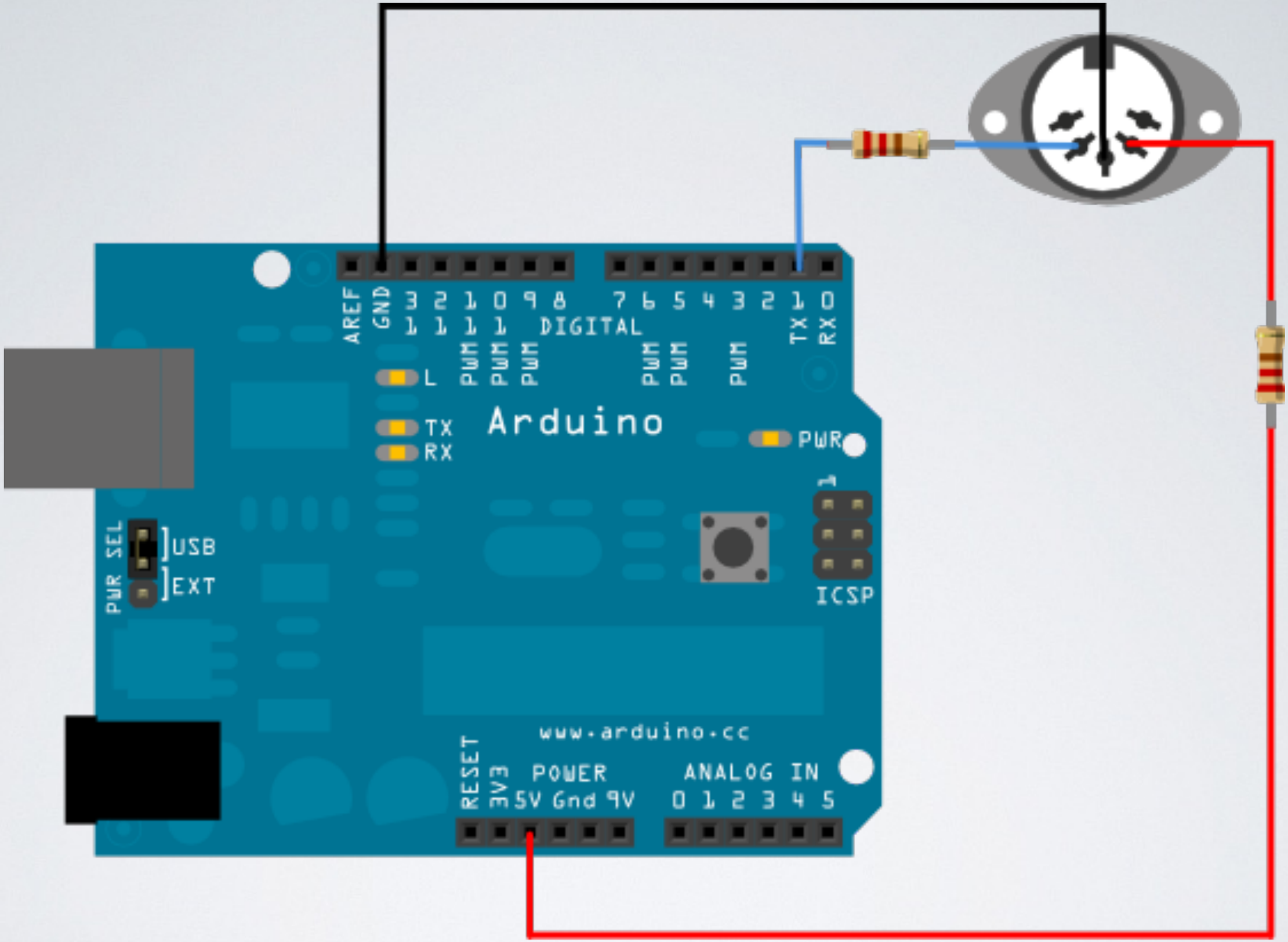
Et l'ordinateur ?

Contrôleur ET module de son

M-Audio Uno

M-Audio MidiSport




AREF
GND
3 2 1 0 9 8 7 6 5 4 3 2 1 0
DIGITAL
TX
RX
L
PWM
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
3V3
POWER
5V Gnd 9V
ANALOG IN
0 1 2 3 4 5

Arduino

Le MIDI

Vu de l'intérieur…

RTFM !

midi.org/specifications

16 canaux

≈16 instruments

éan
tique

France

Castets-
en-Dorthe

Canal de
Garonne

Toulouse

Sète

Canal du Midi

Me
Méditer

Espagne

Message Binaire

1xxxxxxx	0xxxxxxx	0xxxxxxx
Status	Data (optionnel)	Data (optionnel)

Voice messages

On, Off, Pitch Bend, Sustain…

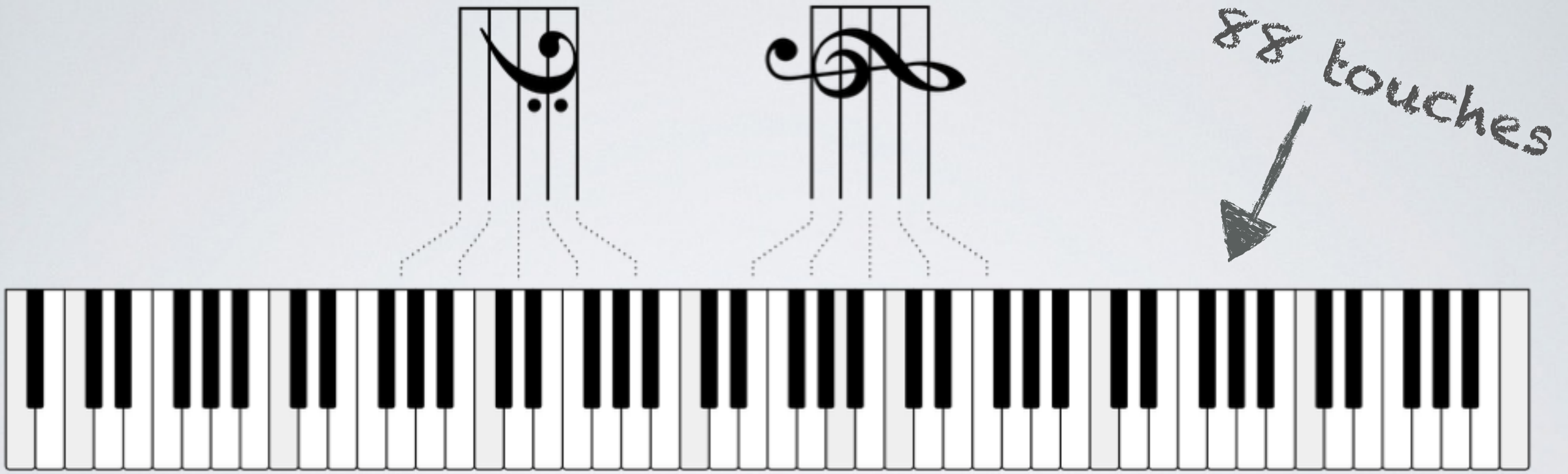
Note On

1001nnnn	0kkkkkkk	0vvvvvvv
Canal	Note	Vélocité

Note Off

1000nnnn	0kkkkkkk	0vvvvvvv

Canal Note Vélocité



Fre-quency	MIDI	[inter-national]	German
4186	108	C8	c'''''
.....			
2093	96	C7	c'''''
.....			
1046.5	84	C6	c''''
.....			
523.25	72	C5	c'''
.....			
440	69 70 71	A4 A#/Bb4 B4	a' ais/b' h'
.....			
261.6	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108	C4 C#/Db4 D4 D#/Eb4 E4 F4 F#/Gb4 G4 G#/Ab4 A4 A#/Bb4 B4 C5 C6 C7 C8	c' cis/des' d' dis/es' e' f' fis/ges' g' gis/as' a' ais/b' h' c'' c''' c'''' c'''''
.....			
130.81	48	C3	c0
.....			
65.406	36	C2	C
.....			
32.703	24	C1	C'
.....			
27.5	21 22 23	A0 A#0 B0	A'' Ais'' H''

128 Notes

Do
C

Ré
D

Mi
E

Fa
F

Sol
G

La
A

Si
B

Notation musicale

Program Change

1100nnnn	0ppppppp	
Canal	Programme	Void

Normes d’instruments

GM1, GM2, GS, XG

Canal 10 = Drums

Control Changes

Volume, souffle, expression…

Control Change

1011nnnn	0ccccccc	0vvvvvvv
Canal	Control Change	Valeur

Channel Mode Messages

Polyphonie, reset…

All Notes Off

1011nnnn	01111011	00000000
Canal	All Notes Off	0

System Real Time

Clock, Active sensing…

System Common Messages

Song position, Tune…

System Exclusive

Le standard dans le standard

SysEx

Begin Data End

Web MIDI API

Javascript

PARENTAL ADVISORY EXPLICIT JS

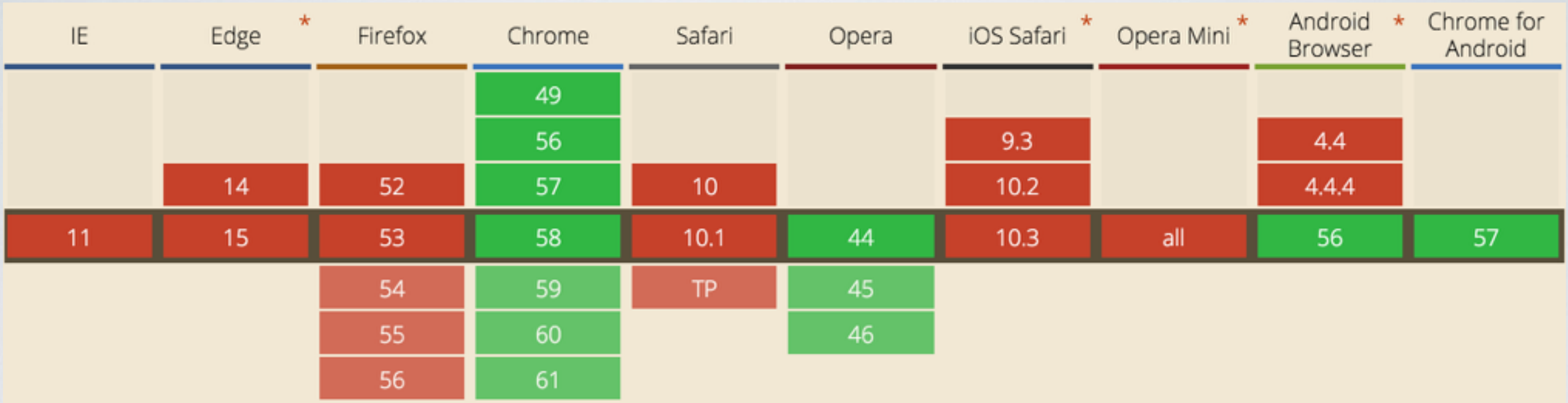
Pourquoi dans le navigateur?

À part la *Hype*

« The Web-MIDI API is the most significant advancement of MIDI since… MIDI itself! »

midi.org

MIDI
MIDI EVERYWHERE
SPACE RANGER
LIGHTYEAR



Compatibilité

Jazz-Plugin

http://jazz-soft.net/

Documentation

https://webaudio.github.io/web-midi-api/

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
partial interface Navigator {

  Promise<MIDIAccess> requestMIDIAccess(
      optional MIDIOptions options
  );

};
```

```
dictionary MIDIOptions {

   boolean sysex;
   boolean software;

};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIAccess: EventTarget {

    MIDIInputMap  inputs;

    MIDIOutputMap outputs;

    EventHandler  onstatechange;

    boolean       sysexEnabled;
};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIInputMap {

  readonly
    maplike<DOMString, MIDIInput>;

};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIInput : MIDIPort {
    EventHandler onmidimessage;

};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIMessageEvent: Event {
    Uint8Array data;

};
```

High Resolution Time

window.preformance.now()

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIOutput : MIDIPort {

  void send(
    sequence<octet> data,
    DOMHighResTimeStamp timestamp = 0
  );

  void clear();

};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

```
interface MIDIPort : EventTarget {
  DOMString     id;
  DOMString?    manufacturer;
  DOMString?    name;
  MIDIPortType type;
  DOMString?    version;
  MIDIPortDeviceState      state;
  MIDIPortConnectionState connection;

  EventHandler onstatechange;

  Promise<MIDIPort> open();
  Promise<MIDIPort> close();
};
```

```
var output = null;

function echoMIDIMessage( event ) {
    if (output) {
        output.send( event.data, event.timeStamp );
    }
}

function onMIDISuccess( midiAccess ) {
    console.log( "MIDI ready!" );
    var input = midiAccess.inputs.values().next().value;
    if (input)
        input.onmidimessage = echoMIDIMessage;
    output = midiAccess.outputs.values().next().value;
    if (!input || !output)
        console.log("Uh oh! Couldn't get i/o ports.");
    else
        console.log(input.name, output.name);
}

function onMIDIFailure(msg) {
    console.log( "Failed to get MIDI access - " + msg );
}

navigator.requestMIDIAccess().then( onMIDISuccess, onMIDIFailure );
```

API bas niveau

Envoyer/Recevoir des messages

API « Moyen » Niveau

Sécurité & Vie Privée

Détournements possibles

Interface virtuelle

La *feature* manquante ?

Démonstrations

Show Time !

Hardware / Software

Présentation

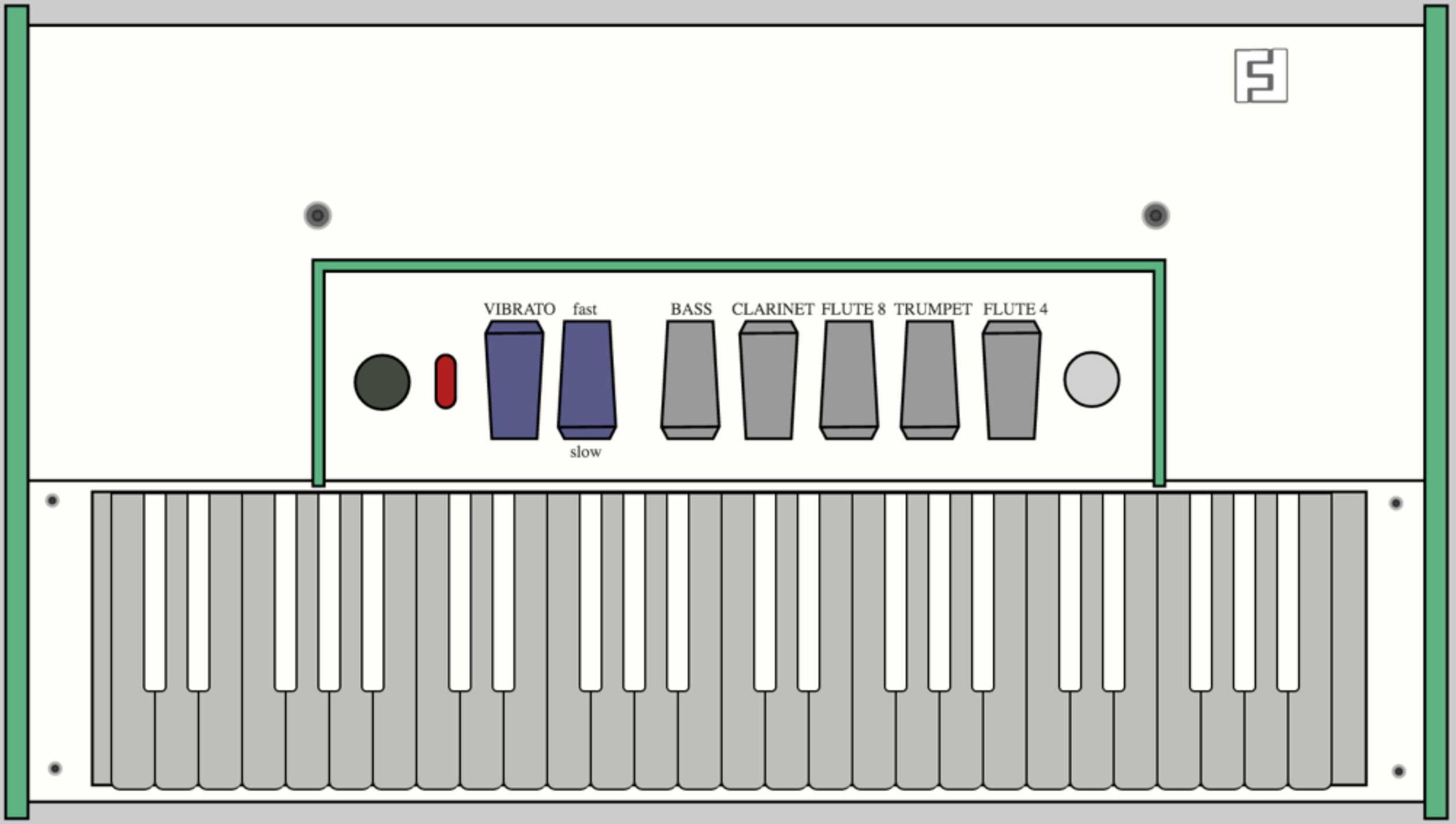
AKAI
professional
LPK25
ARPEGGIATOR
1/4
1/4T
1/8
1/8T
1/16
1/16T
1/32
1/32T
UP
DOWN
EXCL
INCL
ORDER
RAND
ARP OCT 0
ARP OCT 1
ARP OCT 2
ARP OCT 3
PROG 1
PROG 2
PROG 3
PROG 4

Akai LPK25

EWI 5000

This example uses Web MIDI input (from all available ports and channels) to drive a Web Audio monophonic synthesizer. There is no UI. Source available on github.

https://webaudiodemos.appspot.com/monosynth/



http://www.farfisaofthefuture.com/

https://webaudiodemos.appspot.com/midi-synth/

Demo Everywhere!

https://webaudiodemos.appspot.com/

Midi Browser

Sequencer Midi

Pour finir

Quoi retenir ?

Merci

@b_viguier