MIDI

What is MIDI?

- **No Longer** Exclusively the Domain of Musicians.
- Midi provides a very low bandwidth alternative on the Web:
  - transmit musical and
  - certain sound effects data
- also now used as a *compression control language (modified)*
  - See MPEG-4 Section soon




MIDI on the Web

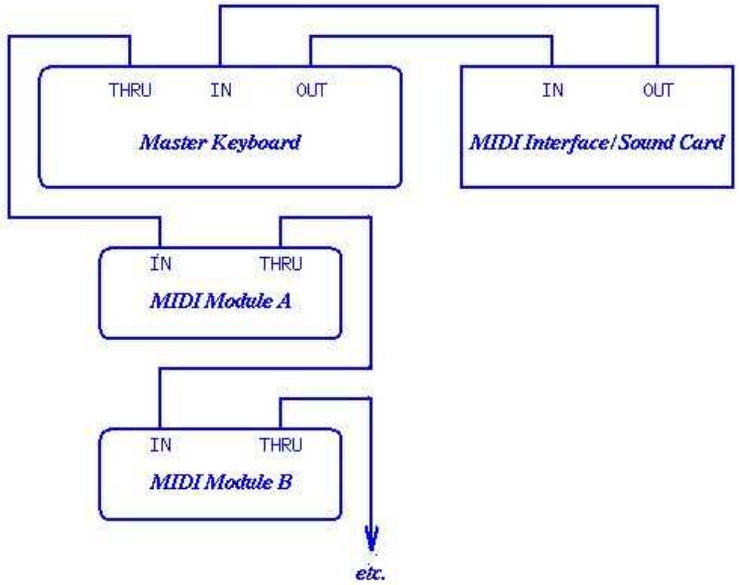
Very Low Bandwidth (few 100K bytes)

- The responsibility of producing sound is moved to the client:
  - Synthesiser Module
  - Sampler
  - Soundcard
  - Software Generated
- Most Web browsers can deal with MIDI.
  - Available as plugins (e.g. Quicktime) and (as of 2013) as Web MIDI API in HTML 5 — (**More Soon**)




Definition of MIDI:

A protocol that enables computers, synthesisers, keyboards, and other musical devices to communicate with each other.



Brief History of MIDI:

Midi is now 30 Years old (as of 2012/3).

However MIDI is still very much alive and kicking.

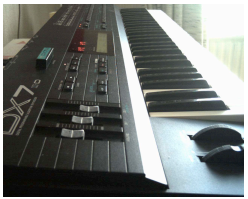
- Old meets new: iPad plays old Commodore Sequencer!
- Brief History: BBC News Web Article
- The protocol is still evolving: High Definition MIDI in Pipeline (2013). (**More soon**)

Not bad for a 30 Year Old Hi-Tec Media Protocol!

Components of a MIDI System

Synthesiser/Sampler:

- It is a sound generator (various pitch, loudness, tone colour)
- Can use a variety of synthesis or Sample-based synthesis to make sound.
- A good (musician's) synthesiser often has a microprocessor, keyboard, control panels, memory, etc.
- For our purposes we define a synthesiser as the tone generation unit.
- It has one or more MIDI INs and MIDI OUTs and/or USB/Firewire connectivity
- Can be software based these days so virtual midi connections.



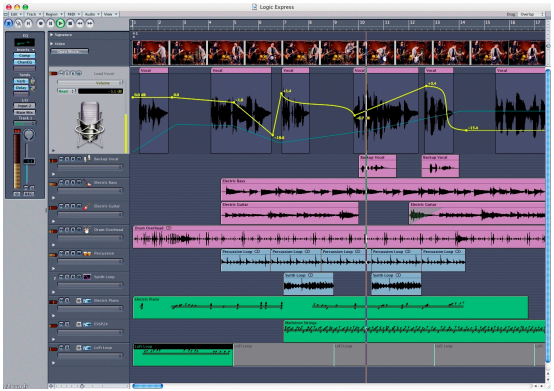

Sequencer:

- It can be a stand-alone unit or a software program for a personal computer. (It used to be a storage server for MIDI data. Nowadays it is more a software *music editor* on the computer.
- It has one or more MIDI INs and MIDI OUTs and/or USB/Firewire connectivity
- If sofware based virtual midi connections

Components of a MIDI System (Cont.)

Computer:

- Heart of a MIDI system
- Controls the scheduling, synchronisation and recording of all data.
- Sequencer usually software based and now part of larger applications that control all aspects of Audio and Midi — Digital Audio Workstation packages such as Cubase, Logic, Sonar, Live, Reason.
- Nowadays, includes many software synthesisers/samplers to make sounds in real time.
- Real time effects
- Control of Video also integral these days.



Components of a MIDI System (Cont.)

Midi Control Input Devices:

- Usually a Keyboard with additional control: sustain, pitch bend,modulation, aftertouch and other controllers
- Can be another musical device e.g. Customised Guitar, Wind Controller
- Can be just a bunch of controllers.
- Can be even more strange: Motion Capture, or virtual input or mind control!!

Components of a MIDI System (Cont.)

Midi Interfaces:

Midi devices (still) need to connect to computer with some interface

- Midi Interface — USB or Firewire
- Often functionality bundled with Keyboard or controller
- Audio Interface via USB or Firewire common
- Even Wireless Keyboards

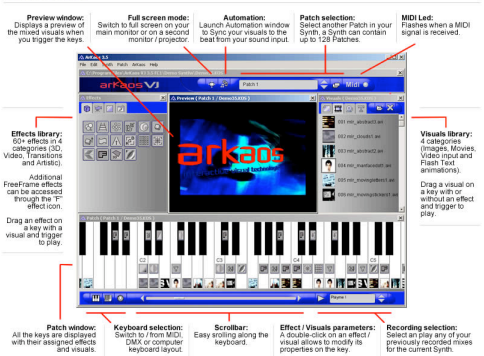
Components of a MIDI System (Cont.)

Midi Control Output Devices:

- Not just making sounds
- MIDI controls other things
- Lighting
- Robotics
  - Even Pat Metheny and his Musical Robot Band: Orchestrion!!
- Video Systems e.g. Video DJing
- MPEG4 Compression — More soon
- Even Hamster Control!!!
- Lots of other applications

  For a full range of MIDI I/o Controllers check out http://www.synthzone.com/ctrlr.htm

ArKaos VJ MIDI



Basic MIDI Concepts

Track:

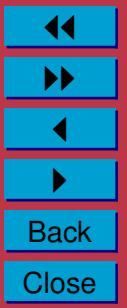
- Track in sequencer is used to organize the recordings.
- Tracks can be turned on or off on recording or playing back.

Channel:

- MIDI channels are used to separate information in a MIDI system.
- There are 16 MIDI channels in one *'cable'*.
- Channel numbers are coded into each MIDI message.

Timbre:

- The quality of the sound, e.g., flute sound, cello sound, etc.
- Multitimbral – capable of playing many different sounds at the same time (e.g., piano, brass, drums, etc.)



Basic MIDI Concepts (Cont.)

Pitch:

- The Musical note that the instrument plays

Voice:

- Voice is the portion of the synthesiser that produces sound.
- synthesisers can have many (12, 20, 24, 36, etc.) voices.
- Each voice works independently and simultaneously to produce sounds of different timbre and pitch.

Patch:

- The control settings that define a particular timbre.

Hardware Aspects of MIDI

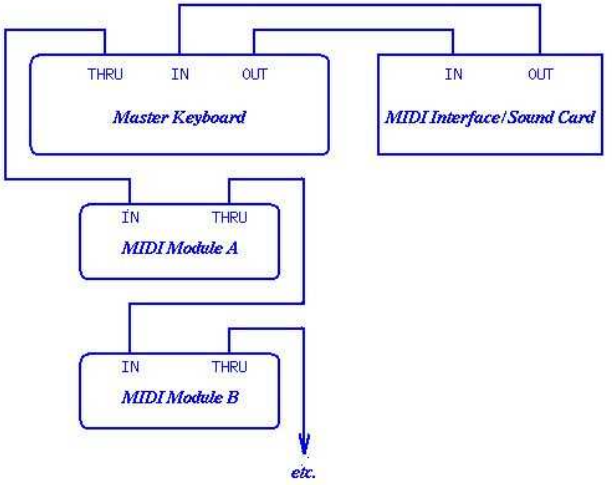
MIDI connectors:

– Three 5-pin ports found on the back of every MIDI unit

- **MIDI IN**: the connector via which the device receives all MIDI data.
- **MIDI OUT**: the connector through which the device transmits all the MIDI data it generates itself.
- **MIDI THROUGH**: the connector by which the device echoes the data receives from MIDI IN.

– As mentioned previously many modern interfaces connect via USB/Firewire

- Many devices bypass direct MIDI IN/OUT/THROUGH and have a direct (or possibly even wireless) USB/Firewire connection to the computer.



MIDI Messages

MIDI messages are used by MIDI devices to communicate with each other.

MIDI messages are very low bandwidth:

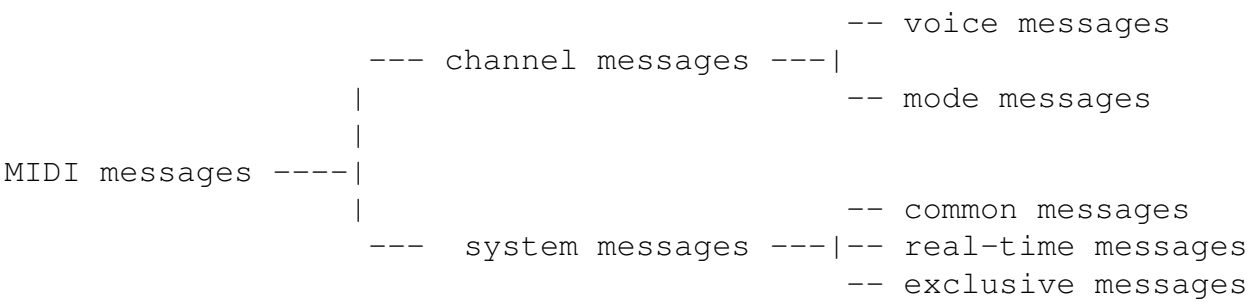
- Note On Command
  - Which Key is pressed
  - Which MIDI Channel (what sound to play)
  - 3 Hexadecimal Numbers
- Note Off Command Similar
- Other command (program change) configure sounds to be played.




Structure of MIDI messages:

- MIDI message includes a status byte and up to two data bytes.
- Status byte
  - The most significant bit of status byte is set to 1.
  - The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
  - The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

Classification of MIDI messages:



Midi Channel messages:

– messages that are transmitted on individual channels rather that globally to all devices in the MIDI network.

Channel voice messages:

- Instruct the receiving instrument to assign particular sounds to its voice
- Turn notes on and off
- Alter the sound of the currently active note or notes




Back
Close

Midi Channel Control Messages

Voice Message	Status Byte	Data Byte1	Data Byte2
Note off	8x	Key number	Note Off velocity
Note on	9x	Key number	Note on velocity
Polyphonic Key Pressure	Ax	Key number	Amount of pressure
Control Change	Bx	Controller number	Controller value
Program Change	Cx	Program number	None
Channel Pressure	Dx	Pressure value	None
Pitch Bend	Ex	MSB	LSB

Notes: ‘x’ in status byte hex value stands for a channel number.

Midi Command Example

A Note On message is followed by two bytes, one to identify the note, and on to specify the velocity.

To play:

- Note number 80 (HEX `50`)
- With maximum velocity (127 (Hex `7F`)
- On channel 13 (Hex `C`),

The MIDI device would send these three hexadecimal byte values:

```
9C   50   7F
```





Midi Channel mode messages:

- Channel mode messages are a special case of the Control Change message (`Bx` (Hex) or `1011nnnn` (Binary)).
- The difference between a Control message and a Channel Mode message, is in the first data byte.
  - Data byte values 121 through 127 have been reserved in the Control Change message for the channel mode messages.
  - Channel mode messages determine how an instrument will process MIDI voice messages.




System Messages:

- System messages carry information that are not channel specific, Examples:
  - Timing signal for synchronization,
  - Positioning information in pre-recorded MIDI sequences, and
  - Detailed setup information for the destination device
  - Setting up sounds, Patch Names etc.




Midi System Real-time Messages

- These messages are related to synchronization/timing etc.

System Real-Time Message	Status Byte
Timing Clock	F8
Start Sequence	FA
Continue Sequence	FB
Stop Sequence	FC
Active Sensing	FE
System Reset	FF

System common messages

- These contain the following (unrelated) messages

System Common Message	Status Byte	Number of Data Bytes
MIDI Timing Code	F1	1
Song Position Pointer	F2	2
Song Select	F3	1
Tune Request	F6	None




Back
Close

Midi System exclusive messages

- Messages related to things that cannot be standardized:
  - System dependent creation of sound
  - System dependent organisation of sounds (**Not General Midi Compliant? (more soon)**)
- An addition to the original MIDI specification.
- Just a stream of bytes
  - all with their high bits set to 0,
  - bracketed by a pair of system exclusive start and end messages:

    `F0` — **Sysex Start**

    `F7` — **Sysex End**
  - Format of message byte stream system dependent.




General MIDI (GM)

Problem: Midi Music may not sound the same everywhere?
Basic GM Idea:

- MIDI + Instrument Patch Map + Percussion Key Map –> a piece of MIDI music sounds (more or less) the same anywhere it is played
  - Instrument patch map is a standardised list consisting of 128 instruments (patches).
    Same instrument type sounds similar if not identical sound
  - Percussion map specifies 47 percussion sounds.
    Same Drum type sounds on keyboard map
  - Key-based percussion is always transmitted on MIDI channel 10 (Default)
    Can be transmitted on other channels as well




Requirements for General MIDI Compatibility

- Support all 16 channels — Default standard Multitimbral MIDI Specification
- Each channel can play a different instrument/program — multitimbral
- Each channel can play many notes — polyphony
- Minimum of 24 (usually much higher 64/128) fully dynamically allocated voices — shared across all channels




General MIDI Instrument Patch Map

Prog No.	Instrument	Prog No.	Instrument
(1-8	PIANO)	(9-16	CHROM PERCUSSION)
1	Acoustic Grand	9	Celesta
2	Bright Acoustic	10	Glockenspiel
3	Electric Grand	11	Music Box
4	Honky-Tonk	12	Vibraphone
5	Electric Piano 1	13	Marimba
6	Electric Piano 2	14	Xylophone
7	Harpsichord	15	Tubular Bells
8	Clav	16	Dulcimer
(17-24	ORGAN)	(25-32	GUITAR)
17	Drawbar Organ	25	Acoustic Guitar(nylon)
18	Percussive Organ	26	Acoustic Guitar(steel)
19	Rock Organ	27	Electric Guitar(jazz)
20	Church Organ	28	Electric Guitar(clean)
21	Reed Organ	29	Electric Guitar(muted)
22	Accordion	30	Overdriven Guitar
23	Harmonica	31	Distortion Guitar
24	Tango Accordian	32	Guitar Harmonics
(33-40	BASS)	(41-48	STRINGS)
33	Acoustic Bass	41	Violin
34	Electric Bass(finger)	42	Viola
35	Electric Bass(pick)	43	Cello
36	Fretless Bass	44	Contrabass
37	Slap Bass 1	45	Tremolo Strings
38	Slap Bass 2	46	Pizzicato Strings
39	Synth Bass 1	47	Orchestral Strings
40	Synth Bass 2	48	Timpani



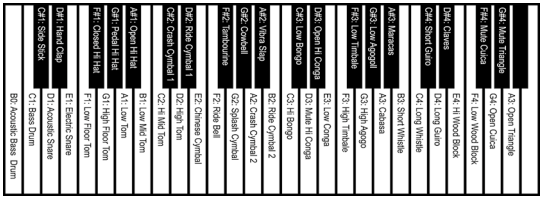

	(49-56 ENSEMBLE)		(57-64 BRASS)
49	String Ensemble 1	57	Trumpet
50	String Ensemble 2	58	Trombone
51	SynthStrings 1	59	Tuba
52	SynthStrings 2	60	Muted Trumpet
53	Choir Aahs	61	French Horn
54	Voice Oohs	62	Brass Section
55	Synth Voice	63	SynthBrass 1
56	Orchestra Hit	64	SynthBrass 2
	(65-72 REED)		(73-80 PIPE)
65	Soprano Sax	73	Piccolo
66	Alto Sax	74	Flute
67	Tenor Sax	75	Recorder
68	Baritone Sax	76	Pan Flute
69	Oboe	77	Blown Bottle
70	English Horn	78	Skakuhachi
71	Bassoon	79	Whistle
72	Clarinet	80	Ocarina
	(81-88 SYNTH LEAD)		(89-96 SYNTH PAD)
81	Lead 1 (square)	89	Pad 1 (new age)
82	Lead 2 (sawtooth)	90	Pad 2 (warm)
83	Lead 3 (calliope)	91	Pad 3 (polysynth)
84	Lead 4 (chiff)	92	Pad 4 (choir)
85	Lead 5 (charang)	93	Pad 5 (bowed)
86	Lead 6 (voice)	94	Pad 6 (metallic)
87	Lead 7 (fifths)	95	Pad 7 (halo)
88	Lead 8 (bass+lead)	96	Pad 8 (sweep)

(97-104 SYNTH EFFECTS)		(105-112 ETHNIC)	
97	FX 1 (rain)	105	Sitar
98	FX 2 (soundtrack)	106	Banjo
99	FX 3 (crystal)	107	Shamisen
100	FX 4 (atmosphere)	108	Koto
101	FX 5 (brightness)	109	Kalimba
102	FX 6 (goblins)	110	Bagpipe
103	FX 7 (echoes)	111	Fiddle
104	FX 8 (sci-fi)	112	Shanai
(113-120 PERCUSSIVE)		(121-128 SOUND EFFECTS)	
113	Tinkle Bell	121	Guitar Fret Noise
114	Agogo	122	Breath Noise
115	Steel Drums	123	Seashore
116	Woodblock	124	Bird Tweet
117	Taiko Drum	125	Telephone Ring
118	Melodic Tom	126	Helicopter
119	Synth Drum	127	Applause
120	Reverse Cymbal	128	Gunshot

Back
Close

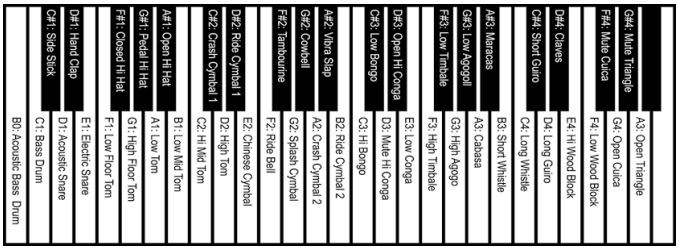
General MIDI Percussion Key Map

MIDI Key	Drum Sound	MIDI Key	Drum Sound
35	Acoustic Bass Drum	59	Ride Cymbal 2
36	Bass Drum 1	60	Hi Bongo
37	Side Stick	61	Low Bongo
38	Acoustic Snare	62	Mute Hi Conga
39	Hand Clap	63	Open Hi Conga
40	Electric Snare	64	Low Conga
41	Low Floor Tom	65	High Timbale
42	Closed Hi-Hat	66	Low Timbale
43	High Floor Tom	67	High Agogo
44	Pedal Hi-Hat	68	Low Agogo
45	Low Tom	69	Cabasa
46	Open Hi-Hat	70	Maracas
47	Low-Mid Tom	71	Short Whistle
48	Hi-Mid Tom	72	Long Whistle
49	Crash Cymbal 1	73	Short Guiro
50	High Tom	74	Long Guiro
51	Ride Cymbal 1	75	Claves
52	Chinese Cymbal	76	Hi Wood Block
53	Ride Bell	77	Low Wood Block
54	Tambourine	78	Mute Cuica
55	Splash Cymbal	79	Open Cuica
56	Cowbell	80	Mute Triangle
57	Crash Cymbal 2	81	Open Triangle
58	Vibraslap		



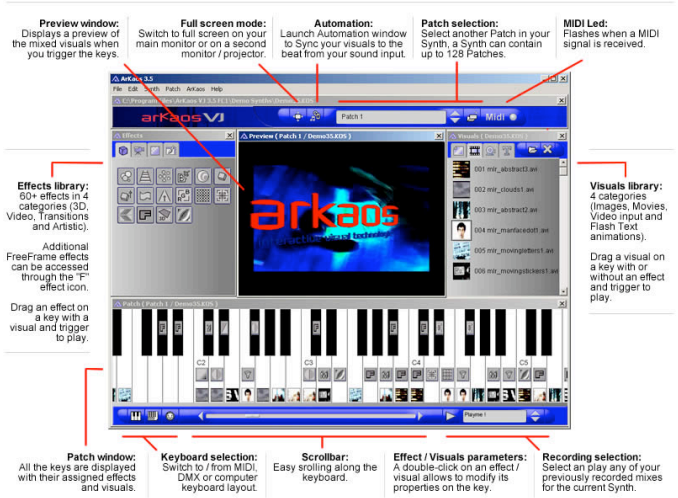
MIDI Percussion Key Mapping

- Each key is essentially a switch
- No **Pitch** information relevant — usually
- Can be extended to control other stuff e.g. Video DJ (VJ) application



	Pitch	Instrument
	C1	Bass Drum
	C#1	Rim
	D1	Snare 1
	E1	Snare 2
	F#1	Hi-hat Closed
	G#1	Hi-hat Pedal
	A#1	Hi-Hat Open
	A1	Tom Low
	C2	Tom Mid
	D2	Tom High
	D#2	Ride
	C#2	Crash
	C-2	Sound 1

ArKaos VJ MIDI



Limitations of Conventional MIDI

- Limited Number of Channels and Controllers
- Limited resolution in data values
  - Most midi numbers are 8-bit

Solutions:

- Some MIDI manufacturer utilities two midi data values to allow for large range of values

  E.g. Use values as Least and Most Significant Bytes: **16 bit range**

- Open Sound Control (OSC) — been around a while, MIDI still rules?
- High Definition MIDI — fixes the above and adds more features.




Digital Audio, Synthesis, Midi and Compression — MPEG 4 Structured Audio

- We have seen the need for compression already in Digital Audio — Large Data Files
- Basic Ideas of compression (next lecture) used as integral part of audio format — MP3, real audio *etc.*
- Mpeg-4 audio — actually combines compression synthesis and midi to have a massive impact on compression.
- Midi, Synthesis encode what note to play and how to play it with a small number of parameters

  — Much greater reduction than simply having some encoded bits of audio.
- Responsibility to create audio delegated to generation side.




MPEG 4 Structured Audio

A newer standard than MP3 Audio — which we study in detail later

MPEG-4 covers the the whole range of digital audio:

- From very low bit rate speech
- To full bandwidth high quality audio
- Built in anti-piracy measures
- **Structured Audio**
- Relation to MIDI so we study MPEG 4 audio here




Structured Audio Tools

MPEG-4 comprises of 6 *Structured Audio tools* are:

SAOL – the Structured Audio Orchestra Language

SASL – the Structured Audio Score Language

SASBF – the Structured Audio Sample Bank Format

Set of MIDI semantics — describe how to control SAOL with MIDI

Scheduler – describe how to take the above parts and create sound

AudioBIFS – part of BIFS, which lets you make audio soundtracks in MPEG-4 using a variety of tools and effects-processing techniques




SAOL
(Structured Audio Orchestra Language)

- Pronounced "sail"
- The central part of the Structured Audio toolset.
- A new software-synthesis language
- A language for describing synthesisers, a program, or instrument
- Specifically designed it for use in MPEG-4.
- Not based on any particular method of synthesis – supports many underlying synthesis methods.




SAOL Synthesis Methods

- Any known method of synthesis can be described in SAOL (Open Support).
  - FM synthesis,
  - physical-modeling synthesis,
  - Sample-based synthesis,
  - granular synthesis,
  - subtractive synthesis,
  - FOF synthesis, and
  - hybrids of all of these in SAOL.

SASL (Structured Audio Score Language)

- A very simple language to control the synthesisers specified by SAOL instruments.
- A SASL program, or score, contains instructions that tell SAOL:
  - what notes to play,
  - how loud to play them,
  - what tempo to play them at,
  - how long they last, and how to control them
- Similar to MIDI
  - doesn't suffer from MIDI's restrictions on temporal resolution or bandwidth.
  - more sophisticated controller structure

SASL (Structured Audio Score Language) (Cont.)

- Lightweight Scoring Language: Does not support:
  - looping,
  - sections,
  - repeats,
  - expression evaluation,
  - some other things.
  - most SASL scores will be created by automatic tools




SASBF
(Structured Audio Sample Bank Format)

- A format for efficiently transmitting banks of sound samples
- Used in wavetable, or sample-based synthesis.
- Partly compatible with the MIDI Downloaded Sounds (DLS) format
- The most active participants in this activity are EMu Systems (sampler manufacturer) and the MIDI Manufacturers Association (MMA).




Back
Close

MPEG-4 MIDI Semantics

SASL can be controlled by

- SASL Scripts
- MIDI
- Scores in MPEG-4

Reasons to use MIDI:

- MIDI is today's most commonly used representation for music score data,
- Many sophisticated authoring tools (such as sequencers) work with MIDI.




MPEG-4 Midi Control

- MIDI syntax external to MPEG-4 Structured Audio standard
- Use MIDI Manufacturers Association's standard.
- *Redefines* the some semantics for MPEG-4.
- The new semantics are carefully defined as part of the MPEG-4 specification.




MPEG-4 Scheduler

- The main body of the Structured Audio definition.
- A set of carefully defined and somewhat complicated instructions
- Specify how SAOL is used to create sound when it is driven by MIDI or SASL.




AudioBIFS

- BIFS is the MPEG-4 *Binary Format for Scene Description*.
- Describes how the different ”objects” in a structured media scene fit together:
  - MPEG-4 consists also of the video clips, sounds, animations, and other pieces of multimedia
  - Each have special formats to describe them.
  - Need to put the pieces together
  - BIFS lets you describe how to put the pieces together.




Back
Close

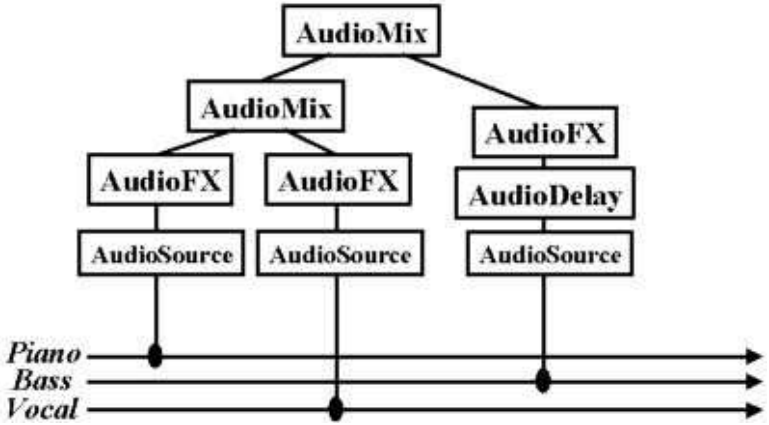
AudioBIFS (Cont.)

- AudioBIFS is designed for specifying the mixing and post-production of audio scenes as they're played back.
- For example,
  - we can specify how the voice-track is mixed with the background music, and
  - that it fades out after 10 seconds and
  - this other music comes in and has a nice reverb on it.
- Extended version of VRML: capabilities for
  - streaming and
  - mixing audio and video data
- Very advanced sound model.




AudioBIFS (Cont.)

How a simple sound is created from three elementary sound streams:



HTML 5 MIDI

A new Web MIDI API (2013)[1]:

- Part of general web audio development of HTML 5

The Web MIDI API specification

- Defines a means for web developers to manipulate and access MIDI devices
  - Midi Input and Output to hardware (outboard) and software.
  - Audio Synthesis available in Browser.
  - Total Web-Mid Control.
  - JavaScript Programming.

[1]Support of Web MIDI API is not that well developed. Not all browsers support it. See here for an example of how to install



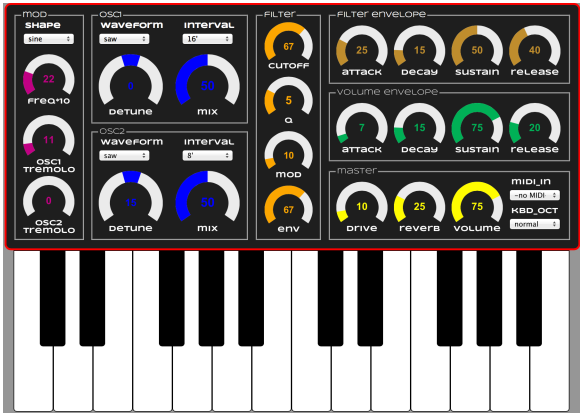

Some HTML 5 MIDI Examples: Moog Doodle

The first app was the Google Doodle for the Mini Moog.

- Uses Web Audio/MIDI API to recreate a Moog Synthesiser.
- Subtractive Sythesis on Web — code here.
- Celebrated Bob Moog's 78th Birthday.
- Spawned a whole community.

Some HTML 5 MIDI Examples: MIDI Controlled Subtractive Synthesis

A fully fledged MIDI controlled Subtractive Synthesiser



- Configurable MIDI input — needs JAVA[2]
- Polyphonic Synthesiser

[2] see for details how run this




Back
Close

Some Other HTML 5 MIDI Examples

Drum Machine :

Web Audio Drumming

Wavetable Synthesis :

Controllable Wavetables

Granular Synthesis :

Simple Granular Synth

More Examples :

webaudiodemos.appspot.com

Some More Examples :

jazz-soft.net/demo