

Music Technology Lecture 4 Sequencers

Lecturer: Matthew Yee-King

Contact: m.j.yee-king@sussex.ac.uk

URL:
<http://www.informatics.sussex.ac.uk/courses/mt>

1

Last week's lecture

- Synths in the 80s
 - Wavetable synths
 - Cheap analogue synths
 - FM synths
- Synthesis techniques
 - FM synthesis
 - Additive synthesis

2

Lecture 4 Summary

- What is a sequencer?
- History of sequencers
 - Mechanical
 - Analogue
 - Digital
- The MIDI protocol
- The OSC protocol

3

What is a sequencer?

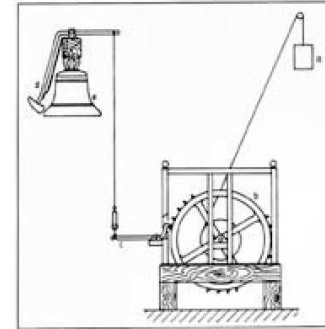
- A music recording and playback system with a programmable memory
- It stores information about notes as opposed to the audio data itself – **control data**
 - What control data might be required to play a note?
- It allows the automated playing of synthesizers and other musical instruments
- It can be a dedicated hardware device or some software on a general purpose computer

4

Early mechanical sequencers

- The first sequencers were automatic bell ringers used to play 'Carillons'
- A Carillon consists of 2+ octaves of tuned, copper based bells with well defined partials
- Automated Carillon players were seen as early as 1381 in Belgium
- Some players had up to 9000 bit memories
 - How many beats is that, with 2 octaves of bells?
- Programmed using wooden pegs placed in a rotating barrel
- The forerunner to 'step mode' sequencing

A mechanical Carillon sequencer



6

The Pianola

- An automated piano player
- First seen in the late 19th century (Edwin Votey)
- Note sequence defined on a rolled up punch card – the 'piano roll'
- First models placed in front of a normal piano
- Later models were built into a piano – the 'player piano'
- Piano rolls produced manually by technicians
- Pneumatic paper punching system appeared in the early 20th century, which allowed a performance on a keyboard to be recorded to paper tape

7

Edwin Votey's Pianola - 1898



8

The Coupleaux -Givelet Organ - 1930

- First synthesizer with a self playing capability
- Used punched cards to control vacuum tube oscillators



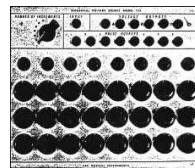
9

Analogue sequencers

- Voltage Control – voltages are used to control different aspects of the sound
- Modular synths had analogue sequencer modules
- The modules communicate using varying voltages
- The Buchla 100 Series, 1963 was one of the first voltage controlled analogue sequencers
- From the PD demo, can you think of limitations of such systems?

10

Buchla 100 Series - 1963



Model 123 Sequencer module

11

Limitations of analogue sequencers

- No way of recording a sequence into the device
- Limited number of steps
 - e.g. Model 123 had 24 steps for all parameters
 - Playback is highly quantized
- Sequence cannot be rapidly changed
- Polyphony something of a challenge

12

Digital sequencers

- Back to Max Mathews!
- GROOVE (1970) was an analogue-digital hybrid system
 - Functions of time stored digitally and played back
 - Performer manipulates the functions in real time using analogue controls

13

EMS digital sequencers

- Synthi sequencer 256 (1971)
 - Could sequence EMS synths or Moogs
 - Five Octave Velocity Sensitive Keyboard
 - 256 Event x 42 bit Memory
 - Editing



14

Digital sequencers in the 70s

- Roland MC-8 1977
 - Intel 8080 CPU
 - 4k RAM
 - 8 tracks, up to 1,100 notes
 - Programmed via the calculator keyboard
 - Cost \$4500!

15

The birth of MIDI

- Musical Instrument Digital Interface
- Developed by Roland as a format for their new IBM PC – synthesizer interface, the MPU401
- Roland got together with Sequential Circuits as NAMM in 1983 and called it MIDI
- Developed into a universal communication protocol for digital music equipment

16

The birth of MIDI

- The first MIDI synth was the Sequential Circuits Prophet 600 in 1983
- Other manufacturers joined in, e.g. Yamaha with the DX7 at the end of 1983
- Voltage controlled systems were largely made obsolete overnight

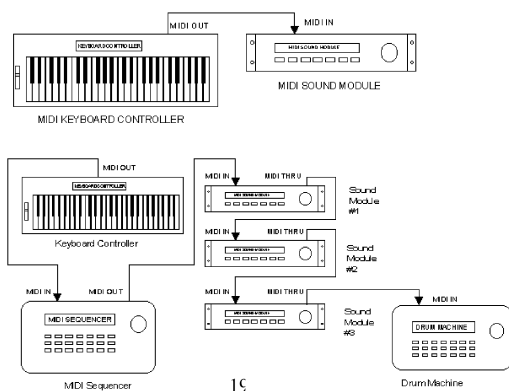
17

The MIDI protocol

- Digital protocol
- Specially designed for music systems
- Sent in 8 bit words
- MIDI messages are made from one or more words - status byte+(data byte(s))
- Uses 5 pin DIN cables
- Unidirectional, asynchronous bit stream
- Bit rate is 31.25 kbit/s
- MIDI connectors are in out and thru

18

MIDI connection examples



19

Overview of MIDI messages

- Channel messages
 - voice messages
 - note on, note off, voice control data
 - mode messages
 - define how to respond to the data
- System messages
 - common messages
 - time code
 - real time messages
 - more synchronization
 - system exclusive messages
 - hardware specific

20

Channel messages

- Status byte:
 - 4 bits for message type
 - 4 bits for channel number (0-15)
- Data bytes 1 and 2 define 2 parameters between 0 and 127
- E.g. to play a middle C on channel 2 with maximum velocity:
 - S = 9,1
 - D1 = 3C / 60
 - D2 = 7F / 127

21

Other channel voice 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

x = channel number

22

Channel mode messages

- Tell the synth how to respond to incoming notes
 - e.g. polyphonic or mono mode, omni mode
- Turn local control on or off
- Reset the controllers

1st Data Byte	Description	Meaning of 2nd Data Byte
79	Reset all controllers	None; set to 0
7A	Local control	0 = off; 127 = on
7B	All notes off	None; set to 0
7C	Omni mode off	None; set to 0
7D	Omni mode on	None; set to 0
7E	Mono mode on (Poly mode off)	**
7F	Poly mode on (Mono mode off)	None; set to 0

23

System messages

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 real time 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

system common messages

24

System exclusive messages

- Hardware specific
- Can contain a long stream of data bytes
- Example applications:
 - To access synthesis features that are outside of the standard MIDI specification
 - To build graphical personal computer based editors for synths

25

OSC - OpenSoundControl

- Client/ Server scheme
 - control device (e.g. keyboard)
 - synth as server
- Network based
 - protocol independent (TCP/IP, UDP etc)
- Transmitted in OSC packets
- Open protocol - think XML

26

OSC messages

- A message consists of:
 - an address pattern - specifies the part of the server to which the message is addressed, like a URL
 - a type tag string - defines the data types to be found in the arguments
 - arguments - the real (i.e. not meta) data

27

OSC data types

- ASCII strings
- 32 bit floats and ints
- blobs - chunks of binary data
- More data types will become available as new OSC implementations arrive

28

OSC messages

- Example note playing message
- address pattern: /voice/3/freq
- type tag string: float
- argument: 452.24

25

More OSC

- Address patterns support regular expression pattern matching
 - e.g. instead of /voice/3/freq, /voice/*/freq
- Querying
 - a client can query a server as to its address space and capabilities
- Bundles
 - bundling of messages allows their synchronous execution

30

Lecture 4 Summary

- What is a sequencer?
- History of sequencers
 - Mechanical
 - Analogue
 - Digital
- The MIDI protocol
- The OSC protocol

31

References

- Good intro to MIDI:
 - <http://www.harmony-central.com/MIDI/Doc/tutorial.html>
- Latest information about OpenSoundControl
 - <http://www.cnmat.berkeley.edu/OpenSoundControl/>
 - especially this:
 - http://www.cnmat.berkeley.edu/Research/NIME2003/NIME03_Wright.pdf

32