

CTP 431 Music and Audio Computing

Music Interface

Graduate School of Culture Technology (GSCT)

Juhan Nam

Introduction

- Musical Instrument

- Sound generation: we have focused only on the resulting sounds
- Interface: another important aspect
 - Basic actions: key striking, plucking, blowing, bowing
 - Expressions: continuous/discrete, dynamics, subtlety
 - Monophonic / polyphony
 - Human side: learning curve, virtuosity, challenge/boredom

- These two aspects of musical instruments are decoupled in electrical form of instruments
 - Here we focus on music interface: capture musician's gestures and convert them to digital data

Keyboard Controllers



M-Audio KeyStudio49



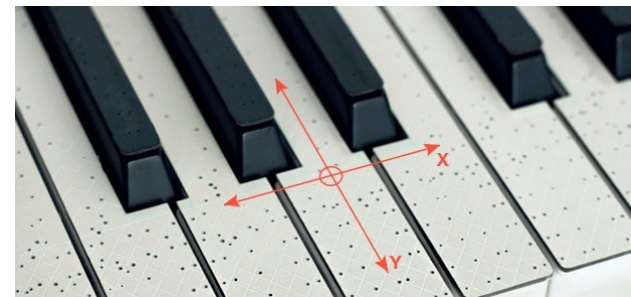
AXiS-64

<https://www.youtube.com/watch?v=pQ4nPcGCGIs>



Haken Continuum

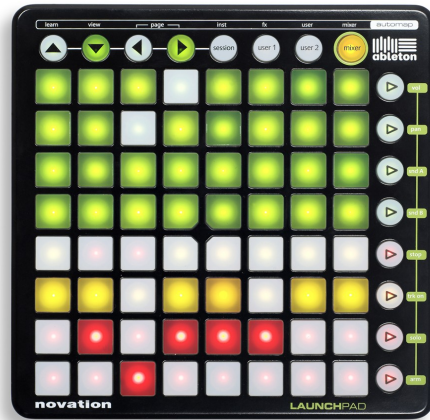
<https://www.youtube.com/watch?v=PnBhR8RLJN8>



Touchkeys

<https://www.kickstarter.com/projects/instrumentslab/touchkeys-multi-touch-musical-keyboard>

Sequencer-Type / Pad Controllers



Novation Launch Pad



AKAI MPC

https://www.youtube.com/watch?v=4g_OTCbO2eU



Tenori-On

<https://www.youtube.com/watch?v=hzcptMO0CrI>



Monome

<http://monome.org/>

<https://www.youtube.com/watch?v=2uZlqRU0srU&index=12>

Remote Controllers



Radio Baton (by Max Mathews)

<https://www.youtube.com/watch?v=3ZOzUVD4oLg>



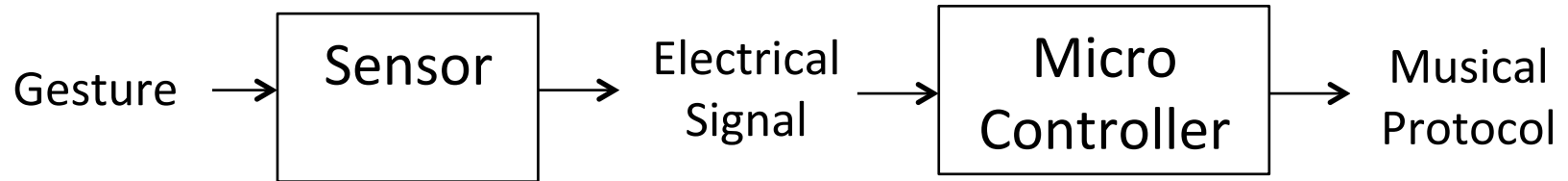
Virtual Slide Guitar

<https://www.youtube.com/watch?v=aIJ-8kd8rFs>

Other Controllers

- Wind controller
- Guitar controller
- Drum controller
- EEG-based controllers
- Wii-mote controller
- Touchscreen controllers
- And so many “creative” controllers...
 - Reactable: <https://www.youtube.com/watch?v=Mgy1S8qymx0>
 - NIME (New Interfaces for Music Expressions): <http://www.nime.org/>

Digital Interface



■ Sensors

- Button (or switch), potentiometer (knob), resistive (rubber, photo-resister), optical, accelerometer, gyro, microphone, camera...
- Physical actions: push, rotation, velocity, pressure, location, ...
- <http://www.sensorwiki.org/doku.php>

■ Micro controllers

- AD converter (continuous input), input port (discrete input)
- Map the captured input to musical protocols: MIDI, OSC

MIDI

- Musical Instrument Digital Interface
 - Standard music protocol
 - <http://www.midi.org/>
- Why MIDI?
 - Need of musical communication among different vendors' instruments
 - Store music data (notation or performance) for composers
- Hardware
 - 5-pin cables, separate in/out in connection
 - 31250 bits per second
- Software
 - MIDI messages: note number/velocity, control data

MIDI Message

- Format

	Status Byte	Data Byte1	Data Byte2
Note Off	1000 xxxx	Note Number	Velocity
Note On	1001 xxxx	Note Number	Velocity
Note Pressure	1010 xxxx	Note Number	Velocity
Control Change	1011 xxxx	Ctrl. Number	Ctrl Value
Program Change	1100 xxxx	Prog. Number	-

xxxx: channel number (0-15)

Data byte: 0-127 (MSB is 0)

MIDI Sequencer

- Tempo and Time stamps are added
 - Relative timestamp given tempo
 - To avoid recalculating timestamps for different tempos
 - Tempo: beats per minutes (BPM)
 - Beat: quarter note
 - Tick: minimum time resolution: ticks per beat
 - Time-stamped MIDI messages are stored as a MIDI file
 - They are edited and played back by MIDI sequencers
- MIDI Representations
 - MIDI Events
 - Piano Rolls
 - Music Notation

Web MIDI

- Support MIDI on web browsers
 - W3C Editor's draft: <http://webaudio.github.io/web-midi-api/>

OSC

- Open Sound Control

- An alternative to the 1983 MIDI standard
- Can define your own “status byte”
- Support various data types: integer, floating-point, strings, blob
- Designed for streaming control data over networks
 - Among computers, software programs and other multimedia devices: e.g.) Arduino + PureData + Processing
- <http://opensoundcontrol.org/introduction-osc>

- Format

- Address Pattern (with '/') + Tag type string (begin with ',') + Arguments (variable-length data)
 - Example: /sensor/accelerometer/xy ,ii 123 456