MUSC 4820/5820 Digital Music Techniques 001

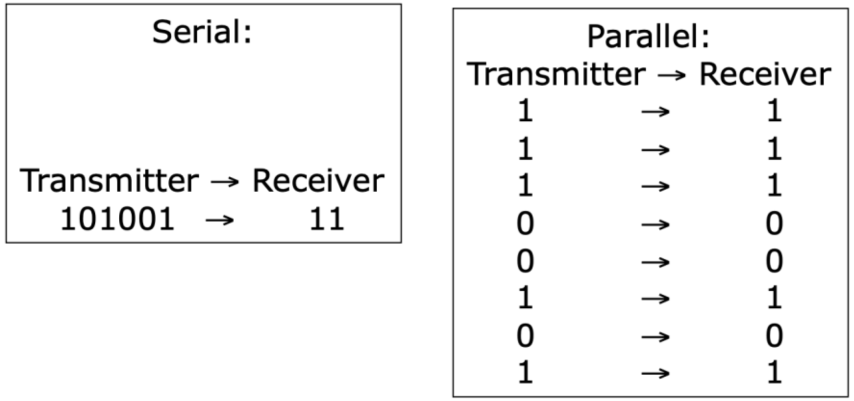
Week 13: Controllers, Concepts of Input, Control, and Mapping for Digital Music Instrument

College of Arts & Media
UNIVERSITY OF COLORADO DENVER

Dr. Jiayue Cecilia Wu
Assistant Professor
Department of Music & Entertainment Industry Studies
University of Colorado, Denver

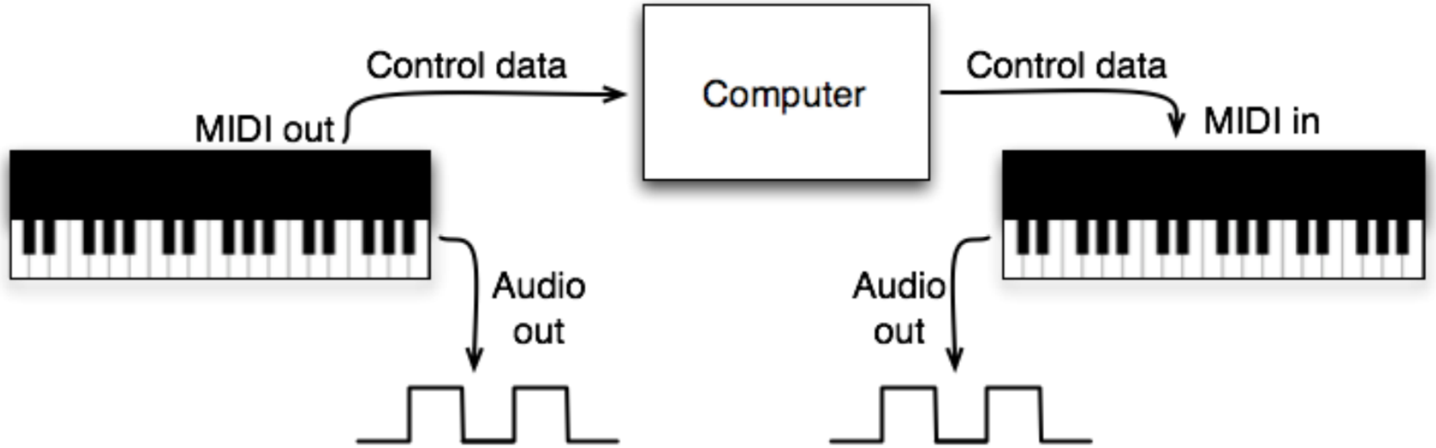
Musical Instrument Digital Interface (MIDI)

- MIDI is a communication protocol for remote/synchronized control mechanism and for storing information
- It’s a standard established in 1983
- It’s extensible, many uses beside MIDI notes
- Max 1.5 kHz bandwidth
- Uni-directional serial interface and asynchronous communication
- These often used one port for timing and another for note triggering a
- Max cable length 15 m, USB and wireless MIDI connections also work
- Long cables and slow wifi connection cause unwanted signal distortion
- MIDI/computer Interface: External MIDI/USB devices such as are widely used for multi-port interfaces, which are able to handle a number of MIDI streams (each controlling up to 16 channels) and distribute separately. They also allow the synchronized handling of several devices (video recorder, automated mixer, effects, samplers, etc)
- Message scheme based on keyboard’s model



Musical Instrument Digital Interface (MIDI)

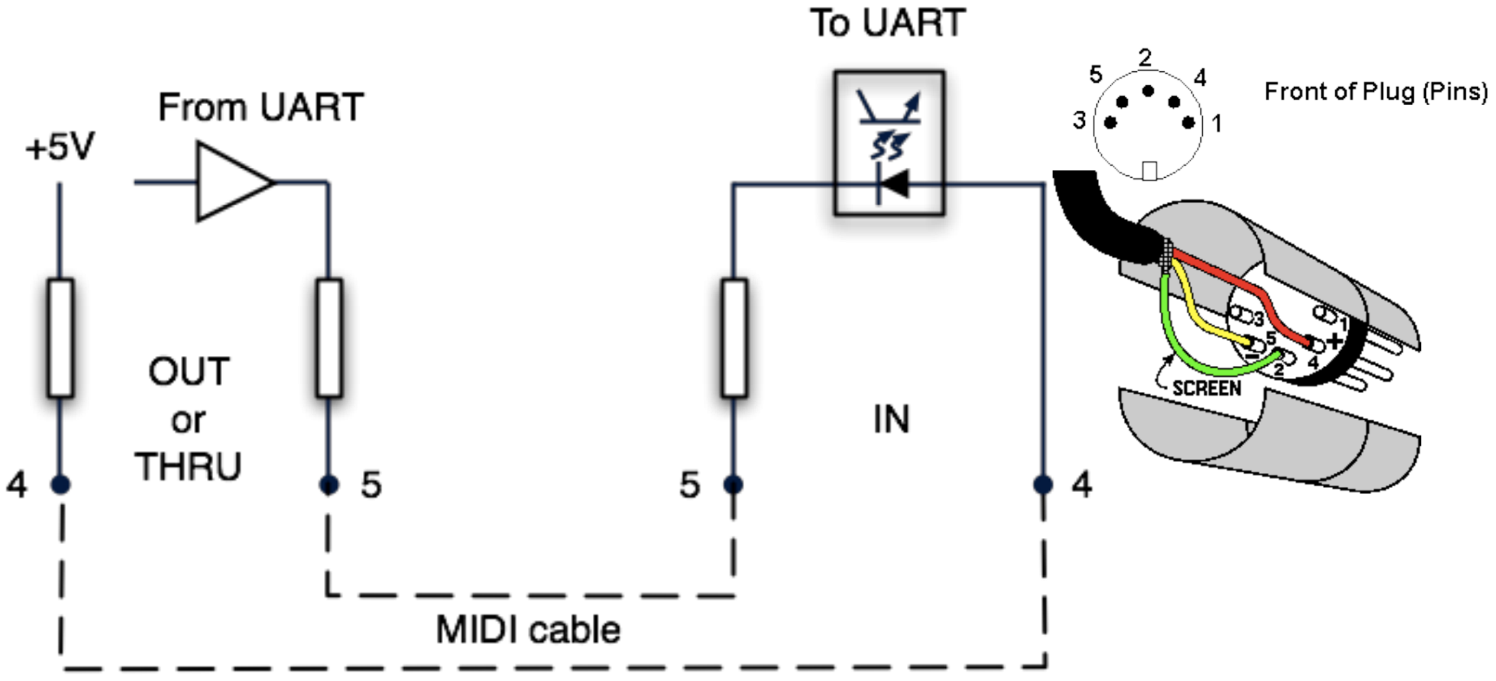
- In MIDI, processing and storage also occurs in the digital domain, but the information being processed is not the audio signal but the control data used to generate it.
- An electronic instrument is needed to reproduce the sound, which means that unless we use the exact same synthesis engine, MIDI-generated sounds are never the same.



- Because it comprises control data only, MIDI uses significantly less memory space than digital audio

Musical Instrument Digital Interface (MIDI)

- There are 3 kinds of MIDI ports: IN, THRU, and OUT. The IN port accepts input to a device, the THRU port passes an amplified copy of the input signal along, and the OUT port is used to transmit the device's output.
- The hardware uses cables terminated in 180-degree 5-pin DIN connectors, of which only three pins are used (5, 4 and 2). (Pin 2 is connected to earth in OUT and THRU only)

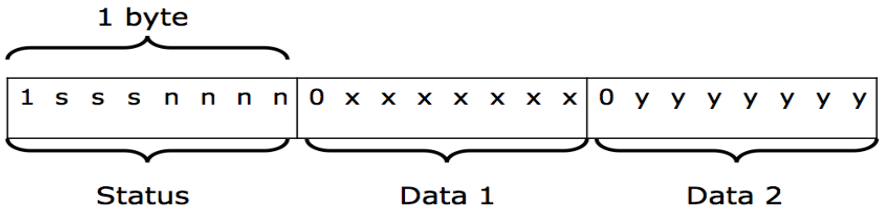


* **UART** stands for "Universal Asynchronous Receiver/Transmitter". It is a piece of digital hardware that transports bytes between digital devices, commonly found as a peripheral on computer and microcontroller systems. It is the device that underlies a serial port, and it is also used by **MIDI**.

Musical Instrument Digital Interface (MIDI)
MIDI Transmission Standard

- A maximum transmission rate of 31250 bits per second (3125 bytes per second), which means no more than 651 notes can be transmitted /second.
- 16 channels limits due to bandwidth
- If first bit =1, the following byte is a “status” byte

  If first bit = 0, the following byte is a “data” byte



- The status byte determines the length of most messages, which are usually 1,2,or 3 bytes in length
- System exclusive messages are of variable length and have a start and ending status byte.
- MIDI uses hexadecimal system: Hexadecimal numbers are a base-16 representation of numbers and are useful for humans when dealing with binary numbers.

  Every 4 binary digits are represented by 1 hexadecimal digit

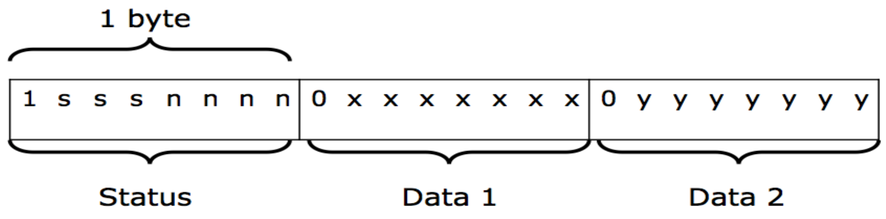
number base equivalences

dec	hex	bin	dec	hex	bin
0	0	0	8	8	1000
1	1	1	9	9	1001
2	2	10	10	A	1010
3	3	11	11	B	1011
4	4	100	12	C	1100
5	5	101	13	D	1101
6	6	110	14	E	1110
7	7	111	15	F	1111

Musical Instrument Digital Interface (MIDI)
MIDI Transmission Standard

- A maximum transmitted as asynchronous bytes at 31250 bits per second. One start bit, eight data bits, and one stop bit result in a maximum (3125 bytes per second).
- If the first bit =1, the following byte is a "status" byte

  If the first bit = 0, the following byte is a "data" byte
- 16 channels limits due to bandwidth



- The status byte determines the length of most messages, which are usually 1,2,or 3 bytes in length
- System-exclusive messages are of variable length and have a start and ending status byte.
- MIDI uses a hexadecimal system: Hexadecimal numbers are a base-16 representation of numbers and are useful for humans when dealing with binary numbers.

  Every 4 binary digits are represented by 1 hexadecimal digit

number base equivalences

dec	hex	bin	dec	hex	bin
0	0	0	8	8	1000
1	1	1	9	9	1001
2	2	10	10	A	1010
3	3	11	11	B	1011
4	4	100	12	C	1100
5	5	101	13	D	1101
6	6	110	14	E	1110
7	7	111	15	F	1111

MIDI Transmission Standard

T23 – MIDI-Code

Status Bytes	Data Bytes	Command type
1000nnnnb (=8mh)	pitch number [0–127], force of attack* [0–127]	Note Off
1001nnnnb (=9mh)	pitch number [0–127], force of attack* [0=off, else 1–127]	Note On
1010nnnnb (=Amh)	pitch number [0–127], force of attack* [0–127]	After-touch
1011nnnnb (=Bmh)	control nummer [0–121: e.g. 7=Volume,....], control value [0–127]	Control Change
1100nnnnb (=Cmh)	program number [0–127] (only 1 data byte!)	Program Change
1101nnnnb (=Dmh)	pressure value [0–127] (only 1 data byte!)	Channel Pressure
1110nnnnb (=Emh)	lower byte [0–127], upper byte [0–127]	Pitch Wheel
11110nnnb (=Fmh)	brand dependent	System Exclusive

* 'velocity' for technocrats

MIDI commands

0x80	Note Off
0x90	Note On
0xA0	Aftertouch
0xB0	Continuous controller
0xC0	Patch change
0xD0	Channel Pressure
0xE0	Pitch bend
0xF0	(non-musical commands)

MIDI 1.0
and
MIDI 2.0

Feature	MIDI (1.0)	MIDI 2.0
Resolution	7-bit (128 levels)	Up to 32-bit (massive precision)
Communication	One-way	Bidirectional
Per-Note Control	Global	Independent per note
Profiles and Configurations	No	Supported
Extended Messages	Fixed set	Customizable
Channel Count	16 channels	Still 16 (scalable for future)
Timing	Standard (lower precision)	Enhanced timing accuracy
Backward Compatibility	N/A	Fully backward-compatible

Open Sound Control (OSC)

Communication protocol

Flexible (wireless)

Big community not only in music but also in visual arts

Same network! IP

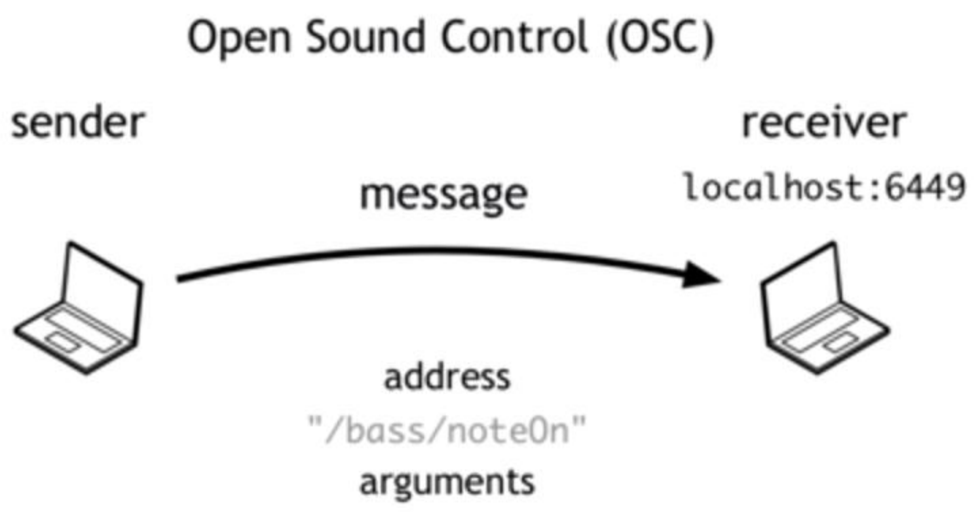
Different address

Different OSC objects

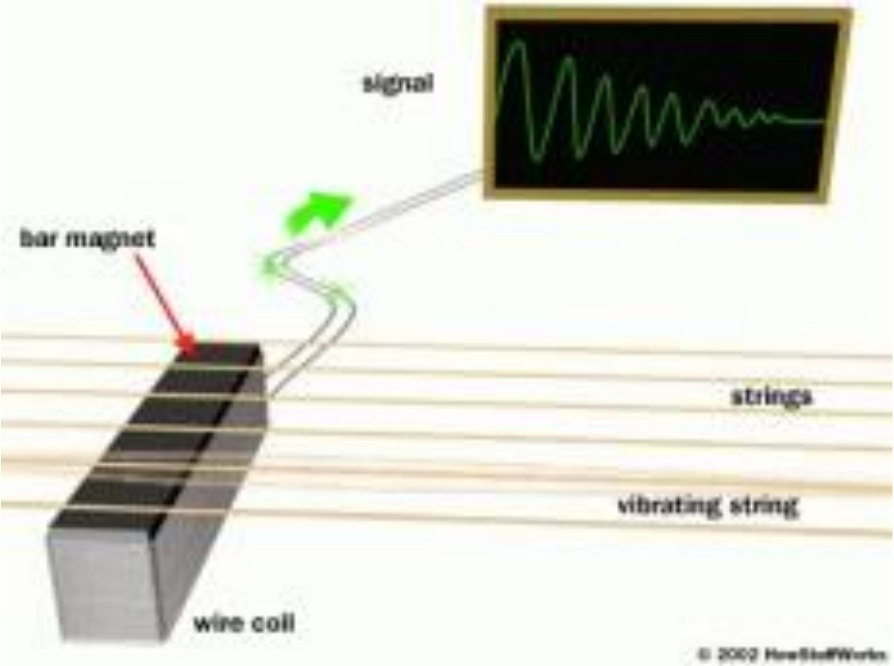
Different control

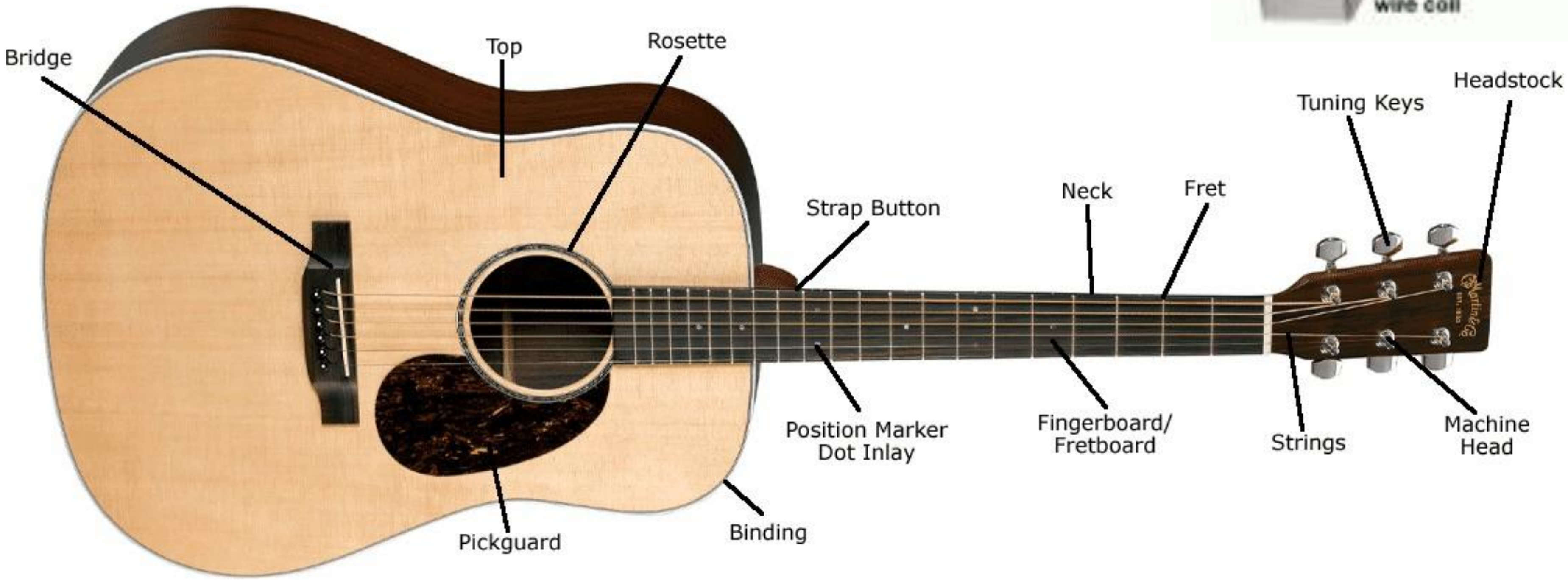
Synchronous events!! (UDP protocol via Internet faster than TCP)

High resolution (32 bits flouting points)

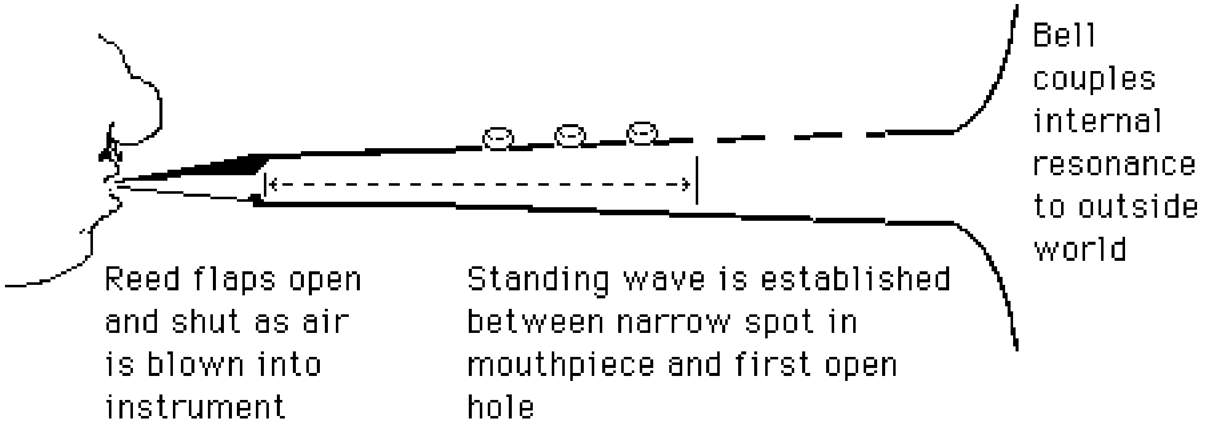


How does an acoustic instrument make sound
Guitar?
Flute?
Piano?





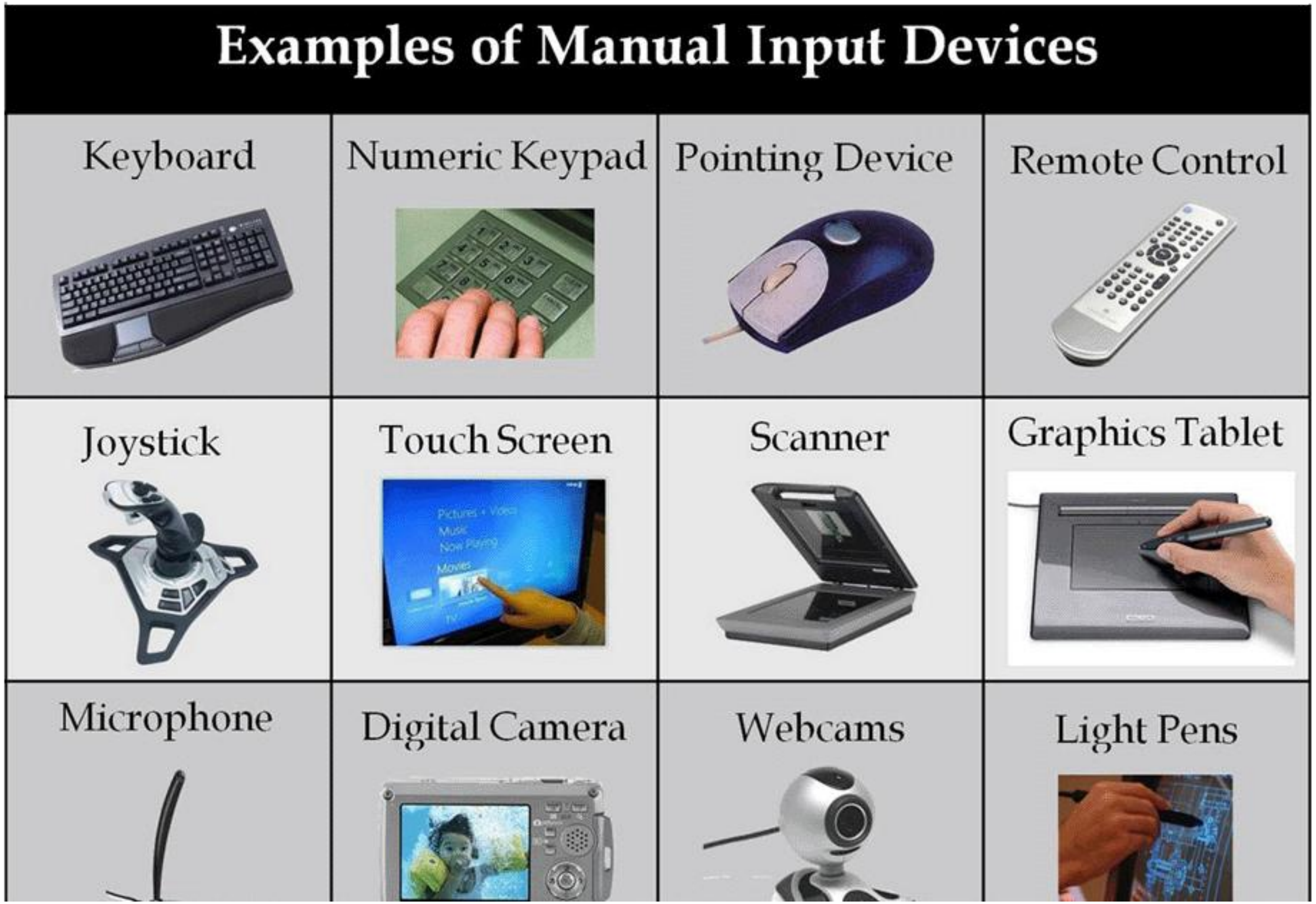
Guess: Which part is Input, Control, and Mapping?



Digital Music Instrument (DMI)

Input: they are data~ not device~

- Continuous (Slider input)
- Discrete (Key/button/switch input)



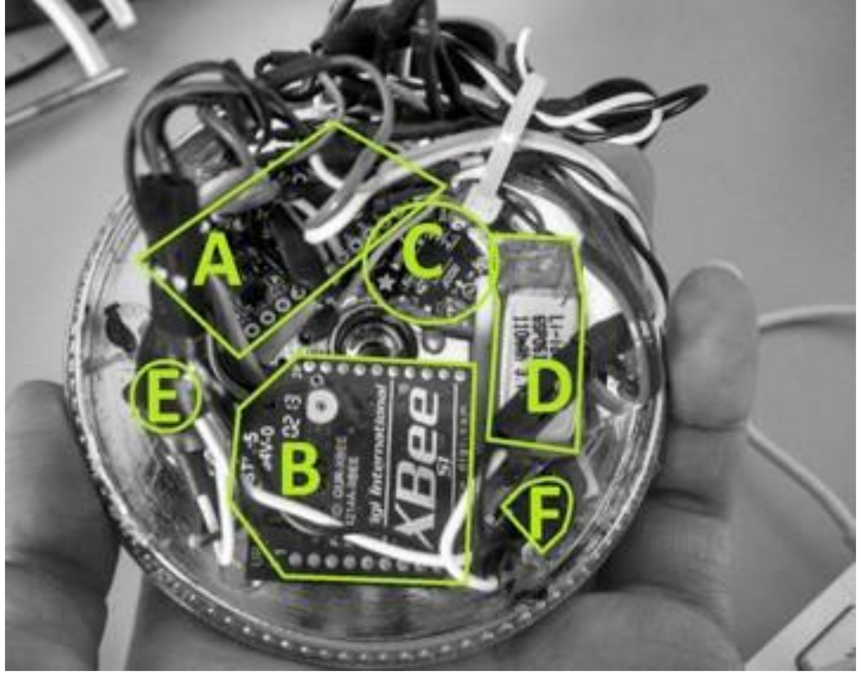
Input Devices

Novel Digital Music Instrument and Controller Design

Example 1

Example 2

Example3



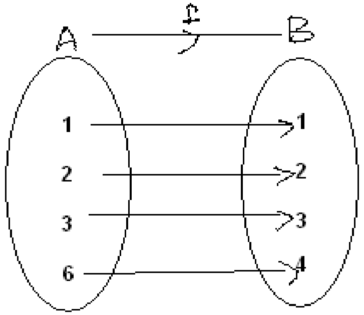
What is Control?

- It's a system, software architecture + hardware system configuration. It's a mechanism.
- It's how you manipulate the virtual "body" of your instrument.

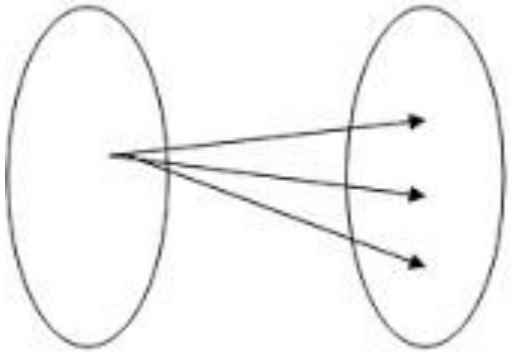
Input data -> Controller -> SoundMapping Strategy -> Creative Coding (MUSC 4360)

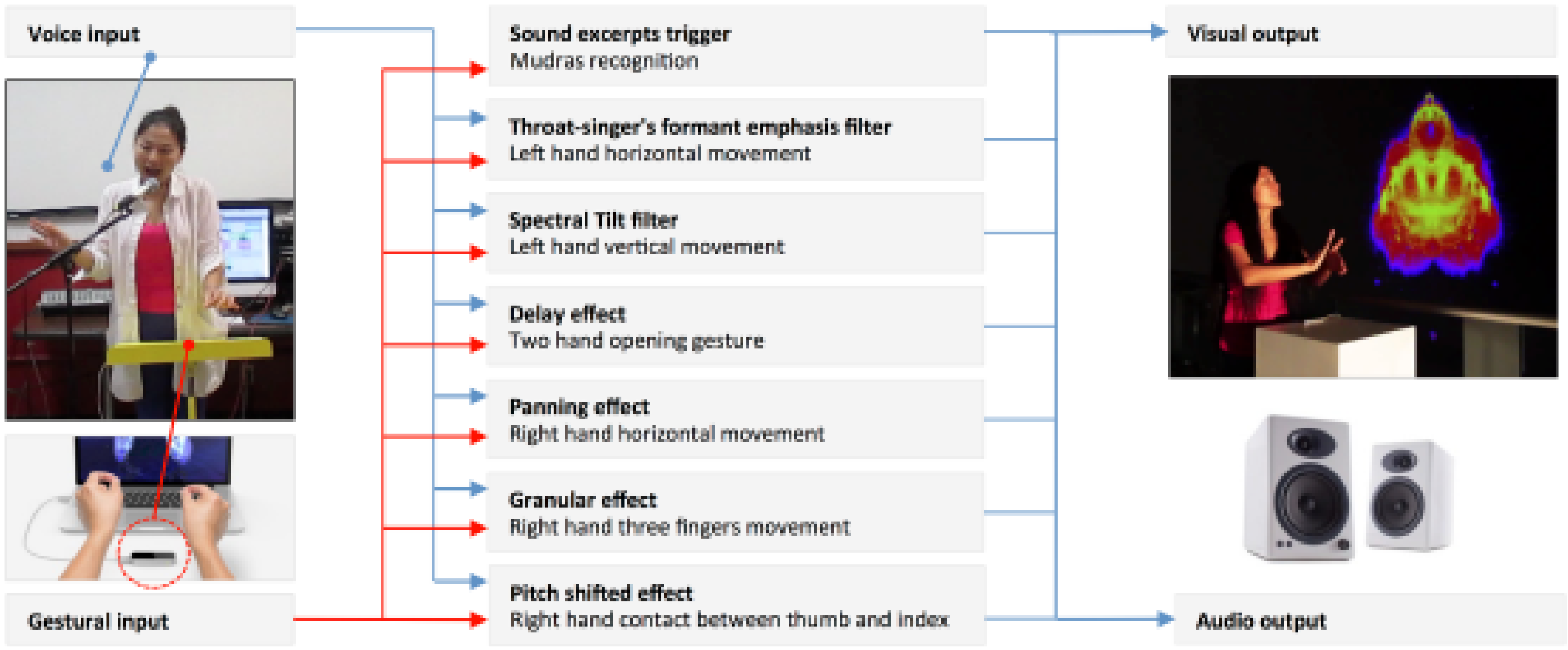
Magic of Gestural Input, Control, and Mapping

One-to-One mapping



One-to-Many mappings





Embodied Sonic Meditation (2019)

Modular Synth and gestural controller

Example 1
Example 2

Latency must be less than 10-20 ms!!

Input data -> Controller -> SoundMapping Strategy -> Creative Coding (MUSC 4360)

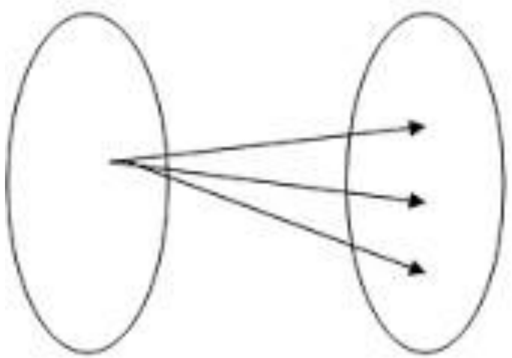
Magic of Gestural Input, Control, and Mapping

Embodied Sonic Meditation (2019)

Input data -> Controller -> SoundMapping Strategy -> Creative Coding (MUSC 4360)

Magic of Gestural Input, Control, and Mapping

One-to-Many mappings



Mandala (2018)