MIDI

EECS 4462 - Digital Audio

Click to edit Master text styles

Second level

Third l

F

Fifth level

September 17, 2018

What is MIDI?

- Stands for **M**usical **I**nstrument **D**igital **I**nterface
- A standardized digital communications protocol that allows computers, electronic musical instruments, and other devices to communicate music event information
- “Music event information”, **NOT** music
- MIDI 1.0 published in October 1983
- MIDI 2.0 rumoured for years

MIDI specification

- Specification of events, such as Note On, Note Off, Velocity (how hard the note was struck), Pitch Bend etc
  - MIDI controllers can create these events live
  - Instruments (physical or virtual) transform these events into audio
- Specification of a physical interface

MIDI physical interface

- A computer can send/receive MIDI events to/from physical instruments through USB

- These days though, virtual instruments are usually used to transform MIDI events into audio

MIDI Channels

- A MIDI signal is composed of 16 channels of events
- Similar to TV, each instrument (physical or virtual) can choose which channel it listens to
- Unlike TV, an instrument can listen to events on all channels at the same time (Omni mode)

MIDI Events

- Each MIDI event consists of (up to) 3 bytes
- The first byte is the STATUS byte
  - First bit always 1
  - 3 next bits: Type of event
  - 4 last bits: Channel number
- The other two bytes are DATA bytes
  - First bit always 0
  - Parameter values for the particular type of event (0-127)
  - May not be there

MIDI event example

10010011	**00011011**	**01111111**
STATUS	DATA	DATA
Event type: **001**	Note number: 27	Velocity: 127
Note ON	**D#1**	Loudest possible
Channel #: **0011**	**38.89Hz**	
Channel 3		

MIDI event types

Event type	Status Code	Data 1	Data 2
Note Off	000	Note number	Velocity
Note On	001	Note number	Velocity
Poly Pressure	010	Note number	Pressure value
Control Change	011	Controller number	Controller value
Program Change	100	New program	Not used
Mono pressure	101	Pressure Value	Not used
Pitch bend	110	Coarse value	Fine value
System	111	Event type	Event value

Poly/Mono Pressure

- Such events get created when a player that has already played a note on a keyboard presses harder on the keys
- Poly stands for Polyphonic
  - Greek for “many voices”
  - Individual events are created for each note held
- Mono stands for Monophonic
  - Greek for “one voice”
  - One event is created for all active notes (the highest pressure for all keys pressed)

Pitch bend

- MIDI controllers often have a pitch bending wheel
- This type of event indicates a change in the position of the wheel
- Both data bytes are used to indicate the value, i.e. 14 bits
  - 0000 – 3FFF in hex
  - 2000 in hex is the center position (no pitch change)

Program change

- This event indicates that the instrument (physical or virtual) should change to a different sound
- These different sounds are referred to in many ways
  - Sound modules
  - Programs
  - Voices
  - Patches
- A patch is typically a group of 16 different sounds, one for each MIDI channel

Control change

- A complicated MIDI setup may include several controllers (pedals, levers) that provide input and may affect the sound in different ways, e.g. volume, balance, various filters
- A control change event indicates a change in the value of one of these controllers
- Controllers 120 to 127 are reserved for special messages, such as
  - All Sound Off
  - Reset all controllers
  - Omni Mode ON

System events

- Used for various system purposes, such as
  - Timing Clock (for synchronization)
  - Active Sensing (sent to indicate that a connection is alive)
  - Song Position Pointer (beats since the beginning of the song)
  - Tune Request (to calibrate analog synthesizers)
  - Exclusive (reserved for each controller manufacturer to implement their own event system)

General MIDI

- A specification agreed by manufacturers in 1991 to ensure consistency in sound
- Before that, each manufacturer would assign arbitrary sounds to different program numbers

Program #	Instrument	Program #	Instrument
1-8	Piano	65-72	Reed
9-16	Percussion	73-80	Pipe
17-24	Organ	81-88	Synth Lead
25-32	Guitar	89-96	Synth Pad
33-40	Bass	97-104	Synth Effects
41-48	Strings	105-112	Ethnic
49-56	Ensemble	113-120	Percussive
57-64	Brass	121-128	Sound FX

General MIDI Drum Mapping

- Different drum sounds in MIDI are represented by different note numbers
- Before General MIDI there was no standardization
- Channel 10 was also designated for drum sounds

MIDI Note #	Drum Sound
36	Bass Drum
38	Snare Drum
46	Open Hi-Hat
42	Closed Hi-Hat
51	Ride Cymbal
41	Low Tom
50	High Tom

MIDI Controllers

- The typical MIDI controller is a keyboard
- However, many types of devices can generate MIDI events
- Various expression pedals, knobs, faders
- There are MIDI guitars as well
- Virtual controllers are common
  - You can run them on your computer, or use your cellphone as a MIDI controller

Or you can go fancy…

Some free software to play with

- Windows
  - Virtual MIDI Piano Keyboard
    https://sourceforge.net/projects/vmpk/
  - MIDI-OX
    http://www.midiox.com/
- Mac
  - Virtual MIDI Piano Keyboard
    https://sourceforge.net/projects/vmpk/
  - MIDI Monitor
    https://www.snoize.com/MIDIMonitor/

Instructions

Next week

- Monday: Guest Lecture by Ricardo Reimao on Deep Learning and Audio
  Detecting Fake Speech
- Wednesday: First look at first assignment and the framework we will use for both MIDI and raw audio development

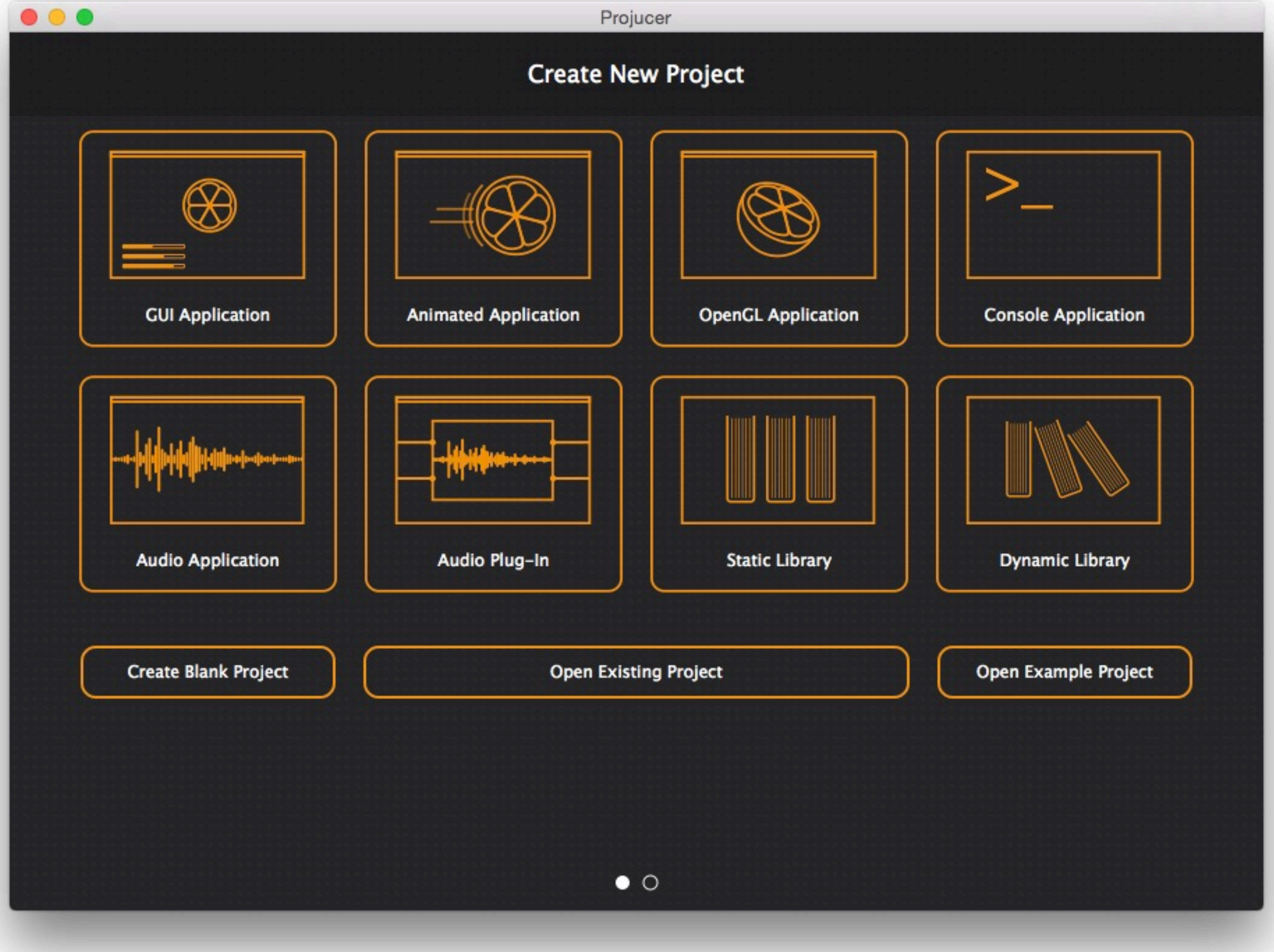
Preparing for the first assignment

- We will use a framework called JUCE to read and modify MIDI events
- Download and install JUCE here: https://shop.juce.com/get-juce
- It is also installed in the lab
- You will need a C++ IDE to go with it
  - Mac: Xcode
  - Windows: Visual Studio or Code::Blocks
  - Linux: Code::Blocks

Getting started

- Lots of tutorials are available at: https://juce.com/learn/tutorials
- Start with **Getting started with the Projucer**
- The Projucer is a wizard that will guide you to create all the files you will need to compile your code
- Before running it, create an account at my.roli.com
- Once you run and login, you will see…

Projucer wizard



Click on Audio Plugin

Projucer Wizard

- Select the location for the code and your IDE
- Click Create…
- Enable
  - Plugin wants MIDI input
  - Plugin produces MIDI output
- Click Save Project and open in IDE…
- Your IDE should open with all the generated code
- More on this next week…