มหาวิทยาลัยอัสสัมชัญ
ASSUMPTION UNIVERSITY
of THAILAND

VINCENT MARY
SCHOOL of SCIENCE and
TECHNOLOGY

# MIDI Signals Manipulating Console for Live Multiple-Keyboard Performance

## Senior Project I

**By**

| | |
|---|---|
| Taechasit Sarasitt | 6110032 |
| Napatsorn Kawaree | 6111099 |
| Kanyarat Nalucupchanchai | 6115308 |

**Project Advisor**
Asst. Prof. Dr. Thitipong Tanprasert

**Committee Members**
Asst. Prof. Dr. Benjawan Srisura
A. Chayapol Moemeng

# Introduction

What is MIDI ?

Echo
Positif
Grand-Orgue
Solignac
extended version
Coupleurs
Pedal
Sébastian Halas (recorded & GO version)
Augustine's V.O., 2020 (HW. version)

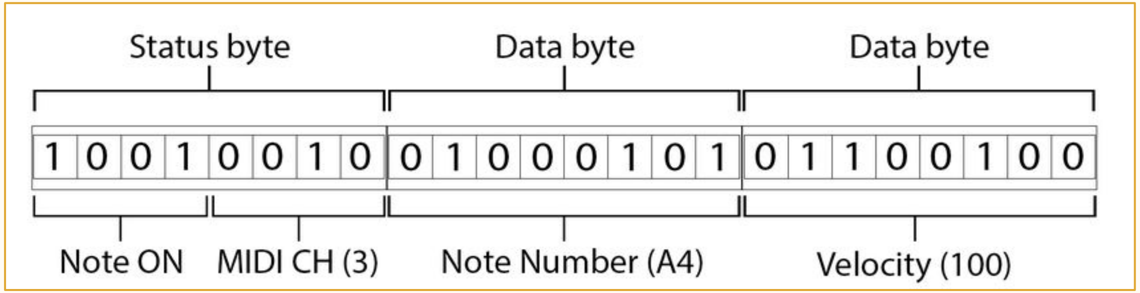
Musical Digital Interface

# Protocol of MIDI

It is a standard protocol invented in 1982 as a music communication system of electronic music devices such as computers, synthesizers and so on.

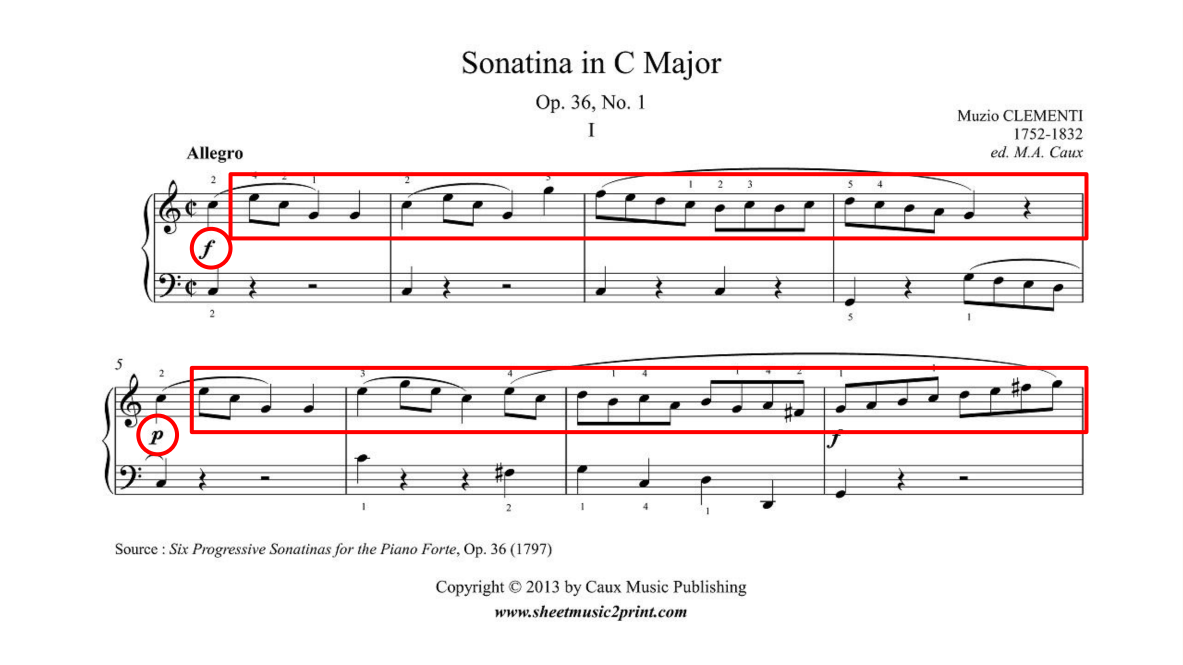
# MIDI messages

The MIDI message is up to 3-bytes long and consists of 3 parts, status byte, data byte 1, and data byte 2. The status byte is mandatory, it describes the functionality of the message alongside with the channel it operates on. While the optional data byte indicates the detail of the message.



| Status byte | | Data byte | Data byte |
|---|---|---|---|
| 1 0 0 1 | 0 0 1 0 | 0 1 0 0 0 1 0 1 | 0 1 1 0 0 1 0 0 |
| Note ON | MIDI CH (3) | Note Number (A4) | Velocity (100) |

# Example of Data byte



## Sonatina in C Major

Op. 36, No. 1

I

Muzio CLEMENTI
1752-1832
*ed. M.A. Caux*

Source : *Six Progressive Sonatinas for the Piano Forte*, Op. 36 (1797)

○ Data byte 1

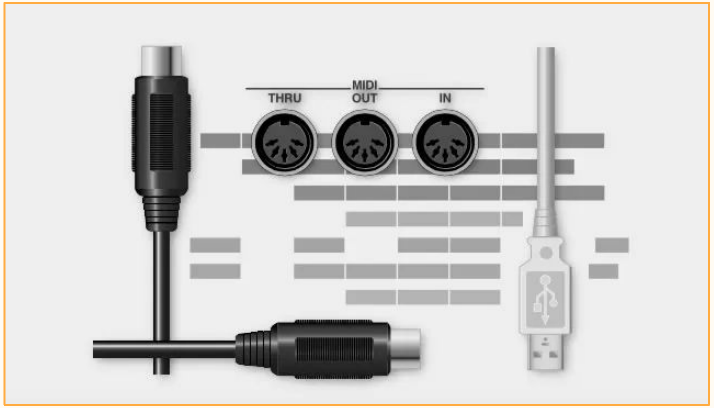
▭ Data byte 2

# 🪇 MIDI transport

The original MIDI 1.0 Specification called for using a 5-Pin DIN cable to connect MIDI compatible devices but today there are many different "transports" capable of carrying MIDI data and the specification for 5-Pin DIN has been updated. So MIDI 2.0 was there to change the musical life.

Significant updates on MIDI 2.0

- Adds bidirectional communication while maintaining
- backwards compatibility
- High-speed transports such as USB



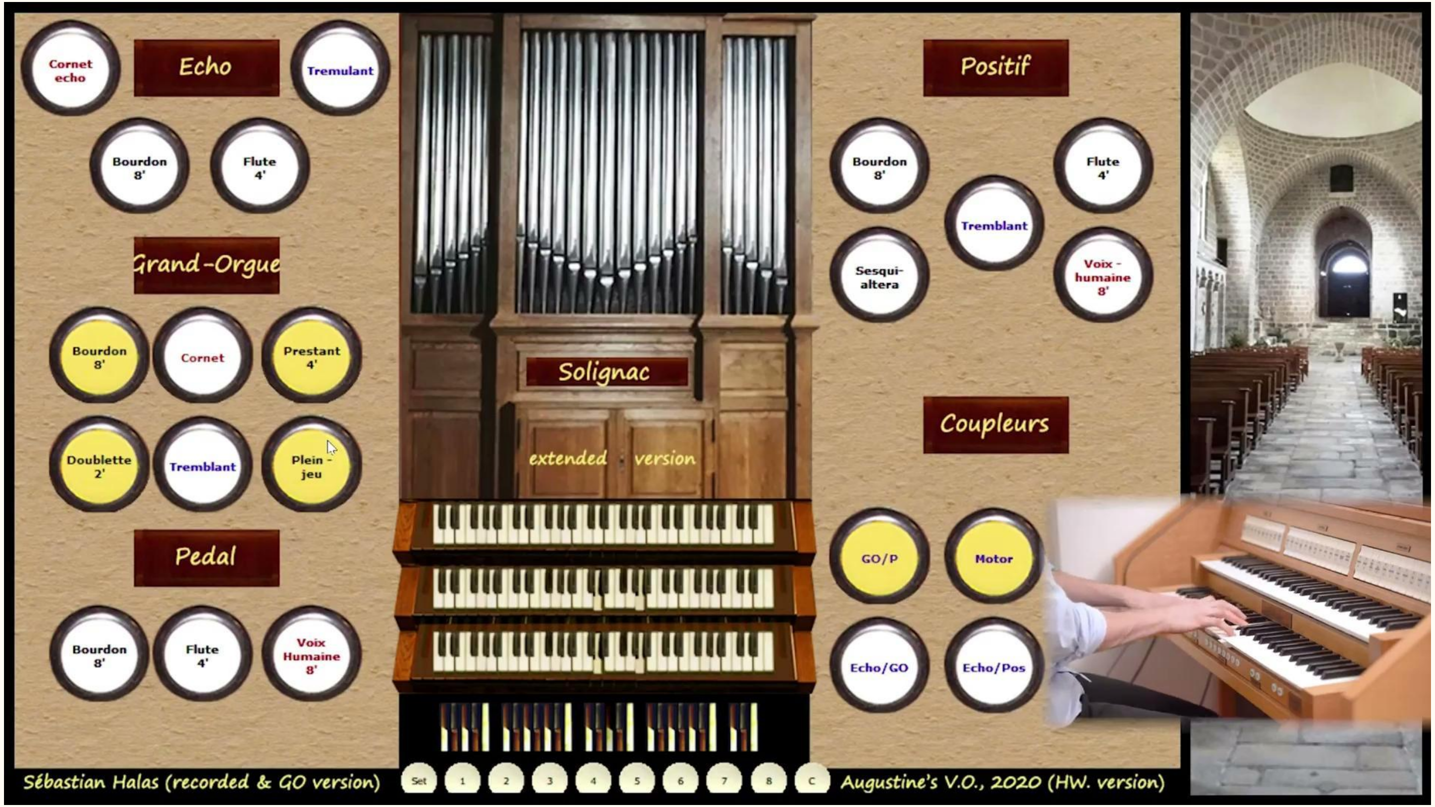
# ♪ Related work 🥁

The King of All Virtual Instruments

Software for your Mac & PC

Hauptwerk is a German computer program available from Milan Digital Audio designed to allow playback or live playback of pipe organ music using MIDI and recorded audio samples.

# Sampleset of Haupwerk Free Version



# Problem Statement

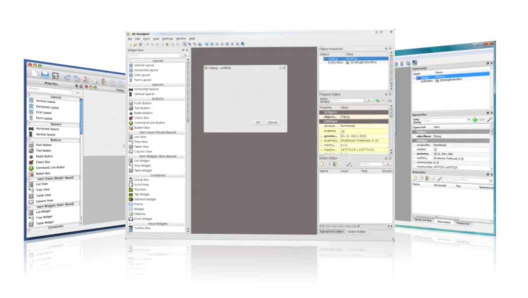
In order to facilitate digital audio technology in providing flexible arrangement of sounds for a musician to perform in live mode, while keeping minimal dependence on specific hardware. A software is to be developed as a generic console of a multiple-keyboard musical instrument.

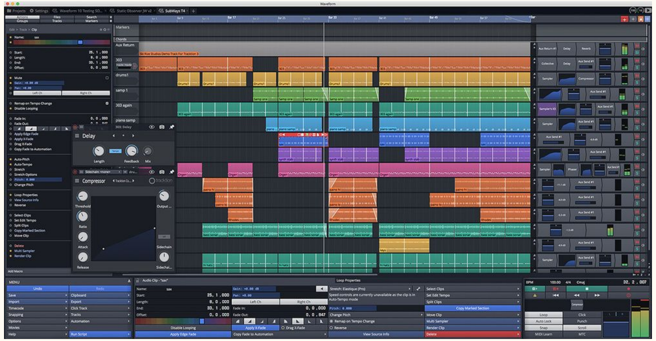
- The user can connect MIDI-controller keyboards to the PC running the software.
- The software provides user-friendly interface for selecting and mixing sounds, for selecting rhythm and tempo.
- The user interface is expected to utilize touch-screen technology.
- Saving and restoring sounds arrangements as presets that can be recalled on the fly, and manages all the MIDI outputs to produce the actual sounds through Digital Audio Workstation.

# 🎺 Scope of the project 🎹

- Using Python as programming language, Mido as MIDI library, and Qt designer as user interface design software.
- Tracktion Waveform Digital Audio Workstation (DAW) to turn MIDI signals into sound.
- The PC’s specification must support ASIO audio standard or a compatible one.

python™





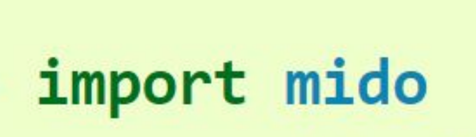
# Methodology

# Mido

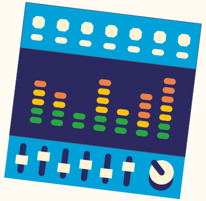
Mido (MIDI Objects for Python) library to interact with midi. Mido is created by Ole Martin Bjørndalen and many other contributors. It is released as an open source library under the terms of MIT license.

Mido has support for multiple backends:

- RtMidi (default)
- PortMidi
- Pygame
- rtmidi-python
- Amidi



```
import mido
```

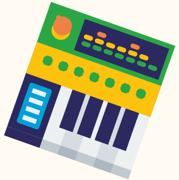


# Qt

Qt is a cross-platform application development framework for desktop, embedded and mobile. Supported Platforms include Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS and others.

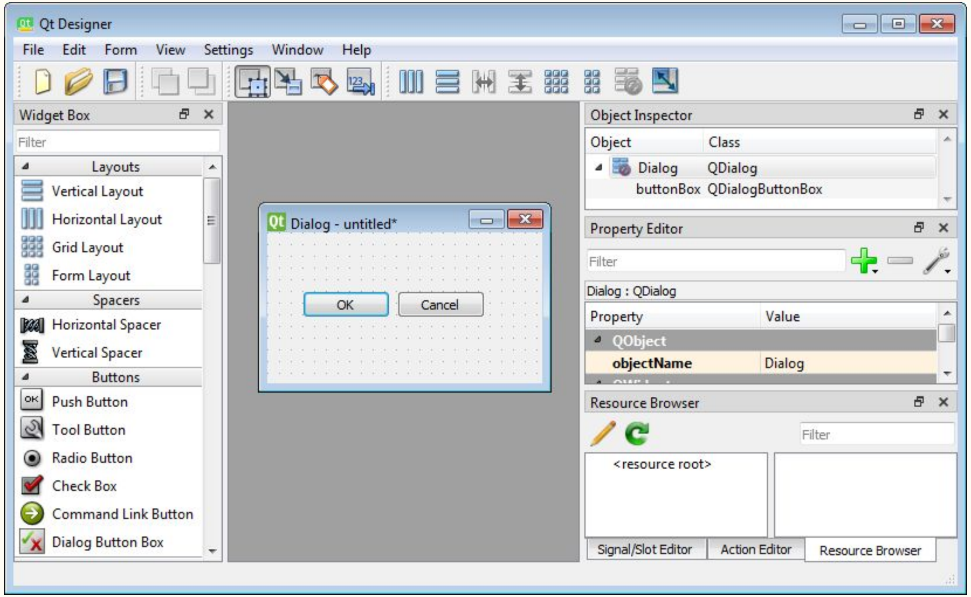
Qt is also a graphical user interface (GUI) framework.

# Qt designer



Qt Designer is a tool for quickly building graphical user interfaces with widgets from the Qt GUI framework.

Simple drag-and-drop interface for laying out components such as buttons, text fields, combo boxes and more.

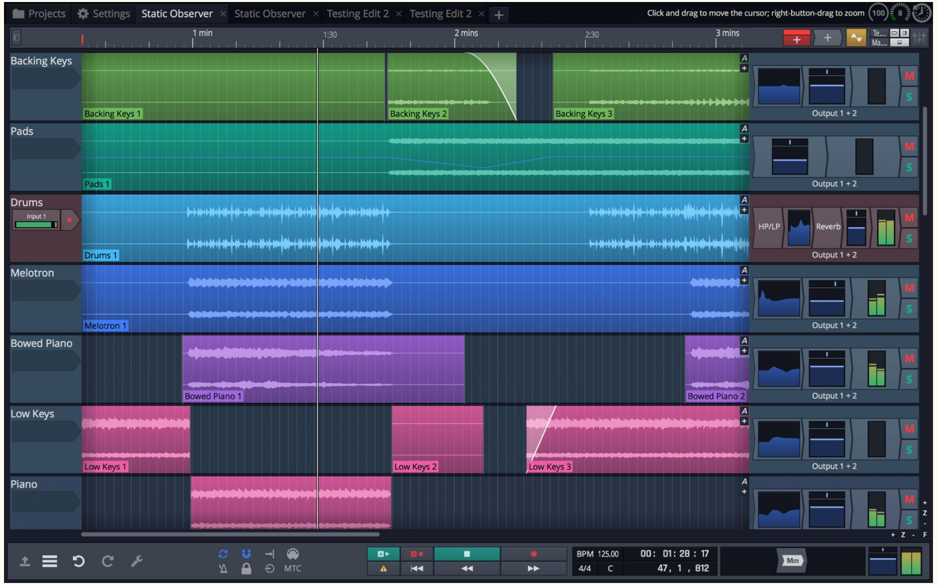


# Tracktion Waveform

Tracktion Waveform is a Digital Audio Workstation for recording and editing audio and playing MIDI.

The software is cross-platform, running on Apple macOS, Microsoft Windows and Linux.

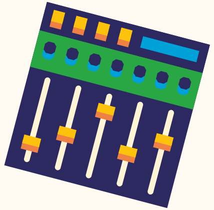
Turn the midi signal that has been manipulated from our software into audible sound.



# LoopMIDI

LoopMIDI is an application that can create virtual loopback MIDI ports. The ports created are unique for each user and only exist while the loopMIDI-application is running.

So if we log-off, the created ports cease to exist. LoopMIDI comes from the fact that the ports created with this tool will cease to exist once the app is closed.



# MIDI-OX

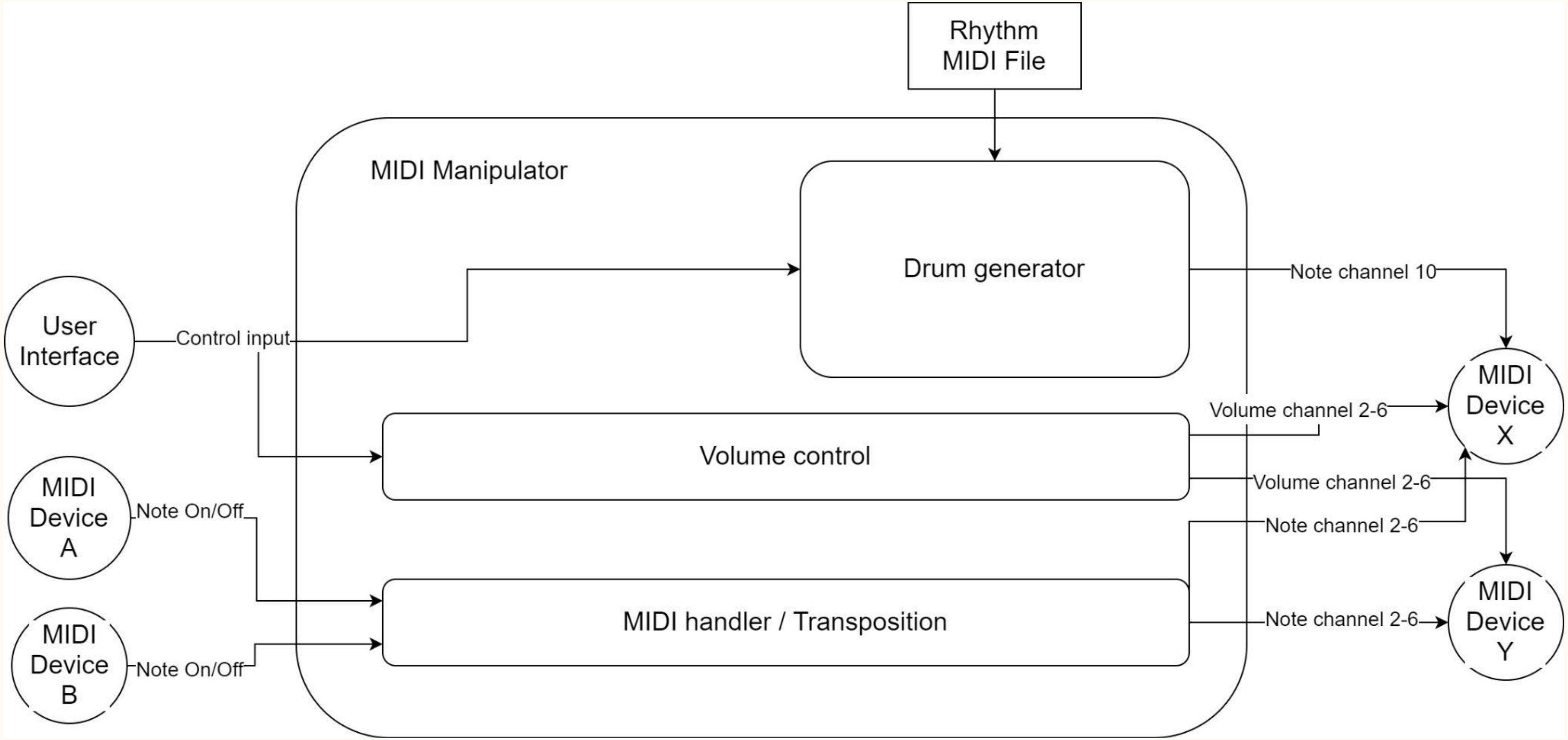
MIDI-OX is a versatile utility that is great for troubleshooting faulty MIDI hardware devices. It displays incoming MIDI streams, and passes the data to a MIDI output driver or the MIDI Mapper.

- Used to debug the message sent from our software.

# User Interface & Function breakdown

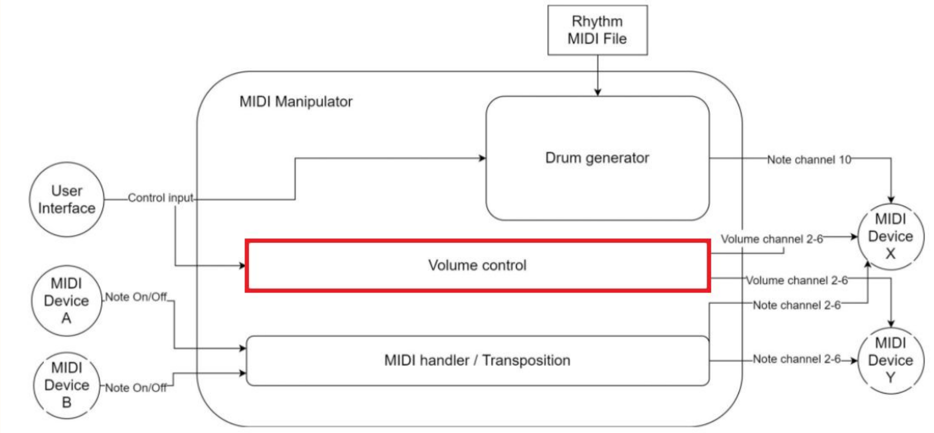
## MIDI Signals Manipulating Console

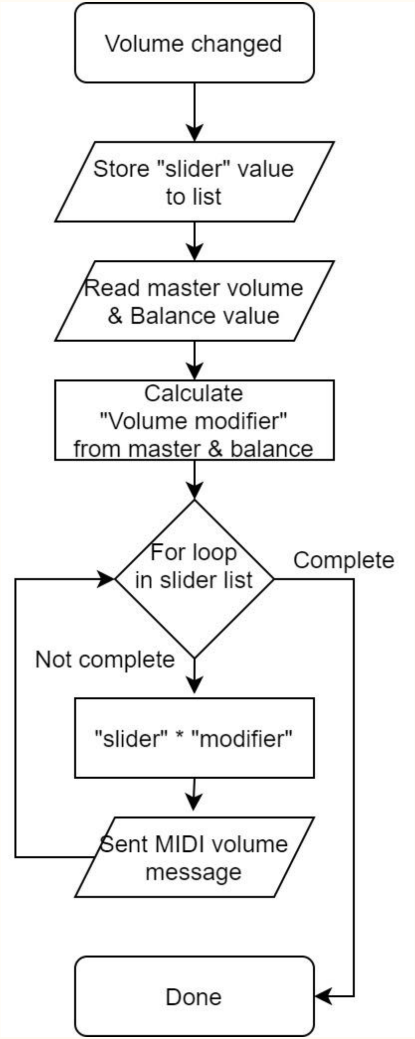
# ♫ System architecture



# Volume control

- Volume control is used to send the midi control (CONTROL/MODE CHANGE) message to the destination midi devices.
- It will be called once any of volume-related UI is change.
- The “slider” value is 0-127, which is the minimum / maximum value for midi messages.
- Then it will be modified by master and balance value before sending to the destination midi device



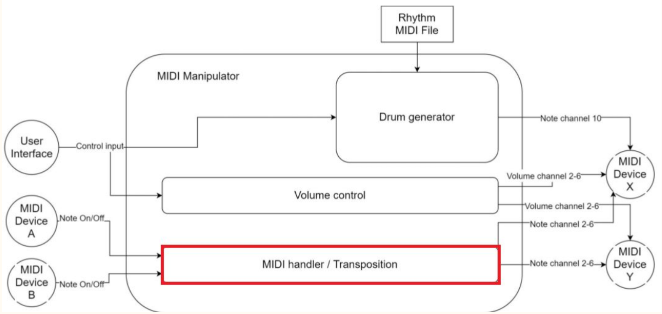

Volume changed
Store "slider" value to list
Read master volume & Balance value
Calculate "Volume modifier" from master & balance
For loop in slider list
Complete
Not complete
"slider" * "modifier"
Sent MIDI volume message
Done

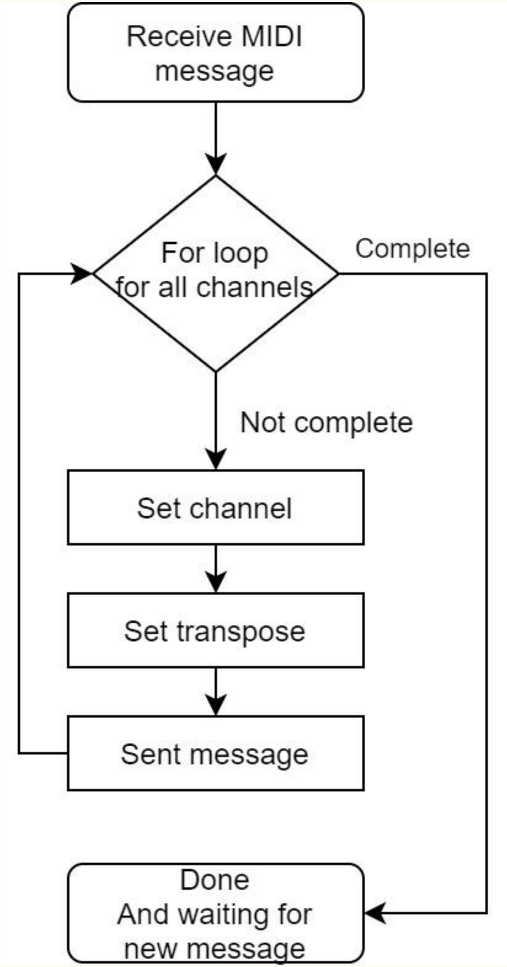
# MIDI handler

- It responsible for duplicating and forwarding midi signals from source device to the destination device in real time.

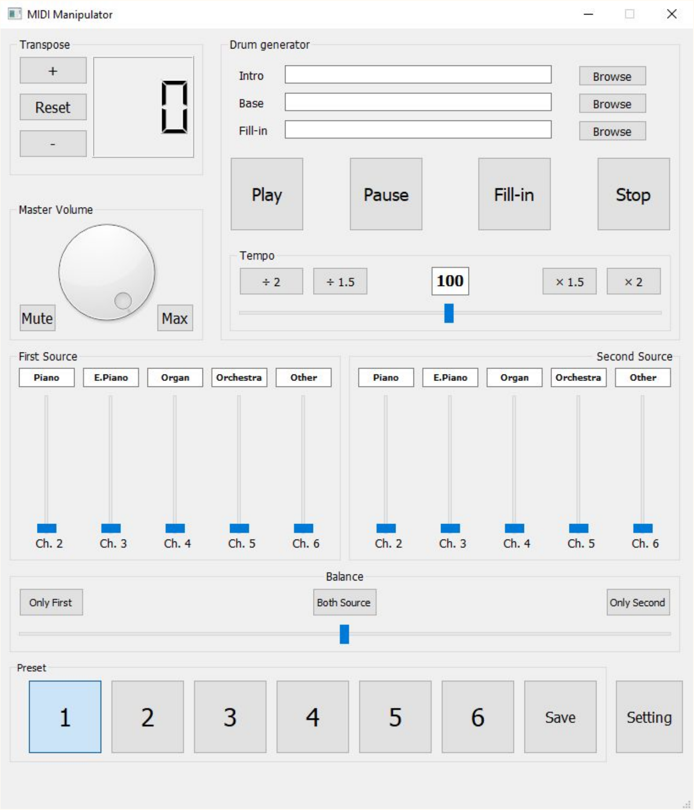
- The transposition happens in the MIDI handler by modifying the midi NOTE ON/OFF message data according to the transpose set in software.

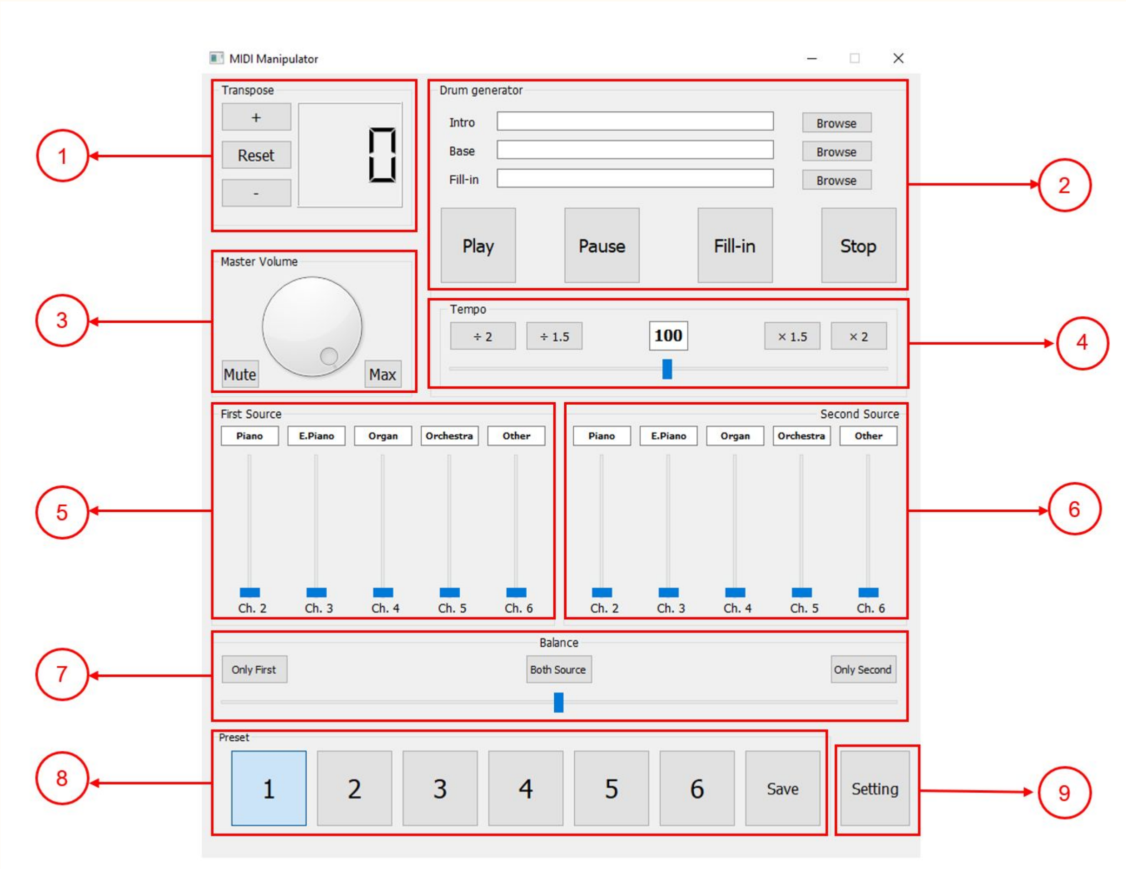
- MIDI handler function is operated by using the “callback” feature in mido library. Which is the function that only acts when the midi message is sent to the port that callback operates on.





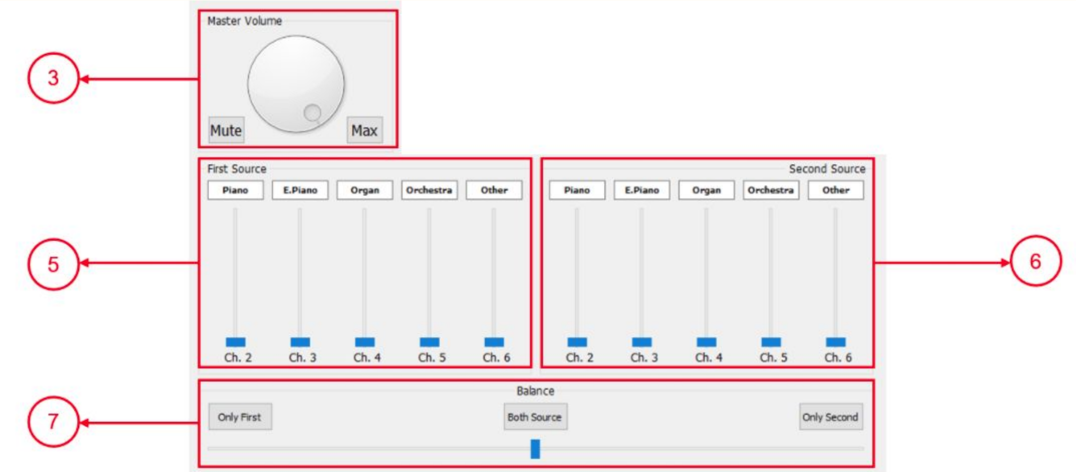
# User interface





# User interface - Transpose

# User interface - Volume control

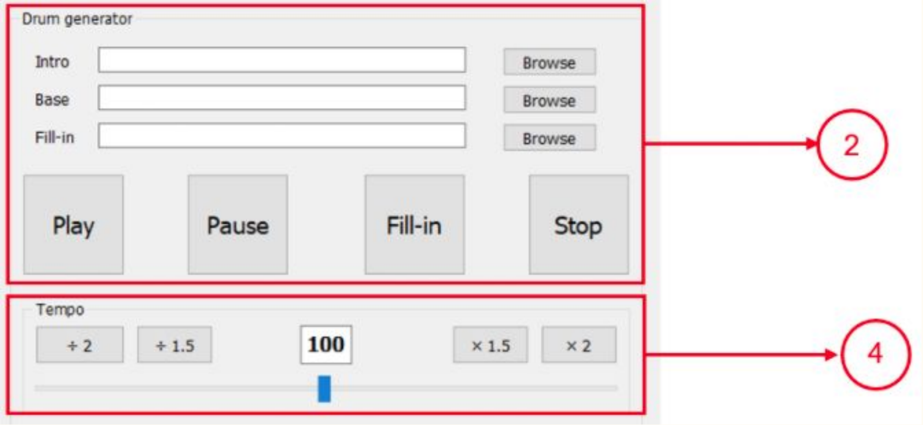


The volume control system is consist of 3 parts

- Slider (5), (6) is used to control the volume of each channel
- Balance (7) is used to control the difference is volume level that output from (5) and (6)
- Master volume (3) is used to control the volume of every component
- Volume calculation = Slider value * Modifier
- Modifier is calculated from Master volume and Balance

# User interface - Volume control

# User interface - Drum generator



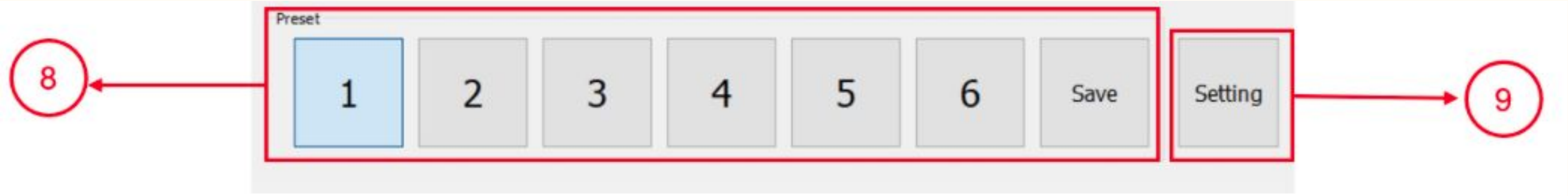
- Function that can play Midi file, loop, and set the tempo of that particular file played.

- It will play the intro file followed by the looped base, and it will play fill-in when the user clicks the fill-in button.

- The tempo of the drum generator is set by the slider in (4). If users want to increase or decrease the tempo by half or one fold, they can easily click the button above the slider.

# User interface - Drum generator

# User interface - Preset

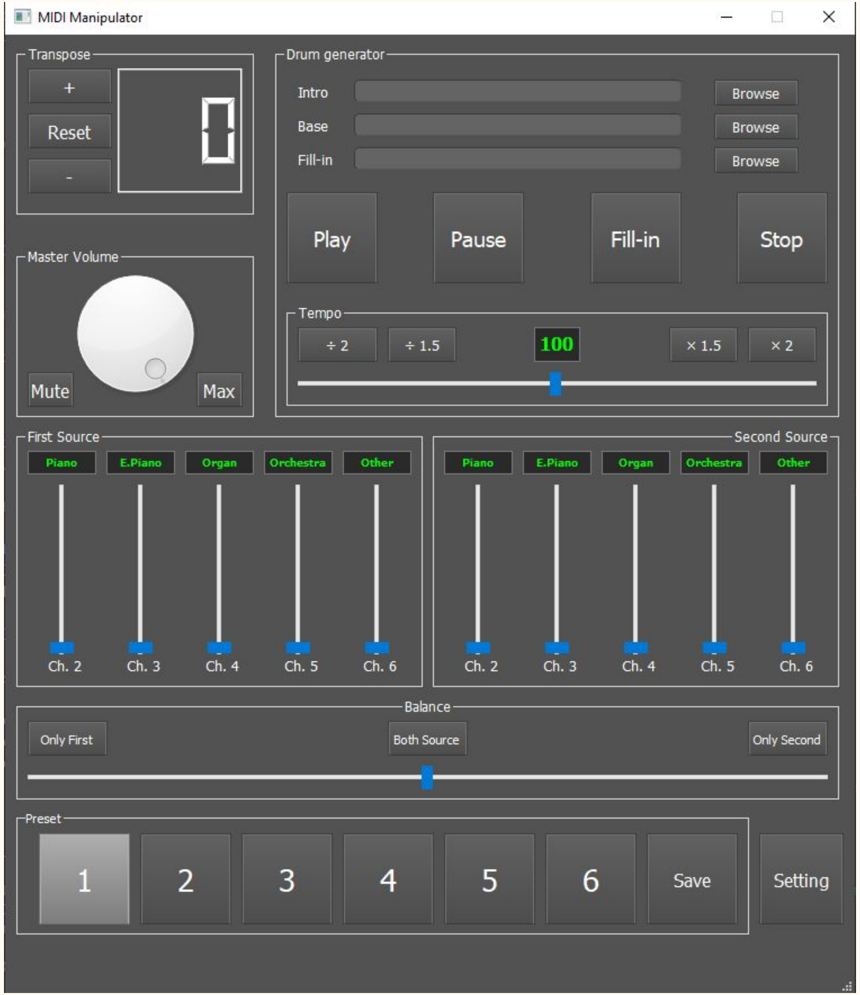


- It allows the user to save the setting in (3), (5), (6), (7) into the preset.
- They can change the setting by just single clicking, which will be useful in the live-performance.
- Player can save the setting into preset by clicking the Save button.

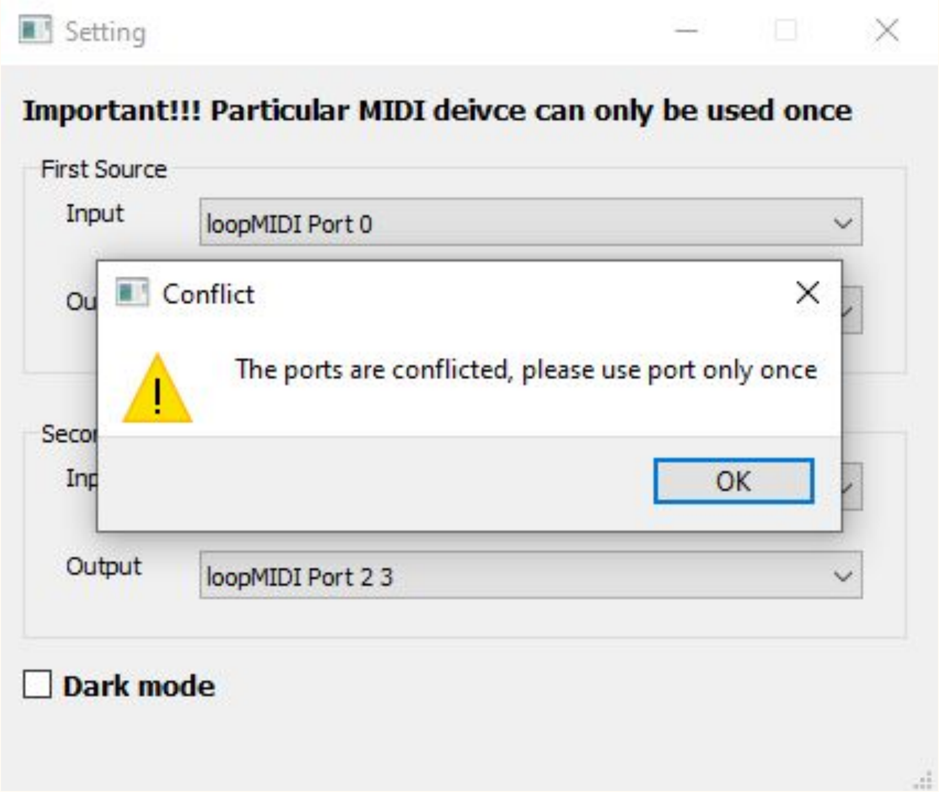
# User interface - Profiling

# User interface

## Dark mode



# User interface - Protection



Setting

**Important!!! Particular MIDI deivce can only be used once**

First Source

Input: loopMIDI Port 0

Ou

Conflict

The ports are conflicted, please use port only once

Secon

Inp

Output: loopMIDI Port 2 3

□ **Dark mode**

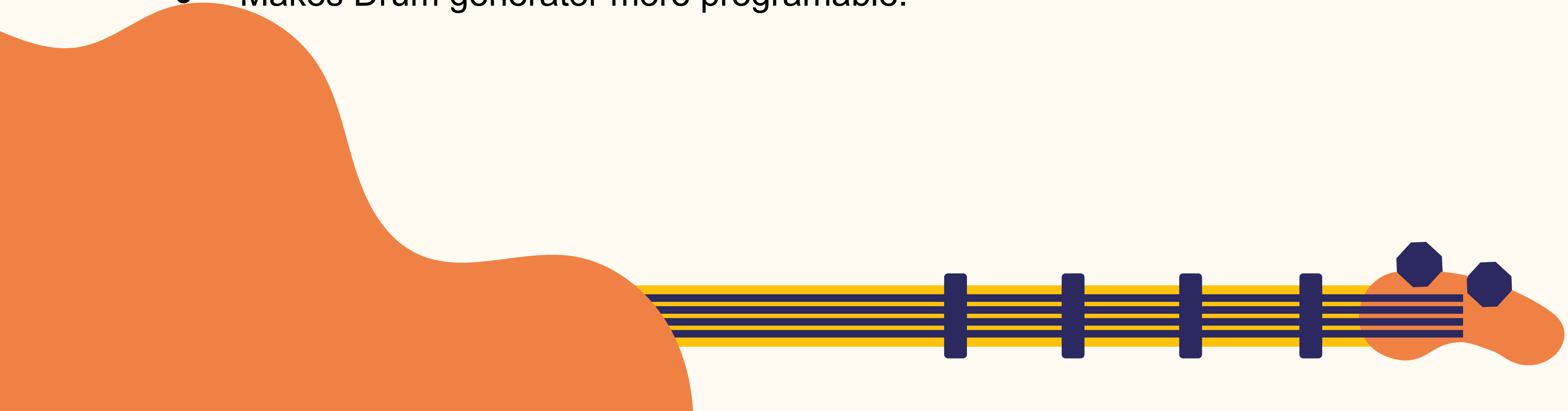
# Real-life performance

# Conclusion

# Conclusion

There are suggestions of what can be in the further stage

- The design can be improved in the future to be more creative and friendlier looking.
- The output channel for each port will be flexible for the users in the future.
- Addressing the compatibility issues of the Drum generator.
- Makes Drum generator more programable.



# Source code

https://github.com/Taechasit1001/AU_SeniorProject_1

# Thank you