

MIDI Command Controller Interface

Heyan Huang

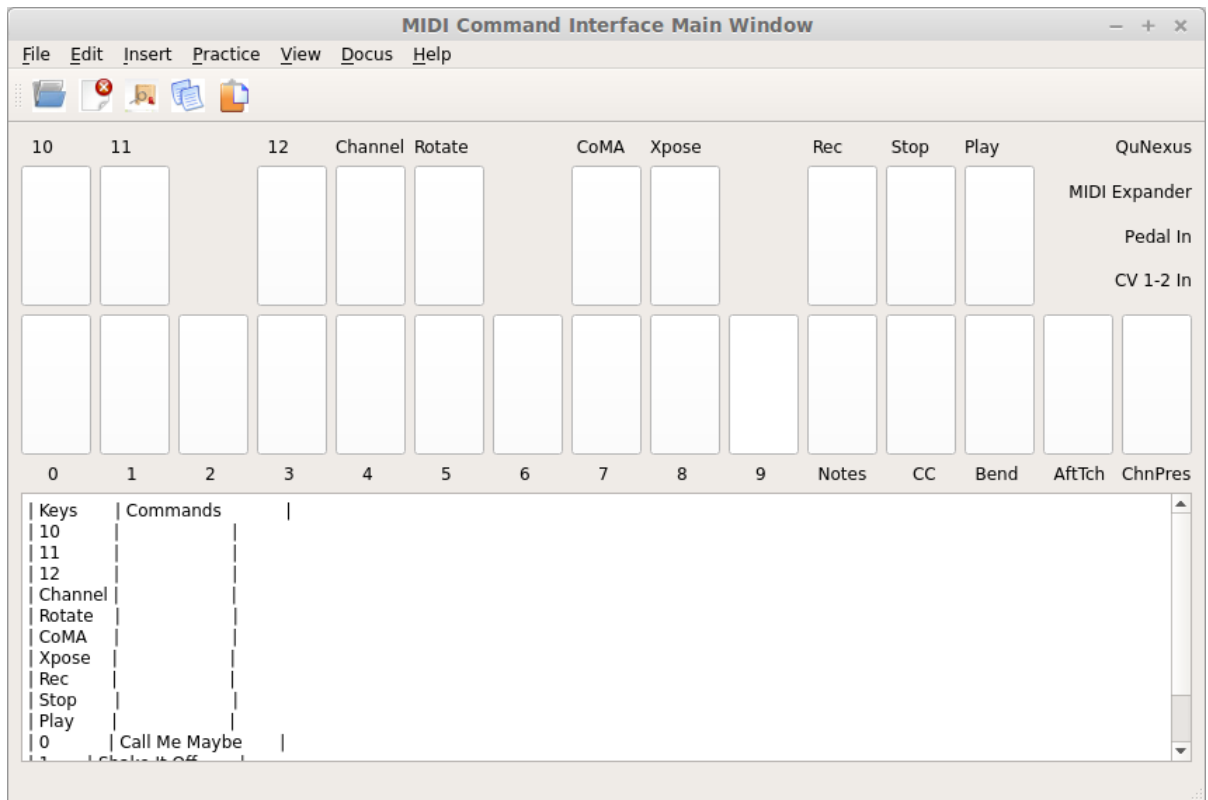
November 23, 2014

Contents

1	Update 11/23/2014, updates include	1
2	Review 11/21/2014, updates include	2
2.1	Review Contents	2
2.2	Detailed Requirements	2
3	Project Requirements	2
4	main functionality	3
4.1	Read data from MIDI	3
4.2	Write data back to MIDI	3
5	Programming Language	3
5.1	Qt	3
5.2	c++	3
6	Interface Design	3
7	Midi keys and corresponded operations	3
8	Interface Guide	3
9	References	3

1 Update 11/23/2014, updates include

- Cleaned repository so that it looks clean and nice;
- Remove menubar as suggested by advisor;
- Removed topright four line texts cause it's not necessary;
- Shifted top line keys so that they look like original midi controller layout;
- Changed PlainTextEdit so that they satisfies the requirements;
- Added left side 8 keys, just that three keys **Bend**, **-Oct**, **Oct+** are **NOT** like the original shapes yet, need work on them later on;
- Will link possible functionalities to make it a functional softwares first, and then updates minus issues.
- Current layout looks as below snapshotted:



2.2 Detailed Requirements

- menubar is NOT necessary, and could be removed away;
- Interface topright four line texts are not necessary, could be removed away;
- Interface top line keys should shift to the right by half key width so that the interface looks similar to the original midi controller keyboard;
- PlainTextEdit should be changed to be array of 25/33 lines of (text label, file name editor, browse QPushButton layout);
- Left handside 8 keys should be included in the midi interface even functionalities are not required at this moment;
- When finished the above basic ones, if I have extra time, could explore the left side 8 keys to test if it is possible to use them to set a bunch of sequence so that save time when needed compared with set sequence one by one from the basic 25 keys.

3 Project Requirements

- Use QuNexus Midi controller as a command controller to manipulate play sequence for tower lights show;
- Besides the main functionalities, create a Qt Creator Interface to help facilitate the tower light playing process for clients convenience.

4 main functionality

4.1 Read data from MIDI

- Use the MIDI Controller as a special Controller that can be operated to play specific songs sequence, or do some specific work.
- play specific sequence may be the work for keys 0-9, and 10-12, how about other 20 keys? Do they require specific work to be done?

4.2 Write data back to MIDI

- When a key was pushed, the specific Controller key's LED is supposed to be on to indicate the operation.
- Trick about the LED to be continuously on is that when a key is pressed, that is 1 byte that indicates the "Duration" of the key press, I may need to
 - try to set this byte to be a large value, (1 byte, $2^8 = 256$, it has limits!)
 - or continuously reset is to be that large value;
 - or continuously write this key to be pressed data back to MIDI with time intervals

5 Programming Language

5.1 Qt

- the worries that I have by using Qt is that if Qt has the capability to handle the MIDI-Linux connection problems.
- And also Qt-to-Audio (linux) connection things as well. Should it be Qt, or as far as I can set it to work in Linux, just let it be that way then?

5.2 c++

- I believe C++ is the most widely used Language used by those midi sequencer softwares, so I have no better choice than c++ right now.

6 Interface Design

7 Midi keys and corresponded operations

Table 1: midi keys and corresponded operations

Keys	Commands
10	
11	
12	
channel	
Rotate	
CoMA	
Xpose	
Rec	
Stop	
Play	
0	Call Me Maybe
1	Shake It Off
2	All About That Bass
3	...
4	
5	
6	
7	
8	
9	
Notes:	
CC	
Bend	
AftTch	
ChnPres	
Togl A	
Velo B	
Preset	
Pres C	
Tilt D	
Bend	
Oct-	
Oct+	

8 Interface Guide

- Give text instructions on how to use the Interface, and what are the corresponded operations by press specific keys.
- Like list the above table in the Interface Guide text area.

9 References

- For circle QPushButton

<http://stackoverflow.com/questions/12734319/change-rectangular-qt-button-to-round>

- Draw circle separate
<https://coderalbert.wordpress.com/2014/03/16/creating-circle-in-linux-using-qt-creator/>
- For Rectangle Arc
<http://stackoverflow.com/questions/20416789/how-to-add-a-small-triangle-at-one-of-the-corn>
- PaintEvent Triangle
<http://stackoverflow.com/questions/20416789/how-to-add-a-small-triangle-at-one-of-the-corn>
<http://stackoverflow.com/questions/3894737/qt4-how-to-draw-inside-a-widget>
<http://qt-project.org/forums/viewthread/1623>
<http://stackoverflow.com/questions/7968269/basic-qt-gui-qpushbutton-for-drawing-a-line>
- QPushButton::drawButton(QPainter *painter);
<https://www.tbi.univie.ac.at/~pmg/tutorials/QT/html/qpushbutton.html>
- QGraphicsSene QGraphicsProxy...
- QPushButton raised enabled
<http://www.qtcentre.org/threads/42852-QStyledItemDelegate-paint-QPushButton-with-styleshee>
- QPushButton two icons
<http://www.qtcentre.org/threads/39445-How-to-add-two-icons-images-to-the-same-QPushButton>
- QPainter
<http://qt-project.org/forums/viewthread/23628>
- QGridLayout ScrollArea
<http://qt-project.org/forums/viewthread/20843>
<http://qt-project.org/forums/viewthread/20924/>
- Leftover five
<http://qt-project.org/doc/qt-4.8/qpainter.html>
<http://qt-project.org/doc/qt-4.8/qwidget.html>
<https://www.tbi.univie.ac.at/~pmg/tutorials/QT/html/qpushbutton.html>
<http://qt.developpez.com/doc/4.7/qpainter/#drawpolygon>
<http://qt.developpez.com/doc/4.7/painting-basicdrawing/>