Arturia Keylab MK2 MC Custom Script

Preinstallation Requirements and Instructions

There are two options in this script, the **FULL** and the **Stripped-Down** one.

In the **Stripped-Down** option, these things are missing:

- No browsing of instruments and effects plugins available via the FN + Jog Wheel turns. This is replaced by the default Cubase's presets browsing functionality.

- No feedback to our Keylab's display for:

  Punch in and out states,

  MIDI Record mode,

  MIDI Cycle Record mode,

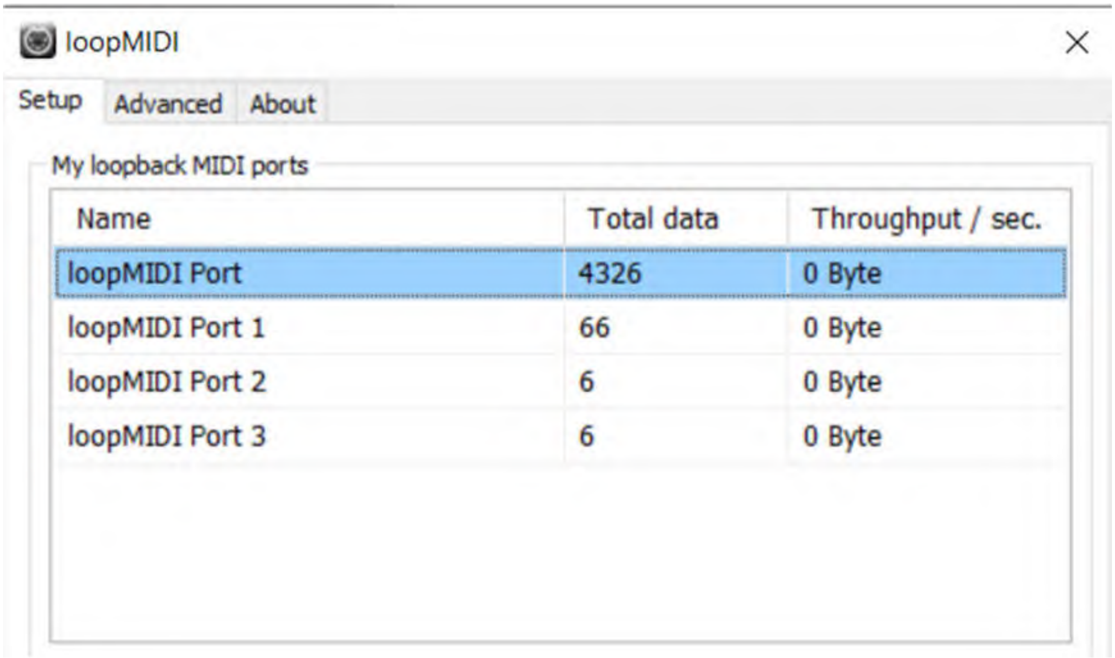
  Metronome Precount.

Apart from the above, all other functions are the same.

[Applies to the FULL implementation. If you're going to use the stripped-down version, please skip this step]

If you're going to use the full implementation of this script you will need 4 virtual MIDI ports (apart from the default ports of our Keylab).

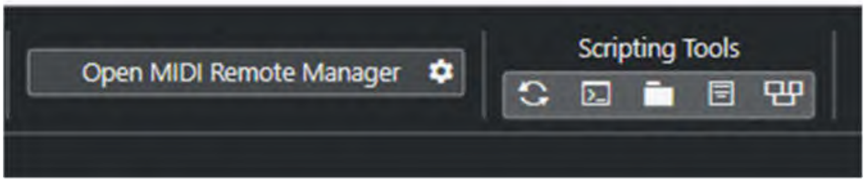
For this, I'm using **loopMIDI** (you can obviously use other virtual ports if you have) and my ports are setup like this:



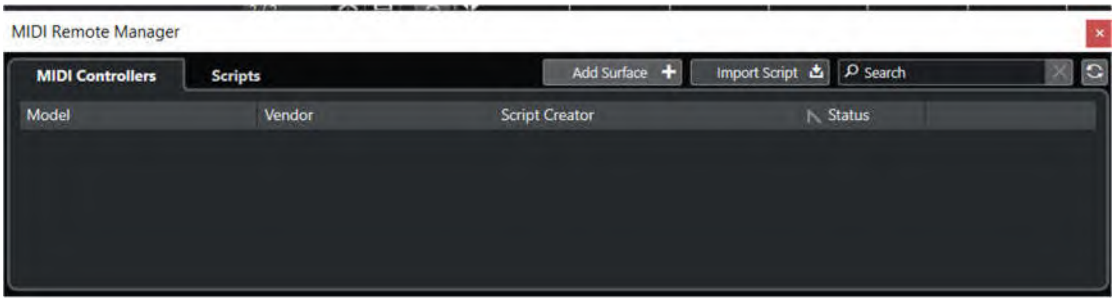
Name	Total data	Throughput / sec.
loopMIDI Port	4326	0 Byte
loopMIDI Port 1	66	0 Byte
loopMIDI Port 2	6	0 Byte
loopMIDI Port 3	6	0 Byte

[Applies to BOTH types of implementation]

Open the **MIDI Remote TAB** in Cubase and then click on **Open MIDI Remote Manager**:



In the Midi Remote Manager window, click on **Import Script:**



In the File Dialog, choose the script's file you've downloaded and Cubase will install the script and its components.

The next step is to make some adjustments if needed.

If you need the FULL version of the script, you have to edit the mapOfGeneralSettings.js file and set:

generalSettings.strippedDown=0 (Save and Close this file after this edit)

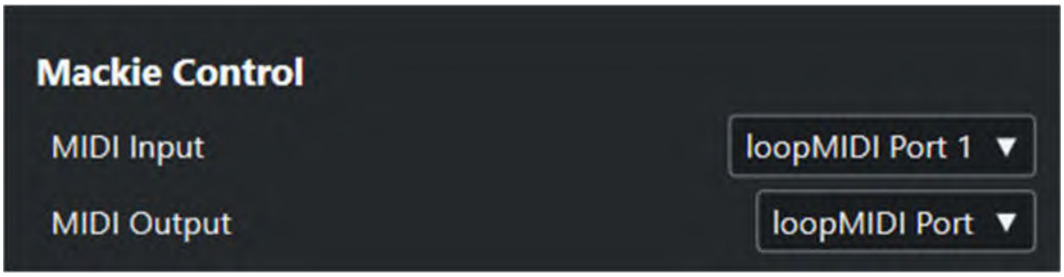
[Applies to the FULL implementation. If you're going to use the stripped-down version, please skip this step]

If we do want to use the full implementation's functionalities, there are two steps involved, apart from setting **generalSettings.useLoopMidiToo=1** in the **mapOfGeneralSettings.js** file:

1. **Setup for allowing changing plugins using our controller**

   Open **Cubase's Studio->Studio Setup**, click on **Add Device** and select **Mackie Control**.

   In the ports of this new item, assign **loopMIDI Port 1** and **loopMIDI port** respectively, as shown in the screenshot below, and click **Apply**.



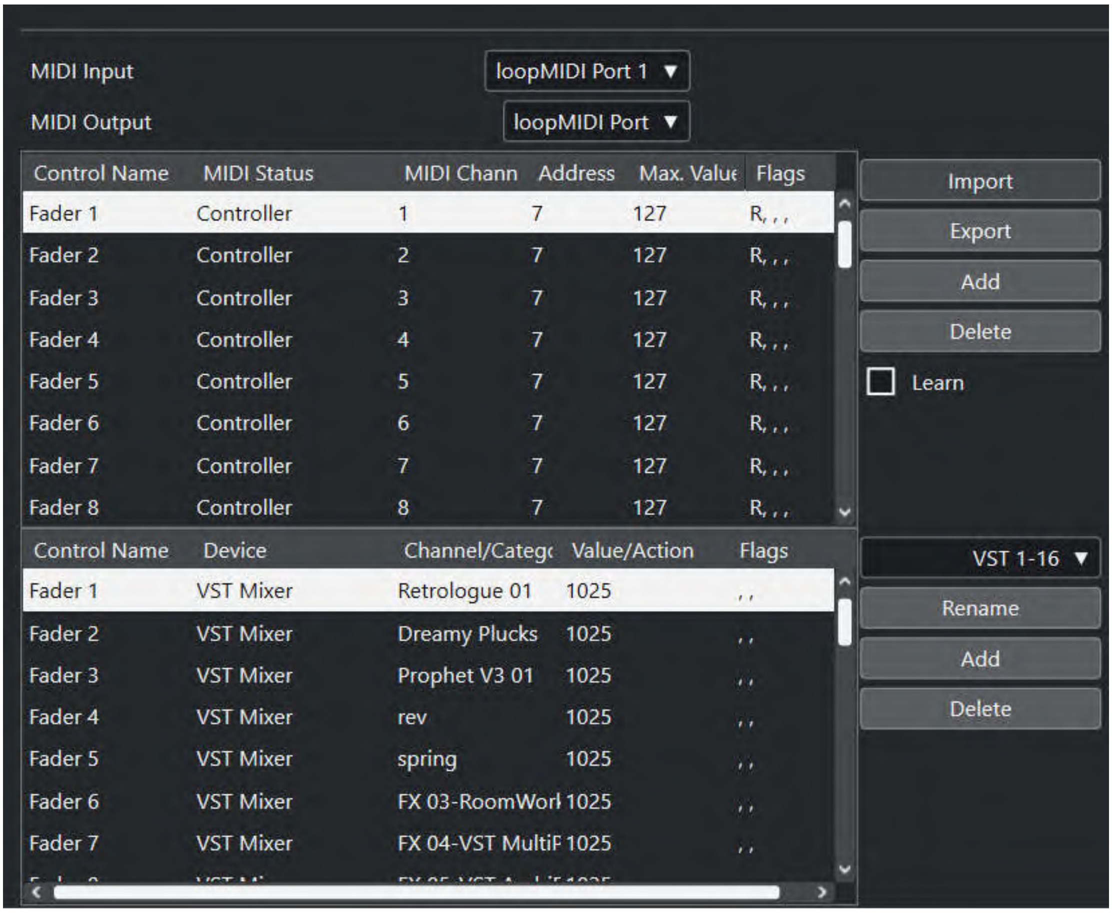
Mackie Control

MIDI Input: loopMIDI Port 1

MIDI Output: loopMIDI Port

2. **Setup for receiving punch Out, Midi Record Mode, Midi Cycle Record Mode and Metronome Pre-Count**:

   Click again on **Add Device** but this time select the **Generic Remote** item.
   Set the Midi In and Out ports as shown in the following screenshot and click on **Import:**



MIDI Input: loopMIDI Port 1 ▼

MIDI Output: loopMIDI Port ▼

Control Name	MIDI Status	MIDI Chann	Address	Max. Value	Flags
Fader 1	Controller	1	7	127	R, , ,
Fader 2	Controller	2	7	127	R, , ,
Fader 3	Controller	3	7	127	R, , ,
Fader 4	Controller	4	7	127	R, , ,
Fader 5	Controller	5	7	127	R, , ,
Fader 6	Controller	6	7	127	R, , ,
Fader 7	Controller	7	7	127	R, , ,
Fader 8	Controller	8	7	127	R, , ,

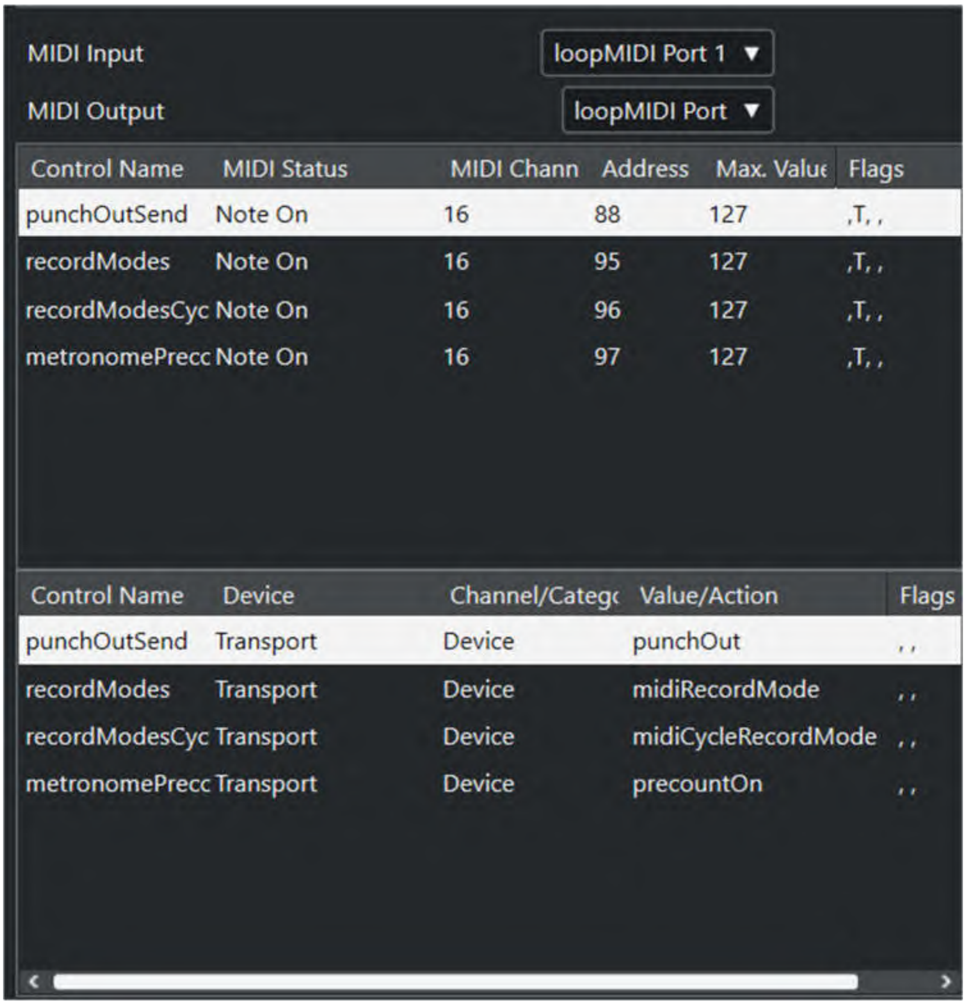
Control Name	Device	Channel/Categ	Value/Action	Flags
Fader 1	VST Mixer	Retrologue 01	1025	, ,
Fader 2	VST Mixer	Dreamy Plucks	1025	, ,
Fader 3	VST Mixer	Prophet V3 01	1025	, ,
Fader 4	VST Mixer	rev	1025	, ,
Fader 5	VST Mixer	spring	1025	, ,
Fader 6	VST Mixer	FX 03-RoomWorl	1025	, ,
Fader 7	VST Mixer	FX 04-VST MultiF	1025	, ,

Browse to the script's folder and select the file:

Arturia_KeyLab_MK2_MC_Custom_Generic_Remote_For_loopMIDI.xml

Hit **Apply** and you're done.

If you don't want to import this generic remote script, you must insert the following entries (**be careful to not miss setting the flag T**), AFTER deleting all the previous default entries:



MIDI Input loopMIDI Port 1 ▼

MIDI Output loopMIDI Port ▼

Control Name	MIDI Status	MIDI Chann	Address	Max. Valu	Flags
punchOutSend	Note On	16	88	127	,T, ,
recordModes	Note On	16	95	127	,T, ,
recordModesCyc	Note On	16	96	127	,T, ,
metronomePrecc	Note On	16	97	127	,T, ,

Control Name	Device	Channel/Catego	Value/Action	Flags
punchOutSend	Transport	Device	punchOut	, ,
recordModes	Transport	Device	midiRecordMode	, ,
recordModesCyc	Transport	Device	midiCycleRecordMode	, ,
metronomePrecc	Transport	Device	precountOn	, ,

[Applies to BOTH implementations]

SETUP

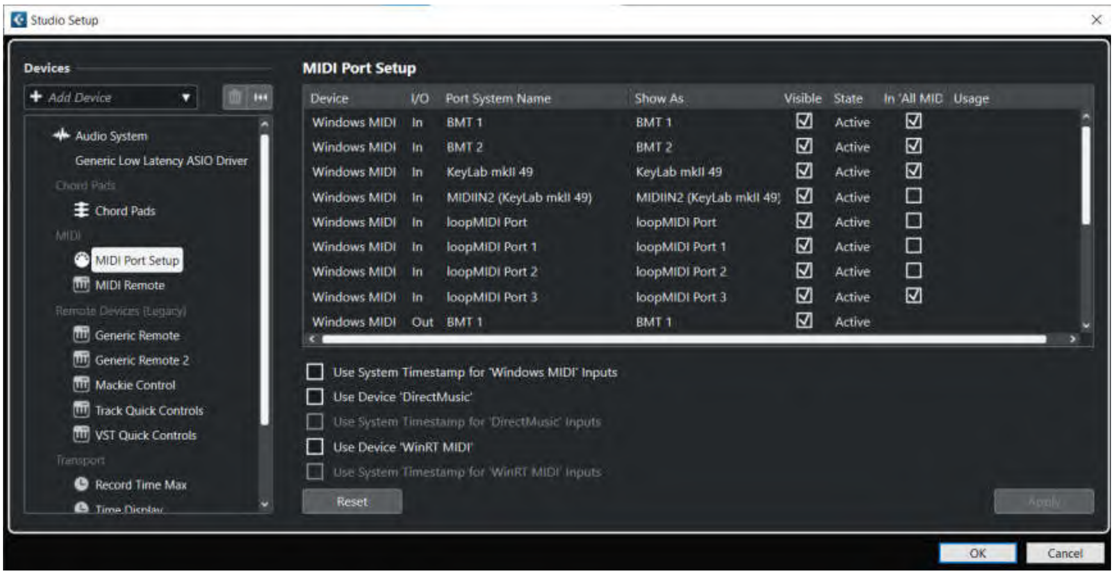
WARNING!

You really have to disable the original Mackie Device of the default Arturia's implementation by setting its ports to None, upon testing my script. Otherwise, there will be conflicts and erratic controller's behavior.

MIDI ports setup

From inside Cubase, go to menu **Studio-> Studio Setup → Tab Midi Port Setup** and **uncheck** the entry **MIDIIN2** of the Keylab (this is its default port for DAW control) from the "**In 'ALL MIDI inputs'**" column.

If using the full implementation, please uncheck the following ports too: **loopMIDI Port**, **loopMIDI Port 1 and loopMIDI Port 2**. **Leave the loopMIDI Port 3 checked**.



MIDI Port Setup

Device	I/O	Port System Name	Show As	Visible	State	In 'All MID	Usage
Windows MIDI	In	BMT 1	BMT 1	☑	Active	☑	
Windows MIDI	In	BMT 2	BMT 2	☑	Active	☑	
Windows MIDI	In	KeyLab mkII 49	KeyLab mkII 49	☑	Active	☑	
Windows MIDI	In	MIDIIN2 (KeyLab mkII 49)	MIDIIN2 (KeyLab mkII 49)	☑	Active	□	
Windows MIDI	In	loopMIDI Port	loopMIDI Port	☑	Active	□	
Windows MIDI	In	loopMIDI Port 1	loopMIDI Port 1	☑	Active	□	
Windows MIDI	In	loopMIDI Port 2	loopMIDI Port 2	☑	Active	□	
Windows MIDI	In	loopMIDI Port 3	loopMIDI Port 3	☑	Active	☑	
Windows MIDI	Out	BMT 1	BMT 1	☑	Active		

□ Use System Timestamp for 'Windows MIDI' Inputs

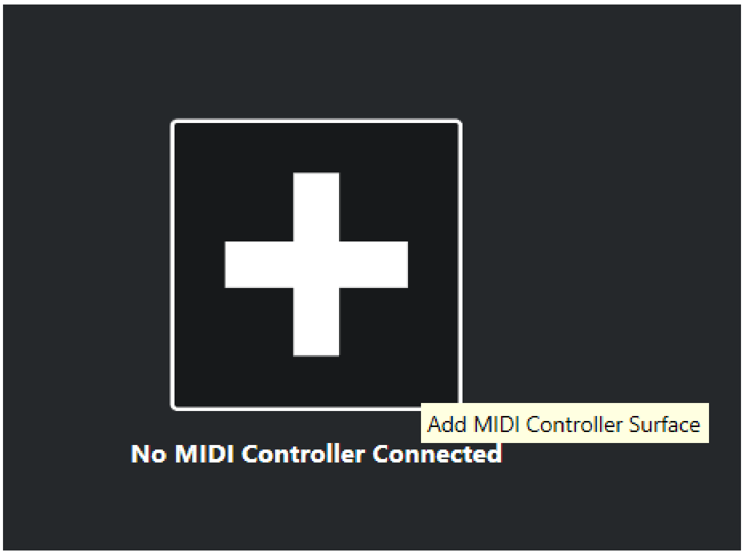
□ Use Device 'DirectMusic'

□ Use System Timestamp for 'DirectMusic' Inputs

□ Use Device 'WinRT MIDI'

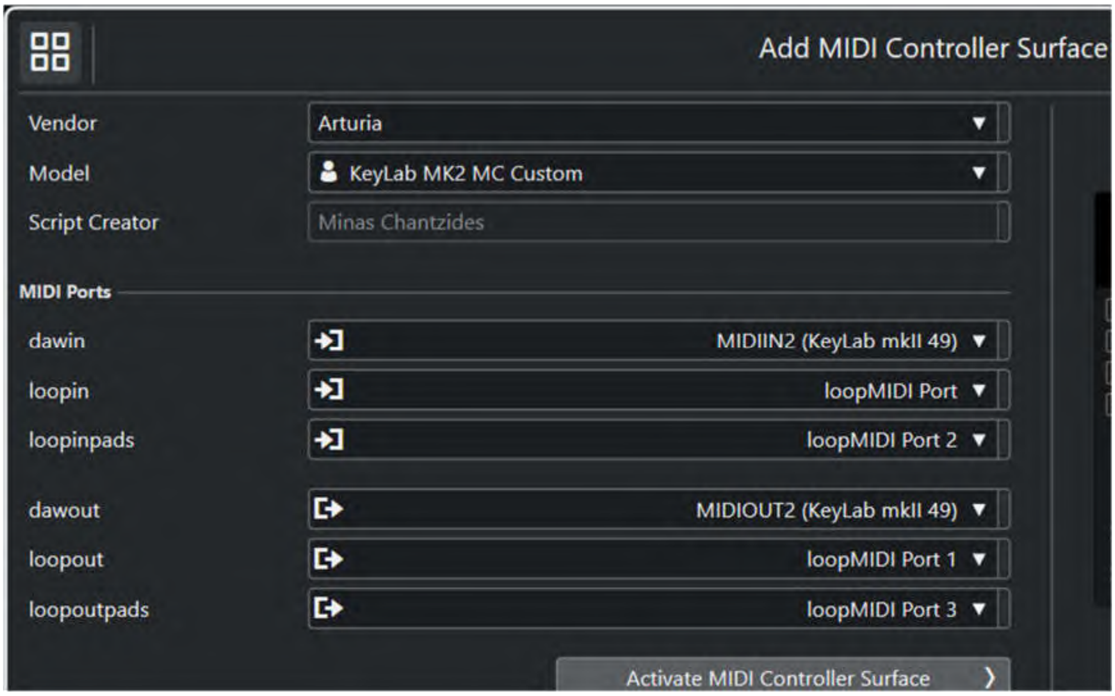
□ Use System Timestamp for 'WinRT MIDI' Inputs

Go back to Cubase and after selecting the **MIDI Editor Tab** click on the big **PLUS** button:



Fill the form as follows and click on the **Activate MIDI Controller Surface** button.

If you're going to use the **full implementation**, you have to fill in the form as follows:



Add MIDI Controller Surface

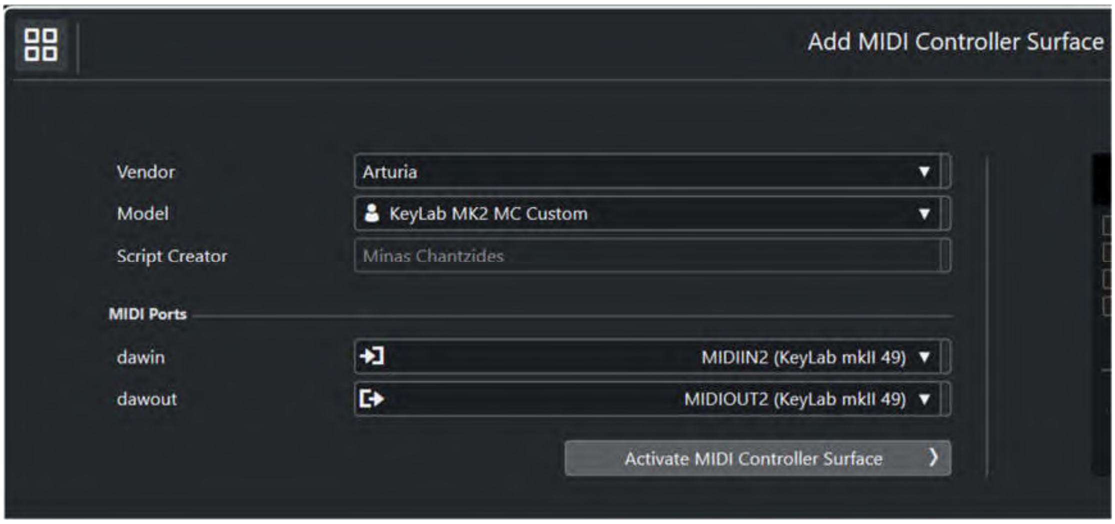
Vendor	Arturia
Model	KeyLab MK2 MC Custom
Script Creator	Minas Chantzides

MIDI Ports

dawin	MIDIIN2 (KeyLab mkII 49)
loopin	loopMIDI Port
loopinpads	loopMIDI Port 2
dawout	MIDIOUT2 (KeyLab mkII 49)
loopout	loopMIDI Port 1
loopoutpads	loopMIDI Port 3

Note (again) that this is the full implementation of the script. In this one we are using apart from the Keylab's MIDI ports, another two sets, created by using the loopMIDI utility, available from Tobias Erichsen's website here: https://www.tobias-erichsen.de/software/loopmidi.html .

If you have selected the **stripped-down version**, you will see this much simpler form, in fact you have to select just the MIDIIN2 and MIDIOUT2 ports of the Keylab (DAW ports):



Vendor: Arturia
Model: KeyLab MK2 MC Custom
Script Creator: Minas Chantzides

MIDI Ports
dawin: MIDIIN2 (KeyLab mkII 49)
dawout: MIDIOUT2 (KeyLab mkII 49)

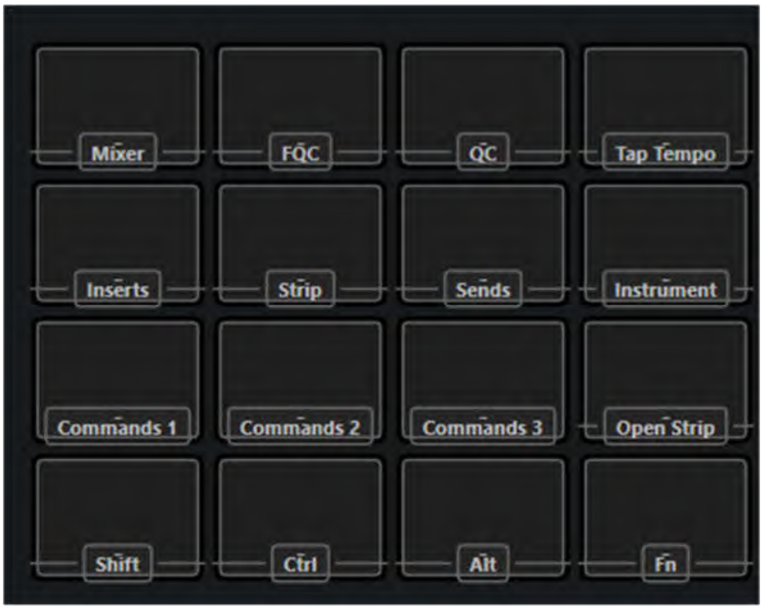
Pages

The script creates the following pages (their short description displayed on pads, in parenthesis):

- **Mixer (Mixer)**
- **Focused Quick Controls (FQC)**
- **Quick Controls (QC)**
- **Channel Strip (Strip)**
- **Insert Effects (Inserts)**
- **Send Effects (Sends)**
- **Commands Set 1 (Commands 1)**
- **Commands Set 2 (Commands 2)**
- **Commands Set 3 (Commands 3)**
- **Commands Set 4 (No pad mapped)**

We can navigate through the various pages using the **Left** and **Right** arrow buttons of the Keylab (the ones placed left and right of the jog wheel):

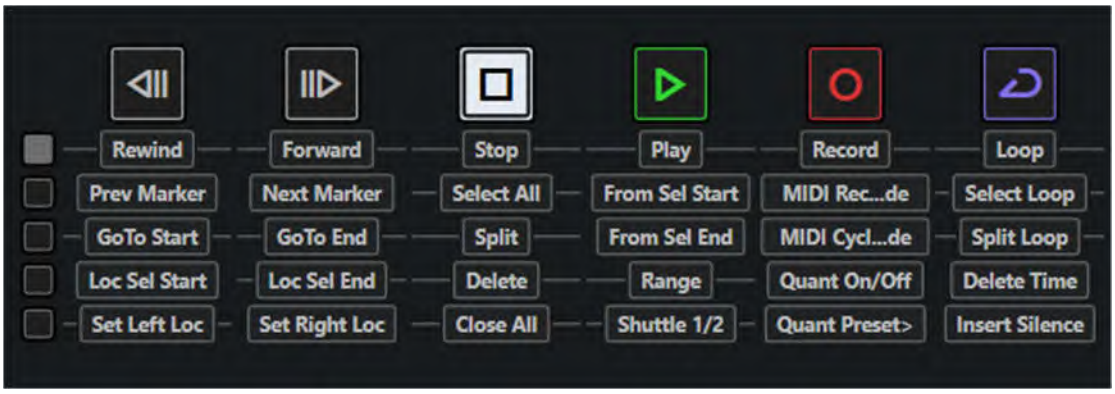
We can also recall pages by using Pads, as shown below:



Mixer	FQC	QC	Tap Tempo
Inserts	Strip	Sends	Instrument
Commands 1	Commands 2	Commands 3	Open Strip
Shift	Ctrl	Alt	Fn

IMPORTANT: In the bottom section of our pads, we see four assignments: Shift, Ctrl, Alt and FN. From now on, when I refer to these, keep in mind that I'm actually referring to these four pads and NOT our keyboard's ones.

Transport Buttons Section



■	Rewind	Forward	Stop	Play	Record	Loop
□	Prev Marker	Next Marker	Select All	From Sel Start	MIDI Rec...de	Select Loop
□	GoTo Start	GoTo End	Split	From Sel End	MIDI Cycl...de	Split Loop
□	Loc Sel Start	Loc Sel End	Delete	Range	Quant On/Off	Delete Time
□	Set Left Loc	Set Right Loc	Close All	Shuttle 1/2	Quant Preset>	Insert Silence

Here we see the mapping of our transport section.

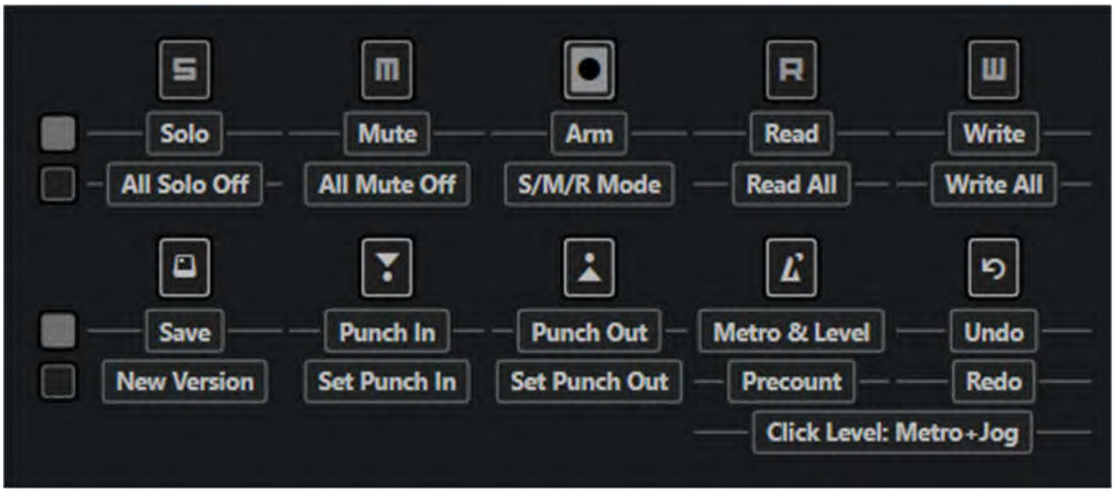
In the very first line of bindings, we have the default actions i.e., Rewind, Forward, Stop, Play, Record and Loop.

If we want to use the other ones, we have to press and hold Shift, Ctrl, Alt or FN, based on the line we want to select.

For example, in order to move to the next marker, we should press Shift + Forward.

Another example is to set left locator to the current cursor position. We should press FN + Rewind, since the "Set Left Loc" resides in the final line of our mapping.

Daw Commands Section



The **S/M/R Mode**, which is toggled by using **Shift** + **Arm** refers to the two modes of Keylab's track select buttons, **SINGLE** and **MULTI**:

SINGLE MODE

In this mode we're using the track select buttons to change the selected track of our current bank.

If we want to solo/unsolo, mute/unmute, arm/unarm our track we have to press the Solo, Mute, Arm button respectively.

Upon changing track, Solo/Mute/Arm leds are updating accordingly.

MULTI MODE

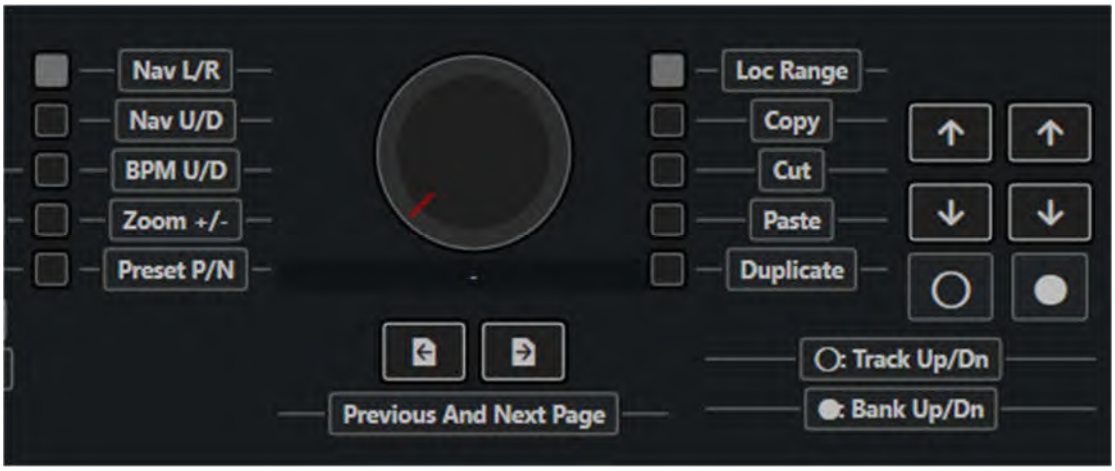
In this mode, we can use the **Solo, Mute, Arm** buttons, to change the actions of our select-track buttons.

Solo: If we enable Solo, (its led turns on), we're using our select buttons to Solo/Unsolo tracks. The color of the buttons changes accordingly (Blue=Solo, Off=Not Solo)

Mute: If we enable Mute, our select buttons are used to Mute/Unmute tracks. The color of the buttons changes accordingly (Green=Mute, Off=Not Mute)

Record: Similar to the previous ones, we're using this one, to record-arm multiple channels (Red = Arm, Off = Disarm)

JOG SECTION



Here we have two sets of actions.

The one on the left is for the jog wheel, while the other one on the right is for the jog push.

The action described in the very first line of each set is performed when we move our wheel or push it.

If we use Shift, Ctrl, Alt or FN we perform the actions of each corresponding line.

For example, if we want to change BPM, we can do it by holding down CTRL and turning our jog wheel.

Another example, if we want to Duplicate our selected event, we have to hold down “FN” and push our jog.

Note that depending on which page we are on, the very first line of actions is altered in ordered to perform tasks suitable for this page.

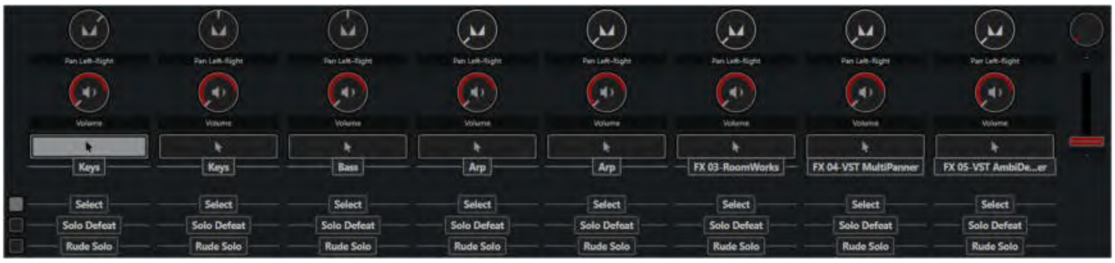
Finally, at the right end of the Jog section, we see four buttons. These correspond to the **Part1/Next** and **Part2/Previous** buttons of the Keylab with and without the **Live/Bank** button pushed.

Again, depending on the selected page, these four buttons perform different tasks.

For example, in mixer mode, the first two are used to move through the tracks of a bank, while the other two are used to move through 8-tracks banks. This is the same with the default Arturia’s DAW implementation, and I kept it this way, cause it’s great.

In other pages, their action is usually set to changing sub pages (when available) and changing banks of the selected plugin.

Mixer Section



In this section, we have different actions based on our selected page.

In Mixer page (as expected) we use faders/knobs/buttons to change volume/pan/selected track.

By using **Shift + Select Buttons**, we set the corresponding channels to **Solo Defeat**. Don't forget to **Shift + Select** again to deactivate the Solo Defeat state when no longer needed, otherwise you won't be able to solo/mute this track.

Ctrl + Select Button sets the corresponding channel to **Rude Solo** (i.e., mutes all other channels, and sets the selected one to solo).

Alt + Select Button activates/deactivates **Monitor** Mode.

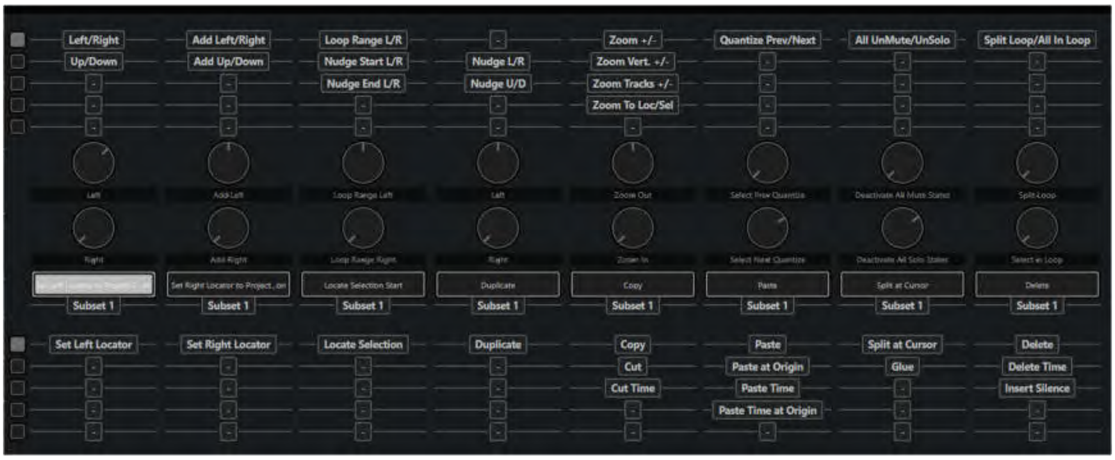
FN + Select Button activates/deactivates **Listen** mode.

Furthermore, we have the 9th Keylab's fader and knob. Depending on our general settings, these are assigned to the currently selected track or to the master channel. We can toggle this assignment on the fly, by holding Shift while moving our fader or knob.

Generally, in all pages, we can see the assignment of each fader, knob and button.

Furthermore, we can see (if available) assignments to our Shift, Ctrl, Alt and FN states.

For example, here's a screenshot of the Commands' Set 1 page:



As previously written, by using our four state pads, we can perform different actions.

In this example, if we want to "Paste At Origin" we have to press Shift + Button 6.

If we want to "Nudge Up/Down", we have to hold down Ctrl and move our 4th knob.

General Settings

In the file **mapOfGeneralSettings.js**, we have the general settings which are used by the script.

You can modify any entry you wish.

I’ve included comments to show what every setting does. You can alter their values to 0 or 1, based on your preference.

Transport section commands

In the file **mapOfTransportSectionCommands.js**, we can alter the actions of our transport buttons (i.e., Rewind, Forward, Stop, Play, Record and Loop) **ONLY** when we are using the state buttons (Shift, Ctrl, Alt and FN). Please see the comments inside this file for details.

Commands Sets

As written earlier, I have three pages of commands' sets. You can alter the commands binded, by editing the file **mapOfCommands.js**. Please follow the instructions provided in the file's comments.

Since this is a script which was built based on my personal workflow, I have setup some Macros and Logical Editor Presets, that are used in these command sets, as well as factory commands.

You can obviously edit these entries and setup your very own.

However, if you want these commands, you have to add them manually.

Inside the scripts folder, I have placed two subfolders **Logical_editor_presets** and MIDI_Logical_editor_presets, containing the logical presets xml files, I use in my implementation. Feel free to import these to your own presets folder if you like them and/or wish to alter them.

But, of course, you then have to create the corresponding macros. This means that you have to manually edit your commands xml file unfortunately, but it surely can be done with not too much effort (hopefully).

IF, however, you wish to try, you can try the instructions inside the file **"Instructions for importing my own logical editor presets.pdf".**

If you need help on this, I'm sure that you can find it in the **Steinberg's forum** and relative FB Pages (I personally follow **Cubase Pro**, **Strictly Cubase** and **Cubase Academy,** but I guess there are other as well). There are great people in all these places willing to help (I will also try to help of course, If I see such requests, since I'm a member of all these places and I personally have been helped there too).

Knobs and States

By altering the file **mapOfActionsOfKnobsByState.js**, we can setup the behavior of our 8 knobs when Shift, Ctrl, Alt or FN is pushed, in all pages except from the ones of the Commands Sets.

However, if you don’t need this functionality, you can deactivate it by setting **generalSettings.allowCustomKnobsStates=0** in our **mapOfGeneralSettings.js** file.

We have three types of mapping for each knob and each state:

Normal MIDI CC. Upon turning a knob, we send its CC value to a CC we’ve set up in the mapOfActionsOfKnobsByState.js file.

Dual MIDI CC. In this one, upon turning a knob, we send a 127 to one of two CCs we have defined, based on our turning direction, i.e., if we turn our knob left or right.

Commands. Upon turning a knob, based on the turning direction, we execute a command assigned to either left or right turn.

Master Fader and Knob States

By altering the file **mapOfMasterFaderAndKnobBindings.js**, we can setup the behavior of our master fader and knob when Ctrl, Alt or FN is pushed, in the mixer page.

Note that this requires knowledge of the API and is not advised for beginners.

A word about PADS

In both FULL and Stripped-down options, pads are used strictly by my script to perform the desired actions.

Obviously, there will be times when we'll want to use our pads for finger drumming (or other stuff we have mapped through our user pages).

In order to do that, we just have to select a USER template from the 10 available by our controller so that we get the normal functionality of pads (if of course we have them set up this way in these templates).