Gig Performer 4 Mackie Control Unit Protocol Extension

Purpose:

This extension is designed to expand and simplify using an external control surface utilizing the Mackie MCU protocol with Gig Performer.

Requirements:

To function correctly there are four requirements:

1. Download the appropriate extension file for your platform (Windows or Macintiosh)
2. Download the required Panel template
3. Copy both to the appropriate directory for Gig Performer extensions
   1. On Windows systems this is C:/Users/Public/Documents/Gig Performer/Extensions
4. Launch Gig Performer and enable the MCU extension when prompted

Optional enhancements:

There are a several OSC templates available that replicate the MCU protocol. These can stand in as a substitute for a hardware MCU unit when a physical unit is not available, or as an adjunct display to use in combination with a hardware unit.

Open Stage Control is an open source OSC platform that runs on Windows, Mac, and Linux. Generally one would run the Open Stage Control server application on the same system as Gig Performer and then run OSC templates on any modern web browser on any device on the same local network. This can be used to control Gig Performer by a mobile phone, tablet, or another PC.

Once Open Stage Control is launched and configured the relevant MCU templates can be loaded through its web interface.

An open source TouchOSC template is also available that replicates a physical MCU unit.

Basic Theory of Operation:

The Gig Performer MCU extension allows:

1. assigning Gig Performer widgets to the knobs, faders, and buttons on the hardware
2. entering and exiting Setlist mode
3. selecting among songs in a setlist, and song parts within a song
4. changing between Rackspaces, and Variations within a Rackspace
5. adjusting tempo with the Jog Wheel

As is customary with many DAWs, the MCU extension works using the concept of “banks” of controls. While there are only 8 knobs and 9 faders on the MCU hardware, by using “bank selects” the user can assign any number of knobs and faders to the MCU and move between them by changing banks.

An example would be to assign one bank of faders to the upper register drawbars on a Hammond B3 VST, a second bank to the lower register drawbars, and a third bank to master volumes in a mixer.

When a Rackspace is loaded the MCU extension searches through all widgets in the Rackspace and Global Rackspace. It identifies widgets to be used by the MCU extension by their “OSC/GPScript Name”.

The naming convention for such widgets uses four components separated by underscore characters. A typical such name would look like “mc\_k\_pan\_7”. The underscores serve as separators between fields.

The first field is the “mc” designation, which indicates the MCU extension should use this widget. If the widget name does not begin with “mc” the extension will ignore it.

The second field indicates the type of control on the control surface the widget should be mapped to. These are:

* k – knob
* f – fader
* rec – buttons on the Record row
* sel – button on the Select row
* solo – buttons on the Solo row
* mute – buttons on the Mute row
* fn – button on the Function row
* view – button on the Views row

The third and fourth fields indicate the BankID and Column number for the widget. The Column numbers range from 0-7 for buttons and knobs, and 0–8 for faders. Fader 8 is mapped to the master fader on the MCU.

A fader widget given the Name “mc\_f\_volume\_0” will be assigned to the leftmost fader on the control surface, and be actively controlled while the BankID “volume” is selected. A fader widget with the name “mc\_f\_eq\_0” would be assigned to the same fader, but controlled when BankID “eq” is selected.

The BankIDs can be any arbitrary names that are meaningful to you.