APCmini Ardour Bindings

Albert Gräf <aggraef@gmail.com>, Dept. of Music-Informatics, JGU Mainz, Germany

This map, to be used with Ardour's Generic MIDI interface, and the accompanying Pd patch turn the AKAI APCmini into a useful DAW controller. The APCmini is a great little device, but it's designed as a performance controller for Ableton Live, not a DAW controller, so a substantial amount of translation is needed to make it useful with Ardour. Most of the translation actually happens in the Pd patch, so you need to hook up this patch to Ardour on one side and the APCmini on the other, rather than connecting Ardour and the APCmini directly. Setting this up isn't too hard, though, please check the instructions further below.

Basically, this map partitions the 8x8 button grid of the APCmini into a 5x8 grid at the top, which can be used to play notes (e.g., like drum pads), and a 3x8 grid at the bottom which is used to operate the solo/mute/rec buttons of the corresponding tracks in Ardour. When engaged, the latter will light up in green, yellow and red, respectively. The nine faders at the bottom of the device are assigned to the corresponding function depending on the selection in the **FADER CTRL** section (**VOLUME**, **PAN** and **SEND**, which apply to the current bank of tracks (faders 1-8) and the master bus (fader 9), respectively. **DEVICE** applies faders 1-8 to the plugin parameters on the currently selected track. The send/plugin number can be selected with the three launch buttons at the bottom (send/plugin 1-3, or 4-6 when SHIFTed). The up/down and left/right arrow keys are used to select the current bank of tracks (for **VOLUME**/**PAN**/**SEND**) and the current track (for **DEVICE**), respectively. The **FADER CTRL** and send/plugin buttons will light up in red and green, respectively, to indicate the current status. The topmost four buttons in the **SOFT KEYS** section also have some convenient bindings to Ardour's transport controls (stop/play/rec/loop, or start/end and previous/next marker when SHIFTed).

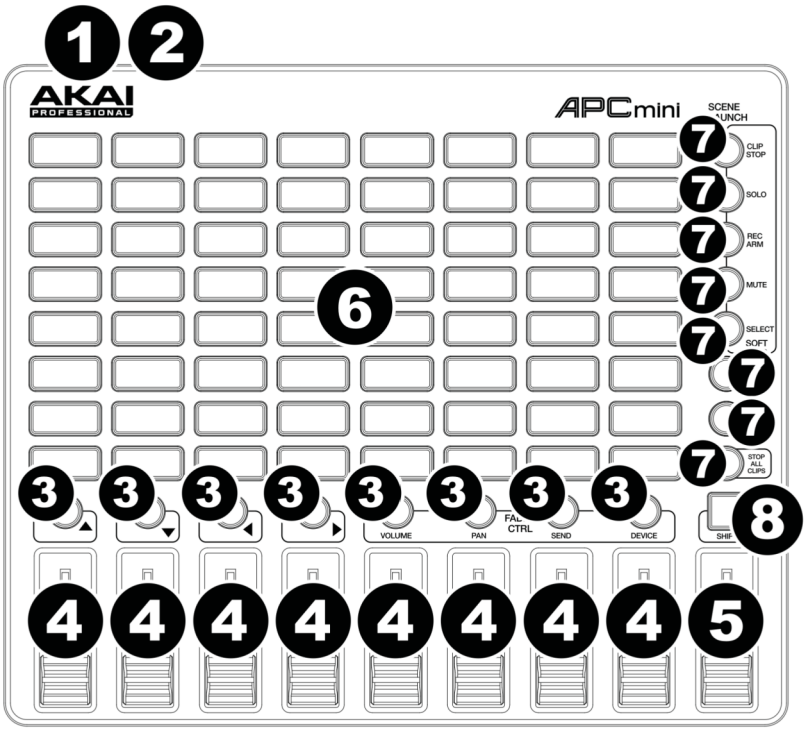
The **SHIFT** key in the bottom-right corner can be used to access alternative functions with some buttons (most of the buttons then start blinking to indicate this special status). As already noted, it can be used to access a second bank of sends/plugins (4-6) with the send/plugin buttons, and alternative transport functions with the topmost four buttons buttons in the SOFT KEYS section. It can also be used with the **DEVICE** key in order to access a second bank of plugin controls (9-16). When used with **VOLUME**, it selects the VCA (rather than track) volume controls. Together with **PAN**, it controls pan width (rather than pan direction). Finally, when pressed together with any button on the 8x8 grid, it sends a special control change message for the corresponding plugin control (CC *n* 127, where *n* is the column number of the pressed button; the row number is ignored, so you can use whatever row of buttons is most convenient). This is most useful in conjunction with controller *switches*, where Ardour expects a single controller value ≥ 64 to toggle the switch, which isn't easy to do with the faders, since they will usually send more than one value when moved.

One more thing: The **SELECT** button (fifth button in the SOFT KEYS section) can be used to kick the faders into a special gear where all fader movements (or SHIFTed pad clicks) are output to

Ardour on MIDI channel 16, which isn't used in the map file. They can then be assigned using Ardour's MIDI learn function, independent from their function in non-SELECT mode. Pressing **SELECT** again restores whatever FADER CTRL configuration is currently active; the current status of this option is indicated by a green light on the button if SELECT mode is currently active. Like DEVICE, this button can also be combined with **SHIFT** to give a second bank of freely assignable controllers.

As you can see, that's quite a lot of functionality crammed into this little device, but all in all I think that this assignment should be rather intuitive, at least after the initial learning curve. More details of the assignment and instructions for setting up can be found in the notes below.

Function Overview



1. USB port
2. Kensington lock
3. ▲: previous bank, ▼: next bank, ◄: select previous track, ►: select next track;
   VOLUME, **PAN**: faders control volume and pan in the corresponding track of the current bank;
   SEND: faders control send #1-6 in the corresponding track of the current bank;
   DEVICE: faders control parameter #1-8 of plugin #1-6 in the currently selected track
4. **Faders** control volume, pan, send or plugin parameters depending on the **FADER CTRL** settings (3)
5. The **rightmost fader** controls volume/pan/sends of Ardour's **master track**.
6. **5x8 grid** at top: sends note messages 24-63 (can be transposed with corresponding control in the Pd patch; output goes to Pd's first MIDI output);
   3x8 grid at bottom: solo/mute/rec controls for the tracks in the current bank;

8x8 grid, when **SHIFT**ed: sends a CC message for the control in the corresponding column (useful for toggling switches in **DEVICE** or **SELECT** mode)

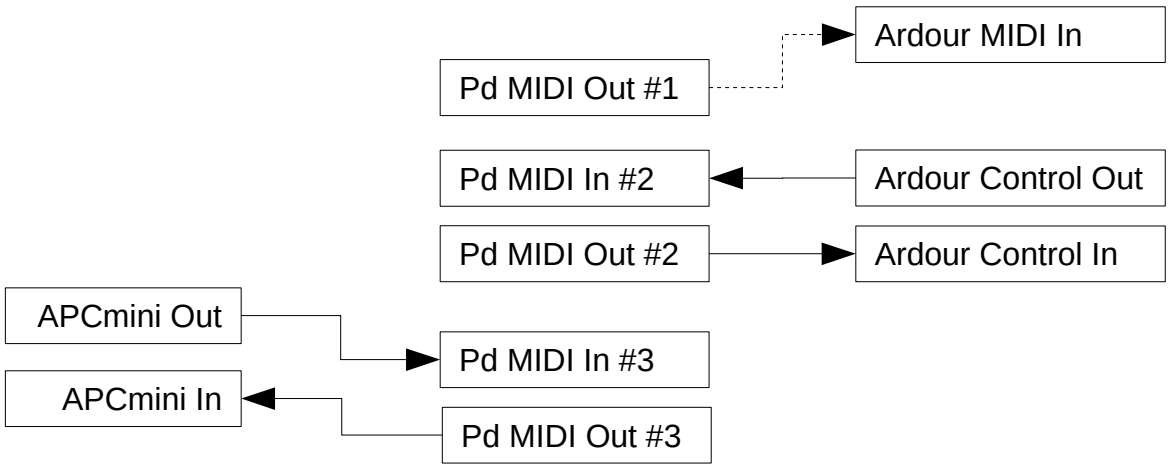
7. **SOFT KEYS** (5 buttons at top): the topmost four of these are assigned to the transport controls stop/play/rec/loop; the fifth button **SELECT** engages a special mode where all controller values are sent to a reserved MIDI channel and can then be freely assigned with Ardour's MIDI learn function; **3 buttons at bottom**: select send or plugin #1-3 in **SEND** and **DEVICE** modes
8. **SHIFT**: when pressed, alters the behavior of the 8x8 grid (6), the **VOLUME**, **PAN** and **DEVICE** buttons (3), as well as the transport buttons, the **SELECT** button and the three send/plugin selection buttons (7) as follows:
   SHIFT + **8x8 grid**: pads send CC messages for the corresponding control;
   SHIFT + **VOLUME**: controls VCA (rather than track) volume;
   SHIFT + **PAN**: controls pan width (rather than pan direction);
   SHIFT + **DEVICE**: selects a second bank of plugin controls (#9-16);
   SHIFT + **SELECT**: selects a second bank of freely assignable controls (#9-16);
   SHIFT + **send/plugin** button: selects a second bank of sends/plugins (send or plugin #4-6 instead of #1-3)
   SHIFT + **transport** buttons: second set of alternative transport functions (start/end and previous/next marker)

Note that the VOLUME, PAN, DEVICE, SELECT and send/plugin buttons start blinking when SHIFTed, to give a visual indication of their special status.

Setup

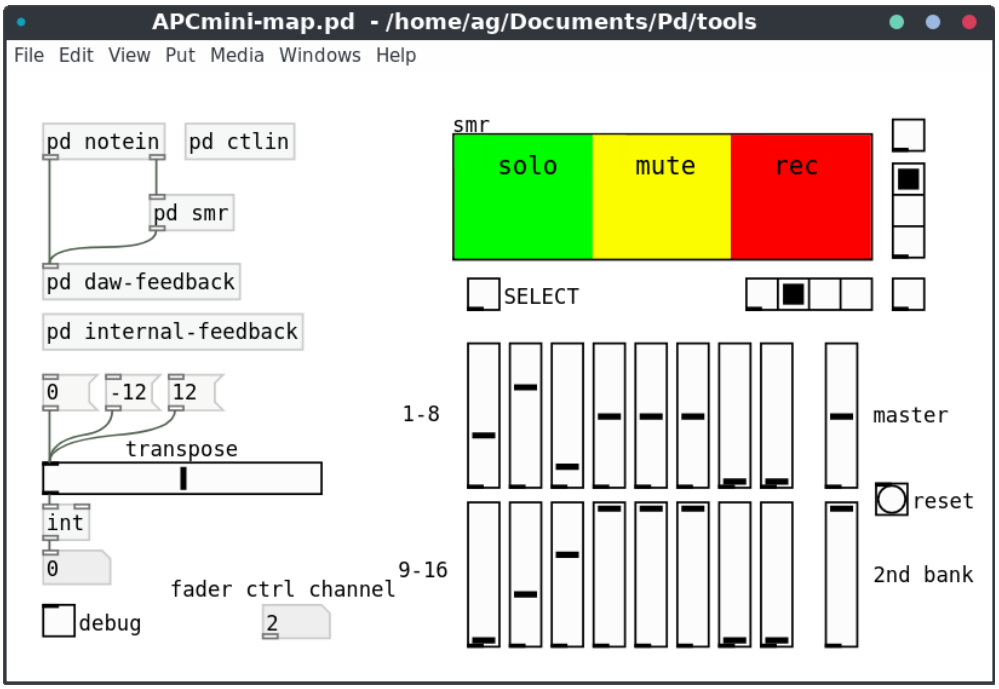
For starters, you need to copy the AKAI_APCmini.map file to the appropriate subdirectory in your Ardour configuration folder, so that it appears in the dropdown list of the generic MIDI control surface configuration in Ardour's preferences. On Linux this is usually ~/.config/ardour5/midi_maps, or /usr/share/ardour5/midi_maps for system-wide installation; please consult the Ardour manual for details.

The map also requires the accompanying Pd patch APCmini-map.pd which actually implements most of the mapping; you can put this anywhere you want. The Pd patch is to be connected to Ardour on MIDI I/O port #2 and to the APCmini on port #3, so your Pd setup needs to provide at least 3 pairs of MIDI inputs and outputs. Here is a quick rundown of the required connections (the outgoing note connection from Pd #1 is optional):



Use your MIDI patchbay to configure the bidirectional connection between Pd's MIDI I/O port #3 and the APCmini, as well as Pd's MIDI I/O port #2 and Ardour's MIDI control inputs and outputs. You might also want to hook up Pd's MIDI output #1 to Ardour's MIDI track inputs as needed, if you want to use the upper 5x8 grid of pads for playing notes. Then enable Ardour's generic MIDI control surface, load this map, make sure that feedback is enabled, and you should be set. (This has only been tested on Linux so far, but I'm sure that the multiple MIDI I/O port setup can be done with Pd on macOS and Windows, too. It should be more or less the same procedure with some loopback/virtual MIDI devices and a suitable MIDI patchbay at hand.)

The patch also provides a basic GUI, depicted below, to give feedback about the current status of the solo/mute/rec buttons (smr array), the send/plugin and FADER CTRL selection along with the corresponding shift status, and the positions of the faders. Note that these elements are only used for display purposes. In particular, the sliders in the GUI make up for the fact that the APCmini doesn't have motorized faders. To find out which faders certain plugin parameters are assigned to, just put the APCmini in DEVICE mode, wiggle the controls in Ardour and observe which faders move in the GUI. The same method works in SELECT mode, to find out which fader, if any, is assigned to a control via Ardour's MIDI learn facility.



In addition, the transpose slider lets you shift the notes output with the upper 5x8 grid (default is 24-63 from bottom left to top right). Useful transposition presets are -12 (1 octave down, which puts the usual drum range starting at note 36 in the two top rows of the 5x8 grid) and +12 (1 octave up, which puts the drum range in the two bottom rows). Finally, the debug toggle, when enabled, causes messages sent between Pd and Ardour to be reported in Pd's main window, which – you guessed it – is useful for debugging purposes.

Caveats and Bugs

The APCmini-map patch was created with Purr-Data a.k.a. Pd-l2ork 2.4, thus the formatting will be somewhat off in vanilla Pd, but it should mostly work there, too. (Vanilla might complain about a missing closebang object, but that only means that the lights on the APCmini won't go off automatically when the patch is closed.) Another thing worth noting is that I've zoomed the patch

for better readability on high-DPI displays, so you may want to turn on the option “save/load zoom level with patch” in Purr Data’s GUI preferences to have the patch displayed as intended.

Also, note that the patch heavily relies on Ardour’s feedback for showing the right fader values and other status information, so you should make sure that feedback is enabled when configuring the controller. Sometimes the Pd GUI may still be out of sync with Ardour, and won’t reflect the actual values until you operate the controls either on the APCmini or in Ardour. Even reloading the session doesn’t seem to cure this, at least with the Ardour version I’m running at the time of this writing (5.12). However, you *can* force Ardour to resend all the current fader values by going to the generic MIDI controller setup in Ardour’s preferences and reloading the APCmini map. (It seems that sometimes it’s enough to just display that dialog, otherwise try switching the protocol to “Reset All” and then back to “AKAI APCmini”.) If the fader display still seems out of whack, try pushing that little “reset” button to the right of the faders to reset them all to zero before reloading the map.

Another minor obstacle arises when tracks are in a group so that they can only be selected together. This interferes with our bindings for the plugin controls, which will only allow you to manipulate controls on the *first* selected track. The only way around this is to edit the track groups in question in order to disable the sharing of the selection status, so that the individual tracks can be selected.

Do you have any comments, ideas for improvements or (preferably) actual code? Mail me at aggraef@gmail.com or toss me a pull request over at Github!