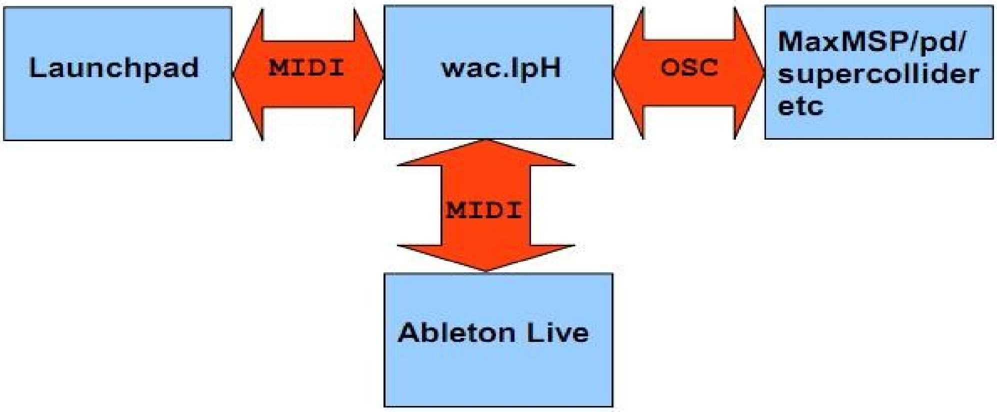
wac_launchpadHandler

A utility for adding an OSC communication layer on top of the default integration offered between a Novation Launchpad and Ableton Live.

The utility (wac_lpHandler) is a 'patch' written in Cycling 74's MaxMSP and can be run under Windows or Os X via the free Max Runtime environment. The utility's functionality is best described diagrammatically:



wac_lpHandler acts as a router and OSC wrapper, passing MIDI between Ableton Live and the Launchpad. It also creates eight additional 'virtual banks' that send and receive OSC formatted data for signalling button pushes and allowing control of the LEDs on the Launchpad surface. These processes run concurrently, with wac_lpHandler able to independently manage up to six devices.

The software is intended to simplify interfacing with the Launchpad by reducing its (complex) MIDI protocol to a set of simple human-readable OSC commands.

Included Files

maxProgrammingExample.maxpat

MaxMSP patch to demonstrate using wac.lpHandler. This should be your first port of call, as the entire OSC protocol used is explained through examples. There is also a simple synth for you to play with the Launchpad.

wac_lpHandler_v1_1_0.maxpat

MaxMSP patch; this is the wac_lpHandler utility, and is loaded in the MaxMSP environment

wac_lpHandler_v1_1_0pd.maxpat

Specialised version of the patch that uses Olaf Matthes' netsend/netreceive objects for MaxMSP. This allows OSC communications with pureData (pd) and some other environments

wac_lpH_v1_1_0.js

javascript file used by the utility. All the data handling/manipulation occurs in here; this can be edited in any text editor if required

wac_launchpadOSCcomms.pd

example pure data patch for demonstrating OSC communications

defaultWacLphSettings.json

File for storing your default wac.lpHandler settings – once saved (via the 'save as default' button in the utility) your MIDI port and network port assignments wlil be automatically recalled on launch.

wlph_basicSynthVoice~.maxpat

Used by the synthesiser in the maxProgrammingExample.maxpat patch

changelog.txt

overview of changes made since the original 0.3 version

Supporting Materials Required

MaxMSP

A full copy or the free Runtime version
http://www.cycling74.com/downloads/max5

netsend/netreceive objects for MaxMSP[Optional]

These are included in the download as they can no longer be obtained from Olaf's site. They are required if you want to interface with Pure Data.
After installing MaxMSP the *netsend/netreceive* files must be unzipped and moved to an appropriate folder that is 'within MaxMSP's searchpath' (see Max MSP manual for an explanation)

- On windows I use:

C:\Program Files\Cycling '74\Max 5.0\Cycling '74\my_Externals

Internal MIDI bus

Allows communication between MIDI applications on the same computer
Os X: uses the IAC bus (included with the operating system)
Windows: third-party application required. I use MIDI Yoke
http://www.midiox.com/index.htm?http://www.midiox.com/myoke.htm

Pure Data [Optional]

Allows the example pd patch to be run. Used for testing/verifying OSC communication layer is working
Freeware environment suitable for developing OSC applications
http://crca.ucsd.edu/~msp/software.html

Installation And Setup

1) Install Max Runtime (or the full version)

2) Unzip the wac_ lpHandler folder and move it to a location on your computer that is"Max's search path"

3) Unzip and place the *netsend/netreceive* externals in a suitable location within Max's search path

4) Under Os X, ensure the IAC bus is enabled. Under Windows, download, install and configure MIDI Yoke (or equivalent internal MIDI bus).

Reference

If you want to change functionality of *wac_lpHandler* , edit the *wac_lpH_v1_1_0.js* file using any text editor and relaunch *wac_lpHandler*. Alternatively, if you own a full version of MaxMSP the source code can be altered to suit your needs.

N.B. The patch includes send and receive objects (called [s OSC_from_lp] and [r OSC_to_lp]) for direct interfacing with MaxMSP. Simply open the patch (within a bpatcher or as an abstraction) within Max and add the corresponding [r OSC_from_lp] and [s OSC_to_lp] objects to your patch.

A programmers reference guide for the Launchpad is available from:
http://www.novationmusic.com/support/viewdownload?download=127

You do not need to do any Max programming to use this. Functionality can be changed by editing the included javascript file. The Cycling74 guide for using javascript in Max should be helpful:
http://www.cycling74.com/docs/max5/vignettes/js/javascriptinmax.html

Software Setup

wac_lpHandler Overview

The wac_lpHandler patch is used to do the following:

- Set up MIDI ports for communicating to/from each Launchpaddsa
- Test communication with each Launchpad via the 'Local Loopback' function
- Set up internal MIDI ports for communicating with Ableton Live and allowing the Launchpad's default functionality/integration to be utilised
- Set the network ports used for bi-directional OSC communication to other software (running on the same computer). By default wac_lpHandler sends on port 8195 and listens on port 8196
- Store and these settings and recall them on subsequent use

wac_lpHandler extends the default integration of the Launchpad with Ableton Live by providing an additonal eight banks of OSC sending buttons and OSC controllable LEDs.

The state of the LEDs on each of these eight OSC banks is internally stored, providing the ability for the Launchpad to generate/respond to OSC messages for 576 uniquely addressed grid buttons/leds

OSC Communication Overview

Open Sound Control (OSC) is a flexible network protocol that is finding increasing usage in music applications. It is transmitted over network infrastructure via udp.

The full protocol understood by wac_lpHandler is detailed in the **OSC Protocol** section at the end of this document.

There are two versions of the wac_lpHandler patch included. They are functionally identical but use two different methods for realising udp communication.

wac_lpHandler_v1_1_0.maxpat

Uses the stock udpsend/udpreceive objects included with MaxMSP. Enables udp communication with MaxMSP, Supercollider, Processing, etc.

wac_lpHandler_v1_1_0pd.maxpat

Uses Olaf Matthes' netsend/netreceive objects for enabling udp communication with Pure Data. May work with other environments too.

You should use the version of the patch appropriate for the software you intend to use the Launchpad with.

Step-by-step guide for use with ONE Launchpad

Setting Up And Accessing OSC Mode

1) Ensure your Launchpad is connected to the computer

2) Load the *wac_lpHandler* with MaxMSP

3) Under 'Launchpad 0' select the correct MIDI in/out ports for your unit from the drop-down menus (typically the port will be called "Launchpad" but this may vary system-to-system)

4) Click the 'Local Loopback' checkbox

5) Press the 'user1' and 'user2' buttons simultaneously on your Launchpad to enter OSC mode

If your MIDI IO is correct the two buttons should light up orange to show you are now in OSC mode. With 'Local Loopback' enabled, buttons pressed on the grid will light up green. Uncheck Local Loopback when you are satisfied.

The eight buttons along the top of the Launchpad are used to access the eight virtual OSC banks. A red LED indicates the current bank.

Pressing 'user1' and 'user2' simaltaneously again reverts back to the default 'Ableton Mode'.

Whilst only one 'user bank' can be focused at any one time by the Launchpad, OSC messages sent to every user bank (to set an led state) are internally stored by wac_lpHandler. When switching user banks the Launchpad surface will be updated to reflect the latest 'state' for that bank.

Testing OSC Communication With Pure Data

1) Load the included Pure Data patch

2) Load the special wac_lpHandler_v1_1_0pd.maxpat version of the patch

3) Set-up MIDI communication between the Launchpad and wac_lpHandler

4) Check the 'Network Connection (on/off)' box in wac_lpHandler

5) Watch the Pure Data window to see button presses being registered

6) Use the Pure Data patch to toggle some LEDs on the Launchpad

Ableton Live Mode

Control of Ableton Live is available by default on loading wac_lpHandler, and can be 'banked into and out of' via simultaneous presses of the 'user1' and 'user2' buttons. The behaviour of the Launchpad with Ableton Live is in no way changed by wac_lpHandler

1) Select an appropriate 'MIDI to Live' port (within wac_lpHandler) for sending MIDI to Ableton Live

2) Select an appropriate 'MIDI from Live' port (within wac_lpHandler) for receiving MIDI from Ableton Live.
    These two ports MUST be different to avoid MIDI feedback.
    For example, send on *MIDI Yoke: 1*
    receive on *MIDI Yoke: 2*

3) Make sure you are in the 'Ableton Mode' on your Launchpad (press user1/user2 simultaneously until both buttons are unlit)

4) Under preferences in Ableton Live, set up a Launchpad control surface using the internal MIDI ports that compliment those selected in wac_lpHandler
    For example, set the control surface's in port to *MIDI Yoke: 1*
    Set the control surface's out port to *MIDI Yoke: 2*
    Do not use the Launchpad's direct hardware ports for the control surface, or wac_lpHandler will be bypassed

If you have your ports set up correctly the 'red ring' should appear in Ableton Live, confirming a connection has been made between Ableton Live and the Launchpad *through* wac_lpHandler

Once a connection has been established you can swap between Ableton/OSC modes as required and the Launchpad surface will update accordingly.

Using Multiple Units

Up to six Launchpads can be managed through wac_lpHandler

The 'local loopback' test should prove invaluable for ensuring you have selected the correct MIDI IO ports for each Launchpad.

The OSC messages sent/responded to each Launchpad are distinguished by the 'device' number in the OSC string.

To use six Launchpad's with Ableton Live, six distinct send/receive internal MIDI bus pairs must be used (e.g. IAC buses 1-6 for sending, IAC buses 7-12 for receiving) between Live and wac_lpHandler.

Care should be taken that EVERY Launchpad is in 'Ableton Mode' when setting up each of the six control surfaces in Ableton (as Ableton refreshes its 'handshake' with ALL control surfaces everytime a new control surface is added/removed).

Enjoy
Questions/comments/suggestions/etc
willycwillydo@gmail.com

wac_lpHandler OSC Protocol

Sends OSC on udp port 8195
Receives OSC on udp port 8196
#1 = device ; Supports 6 devices (0 -> 5)
#2 = bank ; One Ableton bank (0) and eight OSC user banks (1-8)

Commands From lpHandler

/wac.lp/#1/#2/press x, y, state

; button presses from the main grid
; x/y, indexed from top left = 0 -> 8
; state (off/on) = 0 / 1

/wac.lp/#1/#2/bank

; sent after changing bank; #2 indicates the new bank

/wac.lp/#1/hardwarebank state

; state (off/on) = 0 / 1 reports hardware banking status after change

Commands To lpHandler

/wac.lp/#1/#2/led x, y, green, [red], [flash]

; address main grid leds
; x/y, indexed from top left = 0 -> 8
; green (brightness) = 0 -> 3
; red (brightness) = 0 -> 3 [optional]
; Any greater-than-zero 5th element makes that led flash

/wac.lp/#1/#2/show x, y, green, [red], [1]

; change an LED display state as per *led* message but value is not internally stored

/wac.lp/#1/#2/recall x, y

; outputs stored value for specified grid location (restores state after a 'show' message)

/wac.lp/#1/#2/set x, y, green, [red], [1]

; as *led* message where new value is stored but not shown

/wac.lp/#1/#2/refresh

; refresh led states if #2 is the current bank
; *set//refresh* together should be used to simaltaneously update >20 leds
; internally this allows use of the Launchpad's 'rapid update message'
; this is more efficient than sending 20+ individual LED messages

/wac.lp/#1/hardwarebank state

; state (off/on) = 0 / 1 toggles the ability to switch banks via buttons on LP

/wac.lp/#1/#2/bank

; change to a new bank, where bank = #2
; performs the same action as changing banks on the top row of LP buttons

/wac.lp/#1/#2/clear

; clear storage/display for the specified device and bank