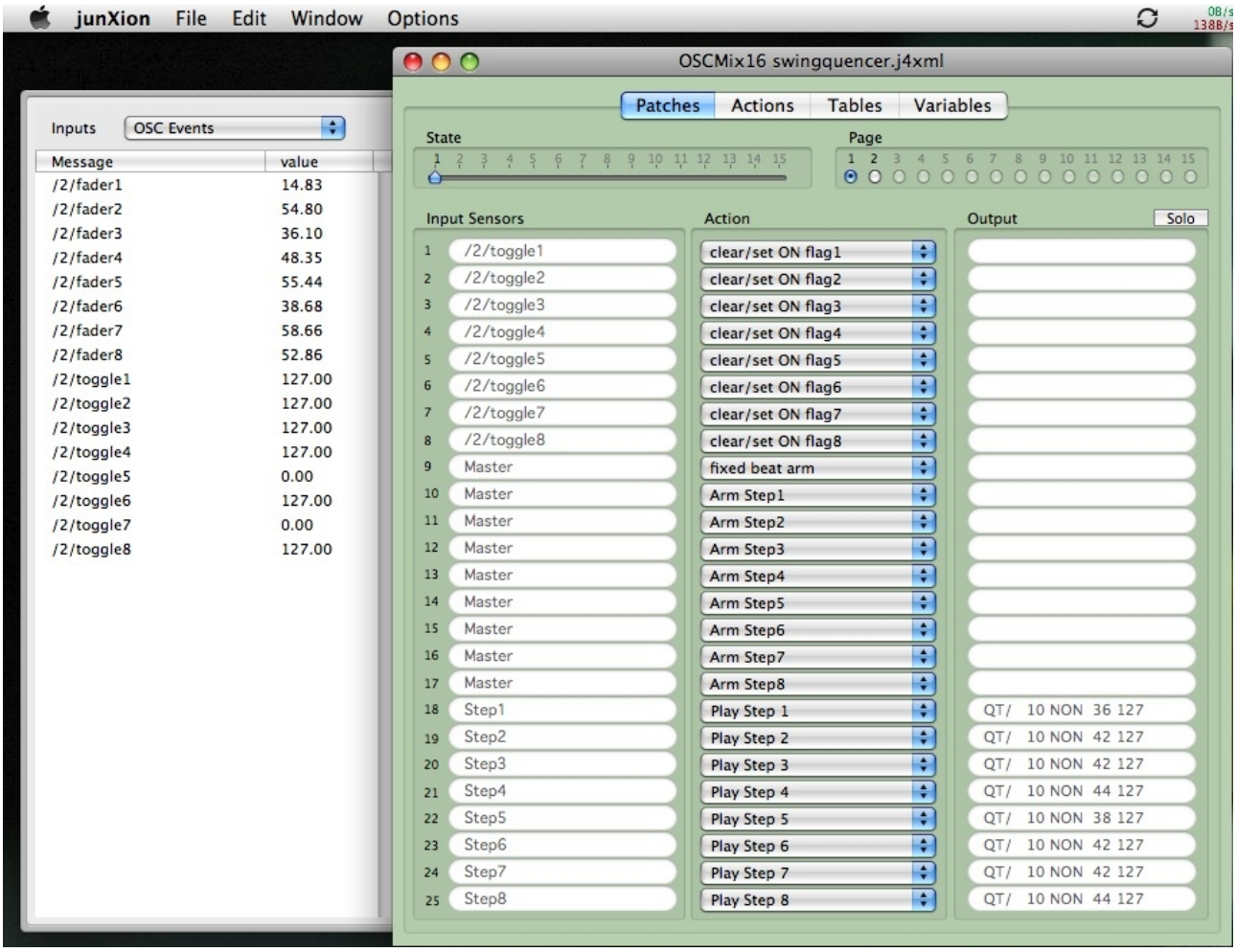
junXion v4

Realworld **T**ool **F**or **M**usicians



© 2008, Steim Foundation

www.steim.nl

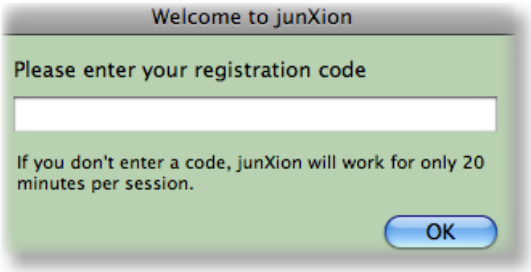
1. Welcome to junXion v4

JunXion v4 is a powerful musical tool developed at STEIM in Amsterdam. It allows you to read, translate, map, reroute and process incoming data from the following sources and convert it into outgoing MIDI or OSC data.

- HID (Human Interface Device) class devices such as Joysticks and Game-Pads, Touch-screens, or the STEIM junXion Board.
- MIDI data from MIDI Keyboards, Control Surfaces and Sequencers.
- Internal data generators called Timers.
- OSC (Open Sound Control) data from Computers or Mobile Devices in a Network.
- Nintendo Wii Remotes
- Audio Inputs, Pitch and Level Data
- Video Inputs, Color and Object Tracking
- Arduino Sensor Board Input

JunXion v4 can be seen as a user friendly programming environment in which you can define the behavior of your physical input devices. junXion v4 employs a programming model where all tasks are performed with the graphical user interface. Entering code or 'scripts' is not necessary.

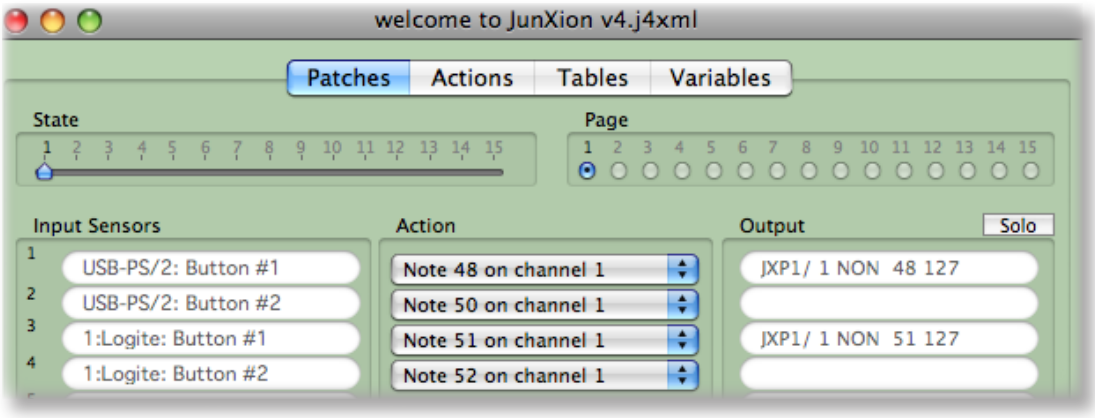
Entering your registration code



The first time you start up junXion v4 you will be asked for a registration code.
If you have received a registration code, copy it from your email into the text field and click OK. If the code is accepted your name will be visible in the 'About junXion' window which is accessible from the 'junXion' menu. To try the program in **demo mode** click OK without entering anything. In demo mode you can use the program up to 15 times for no longer than 20 minutes at a time. Within these restrictions the program works exactly as for registered users. You are permitted to save your work and open it again later.

The junXion v4 programming model

JunXion v4 will work on any Apple Macintosh computer with Mac OSX 10.4 or later and a PowerPC or Intel processor running at 1GHz or higher. A junXion v4 Configuration stores all of the settings for a particular setup, which consists of a number of individual Patches. It is saved in XML Format (Extensible Markup Language) with the extension '.j4xml'. A Configura-



The Overview in the Patches Window

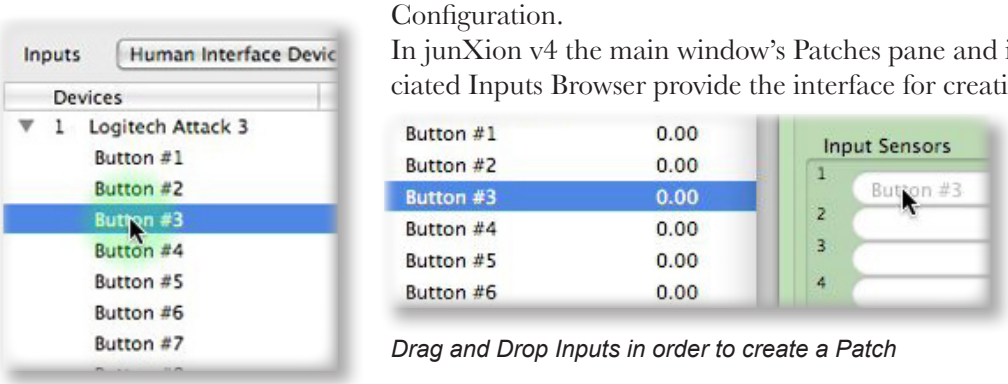
tion contains up to 15 States each of which can include up to 375 Patches. JunXion v4 is able to open older v3 Configurations (extension .jxml) as well.
A Patch associates an Input Sensor with an Action which is responsible for processing the input data. Actions can generate MIDI or OSC output, update internal Variables or Tables, or switch to a different State of junXion v4. Actions can execute conditionally and can also perform detailed data transformation.
The main junXion v4 window consists of a tab view containing four panes named **Patches, Actions, Tables,** and **Variables**. To the left of the main window is the Browser which shows information related to the currently displayed pane.
When the Patches pane is displayed - which is the overview pane - the Browser shows a list of Inputs. Items from this list can be dragged into the 'Input Sensor' column of the Patches pane to create a Patch. The Action column allows the selection of the target Action for each Patch, in which processing including Tables and Variables can be applied. The 'Output' column displays output generated by the patch. Actions, Tables and Variables can be edited on their respective panes.

2. Background

JunXion v4 has been created by Frank Baldé and Michel Waisvisz at STEIM, the Studio for Electro Instrumental Music in Amsterdam.
This studio has today gathered 40 years of experience in the development of electronic musical instruments for instant composition. The direct relationships of the physical, touching and bodily input of a musician into the instrument and its resulting sounds have always been a key challenge here in order to avoid a somewhat too logical or 'offline' approach to electronic music composition. In this manner junXion is designed for musicians who would like to experiment with new and artistically innovative interfacing options, and who share the opinion that a spitting image of a piano keyboard is not the ultimate aproach to today's Samplers and Synths. The development of junXion v4 is historically based on the research on the **STEIM SensorLab** since the late 1980s, one of the world's first 'Real World To MIDI' translators in the form of a mini computer. It could read out analog and digital sensor data and transform it into digital MIDI data, programmed in a scripting language called 'Spider'. The first instrument which has used a prototype of the SensorLab was Michel Waisvisz' famous gestural MIDI controller 'The Hands', as early as in 1984. When Sensor hardware became more and more affordable in widespread commercial game controllers of the 1990s and Laptops became rather powerful and universal, the development of the junXion software eventually replaced STEIM's research on a dedicated hardware for the purpose of reading out and mapping Sensor data.

3. Patches

'Patch' is the name given to a connection between an Input Sensor and an Action within a junXion configuration. When the data of an Input Sensor changes it triggers the Action(s) connected to it. These Actions may result in MIDI or OSC output, update of an internal Variable or Timer, trigger a second 'daisychained' action, or switch to a different State of the program. By using multiple Patches one Input Sensor can be connected to a number of Actions. A number of Input Sensors can also be connected to one same Action. The first case is probably the most useful as it effectively allows you to use your Input device in various ways within one Configuration.
In junXion v4 the main window's Patches pane and its associated Inputs Browser provide the interface for creating and



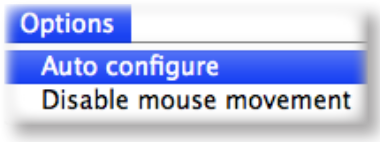
Drag and Drop Inputs in order to create a Patch

modifying patches. Patches are created by dragging Input Sensors from the Inputs browser into empty slots in the Input Sensor column of the Patches pane. Normally a new Action is created for each new Patch.
It is important to realize that junXion v4's Input Sensors will vary in 'resolution', depending on the type of Input sensor. Some HI device's sensors will have a resolution of 8-bit, meaning they can generate 256 different values, others can be 12-bit, thus may generate 4096 different values. The Audio Events are 16-bit, i.e. 65356 different values. In order to work with all this data in a managable way, junXion v4 scales down this raw input into the range 0.000000 - 0.99999999, does all its processing (depending on your Patches) in 32-bit mode and either upscales it to MIDI range (0-127) or leaves it as it is for OSC output. For you as the user however, the data is represented according to the 'user-defined' data range, which by default is from 0 - 127. This range can be set in junXion v4's Preferences.

A ctrl click on the Input Sensor will show you its resolution

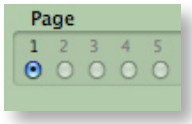
Working with patches

In order to create a Patch simply drag an Input Sensor from the Input Browser onto the 'Input Sensor' column of the 'Patches' pane. You can create multiple patches simultaneously by dragging a whole device or by selecting multiple sensors (using the standard shift and cmd key combinations) and dragging them as a group. This is useful when you want a HI device to spit out MIDI data without doing too much of editing, but simply retrieve some MIDI data in another application.



Selecting 'Auto Configure' in junXion v4's Options menu will automatically create Patches for all connected HI devices on startup, so that you'll immediately get a MIDI Output from all your Inputs. This option is saved in the Preferences when quitting the application.
Normally a new Action is created whenever you drag an Input Sensor into an empty slot. However, if you hold down the Alt / Option key on your Mac's keyboard while dragging, no new Action will be created and the Input Sensor will be connected to the first Action in the Actions list. You can change the Action connected to an Input Sensor at any time by selecting a different one from the pop-up menu in the 'Action' column.
junXion v4 offers maximum flexibility with respect to linking Input Sensors and Actions. You can drag the same Input Sensor into multiple slots to connect one Input Sensor to a number of different Actions. You can also connect an Action to multiple Input Sensors.
Whenever you drag an Input Sensor to a slot containing an existing Patch junXion v4 will replace the Patch's Input Sensor. If you hold down the ctrl-key during this operation, a pop-up will ask you if you want to replace ALL instances of this overwritten sensor.
Patches can be removed by selecting them with the mouse and choosing 'Delete' from the Edit menu, or by using the cmd-Backspace combination on your Mac's keyboard. Patches can be selected by clicking on their slots in the Input Sensors column. Multiple Patches can be selected in the usual way using shift-click to select a continuous group or cmd-click for a non continuous group. You can also use the usual copy and paste operations with selected patches. Keep in mind that deleting a Patch does not delete its corresponding Input Sensor and Action. The Input Sensor will still be in the Inputs menu and the Action will still be in the Actions list.

Pages

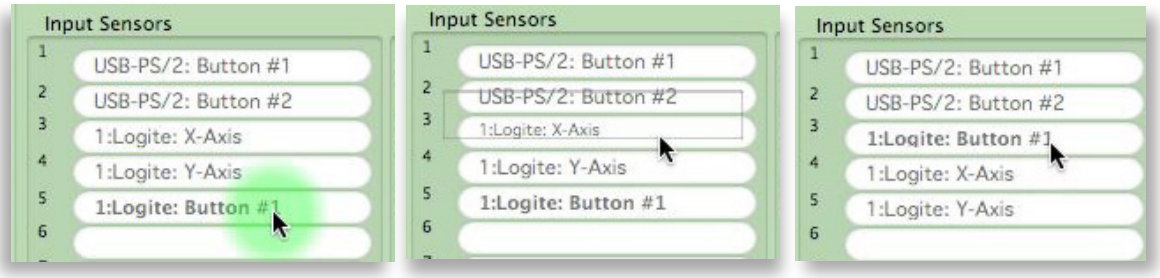


The Pages Button

As soon as the 25th Patch is created junXion v4 will enable Page 2 on the Page selector which is visible at the top right of the Patches pane. Clicking on the page 2 button provides an additional 25 empty slots.
Dragging a group of Input Sensors into the Input Sensor column will automatically create the Patches on new pages if needed.

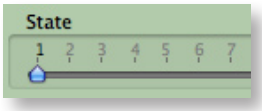
Patch ordering and reordering

It is important to be aware that junXion v4 processes Patches in order, one at a time, starting with Patch 1 and continuing sequentially through to the last Patch, top to bottom. This process takes place 1000 times a second. Indeed, in some cases the order in which patches are processed can be important. For example a Configuration which only generates MIDI Aftertouch from the joystick's Y-axis if a certain button is pressed should process the button Patch before the Y-axis Patch.To change the order of a Patch you can drag it to a new location with the mouse. Make sure that you begin and end the drag in an Input Sensor slot. As you drag the mouse the slot into which the Patch will be inserted is displayed in a smaller font size. If you drag one Action to a different location there will be no resulting gap but the Actions in between will shift accordingly. You can also drag Patches into other pages by moving the mouse cursor over the corresponding page button while dragging. This will make junXion v4 display that page. You can then move the mouse to the slot where you want to move the Patch.
When dragging patches remember that the whole Patch is moved, including both the Input Sensor and the Action it is connected to.



Reordering Patches

States



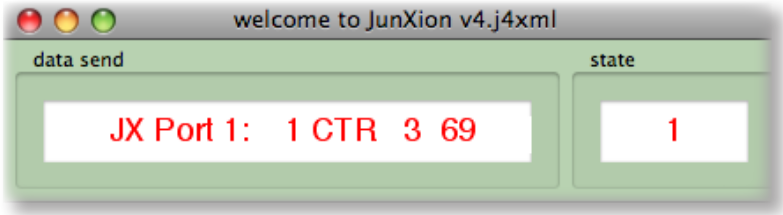
The State Slider

The State slider can be found at the top left of the Patches pane. The slider allows selection of one of 15 possible States, each of which can contain a collection of up to 375 Patches (15 pages with 25 Patches each). At any given time only the Patches in the currently selected State cause Actions to be triggered. Each State can be configured to treat your sensor data in a completely different way.
States can only be filled in serial order. It is not possible to use State 1, have empty States 2, 3 and 4 and then make another collection of Patches in State 5.

In addition to switching states using the State slider, state changes can be triggered by Actions, in order to jump to different 'presets' of your Configuration without looking at the screen. One of an Action's possible Outputs is called State Change, which allows you to use an Input Sensor - most likely a switch - to jump to another State. See the Actions chapter for more detail.

Visual feedback and the Output column

The Output column of the Patches pane allows you to monitor what is being sent by each of the 25 visible patches. The information displayed includes an abbreviation of the MIDI or OSC Output Port name, the channel you are sending on, the event type (NON = Note event, PKP = Poly Key Pressure, CTR = Continuous Controller, PGC = Program Change, PRS = Aftertouch and PBD = Pitch Bend) and its data.
The Output column is updated 25 times each second, which can create a considerable processing load. Although this load



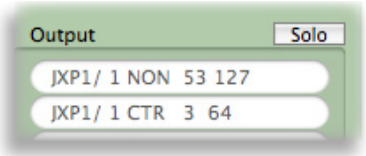
The Status Window

does not affect junXion v4's low level data processing kernel, it may affect other applications running simultaneously.

A simple way to reduce the processing load is to click on the main window's zoom (green (+)) button. This shrinks the main window to a small status window which displays only the last sent MIDI event and the current State.This is probably the best way to use junXion v4 whenever you have created a Configuration and are just using it to control other MIDI programs or devices. To return to the large main window, simply click on the zoom button again.

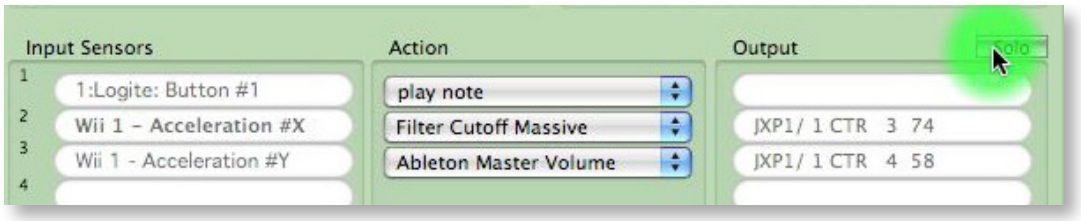
It is also possible to completely close junXion v4's main window without any problems. In this case junXion v4 will run in the background and will use minimal processing power. You can find it back in the Window menu.

The Solo Function



The Solo Button (Patches Window)

Many Software Instruments and Sequencers which you may want to use in connection with junXion offer a so-called MIDI-learn option, which helps to easily assign a MIDI controller to one destination parameter. This is useful when you have a discrete MIDI dataflow. Performing with a multiple sensors device like a 3D accelerometer or a Videotracker, you'll notice that it's almost impossible to only send only the one controller value which you actually want to connect. To avoid a constant overwriting of the MIDI-learn assignment junXion v4 offers a function to single out one output by clicking on its Input Sensor in the Patches Window and then activating the Solo Button. The Solo Patch will be shown in a bold font, like the second Patch in this example:



Singling out one Output of a Configuration

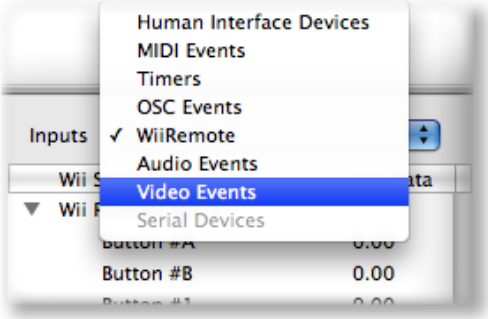
4. Inputs

JunXion v4 supports the following Inputs:

USB **HID** (Human Interface Device) class devices such as Joysticks and Game-Pads, Touchscreens, or the STEIM junXion v4 Box.

Video Inputs, Color and Object Tracking

MIDI data from MIDI Keyboards, Control Surfaces and Sequencers.



The Input Browser in the Patches Window

Internal data generators called **Timers**.

Nintendo **Wii** Remote

Arduino Board Support (Serial Device)

Audio Inputs, Pitch and Level data

OSC (Open Sound Control) data from computers or mobile devices in a network.

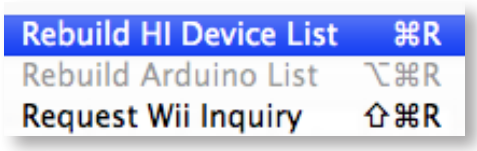
Human Interface Devices

Choosing Human Interface Devices from the Inputs pop-up menu in the browser displays a list of the available HI Devices. Clicking on the disclosure triangle to the left of each Device's name toggles the display of a sub-list listing each of the device's sensors. Note that you can limit the amount of displayed sensors of each HI device in the Preferences.

Display of a Joystick's raw data

The second column of the list displays the current value of each sensor. The values shown are the 'raw' values, scaled to the preferred data range which you can set in the junXion v4 preferences. By default this is the MIDI data range between 0-127 (7 Bit). In general, switches will display their two states as 0 and 127, sensors such as potentiometers or the X-Axis of a Joystick will deliver a continuous data stream. Some sensors have a higher resolution than others, their raw data will be scaled to floating point numbers within the desired data range without actually losing resolution.

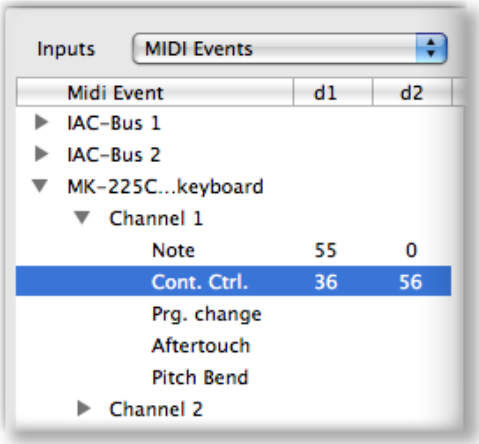
You can discover the resolution of any Input Sensor by control-clicking (or clicking the right mouse button) on the sensor's name in the list. A small yellow note will pop up telling you the sensor's range.



Rebuilding the HI Device list

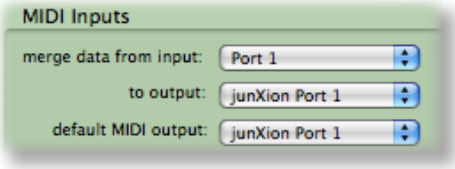
Connecting one or more HI-Devices while junXion v4 is running is not a problem. They will be shown in the Input Browser after rebuilding the HI Device list in the file menu or pressing 'cmd+R' on your Mac's Keyboard. As you can see, this also applies to Arduino Serial Devices and Wii Remotes.

MIDI Events



MIDI Events in the Input Browser

Choosing MIDI Events from the Inputs pop-up menu You can set your default MIDI Ports and also merge ingoing MIDI data through junXion v4. See the ‚Preferences: Midi Input' Chapter for a more detailed description of this feature.



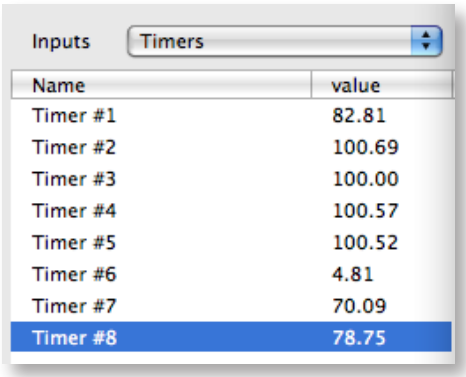
Merging MIDI Inputs

displays a list of available MIDI ports. Clicking on the triangle to the left of each Port's name toggles the display of a sub-list containing the Port's 16 MIDI channels. Each Channel's triangle again toggles display of the five different MIDI event types that can be used as MIDI Input Sensors in junXion v4.

The special case with some MIDI events is that junXion v4 has to deal with two bytes of data, called d1 and d2. For 'Note events' for example both d1 and d2 are displayed, as the data consists of one byte for the note number and another one for the velocity; the same goes for 'Cont. Ctrl' and 'Pitch Bend' sensors. For 'Aftertouch' and 'Prg. change' only d1 is needed and displayed.

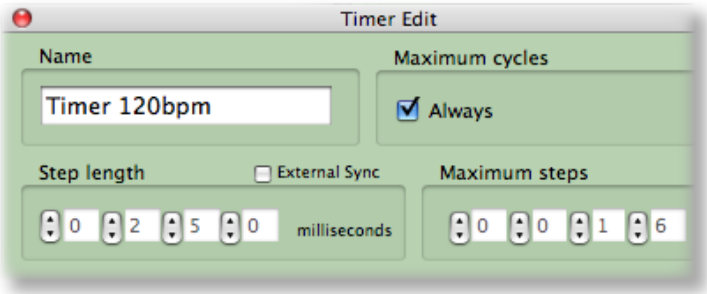
The d1 and d2 columns monitor the most recently received value for each MIDI Input Sensor. If no data has been received nothing will be displayed. All MIDI input values are in the range 0 - 127. If a Patch is created with MIDI Input sensors, the connected Action allows you to choose either d1 or d2 as its data. See the Actions chapter for more info.

Timers



JunXion's default Timers, doubleclick to edit

Choosing Timers from the Inputs pop-up menu displays a list of internal Timers. Timers can be used to create autonomous processes which are not directly triggered by sensor input.
They can be used for many purposes, some examples include: scheduling note off messages after a certain duration, switching to another State after a period of time, and automatically changing the behavior of an Input Sensor if there is no activity for a certain period of time. Another use for Timers is to trigger a sequence of note events in the manner of a classic step sequencer. Note pitches could then be controlled by an Input Sensor or read from a Table, or even another Timer. Timers allow to fire note events and control data very fast or regularly, they can serve well for experimental setups with non static systems. The number of Timers is determined by a setting in junXion v4's Preferences. By default eight Timers are active, a maximum of 100 is available.



Rename and edit the Timer

To edit a Timer's settings, double click on a Timer's name in the list. The Timer Edit floating window will open, and everytime a new timer is selected, the window will adjust to display its settings. The Timer Edit window allows the following settings to be modified.

Name

You can change a Timer's name by clicking in the Name text field and editing it. The new name becomes effective when you hit the 'Enter' or 'Return' key on your Mac's keyboard.

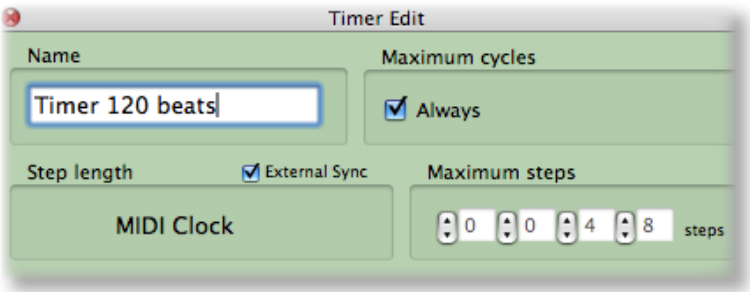
Maximum cycles

Maximum cycles determines the number of times the Timer will cycle from 0 to the maximum number of steps before stopping. A maximum cycles value of one will cause the Timer to stop after the first time it reaches the maximum value. Checking the 'Always' checkbox will cause the Timer to run continuously in a loop as long as junXion v4 is on. The range is from 1 to 999 cycles.

Step length

Step interval time sets the time (in milliseconds) between each increment of the Timer's value. For example setting the interval time to 500 milliseconds will cause the Timer to increment its value twice a second. The range is from 1 to 9999 milliseconds.

External Sync

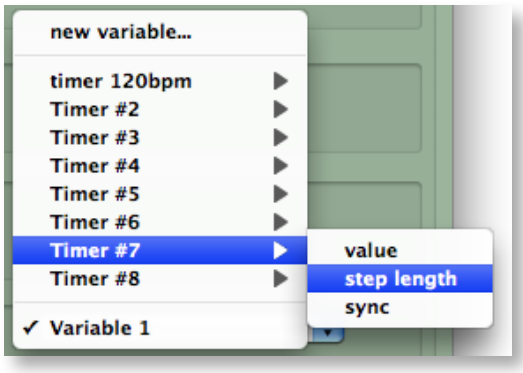


It is possible to link the step length to an external MIDI clock signal. Simply check the checkbox above the 'step length' field. As junXion is not really a Sequencer Application the Timing cannot be set with one click, but needs to be implemented and

calculated within the settings of a Timer. It is nessecary to understand how MIDI clock works, and accordingly set your Timer's number of steps. junXion v4 will not process the external sync information as a sequencer program would do. But the use of Timers can turn junXion v4 into a generator of sequences which can be referred to by a bpm value. MIDI clock is a sequence of pulses which other machines can synchronize to.
For Example:
250 ms - 4 steps will result in a Timer loop of one second.
120 bpm midi clock - 48 steps will result in the same Timer loop.
At 120 bpm, MIDI clock fires 48 clocks/ second.
Note that you will manually have to set the 48 steps in this example.
Accordingly, for any other bpm value the MIDI clock calculation for the Timer's steps is:
x bpm/ 120 * 48 = amount of midi clock pulses/second, for x bpm

Maximum steps

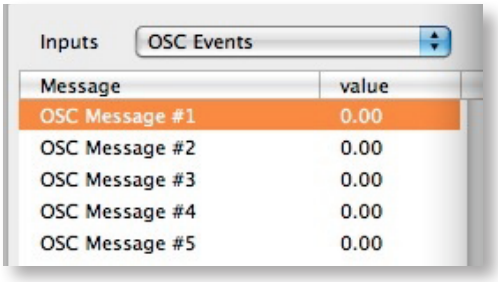
Maximum steps determines the highest step number in a cycle. In each cycle the timer will count from 0 to this value. This effectively determines the resolution of the timer, and with 9999 steps as the maximum this gives a little more than 13 bit resolution. Together with the Step Length time it also determines how quickly the timer cycle completes.
The Timer's list continuously displays the current value of each Timer's internal data variable. An important feature is that these values are accessible as data variables by Actions, and can also be modified by them, for example to reset their value to 0 in order to restart a Timer's cycle. The name of the data variable is the same as the name of the Timer. When the name of a Timer is changed, the new name will also be shown in the Variable's pop-up menus in the Action pane. Once a Timer is defined some of its variables can be changed or reset by Variables in the Actions. These Variables cannot be renamed, because they always refer to one determined parameter.
The three Timer variables 'value', 'step length', and 'sync' (not the same as 'external sync'!) are presented in the Action's Variable popup menus as follows:



Setting Timer Variables in Actions

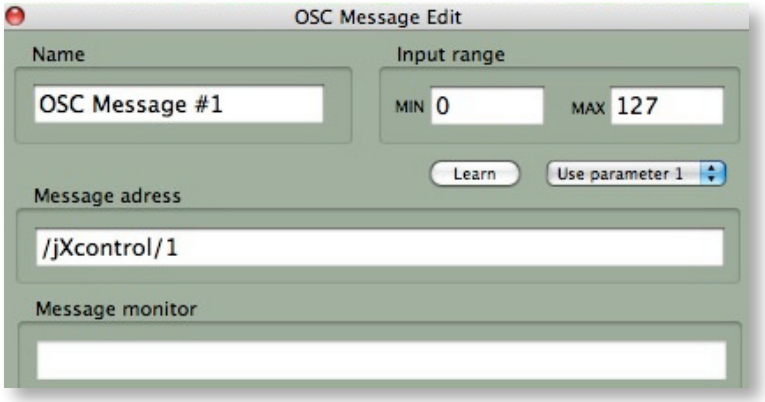
You can choose to use the selected Timer setting its current value, its 'step length' value or its 'sync value'. The 'value' variable sets the timer to a certain value, 0 for example resets it to the beginning. The 'step length' variable contains the value of the Timer's Step Length. This allows you to control the tempo of the Timer loop. The 'sync' value refers to the microtiming of the timers. It is there mainly to be used to reset to zero, since this is the way to precisely synchronize a number of Timers: store a zero into their respective 'sync' variables and connect these Actions to a single Input Sensor like a switch.

OSC Inputs



OSC Events in the Input Browser

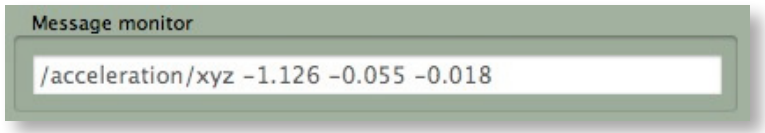
JunXion v4 can read (and also output) OSC messages, if you've set the number of OSC Inputs to other than zero in your Preferences. A maximum of 100 messages can be processed. The implementation of the OSC Message Edit window allows the user to define which parameter to use for the message (if more than one parameter is received at once). Select the OSC Message Edit window by double-clicking on a message in the OSC Input browser.



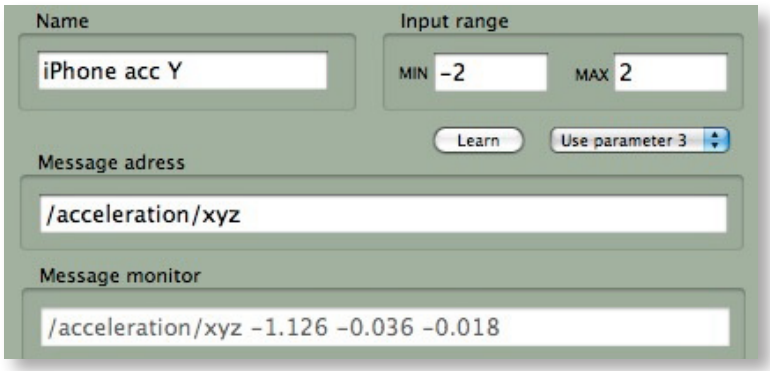
OSC Message Edit Window

In this Edit Window you can access different parameters of an OSC Event, rename it, set the Input Range, and monitor its Output.
Some more info on OSC in general and how it is handled in junXion can be found in the Chapter about OSC Output in this manual.

Start sending your OSC message...

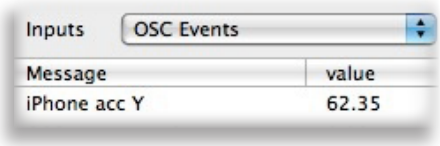


This message tells you that 3 parameters are received: x,y,and z.



The easy way of configuring the OSC message now is to click on the **learn** button, in the above example the 3-parameter message of the iPhone's Accelerometer will be learned, and can be adjusted to your needs: give this message a proper name for your own reference, set the **input range** and select the **parameter** this message needs to use (here either parameter 1, 2 or 3). The Input Range settings determine how your incoming OSC data is scaled and clipped. In the above example, the iPhone's acceleration parmaters actually vary from -3 to +3, by setting the range from -2 to 2 you can use the accelerometer rather as a tilt sensor than as an accelerometer. You may also notice that this raw input range is rescaled in the browser display according to your defined settings.

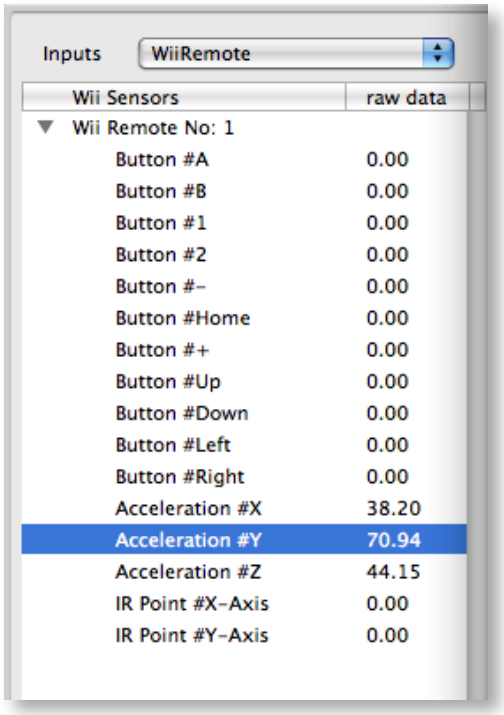
Eventually the OSC Input browser shows something like this, you can drag this Input an use it in your Patch:



! Remember that junXion will not be able to process OSC strings or blobs however, it is limited to the usage of OSC float or integer messages.

Nintendo Wii Remote

JunXion v4 supports the Wii Remote controller, which comes with the Nintendo Wii game console. This controller is kind of special because it uses a three-dimensional accelerometer sensor as its main sensor. When junXion v4 recognizes a Wii controller to be present, in the Inputs menu on top of the sensor browser the menu item named Wii Remote will be enabled. Selecting



Successfully connected Wii Remote in the Input Browser

the Wii Remote menu in the Input browser will show you all available sensors; they can be assigned in Patches just like any other input sensor.
In order to connect the Wii Remote execute the following steps. This description is for Intel-based Macs, for Power PCs see below.

Connect Wii Remote when starting junXion v4 (Intel-Mac):

1) Make sure **Bluetooth is turned on** on the computer (System Preferences - Bluetooth).
2) Start junXion v4.
3) Make sure that in junXion's **Preferences** the maximum amount of Wii Remotes is set to something else than 0.
4) Press **button 1 & 2** on the Wii Remote simultaneously until LEDs start blinking.
5) Wii Remote is connected when **one LED is on** and WiiRemote will be enabled in junXion's Inputs menu.

Connect Wii Remote when junXion is running, but Bluetooth is off:

1) Turn Bluetooth on.
2) In the File menu select Request Wii Inquiry menu item or press Shift-Cmd-R.
3) On the Wii Remote press button 1&2 until LEDs start blinking

Disconnect Wii Remote:

1) Hold Wii 'Power' button pressed until LEDs switch off (recommended if Wii should be connected again later)
2) junXion v4 will continue looking out for a Wii Remote, to reconnect press 1&2 again

Wii Remote, Trouble Shoot:

Wii is connected (LED is on, not blinking), but Wii is not activated in 'Inputs' menu

1) Check if Bluetooth indicates a connection (striped line through Bluetooth sign or in Preferences)
- If YES: Select the menu item Restart Wii Inquiry in junXion v4's File menu
- If NO: check if Wii connected to other system (e.g. to a neighbor who also runs junXion v4)

2) Try to disconnect Wii through holding 'Power' and reconnect by pressing 1&2
3) If still not activated restart junXion v4

Wii is activated in junXion v4, but not all data is read (maybe LEDs are off)

Disconnect Wii Remote by holding 'Power' and reconnect by pressing buttons 1&2

Wii is not connecting to junXion (LEDs start blinking, but turn off)

1) Check if Bluetooth is turned on (if not follow the steps described above)
2) Select the menu item Request Wii Inquiry in junXion v4's File menu and try to connect again
3) If Bluetooth was turned on and off a lot, it often gets stuck, in that case the best thing is to reboot the computer
4) Check batteries of Wii Remote

Nintendo Wii Remote on PowerPC based Macs

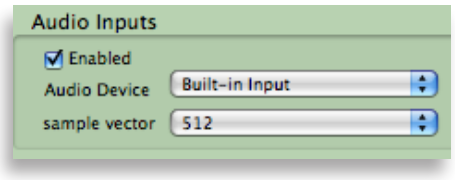
Although Bluetooth usage on PowerPC systems is not without problems, the WiiRemote is also supported on the older machines. However, be aware that it is not very stable and may sometimes be the cause of unexpected results (which is not the case on Intel systems).

Make sure you always initialize using the following steps before starting up junXion v4:
1) Open System Preferences, choose Bluetooth
2) Select the 'Devices' tab.
3) Delete 'Nintendo RVL-CNT-01' (this always needs to be done on a PowerPC machine!)
4) close the System Prerences
5) Start junXion v4
6) Manually perform the Wii Remote request in the 'File' menu.

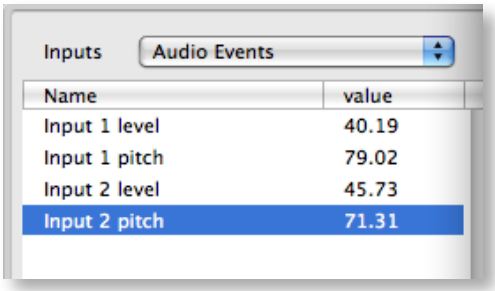
Then continue as on IntelMacs:
1) Press button 1 & 2 on the Wii Remote simultaneously until LEDs start blinking.
2) Wii Remote is connected when one LED is on and WiiRemote will be enabled in junXion v4's 'Inputs' menu.

Audio Events

JunXion v4 is able to use the input signals from any connected or internal Audio Device as an input sensor. First you need to choose which Audio Device you want to use by selecting it in junXion's Preferences window. Make sure the 'Enable' button is checked (you can always temporarily disable the audio input processing using this button, it will save processor power). Each input from the selected device will give you 2 sensors: **Input level** and **Pitch**. This means that if you have an 8 channel external audio device, you can use an extra 16 sensors from microphone or line input in the device. Devices can only be enabled entirely, which means that a multichannel Audio-Interface will cause a serious CPU load because many Inputs will be constantly checked at samplerate. The sample vector determines how many samples are calculated each time for level and pitch analysis. As a general rule: the smaller the vector the higher the CPU load but the faster the response time. The default setting of 512 is a good starting point.



Audio Input Preferences



Audio Events in the Input Browser

Once you have selected the input device and enabled its Audio Inputs, select Audio Events from junXion's main Inputs browser window.
The sensor values are represented in the user defined range (0 -127 default).
The pitch tracker is not created for absolute and accurate pitch tracking, more as a continuous control sensor, its pitch sensitive range is in between 80 Hz to 2000 Hz.

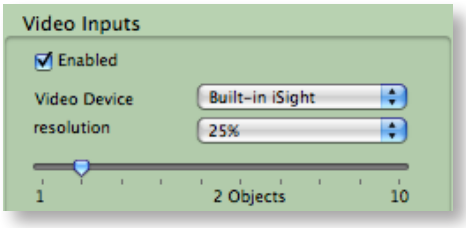
Video Events

JunXion v4 supports video input, as coming from the built-in iSight camera or an external USB or Firewire video camera. junXion v4 is able to analyze the image data and extract several input parameters out of it. This allows you to use a video camera to track one or more particular colored objects or to detect movement in a space. Each tracked Video Object will give you six differents 'events', like X-position, Width, etc. that are treated as any other Input Sensor.

There are a few things to keep in mind however:

JunXion can only track at a maximum 'sampling' rate in between 24 to 30fps, this is dictated by the camera. So virtual fast percussion players won't find any usage here.
Aditionally there is a little latency (although consistent) between your movement and the tracking, this is due to the fact that junXion already gets the original image with some latency, since the OS decoders have to uncompress the data and format it.
When trying to do color tracking in normal daylight you will have a hard time doing that, because daylight constantly fluctuates in color temperature, brightness, etc. However if you are working in a theater you can create perfectly controlled light, so in that case it can work very predictably.
The best way is actually to use 'active' light objects, because then you are not dependent on light reflection etc. A flashlight with a piece of color filter in front of it may give you a perfectly defined color object to track.

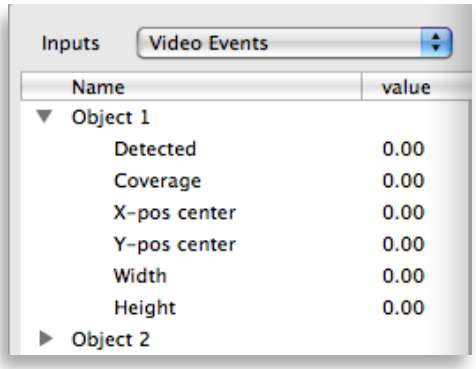
Enabling Video Input



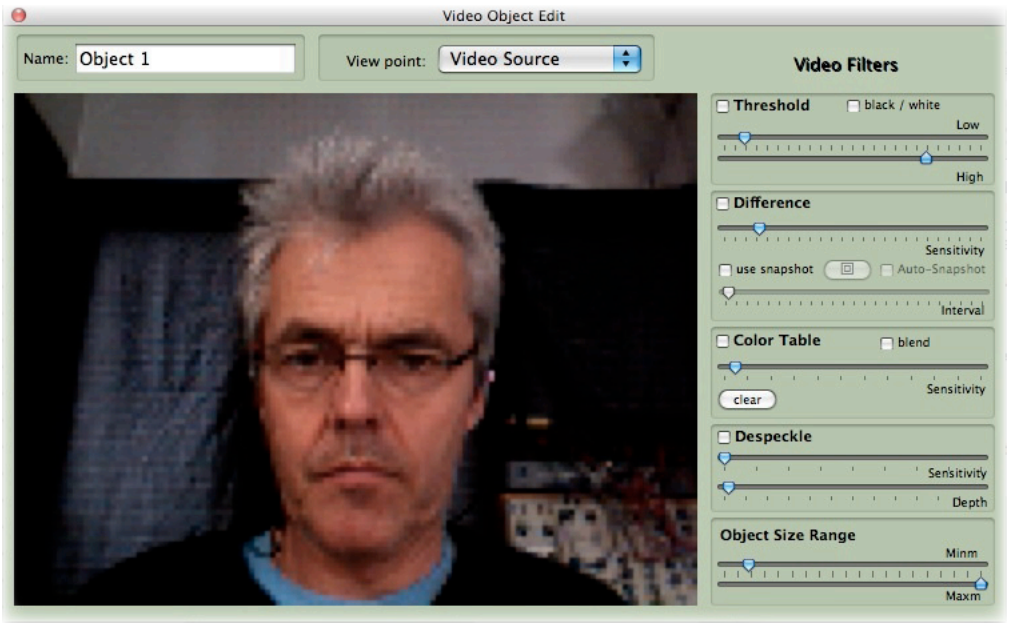
Activate a Camera in the Preferences

By clicking the Enable button (so that its checked) in junXion's Preferences, you are able to use the selected Video Device. For more details refer to the Preferences Chapter of this Manual.
In the Inputs menu you will now find the Video Events menu item enabled, select it and the Input browser will show you the available data streams.
Just like handling any Events in junXion, you can open the Video Object Editor by double-clicking on the object name.

Name (top left) allows you to give your video object a name, always a good idea to maintain a good overview in more complex Configurations.
The main part of the window shows the image from the selected video device, in the example screenshot you can see by the pixelation that the setting is to use 25% of its native resolution. The menu named **View point** allows you to select either the Video Source, one of the activated Video Filters or the Object Tracking.

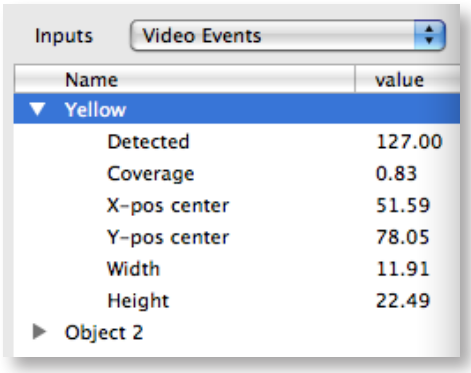


Video Events in the Input Browser



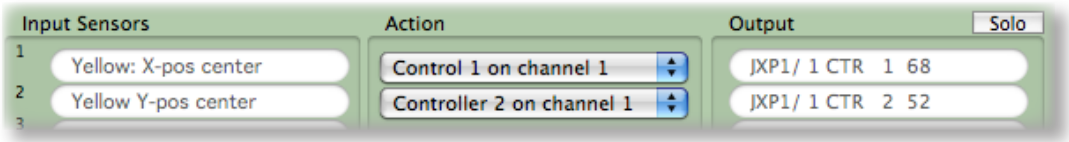
Frank Baldé in a low resolution

Object Tracking

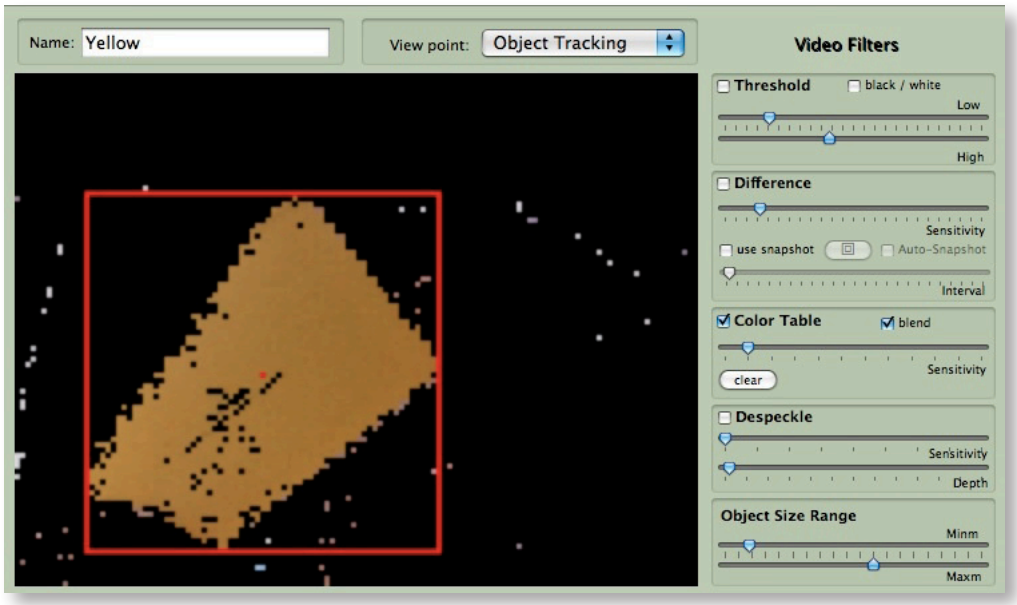


Renamed Video Object and its Sensors

In the following example junXion v4 is tracking a yellow piece of paper, the rectangle being the enclosing of the Object and resulting in the following Yellow object parameters displayed in the Input browser:
In the Input Browser you see that each Video Object generates 6 events, **Detected, Coverage, X-pos center, Y-pos center, Width** and **Height.** Those events can be used as Sensors just like any of the other junXion Input Sensors, so you create a Patch by dragging it onto the Input Sensors column in the Patches tab.
Detected is either on or off; on when an object is tracked, off if no object can be found. **Coverage** is a percentage of actually found matching pixels within your detected object (of which **Width** and **Height** give you the rectangular outline), and **X and Y-pos center** represent the x/y coordinates of the Object center in the Image space. All the values are scaled into your Preferred Data Range.



Dragged onto the Patch you can process and output the Video Object's data



The Object Tracking View point, tracking a yellow piece of paper

The basic premise of the Video Input tracking is to interpret a video image and to convert it into Video Input event (= Sensor) information according to specifications set up by the user. Interpretation is done on a frame by frame basis, the program retains knowledge of previous frames, so the system is able to interpret movement. The user has to specify which objects in the incoming video image are to be looked at and which parts of the image are to be ignored. This is done on the basis of the brightness and/or color attributes of elements in the image.
As you can see in the right part of the Video Object Edit window, there are **four types of video filters**. The central idea behind these image filters is to single out those objects in a scene which are important, i.e. filter out the parts of an image that you don't want junXion to interpret. In the end, when junXion attempts to track objects, the only thing that counts is which pixels in the image are 'on' and which are 'off'. Off pixels being pixels that are absolutely black

and 'on' pixels holding any value other than absolutely black. Color or brightness information is no longer relevant at that stage.
Usually you only need one or two filters per object, if you want to track two different colors you need to create two different Video Objects in the Preferences. Each Video Object may have its own Video Filter settings, described now in more detail:

Threshold

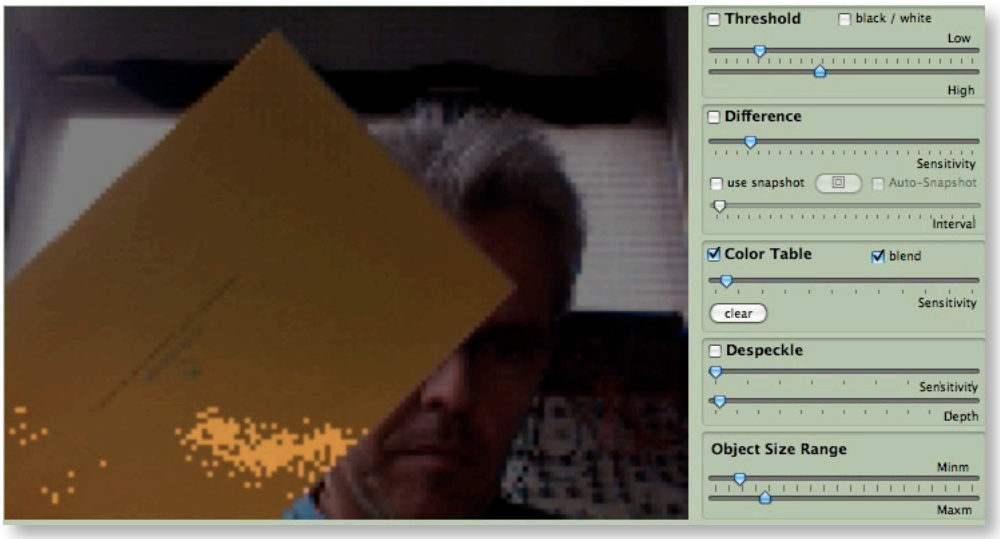
The threshold filter is used to single out objects through their brightness value, light hands against darker clothing and background, for instance. If you have a black and white camera or digitizer the threshold filter replaces the color table filter as the most useful. It can be used with a color camera as well, of course. When selected, the threshold module shows two sliders in the edit area, one for the low threshold; values lower than this are forced to black, and one for the high threshold, which represents the highest brightness value which is not forced to black.

Difference

The difference filter gives you an instrument to pick only moving elements out of an otherwise quite complex scene, independent of color or brightness characteristics. The filter subtracts two consecutive frames from each other and thus retains only the differences between the two frames, which will be the edges of moving objects. With this filter you can really only track movements, beause as soon as the image is *still*, there are no differences any more between the new and previous frames. You can however enable the 'use snapshot' button, which allows you to make a snapshot of the *now* with a mouseclick. From now on all the movement will be compared to this snapshot and give you difference data. When this is enabled you can also enable the auto-snapshot feature, which lets junXion automatically make a new snapshot every new time interval as specified with the Interval slider.
Last but not least there is the Sensitivity slider which forces to black brightness values of the difference below a certain value. Just slide it up and down a bit to see the effect it has in your application.

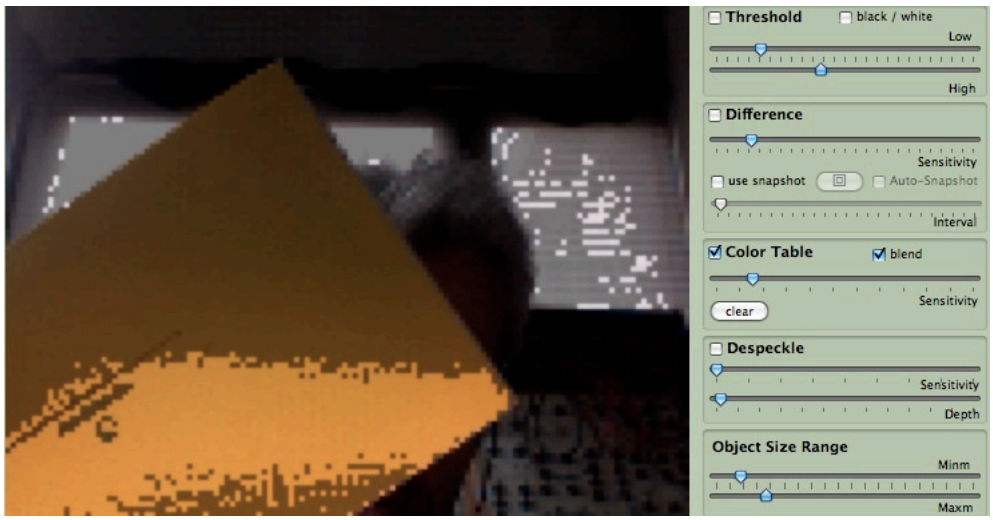
Color Table

This filter allows you to single out objects in the video image on basis of their color, allowing you to look, for instance, only at the green objects in a scene, or just at the red objects. It can also be used to single out colors you don't want to look at, where you have for instance a background of a particular color (or couple of colors) that you're not interested in, and want to interpret at all other colors. When selected by clicking on the Color Table button, you will have to teach junXion which color you want to track. You do this by holding your specific colored object (like the yellow paper in the above picture) in front of the camera and clicking in the Video Image pane on your colored object.



Clicking once selects one color in the Color Table View

As you can see, also automatically the *blend* button will be enabled and a small group of pixels are illuminated, showing you that this color is added to the Color Table. By changing the angle of your object and holding it at different locations you will see that due to light changes and reflection this picked color may not be recognized in every position. You add more color info to the Color Table by clicking a couple of times on your object at different locations and angles. By moving the Sensitivity slider more to the right, you will see that more variations of that color



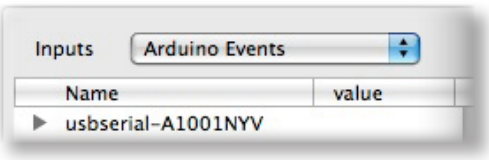
Clicking several times adds more colors to the table

(nearby in the color circle) are added but you also notice that you may have unwanted additional colors (like the window in the picture).
You will have to experiment with the slider settings to find a proper compromise.

Despeckle

This filter is useful to get rid of noise, random pixels, which the previous filters may let through. It blacks out clumps of pixels which are smaller than a certain size. You can set this threshold with the Depth slider. The more this slider is set to the right, the more pixels are 'blacked out'. With the sensitivity slider you can vary the size of the 'clump'. Usually this filter is used in combination with the Color Table. First enable the Despeckle filter, you will notice that the image gets a softer 'out of focus' look, after that enable the Color Table and click on the color you want to add to the table.

! *It is important to remember that the order of the dataprocessing through the Filters is dependent on which Video Filter you enabled first, which one you enable second, etc. The data processing flow can be seen in the View point menu (top to bottom).*



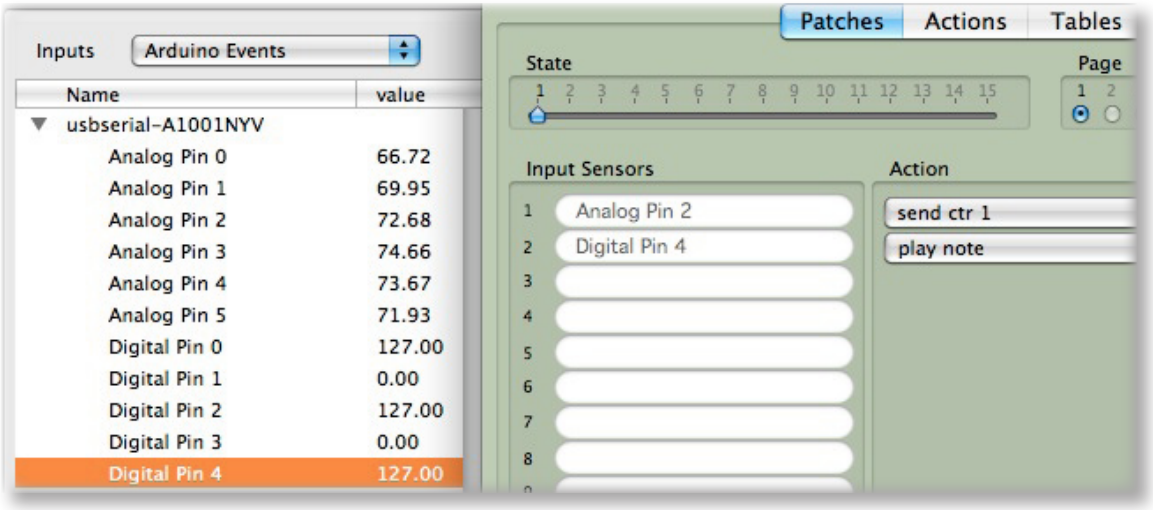
A successfully connected and activated Arduino Board will appear in the Input Browser like this. A click on the triangle shows all available sensors.

! *You can find the Arduino software online: http://arduino.cc/en/Main/Software.*

Arduino Serial Device Input

JunXion v4 supports the widespread Arduino sensor board, which communicates with your Mac through what is called *virtual serial USB*. Although the Arduino system allows you to create a powerful stand-alone sensor system, many users find the programming part quite difficult; so for those users junXion v4 provides a simple way of reading the data from sensors attached to the Arduino board. At this moment we have tested the very popular Arduino *Diecimila* board, other boards should work as well.

When you have bought an Arduino board, you will be supplied with the hardware and you will need



Arduino Events can be used and processed as any other Inputs in junXion

to download the serial port driver software and the Arduino software to initialize your Arduino board for communication with junXion v4.

Before you start using the Arduino software make sure you have installed the FTDIUSB serial driver (included in the Arduino download package). After that, you can connect your Arduino board to a USB port. Now from the junXion v4 folder find the 'Sketch' file named 'junXion_Arduino.pde' and double-click on this file to start up the Arduino program. In the Arduino program make sure to select the right serial port (see the Arduino documentation for more info), compile the Sketch and upload it to the board. Your Arduino board is now ready for communication with junXion v4 and you can quit the Arduino program.

If no sensors are connected you will see a lot of data jittering, this is just noise, once you have properly connected your sensors you should get a stable signal (see Arduino documentation for more info). After that, you can drag any of the Input sensors (Analog Pin 0-5, Digital Pin 0-13) into the Inputs Sensors column to create a Patch, just like any of junXion's input sensors. This way you can use the processing power of junXion v4 to 'intelligently' map the Arduino sensors into MIDI or OSC, without the need of writing a complicated Arduino sketch.

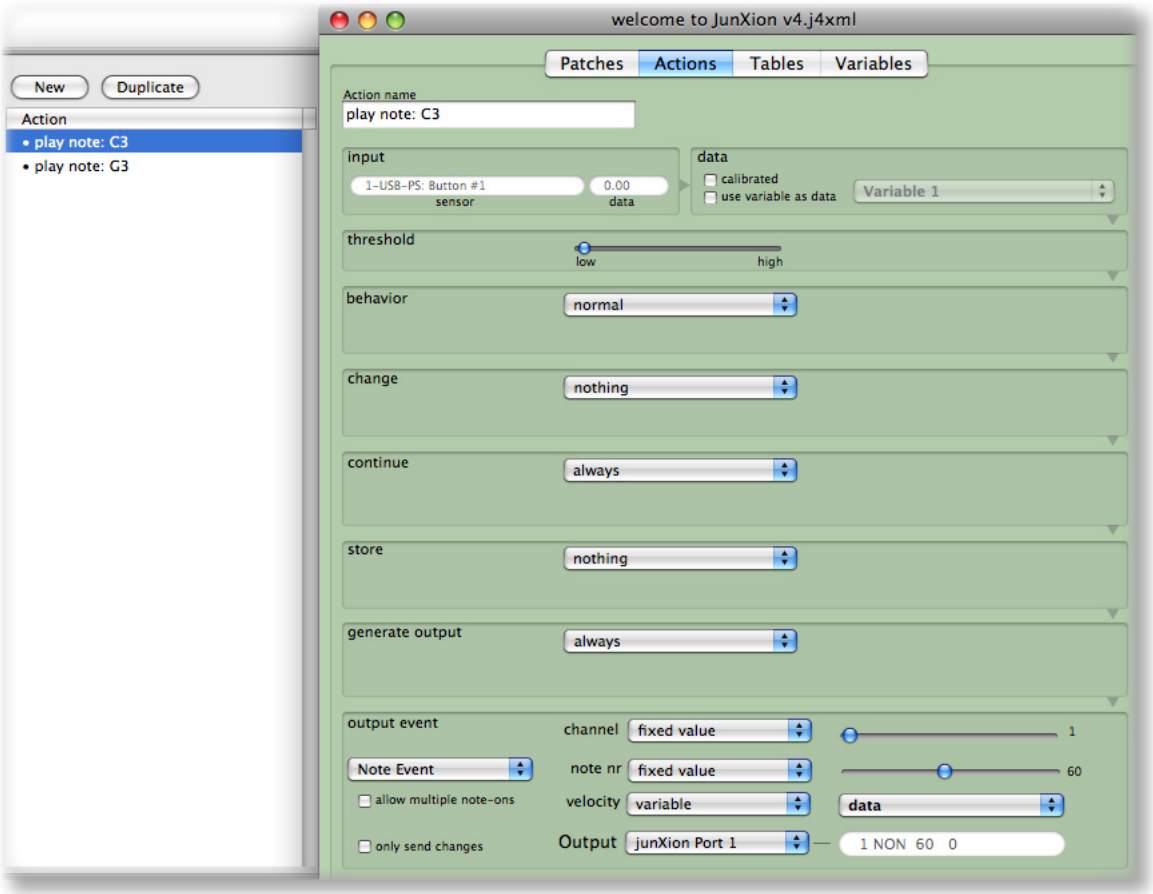
! *Initially Arduino Inputs are not enabled. You'll have to activate the support of the Arduino Serial Device in junXion's Preferences.*

5. Actions

An Action takes an Input Sensor's data and optionally transforms it and generates output. Actions can cause MIDI or OSC output and can modify internal Timers and Variables. Actions can also be used to switch to a different State. A junXion v4 Configuration can contain up to 1000 Actions. Actions are edited using the Actions pane in the main window and its corresponding browser to the left of the window. This browser displays a list of all available actions (initially only one). Clicking on an Action's name immediately displays its settings in the main window.

Working with Actions

When the Actions pane is visible in the main window, the browser to the left of the window displays a list of all available Actions. Actions that are in use and connected to one ore more Inputs show a • before their name. You can create a new Action by clicking on the 'New' button in the drawer or by clicking 'Duplicate' to produce a copy of the currently selected Action. Remember that by default a new action is created automatically for every Patch you create.Actions can be selected by clicking on their names in the list or by using the up/down cursor keys on your keyboard. Whenever an Action is selected, the main window displays the settings for that Action.



The Actions Pane

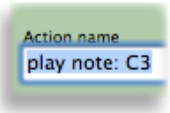
You can 'jump' to an Action from the Patches pane by double-clicking on the Input Sensor column entry for the Action you want to view or edit. junXion v4 will automatically switch to the Actions pane and show the chosen Action.
When an Action is selected in the list you can delete the Action by choosing 'Delete' from the 'Edit' menu or by using the cmd-Backspace keyboard shortcut.
Whenever you use the 'Copy' item in the 'Edit' menu, you actually copy the settings of the currently displayed Action. If you then select another Action and choose 'Paste' from the 'Edit' menu, you replace the current settings of the selected Action with those of the Action you just copied.
In general junXion v4 doesn't care what kind of Input an Action is connected to. All HID, Wii, Audio/ Video, OSC, Timer, Video, and Arduino Board Input Sensors provide only one input data value. However, Note and Cont. Ctr. MIDI events provide two data values (for Note events these are the note number and the note velocity, and for Cont. Ctr. events the controller number and its value). In the case of these two MIDI events an extra setting is displayed in the Threshold group box named 'Trigger'. More details on this special case of Input data can be found in the MIDI Inputs chapter of this manual.

! *'Top to bottom' is the overall calculating principle within junXion v4, remember that also the order of patches in the 'Patches' pane can be relevant for the resulting output.*

Action settings

Each action consists of a series of stages which are executed sequentially, these are: **threshold, behavior, change, continue, store, generate output** and **output event.** These stages appear on the user interface from top to bottom in the same order in which they are executed. In this way you should be able to understand what an Action does by reading the different group box names and their settings from top to bottom. It is important to remember that what is displayed in each group box often depends on the settings of its menus.
The remainder of this chapter describes all of the possible settings for each stage of an Action.

Name



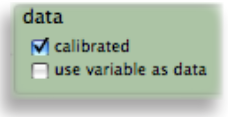
At the top left you will find the Name edit field where you can enter a name for the Action. Click on it to change the name and when finished hit the 'Return' key on your keyboard or click somewhere outside the field. The new name will also be updated in the Action list and the Action menus in the Patches pane.
As in any scripting language, it is important to give your Actions meaningful names. As your Configuration grows in size and has more Actions their names can provide useful reminders of their usage. Don't rely on the default naming of junXion v4. Whenever you create a new Action make sure you give it a proper name which is meaningful to you and makes it easy to find it back when your configuration becomes more complex.

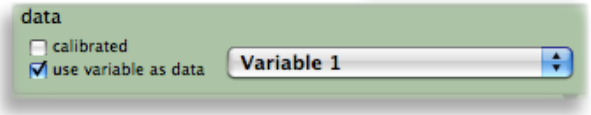
Input Sensor & data monitor

To the right of the Action's name you will find a non-editable display showing which input sensor is connected to the Action. The first number is the State which uses this Action, next is the Input device name, followed by the Input device sensor name. If the Action has been assigned to multiple Input sensors the text 'Multiple sensors' will be displayed. The raw data as displayed in the Patches Pane's Input browser is monitored in the 'data' field right to the sensor.
In the 'data' part on top of the Actions pane is a checkbox named 'calibrated'. When this checkbox is checked, junXion v4 will automatically calibrate the data, meaning that if only a few different values have been received from the Input Sensor, junXion v4 will try to calibrate these values to the full range available. This can be useful if for example your joystick's Y-axis sensor generates raw data in the range 21.42 – 118.53 instead of 0 – 127. With the 'calibrated' option switched on, junXion v4 will auto calibrate values into the full range 0 – 127 range, scaled to floating point numbers within the desired data range you have set in the preferences.

Data variable

Next is a pop-up menu containing the names of all available Variables. This variable list is only enabled when the 'use variable as data' checkbox is checked. In this case the Action will be triggered by the connected Input sensor, however it will use the selected Variable as its data instead of the Input sensor's data. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.





Action Stage 1: Threshold

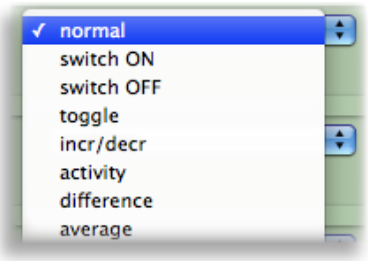
The slider in the **threshold** group box determines how much change in the Input sensor's value is required before the Action is triggered. When the **threshold** is 'low' (the default value is slightly above 'low') the Action will be triggered whenever the sensor's value changes. Sometimes when dealing with 'jittery' sensor data it is useful to set the **threshold** higher to eliminate this jitter. Be aware that a high threshold will cause a loss of resolution. Sometimes it may also be useful to make use of the **average** behavior to handle noisy or jittery input data.

Trigger (only visible with some MIDI events)

As mentioned above in the MIDI Input Chapter, the **threshold** group box will also display the 'trigger' selector if the Input sensor is a MIDI event of type 'Note' or 'Cont. Ctr.'. The trigger selector allows you to select whether the Action is triggered by the first (d1) or second (d2) data byte of the input MIDI event. The value of this data byte will also be used by the Action. In practice this means that two Actions will be necessary to process MIDI Note and Cont. Ctr. sensors. The first Action, triggered by 'd1', is used to store the data value into a Variable, and the

second Action, triggered by 'd2', can then use the stored 'd1' value AND its own 'd2' data value. So, for example, if one wants to rechannelize MIDI note events, the best thing is to use the 'd1' connected Action to store the note number into a variable and the 'd2' connected Action to actually send the rechannelled note event, using the stored 'note nr' variable as the first data byte.

Action stage 2: Behavior



The behavior group boxes allows you to specify how the input sensor should be used. There are eight modes:

Normal

Default setting in an automatically created Action.

Switch ON

This behavior is useful with switching sensors. The Action is only executed when a switch sensor opens, the switch 'close' is ignored.

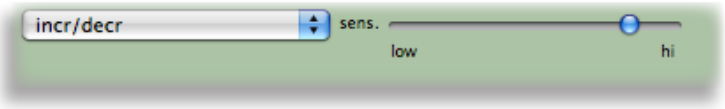
Switch OFF

This behavior is useful with switching sensors. The Action is only executed when a switch sensor closes, the switch 'open' is ignored.

Toggle

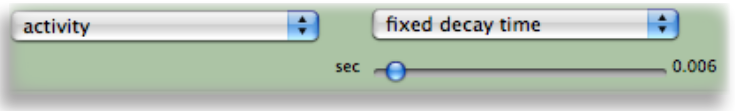
This behavior is useful with switching sensors. The Action only acts on 'switch close' events, always ignoring 'switch open' events. When it first receives a 'switch close' event it outputs 'switch close'. Successively received 'switch close' events alternately output 'switch open' and 'switch close'. Typical toggle switches in everyday life are found in footswitches for electric lights. Pressing and releasing the button will switch the light on, repeating this will switch it off.

Incr/Decr



This **increment / decrement** behavior is useful with input sensors like mice, trackballs, touchpads etc. which generate relative values indicating the amount of change since the last value sent rather than absolute values (such as positions or coordinates). When this behavior is used, junXion v4 will **increment / decrement** an internal variable to give you an absolute data value. The sensitivity slider ('sens') allows you to specify how quickly the data should be **incremented / decremented**.

Activity



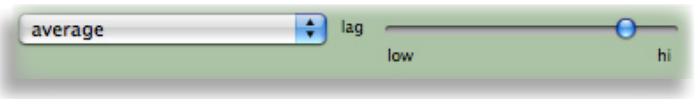
This behavior works best with continuous input sensors (i.e. not with switches). The Action's data value is 'pumped up' according to the amount of activity received from the sensor. The slider determines how quickly the value returns to zero if no sensor activity occurs. You can think of a metaphor like filling up a bucket which actually has a hole in it: you have to keep filling otherwise the bucket will soon empty itself (short decay time = big hole, long decay time = small hole). The **activity decay time** can also be controlled by a variable. If you select variable decay time from the pop-up menu, a second pop-up menu will be displayed allowing you to select the Variable you want to use. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.

Difference

In this mode (formerly called 'Delta') only the Input's **difference**

data is used in the Action, i.e. the difference between the 'data now' and the 'previous data'. These **differences will always be positive**. If after a certain defined reset time no new data is received, the 'data now' and the 'previous data' will be synchronized to avoid sudden big data differences.

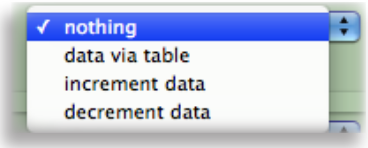
Average



In this mode of behavior junXion v4 **averages the new data with the old data**. With the 'lag' slider the 'weight' of the new data can be controlled. If the 'lag' is set to low, the new data has more direct influence than with a high 'lag'. You can see this behavior as a low pass data filter on the input data. The average behavior can be useful for 'jittery' sensors with a noisy data output.

Action Stage 3: Change

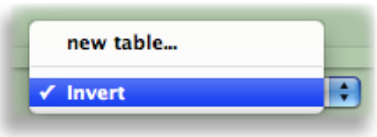
The change stage allows the Action to change its data.
There are four options:



nothing

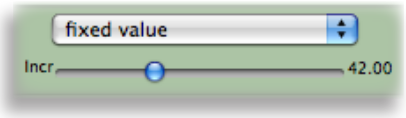
The data is passed through unchanged.
Default setting in an automatically created Action.

change data via table



Using this setting the **data value is transformed to a different value by a lookup table**. Tables allow you to precisely specify the data values you want to get out for each value of the input data. Tables are created and managed in the Tables pane, which is described in more detail in the next chapter. By default there is one Table called 'Invert', which is 65.536 (16 Bit) units long and goes from 127 to 0. If you select the first menu item 'new table...' the program automatically takes you to the Tables pane and creates a new linear (i.e. neutral) Table. More details on Tables can be found in the tables Chapter of this manual.

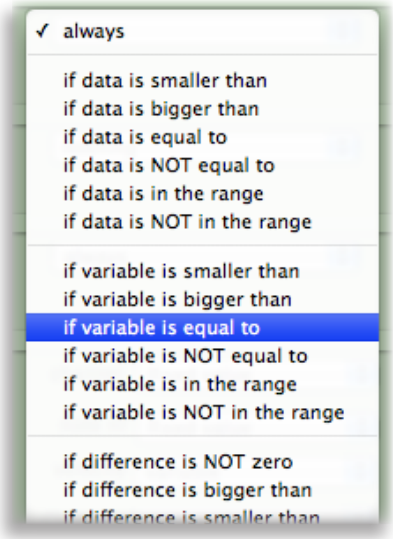
increment data



This setting **adds a fixed value or variable** to the data. The slider specifies the value to add. If the value reaches the maximum possible value it is clamped and does not go any higher.

decrement data

This setting **subtracts a fixed value or variable** from the data. The slider specifies the value to subtract. If the value reaches zero it is clamped and does not go any lower.



Action Stage 4: Continue

Continue is, metaphorically speaking, the first stoplight in an Action. This part of the Action determines whether the subsequent stages of the action (store, generate output and output event) are executed. There are 15 different conditions which can be used to determine if the Action continues or not:

always

The action always **continues**. Default setting in an automatically created Action.

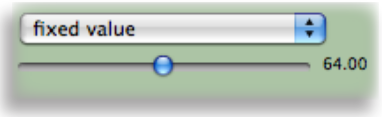
if data is smaller / bigger than

Only **continue** with the rest of the Action if the input sensor's data is smaller / bigger than: fixed value or Variable.
In many Actions you may choose between a fixed or variable value, which means:

Fixed Value: *use the slider to select a value. The slider works in the range from 0 - 127 (or your user-defined setting). If you need to set it to a fractional value, for example 27.45, you can do that by moving the slider so that it reads 27 and then click on the number and move your mouse upwards while holding down the mouse button. This way you can select the fractional part of the number by moving the mouse up and down as a slider.*
Variable: *use the pop-up menu to select a Variable. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference. The Variable can refer to a a fixed value, or a further Variable's value.*

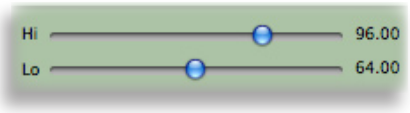
if data is equal to/ if data is NOT equal to

Only **continue** with the rest of the Action if the input sensor's data is (or is NOT) equal to: fixed value or variable.



if data is in the range

Only **continue** if the data is in between the following values. As soon the Lo value is equalled, the Action will continue (including the Lo value) until the Hi value is reached. As soon as the input equals the Hi value, the Action is interrupted (excluding the Hi value).

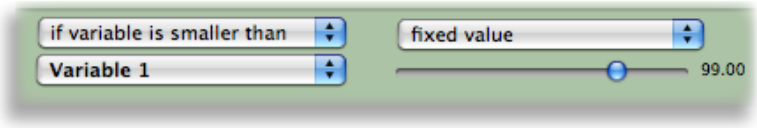


if data is NOT in the range

Only **continue** if the data is NOT in between the following values. As soon the Lo value is equalled, the Action is interrupted (excluding the Lo value) until the Hi value is reached. As soon as the input equals the high value, the Action is continued (including the Hi value).

if variable is smaller / bigger than

Only **continue** with the rest of the Action if the selected Variable's data is smaller / bigger than: fixed value or Variable.



if variable is equal to / if variable is NOT equal to

Only **continue** with the rest of the Action if the selected Variable's data is (or is NOT) equal to: fixed value or Variable

if variable is in the range

Only **continue** if the selected Variable is in between the following values. As soon the Lo value is equalled, the Action will continue (including the Output of the Lo value) until the Hi value is reached. As soon as the Variable equals the Hi value, the Action is interrupted (excluding the Hi value).

if variable is NOT in the range

Only **continue** if the selected Variable is NOT in between the following values. As soon the Lo value is equalled, the Action is interrupted (excluding the Lo value) until the Hi value is reached. As soon as the variable equals the high value, the Action is continued (including the Hi value).

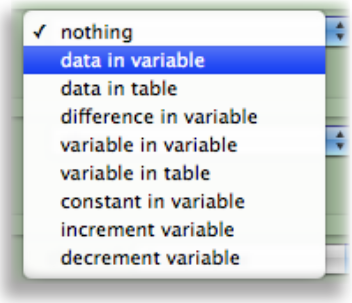
if difference is NOT zero

Only **continue** with the rest of the Action if the input sensor's data has changed since the last time this Action was triggered. The difference between the current data and the last received data measured.

if difference is bigger than/ if difference is smaller than

These features allows you to set a differential threshold, meaning you can decide to only **continue** if the new data is a certain amount bigger than the old data, a bit like the top threshold slider, except that now the data is already processed by the behavior and change stages. In the same way you can decide to only continue if the changes in the data are very small, so you can filter out all the big changes a sensor is generating.

Action Stage 5: Store

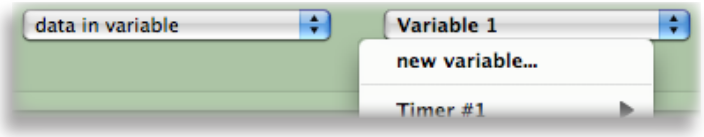


The store stage provides the following options for storing the Action's data:

nothing

The Action's data is **not stored**. Default setting in an automatically created Action.

data in variable



Store the data in the Variable selected in the pop-up menu. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.
Remember that the data you are storing may have already been changed by the change stage. If you want to store the raw data while making use of changed data, use a separate Action connected to the Input sensor which only stores the unchanged data into a Variable and does not generate any output.

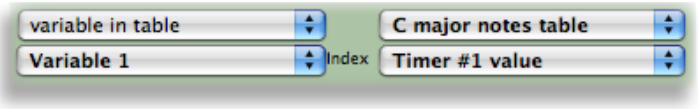
data in table

Store data into the Table selected in the pop-up menu. If you select the first menu item 'new table...' the program automatically takes you to the Tables pane and creates a new linear Table.
The 'Index' Variable pop-up menu allows you to select the variable which will be used to determine the table location into which the data will be stored. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.
Example: You want an Audio Input to write its level data into a table, and the writing shall only proceed if sound is coming in. Therefore you'll first have to store an index variable from your Audio Input Action which increments (wrapped) everytime that audio is being sensed (see: store increment variable, below in this chapter). This index variable is used for a further Action, in which the level data is continuously stored in a table.

difference in variable

Store the difference between the current and previous sensor values into the Variable selected in the pop-up menu. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.

variable in variable

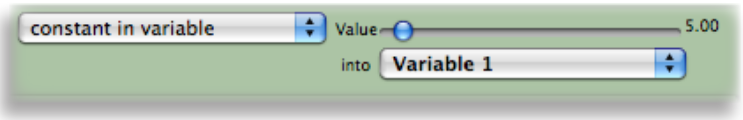


Store the value of one Variable into another. Select the source Variable from the top pop-up menu, and the destination in the lower menu.
You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.
Example: This copy of a Variable can be useful to find out if a note number which is already stored in a Variable has just been played.

variable in table

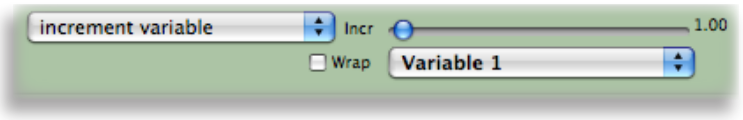
Store the value of a variable into a table. Select the source Variable from the top pop-up menu. The Variable's content will be written into the selected Table at the location as set with the selected index Variable (remember, a Table contains an array of numbers instead of one single number).

constant in variable



Store a fixed value into the selected Variable. This setting can be used to notify other Actions that an event has occurred. Use the Slider to select a fixed value from 0 to 127. The destination Variable can be selected from the pop-up menu. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.

increment variable

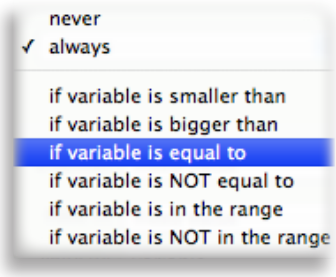


Increment the selected Variable by a fixed value from 0 to 127. The value can be selected with the Slider. The **Wrap** checkbox causes the Variable's value to wrap around to zero after it has reached 127. For example, incrementing a Variable with value 127 by 3 will result in the Variable with the value 2. The target Variable can be selected from the pop-up menu. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.
Example: This setting is especially useful with switching Input Sensors such as buttons. Remember that the Action will usually be executed with every 'switch on' AND 'switch off' event, meaning that a simple press/release of a button will result in 2 executed Actions, and hence 2 increments. If this isn't what you intend make sure to set the behavior to 'switch on', so the 'off' event is ignored.

decrement variable

Decrement the selected Variable by a fixed value from 0 to 127. The value can be selected with the slider. The **Wrap** checkbox causes the Variable's value to wrap around to 127 after it has reached 0. For example, decrementing a Variable with value 0 by 2 will result in the variable having the value 126. The target Variable can be selected from the pop-up menu. You can automatically create a new Variable by selecting 'new variable...' from the menu. Don't forget to give it a meaningful name for later reference.
See the 'increment variable' section above for more information.

Action Stage 6: Generate Output



The 'Generate Output' stage determines whether the Action will generate the output specified by the output event stage. Like in the 'Continue' stage the data stream can be stopped here due to certain specifications. There are 7 different conditions which can be chosen:

never

Never generate output. The output event group box will not be visible.

always

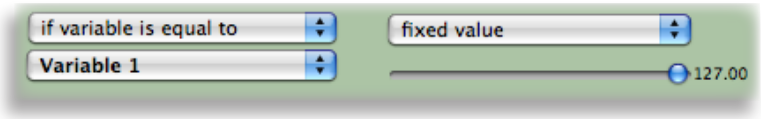
Always generate the output specified in the output event box. Default setting in an automatically created Action.

if variable is smaller than/ if variable is bigger than

Only **generate output** if the selected Variable's data is smaller (or bigger) than: Fixed Value or Variable

if variable is equal to/ if variable is NOT equal to

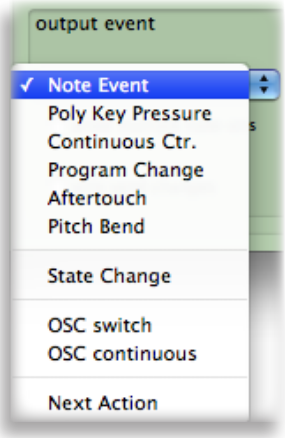
Only **generate output** if the selected Variable's data is (or is NOT) equal to: Fixed Value or Variable



if variable is in the range/ if variable is NOT in the range

Only **generate output** if the selected Variable's data is (or is NOT) within the range specified by the 'Lo' and 'Hi' sliders, each of which can be set to values between 0 and 127.

Action Stage 7: Output Event



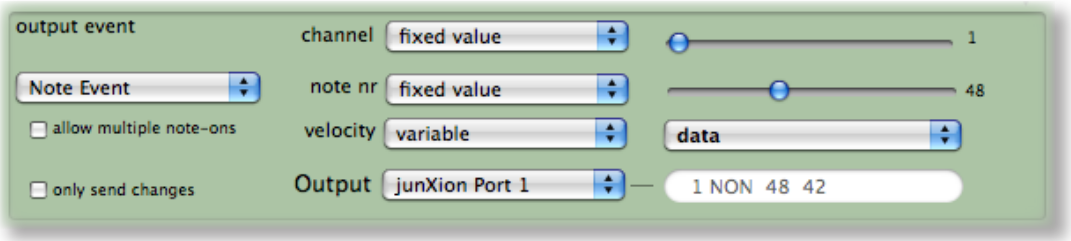
The 'Output Event' stage specifies the final output of an Action which can be either a MIDI or OSC event or an internal State change. Another option is to output the data to a second so-called 'daisychained' action. The pop-up menu at the left of the output event group box allows you to choose between one of the following output events.

Note Event

Output a MIDI note event with the following parameters:

Channel, (available for all MIDI events)

- **Fixed value:** 1 – 16, selected with a slider.
- **Variable**: use the value of the Variable selected in the pop-up menu. The menu contains two additional items named 'data' and 'scaled data'. In all cases the Variable's value is used directly, and clamped to the 1 – 16 range. For example, when the Variable (or data) has value 1 the MIDI channel will be 1, with value 2 the MIDI channel will be 2. If the data value is below 1, the channel will be 1, if the data value is above 16 the channel will be 16. Note that this treatment of Variables is different from the way they are used in other parts of the output event.



Note Nr

for example, note number 69 represents A3 = 440 Hz
For all Fixed Value / Variable selections in the output event the rules are:

Fixed Value: The slider selects a value in the range 0 - 127;
Variable: use the value of the Variable, 'data' or 'scaled data' selected in the pop-up menu. Values will be rescaled into the range 0 – 127. The 'data' option provides the value of the Input Sensor including any modifications applied by the change stage. The 'scaled data' option also takes into account the range of data values actually passed by the continue stage. For example, if the continue stage continues only if the data value is smaller than 64 then the range of values passed will be 0 – 63 and this 0 – 63 range will then be rescaled to 0 – 127.

Velocity

A velocity of 0 means 'note-off', any other value is a 'note-on'.
Select a fixed value or Variable. See the 'note nr' item above for more information about the data options.

Output (available for all MIDI and OSC events)

Specifies the MIDI port to send the event to. The pop-up menu provides a list of all available MIDI ports found by junXion v4 at startup. The list also includes two virtual ports 'junXion Port 1' and 'junXion Port 2' which junXion v4 creates so that you can send MIDI events to other applications. If the Quicktime Synthesizer is enabled in the junXion v4 Preferences it will also be listed.
To the right of the Output menu is a display which shows the MIDI or OSC data generated by the Action whenever it is triggered by a connected Input sensor (same as in the Patches window in the Output Column).

Allow multiple note-ons

Normally an Action can have only one note-on at a time and will automatically send a note-off message to switch the previous note off before sending a new note-on message. When the allow multiple note-ons option is checked the Action will send multiple note-on messages without automatically sending note-off messages. This means that you are responsible for creating Action(s) to send note-off events for every note-on message you send, otherwise you will be left with 'hanging' notes.

Only send changes, (available for all MIDI events)

When this option is checked junXion v4 will only send MIDI data to the selected Output if at least one of the data bytes in the event has changed since the last event sent by the Action. In most cases you will want to have this option enabled, since it thins out the MIDI data stream considerably. However, it is advisable to only use this option for MIDI continuous control data and not for note events, because it will disable junXion v4's feature to automatically send note-off messages and will leave you with the responsibility to avoid chords of 'hanging' notes (hint: the 'Continue: if differential is NOT zero' option can be helpful with note events).

Poly Key Pressure

Output a MIDI poly key pressure event with the following parameters:

Channel/ Note Nr

As described in the Note Event section above.

Pressure

Select the amount of key pressure applied to the note specified by note number.

Output/ Only send changes

As described in the Note Event section above.

Continuous Ctr. (continuous controller)

Output a MIDI continuous controller event with the following parameters:

Channel

As described in the 'Note Nr' section above.

Ctrl Nr

Select the controller number, fixed value or Variable.
(In the MIDI Standard it's e.g.: nr 1 = Modulation Wheel, nr 7 = Volume, etc.).
If you use the MIDI-learn function of your sound generating software you are free to use any values as you want.

Value

Select the value of the controller, fixed value or Variable.

Use double precision

Send double precision (14 bit) controller messages. The 'ctrl nr' slider will be limited to selecting the numbers 0 – 31 as per the MIDI specification. The MIDI specification supports sending 14 bit values for controller numbers 0 to 31 by sending two messages for each control change. The most significant byte (MSB) is sent on the specified controller number, and the least significant byte (LSB) is sent on the corresponding controller number in the range 32 – 63. When this option is selected junXion v4 outputs both controller messages.

Output/Only send changes

As described in the Note Event section above.

Program Change

Output a MIDI program change event with the following parameters.
Note that this event sends only one data byte.

Channel

As described above in the Note Event section above.

Preset

Select a fixed value or Variable. See the 'note nr' item above for more information about the data options.

Output and only send changes

As described in the Note Event section above.

Aftertouch

Output a MIDI aftertouch event with the following parameters.
Note that this event sends only one data byte.

Channel

As described above in the Note Event section above.

Press. (pressure)

Select a fixed value or Variable. See the 'note nr' item above for more information about the data options.

Output/ only send changes

As described in the Note Event section above.

Pitch Bend

Output a 14 bit MIDI pitch bend event with the following parameters:

Channel

As described in the Note Event section above.

LSB / MSB

The LSB is the least significant byte and adresses the lower 7 bits of the final 14 bit value. The MSB (most significant byte) adresses the upper 7 bits of the final 14 bit value.
Select a fixed value or Variable. See the 'note nr' item above for more information about the data options.

Use double precision

Send only the value selected by the MSB menu rescaled to the full 14 bit resolution of the pitch bend event (0 – 16383). The LSB pop-up menu setting will be ignored. This option is only really useful if your Input Sensor's resolution (or the range which this Action is using) is bigger than 7 bits.

Output/ Only send changes

As described in the Note Event section above.

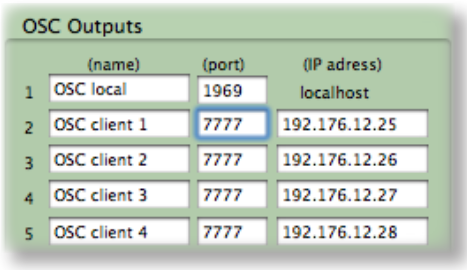
State Change

Switch to one of the 15 States supported by junXion v4.
JunXion v4 supports up to 15 states, each of which can contain up to 375 Patches. At any given time only one state will be active. Switching between States allows you to quickly change the way your Input Sensor's are used.

State

Fixed value: 1 – 15, selected with a slider.
Variable: use the value of the Variable, 'data' or 'scaled data' selected in the pop-up menu. Values are treated as for the channel setting of Note Events described above. This means that the variable's value is used directly, and clamped to the 1 – 15 range. For example, when the variable (or data) has value 1 junXion v4 will switch to State 1, with value 2 junXion v4 will switch to State 2. If the data value is below 1, the State will be 1, if the data value is above 15 the State will be 15 (but you can only jump to State 15 if it exists, meaning there are already 14 States).

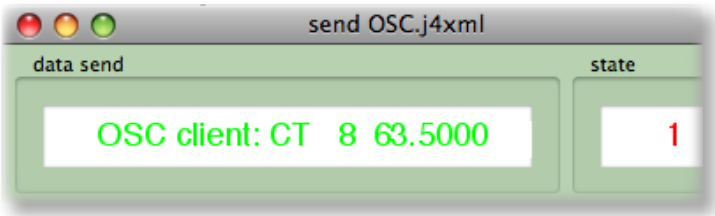
OSC Outputs



The OSC Output Preferences

To set up junXion v4's OSC outputs, open the Preferences window.

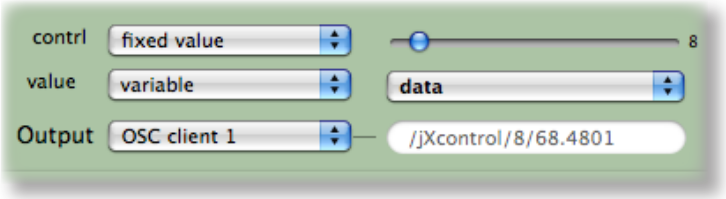
The section named 'OSC Outputs' needs to be configured first before you can send OSC data over your network to other machines. Five OSC outputs are available in junXion v4. Each output can have its own port number.
The OSC output marked as 'localhost' is comparable to MIDI's IAC bus or junXion Port 1 & 2. It can be used to send OSC data to another application on the same machine. The four other OSC Outputs contain a name box (for your own reference) and IP adress and port number boxes where you can enter the IP adress of the machine you want to send your OSC data to and the port number that machine is listening on. The data is sent either via wired ethernet or via WiFi, depending on your network preferences.



OSC Output viewed in the Status Window

OSC Switch or Continuous

There are basically two parameters for each OSC event. With the first parameter you can set the switch or continuous controller number (0 - 127). This parameter can either be fixed or any of junXion v4's Variables. The second parameter contains the 'real' data. This can also be fixed or a Variable. With the Output menu you select to which OSC output this Action will send its data. The data is send as floating point data within the user defined data range. The display to the right of the menu shows you the OSC data that is sent. The data shown is the actual OSC message string that is sent out over the selected port. The switch events can be used for a sound start/ sound stop event for example.



OSC Output. Floating Point data in the format /jXcontrol/x/data

In the Patches pane the Patch Output column will show the sent OSC data. It as well shows the actual OSC message string that is sent out over the selected port. Remember that MIDI and OSC data can be used simultanuously in junXion v4 because the Output can be different for each Action.

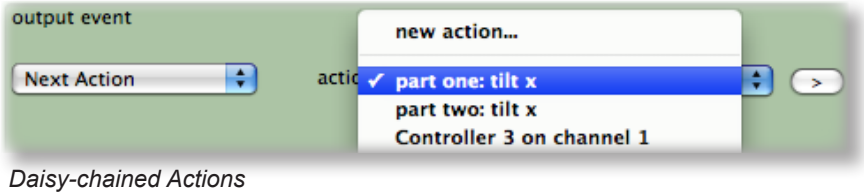
When junXion v4's window is zoomed out to pure status display, all OSC data is shown in green, while MIDI data and the 'state' data is shown in red.

Note that with OSC events we don't have a prestructured language such as MIDI, which means that all the OSC events need to be individually named so that they can adress the target application.

! *The OSC strings that are sent are formatted by junXion v4 in the following way:*
OSC Switch:
„/jXswitch/x/data", x = switch nr, data in the defined user range as floating point messages
OSC Continuous:
„/jXcontrol/x/data", x = contr. nr, data in the defined user range as floating point messages

Next Action

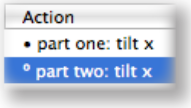
This feature offers the possibility to send an Action's output to another Action (as its input). This way you can check many conditions before you eventually want to send MIDI or OSC, something which cannot always be checked within one single Action. It is also referred to as '**daisy-chaining**' Actions. You can use this feature by selecting Next Action from the output event popup menu.
A new menu is shown where you can select which Action you want to link to. It is also possible to directly create a new Action from this menu. When this menu item is selected the link will be to this new Action and the new created Action will be shown immediately to you for editing.



Daisy-chained Actions

The Action list will show the 'daisy-chained' Action marked with a ° symbol in front of its name, unless the Action is also directly used by

an Input Sensor, in which case the symbol is a •. You can quickly select a daisy-chained Action by clicking on the daisy-button [>] to the right of the linked Actions menu. It might be a good idea to name daisy chained Actions in such a way that you know it is part of the chain, as shown in the example when you have 2 Actions that are connected, naming them 'part one: tilt x' and 'part two: tilt x'.

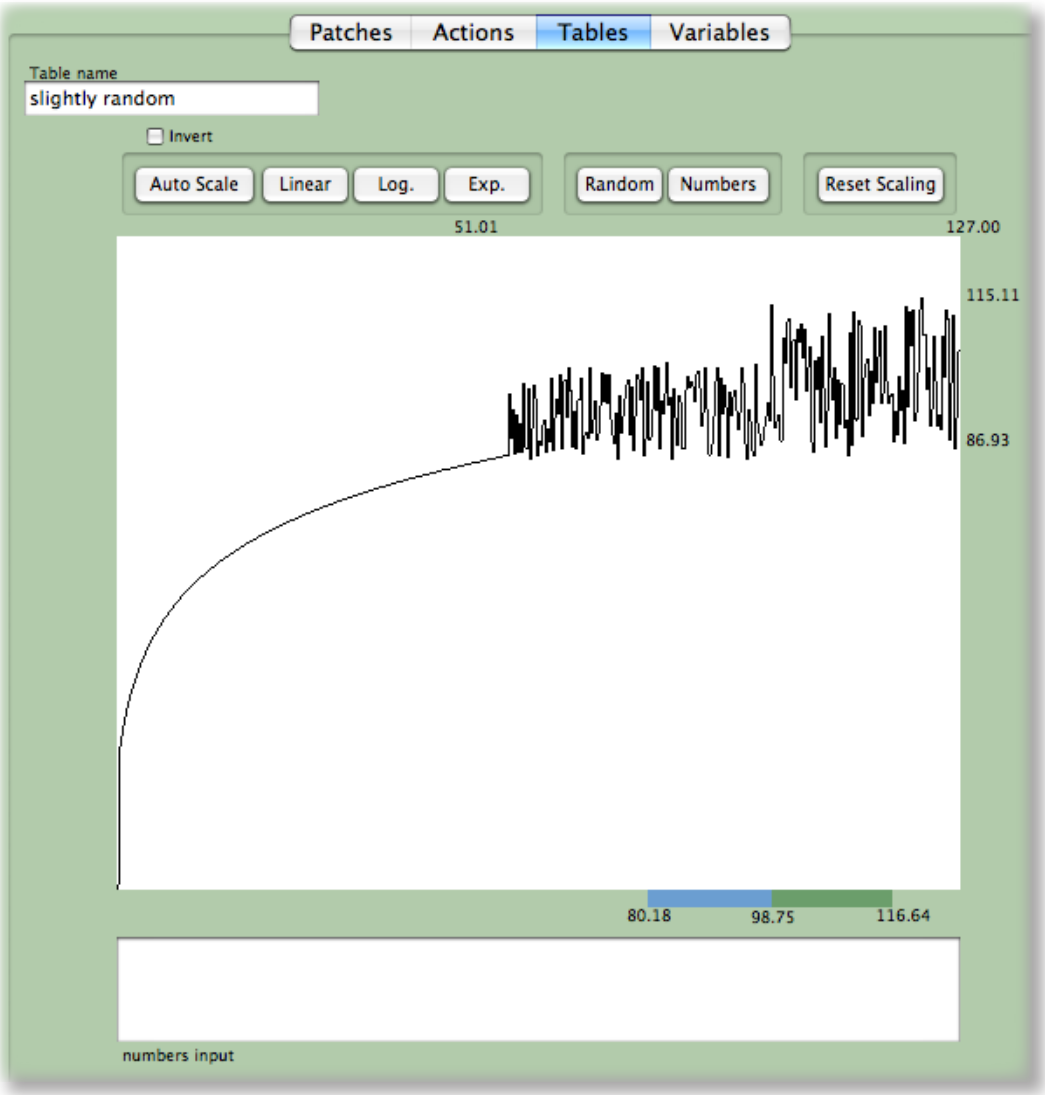


6. Tables

JunXion v4 allows you to create any number of Tables which can be used to transform Input Sensor data values into other values. Actions can also store values into Tables. All junXion v4's Tables are 16-bit Tables. The input range (the horizontal scale) is the same as the output range (vertical scale) and depending on your Preferred Data Range setting. Any higher resolution will be scaled to floating point numbers.

The '**change data via table**' option of an Action's 'change' stage allows an Action to remap its input data by looking up new values in a Table.

An Action's 'store: data in table' option provides a way to dynamically store new data into a Table. See the Actions chapter for further information about these options.



A Table in junXion, the Input values are represented along the X-Axis, Output along the Y-Axis

Input Monitor

As soon as an Input Sensor or Timer is connected and processed by a Table, the Input data will be monitored in a small moving bar below the horizontal input axis.

Organizing Tables

The Tables pane in the main window displays the currently selected Table's name, in some cases its numerical description and always its graphical representation. These are described in more detail below.
When the Tables pane is visible the browser to the left of the main window contains a list of all available Tables. Tables that are currently used by one ore more Actions show a • before their name.
You can create a new Table by clicking on the 'New' button or make a copy of the currently selected Table by clicking on the 'Duplicate' button. Tables can be selected by clicking on their names with the mouse or by using the up/down cursor keys on your keyboard. The main window always displays the settings for the currently selected Table.
The selected Table can be deleted by choosing 'Delete' from the 'Edit' menu, or by using the cmd-Backspace keyboard shortcut. If the Table is being used by any Action you will be asked to confirm the delete operation. Confirming the deletion causes all Actions referencing the deleted Table to be updated to reference the first Table in the list.
Whenever you use the 'Copy' menu item in the 'Edit' menu, you actually copy the settings of the currently displayed Table. If you then select another Table and choose 'Paste' from the 'Edit' menu you replace the settings of the current Table with those of the one you just copied, including its name. Copying does not apply to partial selections within a table.

Table name

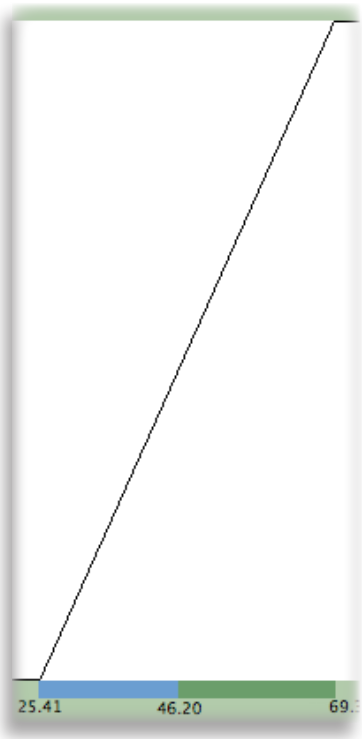
At the top left of the Tables pane is the Table name edit field which allows you to enter a name for the Table. Click on the field to edit the name. Press the 'Enter' or 'Return' key on your keyboard for the change to take effect. The new name will be updated in the Tables list and in the pop-up menus of the Actions that use it. Keep in mind that meaningful names help here to program with more overview as soon as the confuguration gets more complex.

Generating Tables

You can create your own Tables in order to transform input data. The only Table that comes with junXion v4's default Configuration is the 'Invert' Table. Selecting 'new table...' in an Action will produce a new, linear (i.e. neutral) Table which then should be adjusted to achieve your desired transformation.

Auto Scale

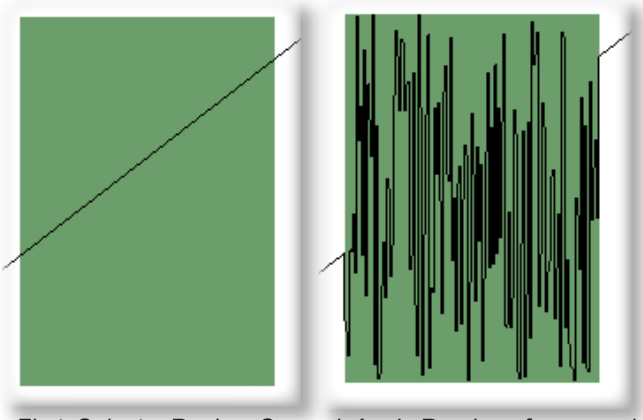
Once a Table is connected to an Input Sensor you will observe the small bar below the Input-Axis which monitors the Input data. If this bar is visible the '**Auto Scale**' button is active. Clicking here will result in a linear table that ranges between the minimum and maximum value of the Input data as it has reached the Table. The minimum will accordingly be mapped to the output 0, the maximum to the output 127. You can reset the scale by clicking on the '**reset scaling**' button and repeat the process. This feature is useful for sensors with a limitation in their output range, e.g. tilt sensors and accelerometers which are used carefully.



Auto Scaled Table

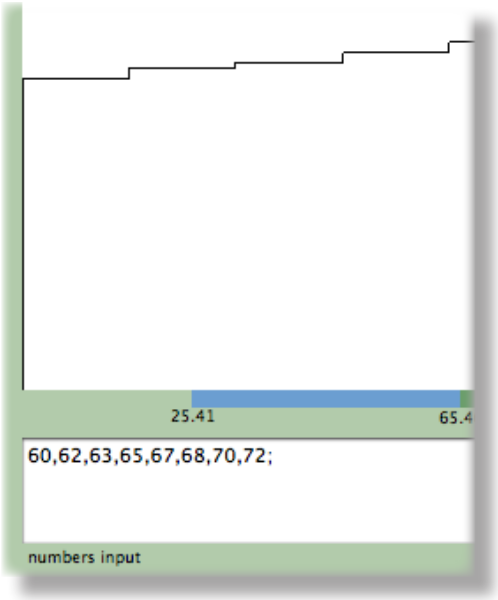
Linear, Exponential and Random Tables

A region can be selected by manually drawing a rectangle onto the Table canvas. The Table generating options explained below will then apply to this selection. If there is no selection (you can erase a selection by clicking on the blank canvas) the Table will be generated over the full data range.



First: Select a Region; Second: Apply Random, for example

Pressing the 'Linear' button will result in a Table that goes straight from the minimum to the maximum values. If you want an inverted table make sure to tick the 'invert' checkbox before clicking.
Pressing '**Log**' or '**Exp**' will result in a progressive Table. This table can also be inverted by activating the checkbox before clicking.
You can also generate a randomized sequence of values for the selection or the entire data range by clicking the '**Random**' button. This process can also be inverted, decide for yourself if that makes sense or not.



Numerical Input allows for precise Table processing

Creating numerical tables

You can enter a sequence of numbers into the '**Numbers Input**' field at the bottom of the Tables pane. They have to be separated by commas and ended with a semicolon; this sequence can be applied to the full range Table or to a drawn selection by clicking on the '**Numbers' button**. Clicking here will create the Table, the numbers will remain editable in the numbers input box.
If a region is selected in the Table and then a Sequence of numbers is applied, the values of the region will determine the horizontal (= input) extension of he new scale, the vertical (=output) extension will be defined by the numbers. In this case a click will erase the numbers in the input box.

!

The actual content of the Table is never saved, since this would require a lot of disk space for every Table. Instead, a Table is created by so-called Table descriptors: every time you generate a Table, either the whole Table or a small selection (= region), the Table descriptor is updated and saved together with your Configuration. This means that after opening your saved Configuration junXion v4 will reconstruct your Table based upon the descriptor. There is a limitation however in length of the descriptor text. This means that if you make a lot of edits and small regions in a Table, its descriptor may become too long. The best thing is, if your not satisfied with your Table and you have already done a lot of region edits, make sure no region is selected, click on the '**Linear**' button, and try again to avoid problems.

7. Variables

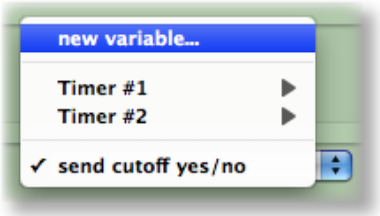
Variables in junXion v4 are used to **store Input sensor values so that they can be shared between different Actions**, set conditions, or be used at a later time. Each Variable stores a floating point value within the set data range.

Working with Variables

When the Variables pane is visible you will see a fairly empty window with merely the Variable's name and the initialization value to be edited. These options are described in more detail below.

The browser to the left of the main window contains a list of all available Variables. Each entry in the list shows the name of the Variable and its current value as a floating point value. You can create a new Variable by clicking on the 'New' button or make a copy of the currently selected Variable by clicking on the 'Duplicate' button. Variables can be selected by clicking on their names with the mouse or by using the up/down cursor keys on your keyboard. The main window always displays the settings for the currently selected Variable.
The selected Variable can be deleted by choosing 'Delete' from the 'Edit' menu, or by using the cmd-Backspace keyboard shortcut.

Variable menus in the Actions

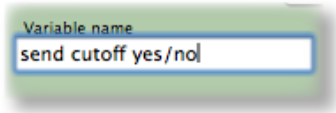


select a Variable in the Actions

All Variables are listed in the Variable pop-up menus used in the Actions pane. A Variable called 'Variable 1' comes with the default Configuration of junXion v4, every additional Variable needs to be created by the user either by selecting 'new variable...' from a Variable's menu in an Action or by clicking on the New button in the Variables browser.

Variable name

At the top left of the Variables pane is the Variable name edit field which allows you to enter a name for the Variable. Click on the field to edit the name. Press the 'Enter' or 'Return' key on your keyboard. The new name will be updated in the Variables list and in the Variable menus of the Actions. It is very useful to make up a meaningful names for each variable in order to handle more complex Configurations



Make up nice names for them

Initialization Value

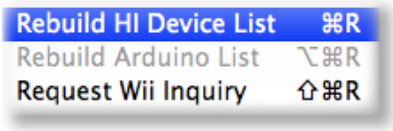
How shall the Variable be set when you open up your Configuration?

The Init Value slider allows you to set the initial value of the Variable in the data range 0 – 127, or the currently preferred data range set in junXion v4's Preferences. This value will be used as the Variable's initial value every time you open the Configuration. It will be overwritten as soon as incoming Input data will be stored into this Variable. If you need to adjust the fractional part of the Init value click and drag up/down on the displayed number to the right of the slider.

8. Configurations

A collection of Patches, States, Actions, Tables and Variables is called a Configuration. You you can save the current Configuration using the 'Save' menu item from the 'File' menu and recall it later using the 'Open' item. In addition to saving the Patch, State, Action, Table and Variable settings each saved Configuration also stores the current window position and zoomed / unzoomed state. The Preferences are globally overwritten to the current settings everytime a Configuration is saved.

Handling missing devices



When opening a saved Configuration at a later time junXion v4 tries to recreate the Patches which you stored. In some cases it may not be possible to recreate all of the Patches because one or more of the Input devices used when the Configuration was saved are not available. If this situation arises junXion v4 will notify you and the Patches which cannot be recreated will be shown in a small typeface. The missing Sensors also show a tiny exclamation mark in fromt of them. In the

case of missing HI devices you can connect the missing devices while junXion v4 is running and choose 'Rebuild HI Device List' from the 'File' menu, or press the 'cmd+R' keyboard shortcut. Similar for missing Wii Remotes and Arduino Boards: Choose 'Request Wii Enquiry' or 'Rebuild Arduino List' from the 'File' menu. If the missing devices are then found the Patches will function normally again and will be shown with the normal typeface. In the case of missing MIDI devices you will need to quit the program, connect the MIDI devices, and open the program again.
If you don't have the missing devices at hand and just want to adjust some Action parameters you can make the changes and save the Configuration. You will be able to test the changes when all devices are properly connected again.

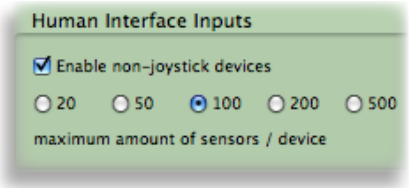
Working with multiple devices of one kind

Up to 15 HI devices and 15 MIDI Input ports can be used simultaneously within junXion v4. When HI devices with unique IDs are used junXion v4 always recognises these devices by their unique IDs and does not care about the order in which they are connected, or to which USB port or USB Hub they are connected.
Many commercial HI devices do not use unique IDs. In this case the devices themselves are indistinguishable to junXion v4 and using more than one of the same device becomes more complicated. To work around this situation junXion v4 remembers the physical USB port that each device is connected to. For this reason it is critical that when working with multiple identical HI devices which don't have unique IDs (for example when using 2 Thrustmaster Firestorm wireless gamepads) that you connect these devices in exactly the same way as they were connected when you saved the Configuration, otherwise things will get mixed up. A simple way to ensure this is to first always connect each device to the same port of a USB hub and only once this is done to connect the USB hub to your computer's USB port.
You can tell that devices have unique IDs if each connected device is displayed in the Input devices list with a different name. If multiple devices are listed with exactly the same name this indicates that they do not have unique IDs.
If you use more than one Wii Remote, they will synchronize as Wii 1-4, and accordingly one to four of the Wii's LEDs will be on when connected, so that you can specify them.

9. Preferences

Preferences in junXion v4 can be used to adjust some global settings. These settings are saved on your computer and used the next time you start up junXion v4. The Preferences are also saved as part of each Configuration. When you load a Configuration the settings will be restored to whatever they were when the Configuration was saved. To open the Preferences window select 'Preferences' from the 'junXion v4' menu or 'cmd + ,'.

Inputs



Human Interface Inputs

The HID Inputs group box allows you to adjust two settings:

Enable non-joystick devices

By default junXion v4 only displays HI Devices which register as joystick type devices. However there are many devices which register as other device types such as mice, keyboards and gamepads. When this checkbox is checked, junXion v4 will also included non-joystick devices such as the Mac's keyboard and mouse in the devices list. The first time junXion v4 is started it will enable this option if no joysticks are found so that the device list is not empty.

maximum amount of sensors / device

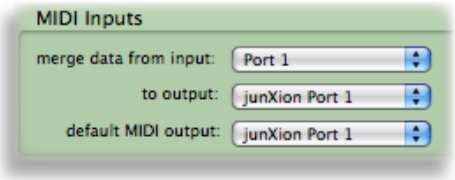
This setting allows you to limit the maximum number of sensors that junXion v4 will scan for each device. Some devices such as the keyboard have an enormous number of sensors, most of

which you will never use. This option also helps you to limit the lengths of each device's Input sensor list to a manageable length.

MIDI Inputs

The MIDI inputs group box allows you to change the following three settings:

merge data from input



This option allows you to select a MIDI input port whose messages will be merged with junXion v4's generated MIDI output.

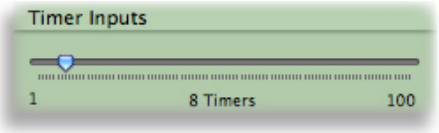
to output

This option allows you to select the destination MIDI port for data coming from the MIDI port selected above. The pop-up menu displays a list of all available MIDI ports found by junXion v4 at startup. The list also includes two virtual ports 'junXion Port 1' and 'junXion Port 2' which junXion v4 creates so that you can send MIDI events to other applications. If the Quicktime synthesizer is enabled (see below) it will also be listed.

default MIDI output

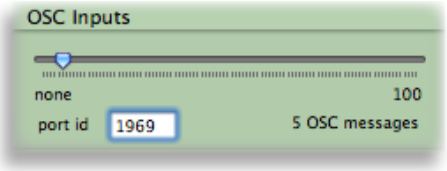
This setting specifies the MIDI output port which will be used by default in newly created Actions.

Timer Inputs



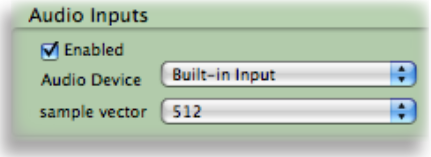
The Timer Inputs group box allows you to specify the number of Timers to use in the Configuration. The slider is adjustable between 1 and 100, and defaults to 8.

OSC Inputs



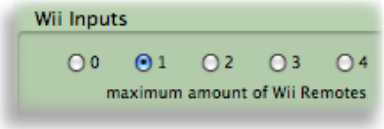
Here you can set the amount of OSC Inputs available in your configuration. The slider is adjustable between 0 and 100, and defaults to 0. Addidtionally, the OSC Input port ID must be set here, to allow communicaction with your OSC output device. This port ID is a freely choosable four digit number.

Audio Inputs



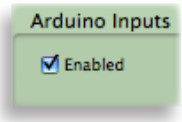
Enabling Audio Inputs activates junXion v4's pitch and envelope followers. Selecting an audio device from the menu will activate ALL available input ports of this device, constantly reading out pitch and level information. Note that this may cause some CPU load, thus this feature should always be disabled when not used in the current configuration. The sample vector defines the amount of samples which are read out to derive control. A large sample vector will result in more accurate data at a higher latency.

Wii Inputs



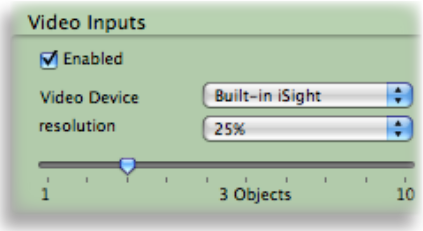
A maximum of four Wii Remotes (+ Nunchuck) is supported in junXion v4. The Wii Remote communicates with the computer via Bluetooth, which should obviously be available and active on your machine in order to use this feature. Setting the amount of Wii Remotes to zero disables this Input and saves some CPU power.
More detail on connecting the Wii and Trouble Shooting can be found in the Inputs part of this Manual.

Arduino Inputs



If this checkbox is checked you will be able to use Arduino Boards in junXion v4. The application will need a few seconds longer to start up in this case.

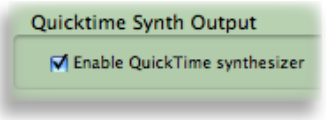
Video Inputs



Enabling Video Inputs activates Object and Color Tracking in junXion v4. The menu shows all the Video devices attached to your computer when you have started up junXion v4. The resolution menu allows you to select either 100%, 50% or 25% of the device's proper resolution. When using 100%, junXion has to process 16 times as much data as when set to 25%. Most DV cameras have a native resolution of 640 x 480, so 50% gives you 320 x 240, and 25% gives you 160 x 120 pixels. Depending on your computer configuration (CPU speed) and your resolution needs you have to decide what setting you want to use.
The slider below tells junXion how many video objects you want to use, your maximum is 10.

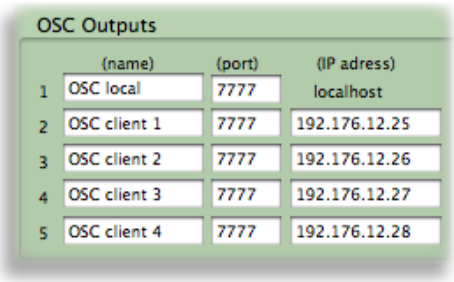
Outputs

Quicktime Synth Output



The Quicktime Synth group box gives you the option to enable the Quicktime synthesizer. When enabled, MIDI data generated by junXion v4 can be sent to your Mac's built-in General MIDI synthesizer. It will show up in the Action's 'output event' menu. This is a relatively powerful GM synth, which among other things always provides a GM drum set on MIDI channel 10. Keep in mind that the Quicktime synth consumes some CPU power and hence should be disabled when not needed.

OSC Output



Five Output ports for outgoing OSC data can be set here, one of which is a local port. You do not have to specify an IP adress for this localhost port. Since OSC is a network protocol you have to do so for your external ports. Additionally you have to give every port its own port ID, a four digit number as in the OSC Input section. If you want you can also make up a name for every port.

Preferred Data Range

It can be useful to set a different data range than the MIDI range which only generates values in between 0-127. For example: if you want to change the step length of a Timer with a Table entering specific numbers, the Data Range displayed in the Table should be adjusted to the Timer's milliseconds range which is between 1-9999. Keep in mind that even in the rather limited data range of MIDI you will not lose any values of your Input Device which might offer a higher resolution since the Input values are always scaled down to floating point numbers within the preferred data range.