Welcome to junXion version 5.3! This document describes all the additions and changes since v4.6 and assumes you are already familiar with that version. If you are new to junXion, please check out the junXion Manual and the manual additions first.....

General

• Starting up junXion v5.3

When you start the application itself, junXion will immediately ask you if you want to create a new Configuration or open an existing one. When you open an existing one, junXion will automatically make a backup of that file, named "xxxxxxxbackup.j4xml". Your opened Configuration will be automatically updated every minute if you make any changes in it. So in general, saving your work is now completely automatic. In previous versions junXion would automatically also create a file with the name "~xxxxxxx.j4xml", this feature is removed since it is obsolete now.

• Quitting junXion

When you made a change just before you quit, junXIon will still ask you if you want to save those changes.

• Help tags

By default now, junXIon will display little help tags when you hoover your mouse above buttons, labels, in other words, the user interface. These Help tags can be switched of by unchecking the **Show help tags** menu, found in the **Options** menu.

• GUI redundancy

In junXion you create Patches by dragging an input sensor in the Patches' view Input Sensors column. Now you can also select an input sensor and click on the **Create Patch** button on top of the browser. If an editor is available for the input sensor type (such as with Timers), you can either double-click on the input sensor or click on the button named **Edit**.

• some changes

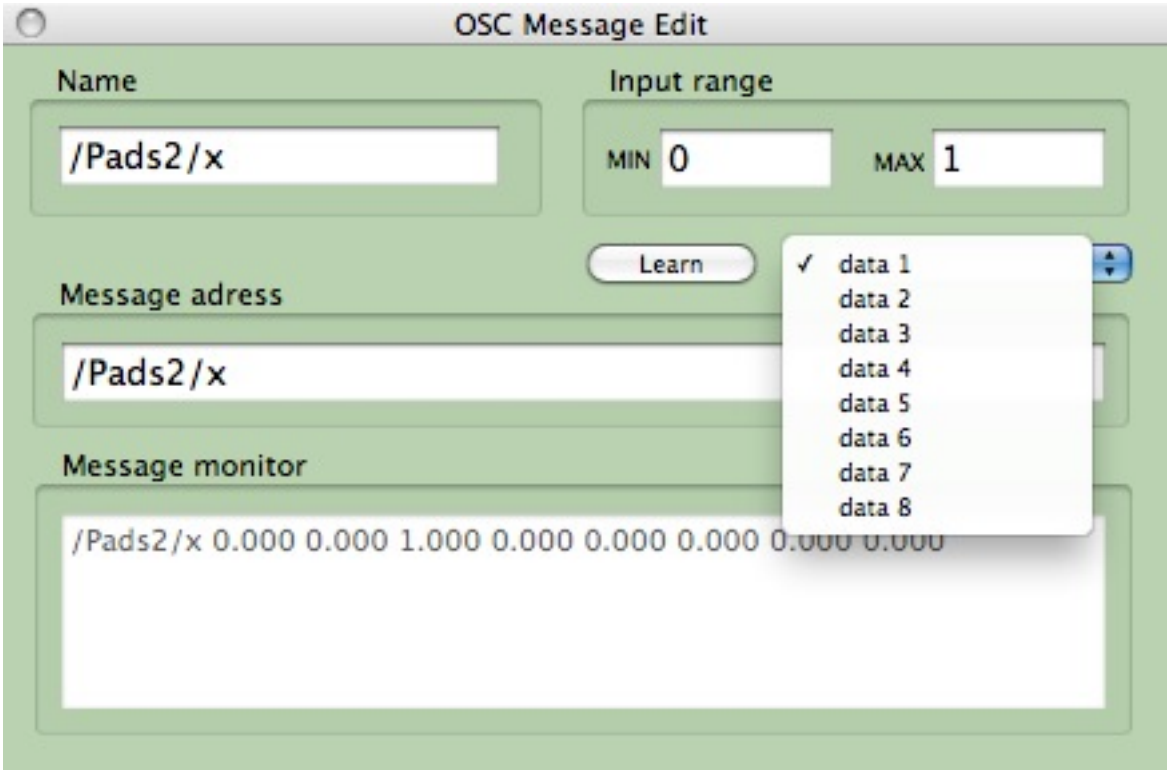
the *Root State* has been renamed to *global state*, since this State is always active. A lot of bugs have been fixed, program has been tested on OSX 10.5,10.6 and Lion.

OSC implementation

• OSC input

In junXion v4.6 the OSC input was seriously limited. Only one parameter messages would be properly decoded, any message more complicated would not work properly.
The OSC input implementation has been greatly improved, meaning that powerful control interfaces like the Lemur (for iPad) or TouchOSC can be used properly as OSC input sensors and their messages can be processed in junXion and for example be translated into MIDI, to be used with Live, Logic, etc.

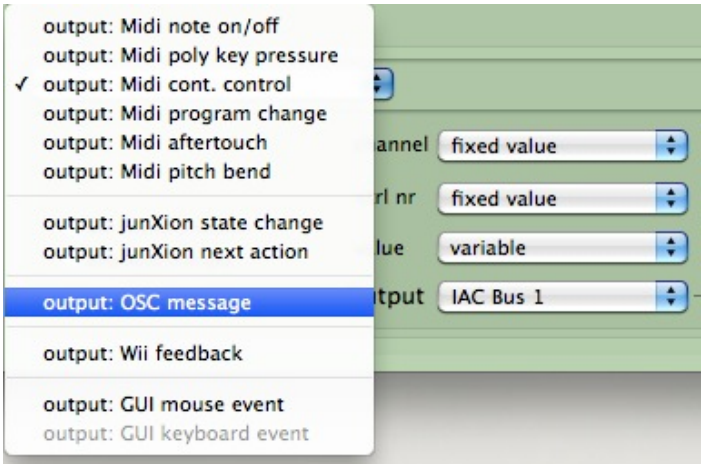
When opening the OSC message editor by selecting an OSC message in the input browser and clicking on the **EDIT** button, the parameter popup menu will be variable in size, depending on how many parameters it recognizes after you click the **Learn** button.



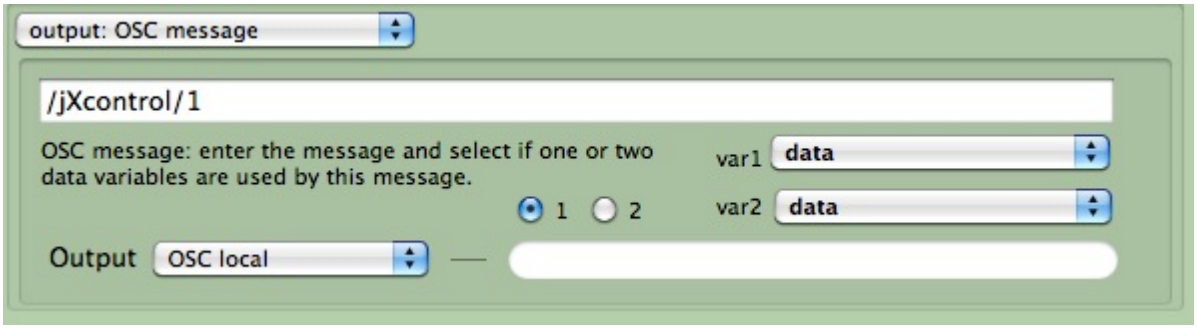
In the shown example, the OSC message contains 8 parameters and you use the popup menu to select which of these parameters you want to use in this OSC input message. That means that for this shown example message, you would need 8 OSC Messages as input sensors, each using another parameter.
junXion v5.2 can deal with OSC messages that can contain up to 16 parameters.

• OSC output

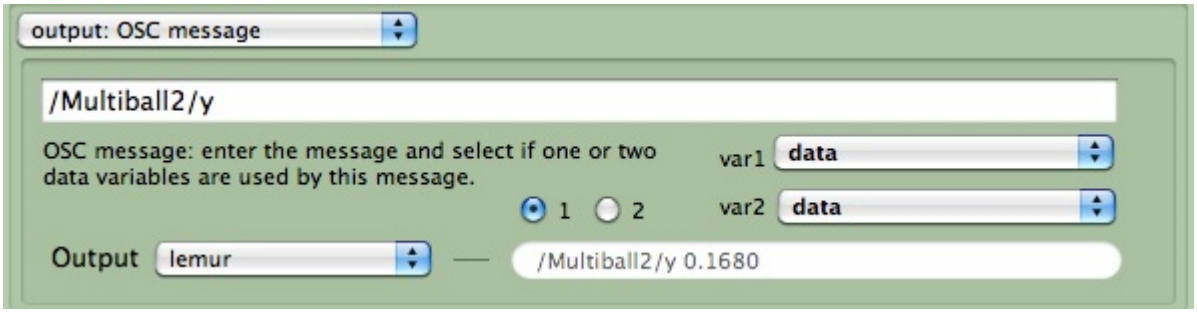
In junXion v4.6 the OSC output was seriously limited. You could select either the message jXswitch or jXcontrol to be sent in a Patch, and if your receiving OSC app or device had no learning option, this was useless...
Starting with junXion v5.2, you can define your own OSC message when using OSC as the output of an Action.
When you want to use OSC as output, select **output: OSC message** from the **Output** popup menu (which has been slightly re-arranged since v4.6)



By default, the message will send out /jXcontrol/1 with one data parameter.
You can enter your own message in the message field and choose if you want to send one or two parameters (data variables) with this message.



So for example:



This message will control the second bouncing XY ball pad in the Lemur factory project named **iPad-Bouncing XY Balls**. The OSC output port named **lemur** has been configured to send its OSC messages to the Lemur iPad software (which btw. always listens to port 8000).
junXion's limitation will be that you can send a maximum of two variable parameters with the created OSC message. The freedom however is that each Action can send its own defined OSC message.

Actions

• browser sorting

A new feature has been added to v5.3, which is the possibility to change the order of the Actions in the browser as to suit your needs. The re-ordering will also be reflected in the Action menus in the Patches view and the **Next Action** popup menu. The new order will be stored into your Configuration so it is permanent. Click-drag an Action that you want to re-order in the browser and release it on top of a new selected one.
Keep in mind that when dragging upwards, the dragged Action will be inserted before the Action that you dropped it on, and when dragging downwards the dragged Action will be inserted after the one you dropped it on.

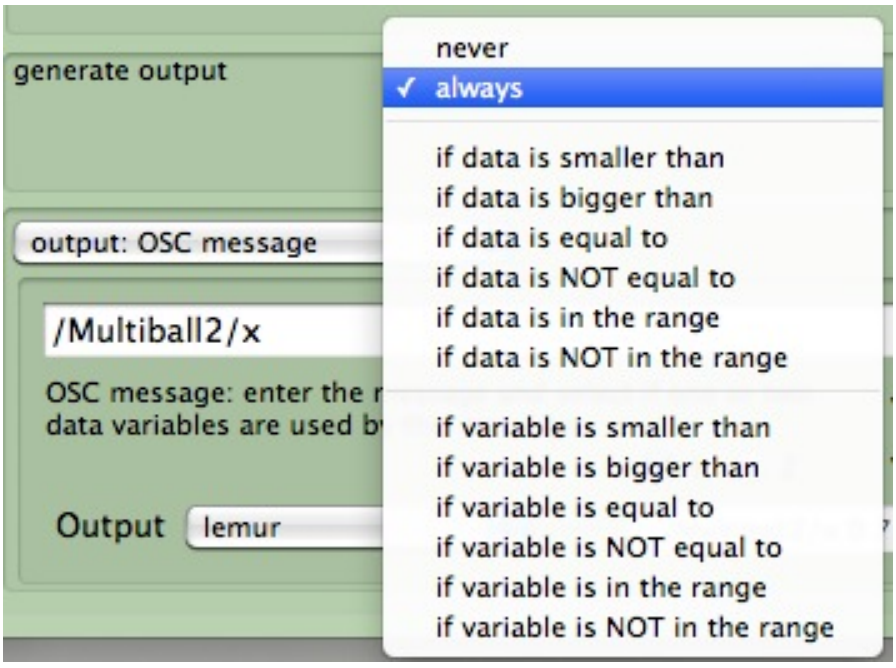
• change

In junXion's Action view, there is a new menu item added to the **change** menu, called ***limit data***.

In the above example, the processed data by **behavior** will be scaled / limited to a maximum value of 75.00, this is no hard clipping but really limited scaling. You may also dynamically limit your data by using a variable instead of a fixed value.

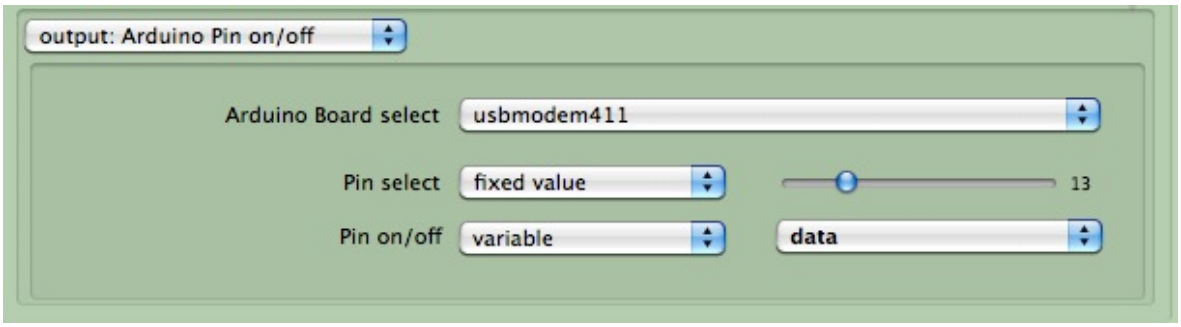
• generate output

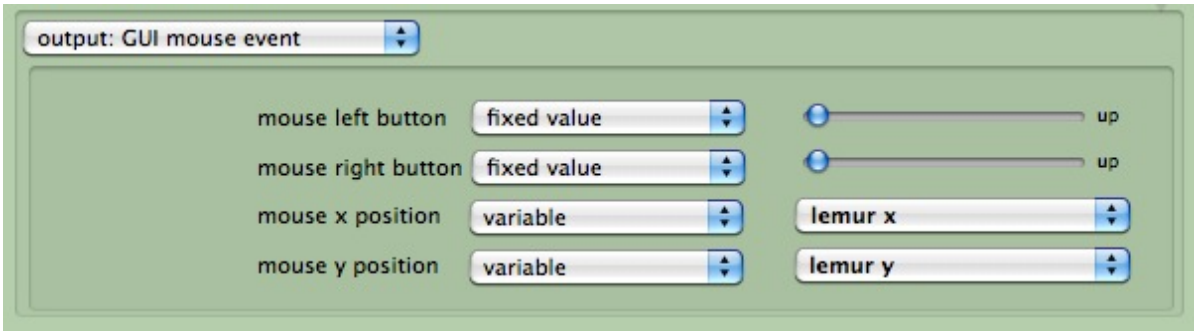
this popup menu has a few menu items added, previously the only condition could be set by comparing a variable, now you can also compare data, meaning almost the same conditioning is possible here as in the **continue** pane.



- **output: GUI mouse event**

  a new menu item has been added to the output menu, named **GUI mouse event**. This output allows you to remotely control your Mac's mouse cursor, independantly of your mouse or trackpad.





In the above example, the Lemur's x and y parameters of one of the bouncing balls are used to control the computer mouse's x and y. Besides the x and y you may also send left and right mouse click events. These mouse events are genuine Apple mouse events, meaning you can remotely control your computer this way. Of course you may not always want this control, in that case hit the Escape key on your Mac's keyboard to 'disconnect' your input sensor from the mouse event control. To re-establish control again, select the **Enable remote mouse control** from the **Options** menu. By default, this option is checked when you start up junXion with your Configuration.

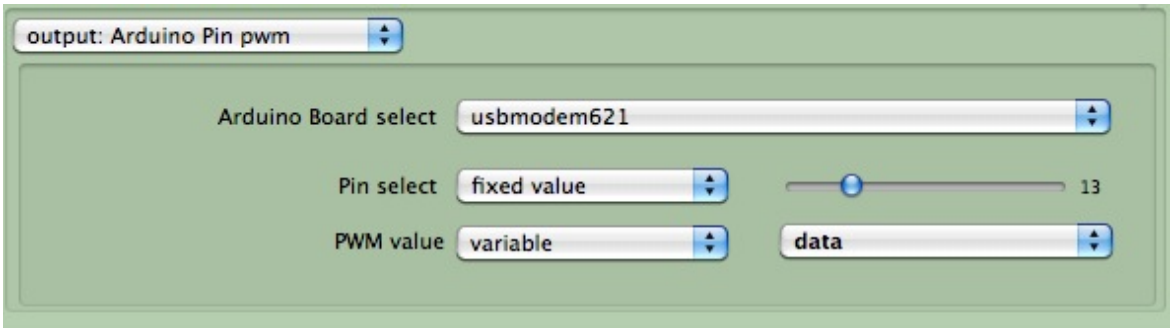
• output: Arduino Pin on/off

a new menu item has been added to the output menu, named **Arduino Pin on/off**. This output allows you to send digital pin on/off messages to a connected Arduino board, allowing you to control things like LED on and off, relays on/off, etc.

- The *Arduino Board select* menu will show you all the Arduino boards connected to your computer.
- The *Pin select* part allows you to define which digital pin on the Arduino board you want to send a message to, you can select Pin 0 - 63 with the slider or use a variable for Pin selection.
- The *Pin on/off* part can determine if the selected Pin's status should be On or Off, this can be either fixed or variable.

- **output: Arduino Pin pwm**

  another new menu item has been added to the output menu, named **Arduino Pin pwm**. This output allows you to send digital pin pulse width modulation (pwm) messages to a connected Arduino board, allowing you to control things like LED brigtness, motor speed, etc. Instead of just on or off, an 8-bit value can be send, a number in the range of 0 - 255.



  - The *Arduino Board select* menu will show you all the Arduino boards connected to your computer. Select which board you want to send the pwm event to.

  - The *Pin select* part allows you to define which digital pin on the Arduino board you want to send the message to, you can select Pin 0 - 63 with the slider or use a variable for Pin selection.

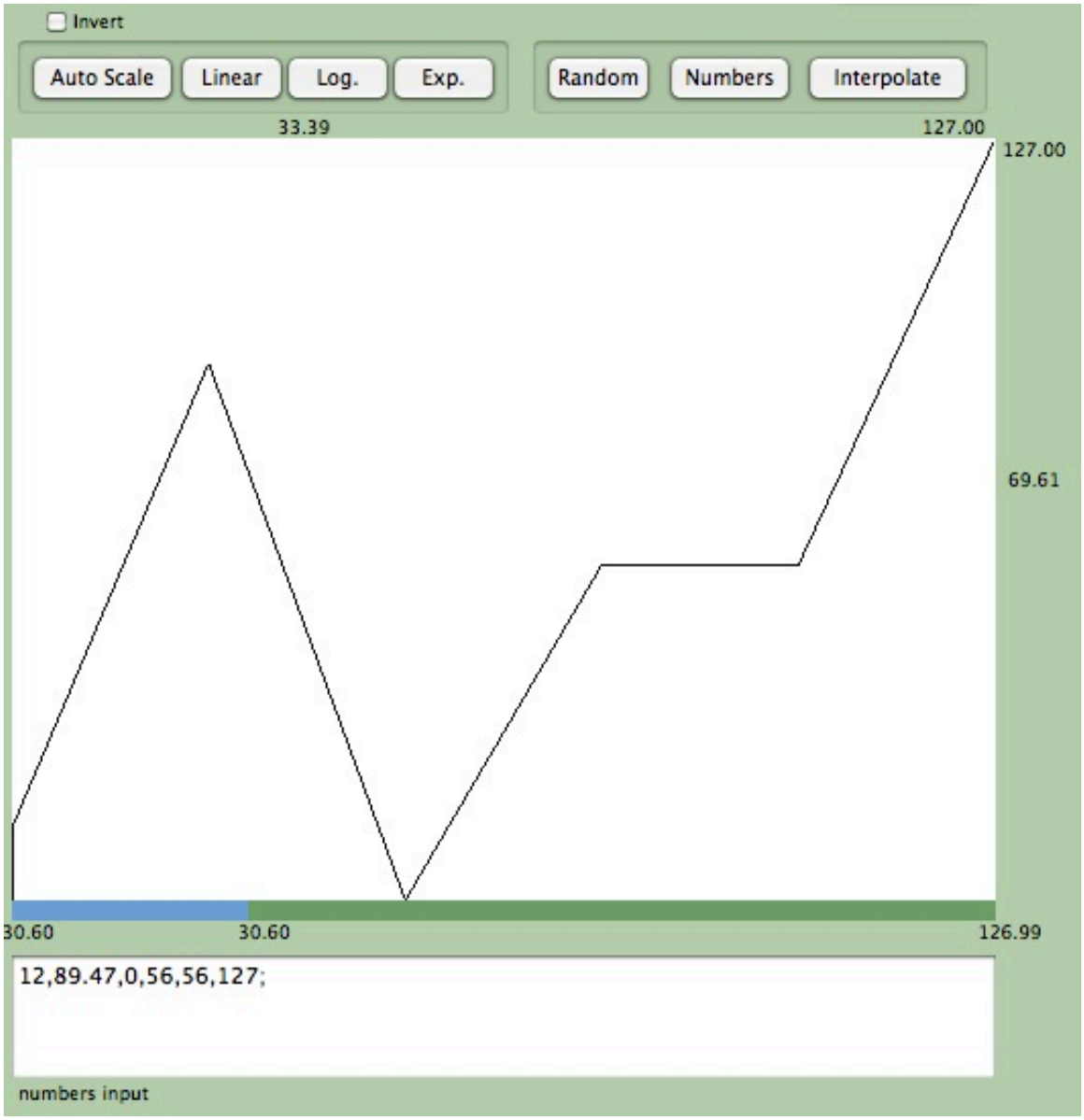
  - The *PWM value* part is used determine the selected Pin's pwm value, this can be either fixed or variable.

Be aware that not all Pins on an Arduino board are suitable for output as switch or as pwm, check out http://arduino.cc to find out which pins are available for your Arduino board. If you try send one of these messages to the wrong Pin, nothing will happen.

Tables

• Interpolate

A new button has been added to generate an interpolated Table. This feature needs at least two numbers entered in the *numbers input* pane, seperated by a comma and ended with a semicolumn.

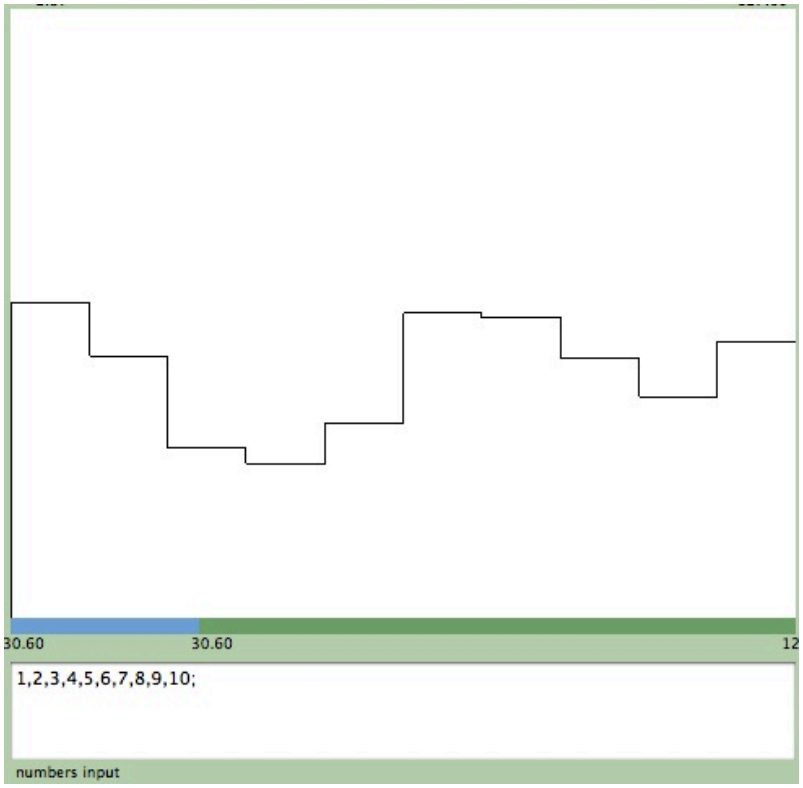


In the shown example six numbers were entered and after that by clicking on the **Interpolate** button the Table was generated, creating 5 line segments between the 6 entered numbers.

• writing data in a Table

Writing data in a Table always caused some issues, since the Tables by default contain 65536 values, and indexing is usually done in the MIDI range, so with 128 different indexes. Starting from junXion v5.2 the program tries to recognize how many entries you actually need in a Table and when writing a value in it will actually write a little line segment in the Table.

So in this example, first a Table was generated using the **Numbers** button after entering the numbers 1 to 10 in the *numbers input* pane.

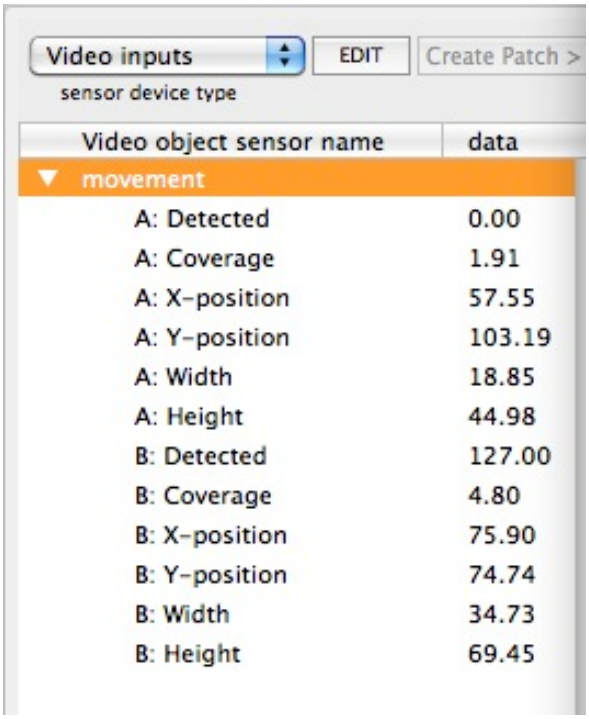


In the Action's **store** pane the menu item ***variable in table*** was selected and this uses the data of variable 'lemur y' and writes it in the 'Invert' table with 'lemur x' as its index. The resulting Table will always have just 10 different segments, since junXion recognized the original Table to contain just ten different values (i.e. ten line segments). This is a great feature if for example you want to store your latest faders positions in a multifader TouchOSC setup. This way you can recall them later and send the values back to the TouchOSC app.

Video tracking

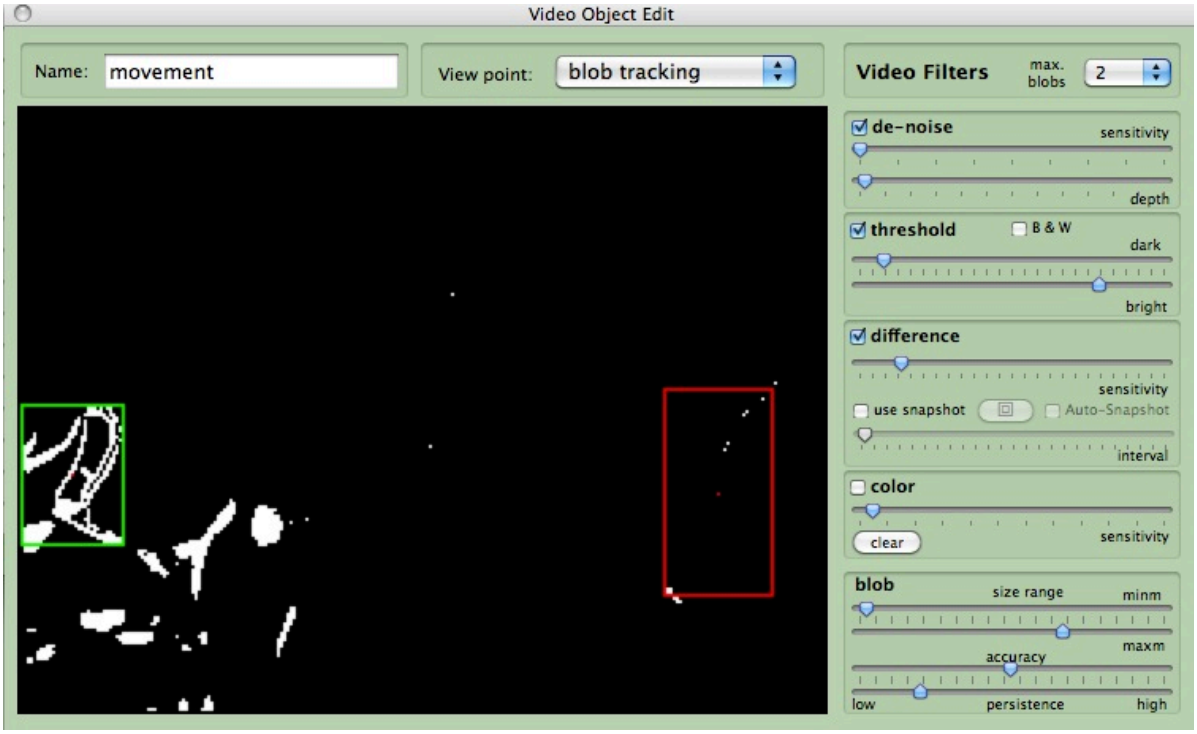
• multi tracking

In junXion v5.2, the Video input Editor window has been slightly re-arranged and renamed and also gives you the possibility to extract up to two so-called ‘blobs’ per Video object. The result is that in the Video input sensor browser each object will now show twelve input sensors, six starting with A: and six starting with B:.
These input sensors represent the data coming from ‘blob’ A and ‘blob’ B, if any.



In the example the object named ***movement*** has detected blob B and is showing the input sensor values as last received by both blobs.

In the **Video Filters** pane you can select either one or two blobs to be extracted from the filters.



The filters are now ordered in such a way that makes more sense to the natural usage order. Normally you pre-process the Video image in the **de-noise** filter

(previously named despeckle), after that you may want to filter out the dark and bright areas in the image (**threshold)** and then usually you use either the **difference** or **color** filter for blob extraction.

Two new faders have been added to define the blob behavior, **accuracy** and **persistence**. When you need a lot of accuracy the responsiveness will be slower. With the persistance fader you may determine the time junXion tries to keep the blob ‘alive’, also slowing down responsiveness. Try different combinations to find the setting that suits you the best.