

## MALINETTE OBJECTS 1.0

Developped by Reso-nance numérique

### VIDEO

**track-areas** : Track changes of luminosity in areas  
**track-color** : Track a color area  
**track-fiducials** : Track 6 fiducial markers  
**track-motion** : Track motion of pixels (blob)  
**video-camera** : Get video from a camera  
**video-color** : Set color balance  
**video-crop** : Get a part of an image (crop)  
**video-effects** : Video effects : inversion - contrast - blur - ...  
**video-eye** : Control the camera positions  
**video-graph** : Display a line like an oscilloscope  
**video-halftone** : Halftone (print point) effect  
**video-images** : Display images  
**video-kalei** : Kaleidoscope effect  
**video-larsen** : Feedback video  
**video-lumaoffset** : Offset pixels depending on the luminance  
**video-mix** : Mix two sources of video  
**video-object** : 2D and 3D objects : cube - sphere - 3d model ...  
**video-out** : Display video elements on the screen (rectangle)  
**video-rec-file** : Record video into a file  
**video-rec** : Record videos in live  
**video-sampler** : Record and play video  
**video-text** : Display text  
**video-xyz** : 3D rotate and translate

### SEQ

**chrono** : Simple chronometer  
**seq-count** : Basic sequencer  
**seq-count2** : Sequencer with more graphical actions  
**seq-nb** : Sequencer with 8 steps of number  
**seq-pulse** : 16 steps sequencers with 3 states  
**seq16-count** : 16 steps sequencer  
**seq16** : 16 steps line  
**seq16-nb** : 16 steps sequence of numbers  
**seq16x3** : Matrix of 16 steps lines  
**seq24-count** : 24 steps sequencer  
**seq24** : 24 steps line  
**seq24-nb** : 24 steps sequence of numbers  
**seq24x6** : Matrix of 24 steps lines  
**seq64-count** : 64 steps sequencer  
**seq64** : 64 steps line  
**seq64x6** : Matrix of 64 steps lines  
**seq64x9** : Matrix of 64 steps x 9 lines  
**seq8-count** : 8 steps sequencer  
**seq8** : 8 steps line  
**seq8-nb** : 8 steps sequence of numbers  
**seq8x3** : Matrix of 8 steps lines

### IN

**arduino-in** : Get analog sensors from an Arduino  
**arduino-in-num** : Get digital sensors from an Arduino  
**joystick** : Get the joystick device  
**key-char** : Get the 10 first alphabetic keys from the keyboard.  
**key-makey** : Get keys from a MakeyMakey device  
**key-num** : Get 10 numerical keys from a keyboard.  
**mouse** : Get the X and Y positions from mouse device

### AUDIO

**audio-attack** : Trigger a sound with a basic envelope  
**audio-bank** : Load a folder of sounds  
**audio-chorus** : Chorus effect  
**audio-comp** : Waveshaping compressor  
**audio-delay** : Simple delay  
**audio-delay4** : 4 bands audio delay  
**audio-disto** : Simple distortion  
**audio-eq13** : Equalizer 13 bands  
**audio-eq3** : Simple 3 band equalizer  
**audio-filter** : Classic audio filter cutoff/resonance  
**audio-flanger** : Flanger effect  
**audio-formant** : Materials and vowels synthesis  
**audio-freeze** : Freeze the sound  
**audio-in** : Get audio input - pitch and sound level  
**audio-lowfi** : Decrease sound quality  
**audio-map** : Map audio signal  
**audio-mix** : Mix two sounds  
**audio-moog** : Moog filter effect  
**audio-nb** : Convert audio to numbers with mapping  
**audio-out** : Audio bus send audio to the [audio-master]  
**audio-pan** : Panning stereo  
**audio-phaser** : Phaser effect  
**audio-pitch** : Pitch the sound  
**audio-rec-file** : Record stereo sound to a file  
**audio-rec** : Record mono sounds  
**audio-reverb** : Reverberation  
**audio-sampler** : Play and record sounds (mono)  
**audio-vocoder** : Vocoder effect  
**audio-vol** : Simple volume mixer  
**audio-vol-st** : Simple volume mixer  
**fftscope~** : See frequencies spectrum of an audio signal  
**hat2** : Simple percussion  
**oscilloscope-big~** : See audio signal in time (big)  
**oscilloscope~** : See audio signal in time  
**out~** : Simple audio out  
**smp-vocoder** :  
**synth-bass** : Bass synthesiser  
**synth-drums** : Synthesis sound of 4 drums : kick - snare - hat - crash  
**synth-emu** : transistor bass emulation (moog ~ tb303)  
**synth-fm** : FM synthesis  
**synth** : Waveforms synthesizer  
**synth-material** : Sound synthesis from physical properties of materials  
**synth-pluck** : String pizzicato synthesis  
**synth-wobble** : Dubstep bass with nice LFO filtering  
**synth-xylo** : Xylophone synthesis.  
**synth~** : Waveforms synthesiser (Hertz version)

### OUT

**arduino-out** : Send data to Arduino outputs : digital + PWM + servo  
**midi-out** : Send MIDI notes  
**osc-out** : UDP/OSC connection

### NUMBERS

**bang-line** : create value ramps in a specified delay with optional random mode  
**bangs** : Send a serie of delayed bangs  
**between** : Let pass numbers between the 2 values  
**bpm-ms** : Convert BPM (Beat per minute) to Ms (Milliseconds)  
**c** : Endless minimize counter  
**chance** : Percentage chance of action  
**count** : Simple counter  
**data-sampler** : Record and play a flow of data from a file  
**fade** : Fade between two values  
**flow** : Switch data flow  
**for** : Send a serie of numbers  
**g-editor** : A tool to edit graphs (add-subtract-etc.)  
**g** : Read and write data with a graphical array  
**g-play** : Read and write data with a breakpoints array  
**g-sampler** : Read and write data with a timeline  
**inverse** : Inverse the value  
**logic** : logic operations and conditions  
**m** : Store and write data (minimalist version of mem)  
**map-auto** : Automap useful for sensor calibration  
**map** : Map a number between the boundaries (rule of three)  
**mem** : Store and read data from a file (presets)  
**mem16** : Memorize presets for seq16  
**midi-hz** : Conversion between Midi and Hertz  
**midi-sampler** : Play and record into a midi file  
**multimeter** : Calculate the Resistance and the Voltage of a sensor  
**notescale** : Convert midi numbers to musical scale  
**notescale2** : Convert midi numbers to musical scale  
**onoff** : Convert a bang to a switch timer  
**oscilloscope** : See data values in time  
**physics** : Simulate physic characteristics like elasticity and gravity  
**rand-bangs** : Send a random number of delayed bangs  
**rand-del** : Random delay into a range time  
**rand** : Output a random value  
**rand-n** : Random N numbers into a range of numbers  
**rand-range** : Random a min and a max value inside a range  
**rand-sel** : Send a bang after a random number of bang  
**rand-walk** : Random walk generator  
**start** : Trigger a bang at startup  
**taptempo** : Get tempo between two triggers