

Digital Orchestra Toolbox

Download Links

[↗ TOOL SITE](#)

This “toolbox” is a collection of Max/MSP abstraction that we have found useful in the IDMIL for creating gesture processing patches for digital musical instruments. Each abstraction is accompanied by a help patch to demonstrate its use. Authors: Joseph Malloch, Stephen Sinclair, and Marlon Schumacher @ the Input Devices and Music Interaction Lab, McGill University.

Collision Detection	
dot.alloc	Using a shared bus, deliberate to allocate a unique identifier.
dot.alloc2	Using a shared bus, deliberate to allocate a unique identifier. Peer with root index suggests identifiers to newcomers.
Control	
dot.for	Outputs a sequence of incremented numbers when banged.
dot.line	Wrapper for the line object so it accepts trajectory lists like line~.
dot.probgate	A probabilistic gate with remote control.
dot.repeat	Repeats a message a specified number of times.
dot.route~	Separates signals from max-messages.
dot.swap	Like the swap object, but for symbols and lists too!
Data	
dot.atoi	Convert a symbol that starts with a decimal number to an integer.
dot.bytetobits	Converts a single decimal byte into 8 binary bits.
dot.bitstobyte	Converts 8 bits into a single decimal byte.
dot.filein	Adds dump command to the filein object.
dot.getindex	Retrieves indexes of coll data entries that match the query.
dot.index	Generates the lowest unused index (for coll storage). Argument sets maximum index.
dot.playabsolute	Play files recorded by dot.recordabsolute .
dot.properties	regexp wrapper for parsing tagged message properties expressed in the form @<label> <data>. Second outlet outputs number of backreferences.
dot.recordabsolute	Record an arbitrary number of datastreams with absolute timestamping.
dot.reg	Like “zl reg” but right outlet bangs when empty.
dot.sparkline	Draws a sparkline from a list or stream onto an lcd object.
dot.typecheck	Route input according to data type.
dot.urn	Generate random numbers without duplicates (like “urn”), but you can put numbers back in the pot.

<code>dot.xmlread</code>	A native-Max XML parser – no externals!
<code>dot.xmlwrite</code>	A native-Max XML parser – no externals!
Filters	
<code>dot.asyncdemod~</code>	Asynchronous demodulation of baseband signal from an AM-carrier.
<code>dot.attackslope</code>	Given two thresholds, determine the slope between the points at which they are crossed in the positive direction. (i.e., attack speed)
<code>dot.autoscale</code>	Rescales signals according to auto-detected maximum and minimum values.
<code>dot.autoscale~</code>	Automatically scale incoming signal to a defined output-range.
<code>dot.boundary</code>	Mutes, clamps, wraps, or folds a stream of numbers at a predefined minimum and/or maximum value.
<code>dot.cartopol3</code>	Converts 3D Cartesian coordinates to polar representation.
<code>dot.centre</code>	Automatically offsets input to re-centre signal around zero, with user-definable delay, ramp time, and time grain.
<code>dot.change</code>	Just like the change object, but works for symbols and lists.
<code>dot.clip</code>	Clips a stream of numbers to a minimum, maximum, or both.
<code>dot.dampedenvelope</code>	Audio-rate envelope-generator with damping.
<code>dot.distance</code>	Finds maximum or minimum distance between a scalar and a list.
<code>dot.dynamicexpression</code>	User-definable expr-based scaling with autoscale.
<code>dot.extrema</code>	Outputs values of local maxima and minima.
<code>dot.fraction</code>	Looks at divisions of a list and outputs the highest value corresponding to nodes at 2 3 4 5 and 6 part divisions. Intended for use with <code>dot.harmonicfilter</code> .
<code>dot.fromsignal~</code>	Samples a signal triggered by change.
<code>dot.harmonicfilter</code>	For damping gains in modal synthesis based on a harmonic series. Handles “string division” at 0.5, 0.3, 0.25, 0.2, 0.167
<code>dot.history</code>	Outputs list of delayed samples: $x[n]$, $x[n-1]$, ..., $x[n-m]$.
<code>dot.hz↔samp~</code>	Converts between frequency in hz (cycles/second) and samples/cycle.
<code>dot.interpolate4~</code>	A sort of spectral interpolation of 4 signals via magnitude and phase vector.
<code>dot.jab</code>	Detect “jabbing” gestures in acceleration data.
<code>dot.leakyintegrator</code>	An accumulator with a hole in it. The leakiness is highly customizable.
<code>dot.leakyintegrator2</code>	Integrator with a leak... handles floating-point numbers, signed values, leak expressions.
<code>dot.mass-spring</code>	Implements a simple mass–spring-damper model.
<code>dot.median</code>	Outputs median value of a user-definable sample size.
<code>dot.mix4~</code>	mixes four signals via XY-coordinates.
<code>dot.normalize</code>	Normalizes a list of ints or floats, or a windowed stream.
<code>dot.polar</code>	Converts x/y into amplitude and angle. Change in angle with wrap-around correction is also calculated.
<code>dot.poltocar3</code>	Converts 3D polar coordinates to Cartesian representation.
<code>dot.rad→norm~</code>	Scales 2pi-radians to normalized range (0-1) with optional wraparound.
<code>dot.region</code>	Outputs and centre of multiple selected areas of a list (binary).
<code>dot.scale~</code>	MSP version of Max’s scale-object (with ‘proper’ exponent).
<code>dot.schmitt</code>	A trigger with hysteresis.

<code>dot.schmitt~</code>	Detect triggers from envelope of a signal.
<code>dot.signaccum</code>	Accumulates positive vs. negative same-sign deltas.
<code>dot.slope</code>	Output the slope between each successive point.
<code>dot.smooth</code>	Simple sample-averaging filter.
<code>dot.split</code>	Right outlet if greater then threshold, left outlet otherwise.
<code>dot.thresh</code>	A little hack to allow thresh to work with symbols.
<code>dot.threshtrig</code>	Output a value only once after passing the threshold in the positive or negative directions.
<code>dot.timedextrema</code>	Outputs the minimum and maximum value received with the last n milliseconds.
<code>dot.timedsmooth</code>	Downsampled audio-rate averaging filter.
<code>dot.timedsmooth2</code>	Time-windowed averaging filter in which each input sample has the same weight.
<code>dot.transfer</code>	Table-based waveshaping with customizable transfer function.
<code>dot.unwrap</code>	Assumes input is polar and lies between arg1 and arg2. Output is the shortest polar distance from the last sample.
<code>dot.vscale</code>	Just like the scale object, but for vectors. Includes clipping-feature.
<code>dot.windowedextrema</code>	Outputs the minimum and maximum value received with the last n samples. Also outputs the order of the extrema in the window.
<code>dot.wrap</code>	Simple offset with wrap-around.
List Processing	
<code>dot.listinterpolate</code>	Interpolates/extrapolates between two lists of equal length.
<code>dot.listinterpolate4</code>	The same for four lists.
<code>dot.listpipe</code>	Delays a stream of input like pipe, but also works for lists.
<code>dot.matchNth</code>	Outputs message if the nth item matches the argument.
<code>dot.matcNth</code>	Outputs message if the nth item address pattern matches the argument.
<code>dot.nth</code>	Works like zl nth, but can match multiple indexes.
MIDI	
<code>dot.combinote</code>	Combines 8 input signals into a single value such that any given combination will be a unique number.
<code>dot.MIDIout</code>	Helper for quickly configuring MIDI output.
<code>dot.MIDIpedal~</code>	Detect triggers from a MIDI-pedal through an audio-input.
OSC	
<code>dot.appendaddr</code>	Append text to the end of the first item of a list.
<code>dot.OSCalias</code>	Shortens long OSC addresses by giving them aliases.
<code>dot.OSCcompress</code>	Simply removes spaces in OSC address strings.
<code>dot.OSCexpand</code>	Simply expands OSC address strings so that the route object can parse them.
<code>dot.OSCroute</code>	Native-max OSC parser allowing multiple OSC addresses to be dynamically added and removed. Arguments or right inlet set addresses to route. Left outlet outputs matches with index corresponding to argument order.
<code>dot.OSCunalias</code>	Restores OSC addresses that have been aliased using dot.OSCalias.
<code>dot.prependaddr</code>	Prepend text to the beginning of the first item of a list.
Quaternions	

dot.jit.quaternion.conjugate	Calculate the conjugate of a quaternion stored in the planes of a jitter matrix.
dot.jit.quaternion.inverse	Calculate the inverse of a quaternion stored in the planes of a jitter matrix.
dot.jit.quaternion.multiply	Multiply two quaternions stored in the planes of a jitter matrix.
dot.quaternion.conjugate	Calculate the conjugate of a quaternion.
dot.quaternion.inverse	Calculate the inverse of a quaternion.
dot.quaternion.multiply	Multiply two quaternions.
dot.quaternion.normalize	Normalize a quaternions.
dot.quaternion.SLERP	Performs Spherical Linear Interpolation between two quaternions.
dot.quaternion2axis	Convert quaternion to axis/angle representation.
Sensors	
dot.fqa	Factored Quaternion Algorithm for calculating orientation from magnetometer and accelerometer data.
dot.orient	Calculates absolute orientation from gravity and magnetic field vectors.
dot.wmp	Process the data from a Wii Motion Plus and remove gyro bias.
Serial	
dot.doubleSLIPdecode	Parses double-ended SLIP-encoded data with user-defined start, end, and escape characters.
dot.doubleSLIPencode	Encodes data using double-ended slip-coding with user-defined start, end, and escape characters.
dot.serial	An abstraction containing the serial object, with menu generation and built-in polling and repeated-reading functionality.
dot.SLIPdecode	Parses slip-encoded data with user-defined delimiter and escape character.
dot.SLIPencode	Encodes data using slip-coding with user-defined delimiter and escape character.
Statistics	
dot.aggregate	Calculates aggregate of a list or a windowed stream.
dot.covariance	Calculates covariance of two windowed streams of numbers.
dot.ema	Calculates the Exponential Moving Average of a stream of numbers.
dot.emd	Calculates the Exponential Moving Deviation of a stream of numbers.
dot.exemplarcovariance	Calculates covariance of a windowed stream of numbers and an example.
dot.phase→freq~	Instantaneous frequency estimation from phase deltas.
dot.sintrack~	Track magnitude, instantaneous phase, and instantaneous freq. of a sinusoid.
dot.std	Calculates the Standard Deviation of a windowed stream of numbers.
Timing	
dot.channelthresh	Combines ID-tagged channels into lists using a delay threshold. Like “thresh,” but keeps channel information.
dot.debounce	Filters multiple messages.
dot.randometro	Like the “metro” object but outputs randomly within a range.
dot.squeuedlim	Limits the speed of messages passing through like speedlim, but queued like zl queue.
dot.wait	bangs when values are over-threshold for wait time.

Additional URLs

PUBLISHED
AUG 14 2012



JOE MALLOCH



A B S T R A C T A U D I O C O N T R I B U T I O N D O C U M E N T S F I L E S F R I I D M J I T L I S T P R O C E S S M A P P M I D I O S Q U A T E I S E N S E R S T A T I T I M C S G



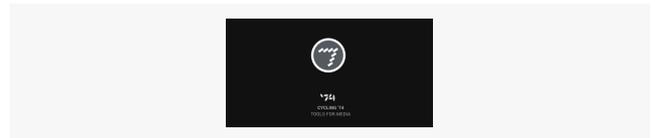
LOG IN TO REPLY

Explore More



XBee4Max

R O B I N A B E I O B 2 0 1 2



Binary, And not Binary, Counter

R O B E R T O A B U E G C E 2 R 1 R A 0 | 1 2

MORE ABOUT MAX

Features and Specifications

Made with Max

Max in Education

Terms & Conditions

About Cycling '74

© Cycling '74