Generic I2C

IIA / IIA address
Set I2C address or get the currently selected address

IIS cmd
Execute the specified command

IIS1 cmd value
Execute the specified command with 1 parameter

IIS2 cmd value1 value2
Execute the specified command with 2 parameters

IIS3 cmd value1 value2 value3
Execute the specified command with 3 parameters

IISB1 cmd value
Execute the specified command with 1 byte parameter

IISB2 cmd value1 value2
Execute the specified command with 2 byte parameters

IISB3 cmd value1 value2 value3
Execute the specified command with 3 byte parameters

IIQ cmd
Execute the specified query and get a value back

IIQ1 cmd value
Execute the specified query with 1 parameter and get a value back

IIQ2 cmd value1 value2
Execute the specified query with 2 parameters and get a value back

IIQ3 cmd value1 value2 value3
Execute the specified query with 3 parameters and get a value back

IIQB1 cmd value
Execute the specified query with 1 byte parameter and get a value back

IIQB2 cmd value1 value2
Execute the specified query with 2 byte parameters and get a value back

IIQB3 cmd value1 value2 value3
Execute the specified query with 3 byte parameters and get a value back

IIB cmd
Execute the specified query and get a byte value back

IIB1 cmd value
Execute the specified query with 1 parameter and get a byte value back

IIB2 cmd value1 value2
Execute the specified query with 2 parameters and get a byte value back

IIB3 cmd value1 value2 value3
Execute the specified query with 3 parameters and get a byte value back

IIBB1 cmd value
Execute the specified query with 1 byte parameter and get a byte value back

IIBB2 cmd value1 value2
Execute the specified query with 2 byte parameters and get a byte value back

IIBB3 cmd value1 value2 value3
Execute the specified query with 3 byte parameters and get a byte value back

Ansible

ANS.G.LED x y
get grid LED buffer at position x, y

ANS.G x y / ANS.G x y z
get/set grid key on/off state (z) at position x, y

ANS.G.P x y
simulate grid key press at position (x, y)

ANS.A.LED n x
read arc LED buffer for ring n, LED x clockwise from north

ANS.A / ANS.A n d
send arc encoder event for ring n, delta d

ANS.APP / ANS.APP x
get/set active app

KR.PRE / KR.PRE x
return current preset / load preset x

KR.PERIOD / KR.PERIOD x
get/set internal clock period

KR.PAT / KR.PAT x
get/set current pattern

KR.SCALE / KR.SCALE x
get/set current scale

KR.POS x y / KR.POS x y z
get/set position z for track z, parameter y

KR.L.ST x y / KR.L.ST x y z
get loop start for track x, parameter y / set to z

KR.L.LEN x y / KR.L.LEN x y z
get length of track x, parameter y / set to z

KR.RES x y
reset position to loop start for track x, parameter y

KR.CV x
get the current CV value for channel x

KR.MUTE x / KR.MUTE x y
get/set mute state for channel x (1 = muted, 0 = unmuted)

KR.TMUTE x
toggle mute state for channel x

KR.CLK x
advance the clock for channel x (channel must have teletype clocking enabled)

KR.PG / KR.PG x
get/set the active page

KR.CUE / KR.CUE x
get/set the cued pattern

KR.DIR / KR.DIR x
get/set the step direction

KR.DUR x
get the current duration value for channel x

ME.PRE / ME.PRE x
return current preset / load preset x

ME.SCALE / ME.SCALE x
get/set current scale

ME.PERIOD / ME.PERIOD x
get/set internal clock period

ME.STOP x
stop channel x (0 = all)

ME.RES x
reset channel x (0 = all), also used as “start”

ME.CV x
get the current CV value for channel x

LV.PRE / LV.PRE x
return current preset / load preset x

LV.RES x
reset, 0 for soft reset (on next ext. clock), 1 for hard reset

LV.POS / LV.POS x
get/set current position

LV.L.ST / LV.L.ST x
get/set loop start

LV.L.LEN / LV.L.LEN x
get/set loop length

LV.L.DIR / LV.L.DIR x
get/set loop direction

LV.CV x
get the current CV value for channel x

CY.PRE / CY.PRE x
return current preset / load preset x

CY.RES x
reset channel x (0 = all)

CY.POS x / CY.POS x y
get / set position of channel x (x = 0 to set all), position between 0-255

CY.REV x
reverse channel x (0 = all)

CY.CV x
get the current CV value for channel x

MID.SLEW t
set pitch slew time in ms (applies to all allocation styles except FIXED)

MID.SHIFT o
shift pitch CV by standard Teletype pitch value (e.g. N 6, V -1, etc)

ARP.HLD h
0 disables key hold mode, other values enable

ARP.STY y
set base arp style [0-7]

ARP.GT v g
set voice gate length [0-127], scaled/synced to course divisions of voice clock

ARP.SLEW v t
set voice slew time in ms

ARP.RPT v n s
set voice pattern repeat, n times [0-8], shifted by s semitones [-24, 24]

ARP.DIV v d
set voice clock divisor (euclidean length), range [1-32]

ARP.FIL v f
set voice euclidean fill, use 1 for straight clock division, range [1-32]

ARP.ROT v r
set voice euclidean rotation, range [-32, 32]

ARP.ER v f d r
set all euclidean rhythm

ARP.RES v
reset voice clock/pattern on next base clock tick

ARP.SHIFT v o
shift voice cv by standard tt pitch value (e.g. N 6, V -1, etc)

White Whale

WW.PRESET x
Recall preset (0-7)

WW.POS x
Cut to position (0-15)

WW.SYNC x
Cut to position (0-15) and hard-sync the clock (if clocked internally)

WW.START x
Set the loop start position (0-15)

WW.END x
Set the loop end position (0-15)

WW.PMODE x
Set the loop play mode (0-5)

WW.PATTERN x
Change pattern (0-15)

WW.QPATTERN x
Change pattern (0-15) after current pattern ends

WW.MUTE1 x
Mute trigger 1 (0 = on, 1 = mute)

WW.MUTE2 x
Mute trigger 2 (0 = on, 1 = mute)

WW.MUTE3 x
Mute trigger 3 (0 = on, 1 = mute)

WW.MUTE4 x
Mute trigger 4 (0 = on, 1 = mute)

WW.MUTEA x
Mute CV A (0 = on, 1 = mute)

WW.MUTEB x
Mute CV B (0 = on, 1 = mute)

Meadowphysics

MP.PRESET x
set Meadowphysics to preset x (indexed from 0)

MP.RESET x
reset countdown for channel x (0 = all, 1-8 = individual channels)

MP.STOP x
reset channel x (0 = all, 1-8 = individual channels)

Earthsea

ES.PRESET x
Recall preset x (0-7)

ES.MODE x
Set pattern clock mode. (0=normal, 1=II clock)

ES.CLOCK x
If II clocked, next pattern event

ES.RESET x
Reset pattern to start (and start playing)

ES.PATTERN x
Select playing pattern (0-15)

ES.TRANS x
Transpose the current pattern

ES.STOP x
Stop pattern playback.

ES.TRIPLE x
Recall triple shape (1-4)

ES.MAGIC x
Magic shape (1= halfspeed, 2=doublespeed, 3=linearize)

ES.CV x
get the current CV value for channel x

Orca

OR.CLK x
Advance track x (1-4)

OR.RST x
Reset track x (1-4)

OR.GRST x
Global reset (x can be any value)

OR.TRK x
Choose track x (1-4) to be used by OR.DIV, OR.PHASE, OR.WGT or OR.MUTE

OR.DIV x
Set divisor for selected track to x (1-16)

OR.PHASE x
Set phase for selected track to x (0-16)

OR.WGT x
Set weight for selected track to x (1-8)

OR.MUTE x
Mute trigger selected by OR.TRK (0 = on, 1 = mute)

OR.SCALE x
Select scale x (1-16)

OR.BANK x
Select preset bank x (1-8)

OR.PRESET x
Select preset x (1-8)

OR.RELOAD x
Reload preset or bank (0 - current preset, 1 - current bank, 2 - all banks)

OR.ROTS x
Rotate scales by x (1-15)

OR.ROTW x
Rotate weights by x (1-3)

OR.CVA x
Select tracks for CV A where x is a binary number representing the tracks

OR.CVB x
Select tracks for CV B where x is a binary number representing the tracks

Just Friends

JF.ADDR x
Sets JF II address (1 = primary, 2 = secondary). Use with only one JF on the bus! Saves to JF internal memory, so only one-time config is needed.

JF.SEL x
Sets target JF unit (1 = primary, 2 = secondary).

JF0: ...
Send following JF OPs to both units starting with selected unit.

JF1: ...
Send following JF OPs to unit 1 ignoring the currently selected unit.

JF2: ...
Send following JF OPs to unit 2 ignoring the currently selected unit.

JF.RAMP
Gets value of RAMP knob.

JF.CURVE
Gets value of CURVE knob.

JF.FM
Gets value of FM knob.

JF.INTONE
Gets value of INTONE knob and CV offset.

JF.TIME
Gets value of TIME knob and CV offset.

JF.SPEED
Gets value of SPEED switch (1 = sound, 0 = shape).

JF.TSC
Gets value of MODE switch (0 = transient, 1 = sustain, 2 = cycle).

JF.TR x y
Simulate a TRIGGER input. x is channel (0 = all primary JF channels, 1..6 = primary JF, 7..12 = secondary JF, -1 = all channels both JF) and y is state (0 or 1)

JF.RMODE x
Set the RUN state of Just Friends when no physical jack is present. (0 = run off, non-zero = run on)

JF.RUN x
Send a 'voltage' to the RUN input. Requires JF.RMODE 1 to have been executed, or a physical cable in JF's input. Thus Just Friend's RUN modes are accessible without needing a physical cable & control voltage to set the RUN parameter. use JF.RUN V x to set to x volts. The expected range is V -5 to V 5

JF.SHIFT x
Shifts the transposition of Just Friends, regardless of speed setting. Shifting by V 1 doubles the frequency in sound, or doubles the rate in shape. x = pitch, use N x for semitones, or V y for octaves.

JF.VTR x y
Like JF.TR with added volume control. Velocity is scaled with volts, so try V 5 for an output trigger of 5 volts. Channels remember their latest velocity setting and apply it regardless of TRIGGER origin (digital or physical). x = channel, 0 sets all channels. y = velocity, amplitude of output in volts. eg JF.VTR 1 V 4.

JF.TUNE x y z
Adjust the tuning ratios used by the INTONE control. x = channel, y = numerator (set the multiplier for the tuning ratio), z = denominator (set the divisor for the tuning ratio). JF.TUNE 0 0 0 resets to default ratios.

JF.MODE x
Set the current choice of standard functionality, or Just Type alternate modes (Speed switch to Sound for Synth, Shape for Geode). You'll likely want to put JF.MODE x in your Teletype INIT scripts. x = nonzero activates alternative modes. 0 restores normal.

JF.VOX x y z
Synth mode: create a note at the specified channel, of the defined pitch & velocity. All channels can be set simultaneously with a chan value of 0. x = channel, y = pitch relative to C3, z = velocity (like JF.VTR). Geode mode: Create a stream of rhythmic envelopes on the named channel. x = channel, y = division, z = number of repeats.

JF.NOTE x y
Synth: polyphonically allocated note sequencing. Works as JF.VOX with chan selected automatically. Free voices will be taken first. If all voices are busy, will steal from the voice which has been active the longest. x = pitch relative to C3, y = velocity. Geode: works as JF.VOX with dynamic allocation of channel. Assigns the rhythmic stream to the oldest unused channel, or if all are busy, the longest running channel. x = division, y = number of repeats.

JF.POLY x y
As JF.NOTE but across dual JF. Switches between primary and secondary units every 6 notes or until reset using JF.POLY.RESET.

JF.POLY.RESET
Resets JF.POLY note count.

JF.PITCH x y
Change pitch without retriggering. x = channel, y = pitch relative to C3.

JF.GOD x
Redefines C3 to align with the 'God' note. x = 0 sets A to 440, x = 1 sets A to 432.

JF.TICK x
Sets the underlying timebase of the Geode. x = clock. 0 resets the timebase to the start of measure. 1 to 48 shall be sent repetitively. The value representing ticks per measure. 49 to 255 sets beats-per-minute and resets the timebase to start of measure.

JF.QT x
When non-zero, all events are queued & delayed until the next quantize event occurs. Using values that don't align with the division of rhythmic streams will cause irregular patterns to unfold. Set to 0 to deactivate quantization. x = division, 0 deactivates quantization, 1 to 32 sets the subdivision & activates quantization.

Faderbank

FADER x FB
Reads the value of the FADER slider x; default return range is from 0 to 16383. Up to four Faderbanks can be addressed; x value between 1 and 16 correspond to Faderbank 1, x between 17 and 32 to Faderbank 2, etc…

FADER.SCALE x y z FB.S
Set static scaling of the FADER x to between min and max.

FADER.CAL.MIN x FB.C.MIN
Reads FADER x minimum position and assigns a zero value

FADER.CAL.MAX x FB.C.MAX
Reads FADER x maximum position and assigns the maximum point

FADER.CAL.RESET x FB.C.R
Resets the calibration for FADER x

ER-301

SC.TR x y
Set trigger output for the ER-301 virtual output x to y (0-1)

SC.TR.POL x y
Set polarity of trigger for the ER-301 virtual output x to y (0-1)

SC.TR.TIME x y
Set the pulse time for the ER-301 virtual trigger x to y in ms

SC.TR.TOG x
Flip the state for the ER-301 virtual trigger output x

SC.TR.PULSE x SC.TR.P
Pulse the ER-301 virtual trigger output x

SC.CV x y
CV target value for the ER-301 virtual output x to value y

SC.CV.OFF x y
CV offset added to the ER-301 virtual output x

SC.CV.SET x
Set CV value for the ER-301 virtual output x

SC.CV.SLEW x y
Set the CV slew time for the ER-301 virtual output x in ms

TELEXi

TI.PARAM x TI.PRM
reads the value of PARAM knob x; default return range is from 0 to 16383; return range can be altered by the TI.PARAM.MAP command

TI.PARAM.QT x TI.PRM.QT
return the quantized value for PARAM knob x using the scale set by TI.PARAM.SCALE; default return range is from 0 to 16383

TI.PARAM.N x TI.PRM.N
return the quantized note number for PARAM knob x using the scale set by TI.PARAM.SCALE

TI.PARAM.SCALE x TI.PRM.SCALE
select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z TI.PRM.MAP
maps the PARAM values for input x across the range y - z (defaults 0-16383)

TI.IN x
reads the value of IN jack x; default return range is from -16384 to 16383 - representing -10V to +10V; return range can be altered by the TI.IN.MAP command

TI.IN.QT x
return the quantized value for IN jack x using the scale set by TI.IN.SCALE; default return range is from -16384 to 16383 - representing -10V to +10V

TI.IN.N x
return the quantized note number for IN jack x using the scale set by TI.IN.SCALE

TI.IN.SCALE x
select scale # y for IN jack x; scales listed in full description

TI.IN.MAP x y z
maps the IN values for input jack x across the range y - z (default range is -16384 to 16383 - representing -10V to +10V)

TI.PARAM.INIT x TI.PRM.INIT
initializes PARAM knob x back to the default boot settings and behaviors; neutralizes mapping (but not calibration)

TI.IN.INIT x
initializes IN jack x back to the default boot settings and behaviors; neutralizes mapping (but not calibration)

TI.INIT d
initializes all of the PARAM and IN inputs for device number d (1-8)

TI.PARAM.CALIB x y TI.PRM.CALIB
calibrates the scaling for PARAM knob x; y of 0 sets the bottom bound; y of 1 sets the top bound

TI.IN.CALIB x y
calibrates the scaling for IN jack x; y of -1 sets the -10V point; y of 0 sets the 0V point; y of 1 sets the +10V point

TI.STORE d
stores the calibration data for TXi number d (1-8) to its internal flash memory

TI.RESET d
resets the calibration data for TXi number d (1-8) to its factory defaults (no calibration)

TELEXo

TO.TR x y
sets the TR value for output x to y (0/1)

TO.TR.TOG x
toggles the TR value for output x

TO.TR.PULSE x TO.TR.P
pulses the TR value for output x for the duration set by TO.TR.TIME/S/M

TO.TR.PULSE.DIV x y TO.TR.P.DIV
sets the clock division factor for TR output x to y

TO.TR.PULSE.MUTE x y TO.TR.P.MUTE
mutes or un-mutes TR output x; y is 1 (mute) or 0 (un-mute)

TO.TR.TIME x y
sets the time for TR.PULSE on output n; y in milliseconds

TO.TR.TIME.S x y
sets the time for TR.PULSE on output n; y in seconds

TO.TR.TIME.M x y
sets the time for TR.PULSE on output n; y in minutes

TO.TR.WIDTH x y
sets the time for TR.PULSE on output n based on the width of its current metronomic value; y in percentage (0-100)

TO.TR.POL x y
sets the polarity for TR output n

TO.TR.M.ACT x y
sets the active status for the independent metronome for output x to y (0/1); default 0 (disabled)

TO.TR.M x y
sets the independent metronome interval for output x to y in milliseconds; default 1000

TO.TR.M.S x y
sets the independent metronome interval for output x to y in seconds; default 1

TO.TR.M.M x y
sets the independent metronome interval for output x to y in minutes

TO.TR.M.BPM x y
sets the independent metronome interval for output x to y in Beats Per Minute

TO.TR.M.COUNT x y
sets the number of repeats before deactivating for output x to y; default 0 (infinity)

TO.TR.M.MUL x y
multiplies the M rate on TR output x by y; y defaults to 1 - no multiplication

TO.TR.M.SYNC x
synchronizes the PULSE for metronome on TR output number x

TO.M.ACT d y
sets the active status for the 4 independent metronomes on device d (1-8) to y (0/1); default 0 (disabled)

TO.M d y
sets the 4 independent metronome intervals for device d (1-8) to y in milliseconds; default 1000

TO.M.S d y
sets the 4 independent metronome intervals for device d to y in seconds; default 1

TO.M.M d y
sets the 4 independent metronome intervals for device d to y in minutes

TO.M.BPM d y
sets the 4 independent metronome intervals for device d to y in Beats Per Minute

TO.M.COUNT d y
sets the number of repeats before deactivating for the 4 metronomes on device d to y; default 0 (infinity)

TO.M.SYNC d
synchronizes the 4 metronomes for device number d (1-8)

TO.CV x
CV target output x; y values are bipolar (-16384 to +16383) and map to -10 to +10

TO.CV.SLEW x y
set the slew amount for output x; y in milliseconds

TO.CV.SLEW.S x y
set the slew amount for output x; y in seconds

TO.CV.SLEW.M x y
set the slew amount for output x; y in minutes

TO.CV.SET x y
set the CV for output x (ignoring SLEW); y values are bipolar (-16384 to +16383) and map to -10 to +10

TO.CV.OFF x y
set the CV offset for output x; y values are added at the final stage

TO.CV.QT x y
CV target output x; y is quantized to output's current CV.SCALE

TO.CV.QT.SET x y
set the CV for output x (ignoring SLEW); y is quantized to output's current CV.SCALE

TO.CV.N x y
target the CV to note y for output x; y is indexed in the output's current CV.SCALE

TO.CV.N.SET x y
set the CV to note y for output x; y is indexed in the output's current CV.SCALE (ignoring SLEW)

TO.CV.SCALE x y
select scale # y for CV output x; scales listed in full description

TO.CV.LOG x y
translates the output for CV output x to logarithmic mode y; y defaults to 0 (off); mode 1 is for 0-16384 (0V-10V), mode 2 is for 0-8192 (0V-5V), mode 3 is for 0-4096 (0V-2.5V), etc.

TO.CV.CALIB x
Locks the current offset (CV.OFF) as a calibration offset and saves it to persist between power cycles for output x.

TO.CV.RESET x
Clears the calibration offset for output x

TO.OSC x y
Targets oscillation for CV output x to y

TO.OSC.SET x y
set oscillation for CV output x to y (ignores slew)

TO.OSC.QT x y
targets oscillation for CV output x to y

TO.OSC.QT.SET x y
set oscillation for CV output x to y, quantized to the current scale (ignores slew)

TO.OSC.N x y
targets oscillation for CV output x to note y

TO.OSC.N.SET x y
sets oscillation for CV output x to note y (ignores slew)

TO.OSC.FQ x y
targets oscillation for CV output x to frequency y in Hertz

TO.OSC.FQ.SET x y
targets oscillation for CV output x to frequency y in Hertz (ignores slew)

TO.OSC.LFO x y
Targets oscillation for CV output x to LFO frequency y in millihertz

TO.OSC.LFO.SET x y
Targets oscillation for CV output x to LFO frequency y in millihertz (ignores slew)

TO.OSC.CYC x y
targets the oscillator cycle length to y for CV output x with the portamento rate determined by the TO.OSC.SLEW value; y is in milliseconds

TO.OSC.CYC.SET x y
sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in milliseconds

TO.OSC.CYC.S x y
targets the oscillator cycle length to y for CV output x with the portamento rate determined by the TO.OSC.SLEW value; y is in seconds

TO.OSC.CYC.S.SET x y
sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in seconds

TO.OSC.CYC.M x y
targets the oscillator cycle length to y for CV output x with the portamento rate determined by the TO.OSC.SLEW value; y is in minutes

TO.OSC.CYC.M.SET x y
sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in minutes

TO.OSC.SCALE x y
select scale # y for CV output x; scales listed in full description

TO.OSC.WAVE x y
set the waveform for output x to y; y range is 0-4500, blending between 45 waveforms

TO.OSC.RECT x y
rectifies the polarity of the oscillator for output x to y; 0 is no rectification, +/-1 is partial rectification, +/-2 is full rectification

TO.OSC.WIDTH x y
sets the width of the pulse wave on output x to y; y is a percentage of total width (0 to 100); only affects waveform 3000

TO.OSC.SYNC x
resets the phase of the oscillator on CV output x (relative to TO.OSC.PHASE)

TO.OSC.PHASE x y
sets the phase offset of the oscillator on CV output x to y (0 to 16383); y is the range of one cycle

TO.OSC.SLEW x y
sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in milliseconds

TO.OSC.SLEW.S x y
sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in seconds

TO.OSC.SLEW.M x y
sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in minutes

TO.OSC.CTR x y
centers the oscillation on CV output x to y; y values are bipolar (-16384 to +16383) and map to -10 to +10

TO.ENV.ACT x y
activates/deactivates the AD envelope generator for the CV output x; y turns the envelope generator off (0 - default) or on (1); CV amplitude is used as the peak for the envelope and needs to be > 0 for the envelope to be perceivable

TO.ENV x y
trigger the attack stage of output x when y changes to 1, or decay stage when it changes to 0

TO.ENV.TRIG x
triggers the envelope at CV output x to cycle; CV amplitude is used as the peak for the envelope and needs to be > 0 for the envelope to be perceivable

TO.ENV.ATT x y
set the envelope attack time to y for CV output x; y in milliseconds (default 12 ms)

TO.ENV.ATT.S x y
set the envelope attack time to y for CV output x; y in seconds

TO.ENV.ATT.M x y
set the envelope attack time to y for CV output x; y in minutes

TO.ENV.DEC x y
set the envelope decay time to y for CV output x; y in milliseconds (default 250 ms)

TO.ENV.DEC.S x y
set the envelope decay time to y for CV output x; y in seconds

TO.ENV.DEC.M x y
set the envelope decay time to y for CV output x; y in minutes

TO.ENV.EOR x n
at the end of rise of CV output x, fires a PULSE to the trigget output n

TO.ENV.EOC x n
at the end of cycle of CV output x, fires a PULSE to the trigget output n

TO.ENV.LOOP x y
causes the envelope on CV output x to loop for y times

TO.TR.INIT x
initializes TR output x back to the default boot settings and behaviors; neutralizes metronomes, dividers, pulse counters, etc.

TO.CV.INIT x
initializes CV output x back to the default boot settings and behaviors; neutralizes offsets, slews, envelopes, oscillation, etc.

TO.INIT d
initializes all of the TR and CV outputs for device number d (1-8)

TO.KILL d
cancels all TR pulses and CV slews for device number d (1-8)

Crow

CROW.SEL x
Sets target crow unit (1 (default), to 4).

CROWN: ...
Send following CROW OPs to all units starting with selected unit.

CROW1: ...
Send following CROW OPs to unit 1 ignoring the currently selected unit.

CROW2: ...
Send following CROW OPs to unit 2 ignoring the currently selected unit.

CROW3: ...
Send following CROW OPs to unit 3 ignoring the currently selected unit.

CROW4: ...
Send following CROW OPs to unit 4 ignoring the currently selected unit.

CROW.V x y
Sets output x to value y. Use `V y` for volts.

CROW.SLEW x y
Sets output x slew rate to y milliseconds.

CROW.C1 x
Calls the function `ii.self.call1(x)` on crow.

CROW.C2 x y
Calls the function `ii.self.call2(x, y)` on crow.

CROW.C3 x y z
Calls the function `ii.self.call3(x, y, z)` on crow.

CROW.C4 x y z t
Calls the function `ii.self.call4(x, y, z, t)` on crow.

CROW.RST
Calls the function `crow.reset()` returning crow to default state.

CROW.PULSE x y z t
Creates a trigger pulse on output x with duration y (ms) to voltage z with polarity t.

CROW.AR x y z t
Creates an envelope on output x, rising in y ms, falling in z ms, and reaching height t.

CROW.LFO x y z t
Starts an envelope on output x at rate y where `0` = 1Hz with 1v/octave scaling. z sets amplitude and t sets skew for assymetrical triangle waves.

CROW.IN x
Gets voltage at input x.

CROW.OUT x
Gets voltage of output x.

CROW.Q0
Returns the result of calling the function `crow.self.query0()`.

CROW.Q1 x
Returns the result of calling the function `crow.self.query1(x)`.

CROW.Q2 x y
Returns the result of calling the function `crow.self.query2(x, y)`.

CROW.Q3 x y z
Returns the result of calling the function `crow.self.query3(x, y, z)`.

W/1.0

WS.PLAY x
Set playback state and direction. `0` stops playback. 1 sets forward motion, while `-1` plays in reverse

WS.REC x
Set recording mode. `0` is playback only. 1 sets overdub mode for additive recording. `-1` sets overwrite mode to replace the tape with your input

WS.CUE x
Go to a cuepoint relative to the playhead position. `0` retriggers the current location. 1 jumps to the next cue forward. `-1` jumps to the previous cue in the reverse. These actions are relative to playback direction such that `0` always retriggers the most recently passed location

WS.LOOP x
Set the loop state on/off. `0` is off. Any other value turns loop on

W/2.0

W/.SEL x
Sets target W/2.0 unit (1 = primary, 2 = secondary).

W/1: ...
Send following W/2.0 OPs to unit 1 ignoring the currently selected unit.

W/2: ...
Send following W/2.0 OPs to unit 2 ignoring the currently selected unit.

W/2.0 tape

W/T.REC active
Sets recording state to `active` (s8)

W/T.PLAY playback
Set the playback state. -1 will flip playback direction (s8)

W/T.REV
Reverse the direction of playback

W/T.SPEED speed deno
Set speed as a rate, or ratio. Negative values are reverse (s16V)

W/T.FREQ freq
Set speed as a `frequency` (s16V) style value. Maintains reverse state

W/T.ERASE.LVL level
Strength of erase head when recording. 0 is overdub, 1 is overwrite. Opposite of feedback (s16V)

W/T.MONITOR.LVL gain
Level of input passed directly to output (s16V)

W/T.REC.LVL gain
Level of input material recorded to tape (s16V)

W/T.ECHOMODE is_echo
Set to 1 to playback before erase. 0 (default) erases first (s8)

W/T.LOOP.START
Set the current time as the beginning of a loop

W/T.LOOP.END
Set the current time as the loop end, and jump to start

W/T.LOOP.ACTIVE state
Set the state of looping (s8)

W/T.LOOP.SCALE scale
Mul(Positive) or Div(Negative) loop brace by arg. Zero resets to original window (s8)

W/T.LOOP.NEXT direction
Move loop brace forward/backward by length of loop. Zero jumps to loop start (s8)

W/T.TIME seconds sub
Move playhead to an arbitrary location on tape (s16)

W/T.SEEK seconds sub
Move playhead relative to current position (s16)

W/T.CLEARTAPE
WARNING! Erases all recorded audio on the tape!

W/2.0 delay

W/D.FBK level
amount of feedback from read head to write head (s16V)

W/D.MIX fade
fade from dry to delayed signal

W/D.FILT cutoff
centre frequency of filter in feedback loop (s16V)

W/D.FREEZE is_active
deactivate record head to freeze the current buffer (s8)

W/D.TIME seconds
set delay buffer length in `seconds` (s16V), when rate == 1

W/D.LEN count divisions
set buffer loop length as a fraction of buffer time (u8)

W/D.POS count divisions
set loop start location as a fraction of buffer time (u8)

W/D.CUT count divisions
jump to loop location as a fraction of loop length (u8)

W/D.FREQ.RNG freq_range
TBD (s8)

W/D.RATE multiplier
direct `multiplier` (s16V) of tape speed

W/D.FREQ volts
manipulate tape speed with musical values (s16V)

W/D.CLK
receive clock pulse for synchronization

W/D.CLK.RATIO mul div
set clock pulses per buffer time, with clock `mul/div` (s8)

W/D.PLUCK volume
pluck the delay line with noise at volume (s16V)

W/D.MOD.RATE rate
set the multiplier for the modulation rate (s16V)

W/D.MOD.AMT amount
set the `amount` (s16V) of delay line modulation to be applied

W/2.0 synth

W/S.PITCH voice pitch
set `voice` (s8) to `pitch` (s16V) in volts-per-octave

W/S.VEL voice velocity
strike the vactrol of `voice` (s8) at `velocity` (s16V) in volts

W/S.VOX voice pitch velocity
set `voice` (s8) to `pitch` (s16V) and strike the vactrol at `velocity` (s16V)

W/S.NOTE pitch level
dynamically assign a voice, set to `pitch` (s16V), strike with `velocity`(s16V)

W/S.POLY pitch level
As `W/S.NOTE` but across dual W/. Switches between primary and secondary units every 4 notes or until reset using `W/S.POLY.RESET`.

W/S.POLY.RESET
Resets `W/S.POLY` note count.

W/S.AR.MODE is_ar
in attack-release mode, all notes are `plucked` and no `release` is required'

W/S.LPG.TIME time
vactrol `time` (s16V) constant. -5=drones, 0=vtl5c3, 5=blits

W/S.LPG.SYM symmetry
vactrol attack-release ratio. -5=fastest attack, 5=long swells (s16V)

W/S.CURVE curve
cross-fade waveforms: -5=square, 0=triangle, 5=sine (s16V)

W/S.RAMP ramp
waveform symmetry: -5=rampwave, 0=triangle, 5=sawtooth (NB: affects FM tone)

W/S.FM.INDEX index
amount of FM modulation. -5=negative, 0=minimum, 5=maximum (s16V)

W/S.FM.RATIO num den
ratio of the FM modulator to carrier as a ratio. floating point values up to 20.0 supported (s16V)

W/S.FM.ENV amount
amount of vactrol envelope applied to fm index, -5 to +5 (s16V)

W/S.PATCH jack param
patch a hardware `jack` (s8) to a `param` (s8) destination

W/S.VOICES count
set number of polyphonic voices to allocate. use 0 for unison mode (s8)

Disting EX

EX / EX x
get or set currently selected unit to x (1-4)

EX1: ...
send following Disting ops to unit 1 ignoring the currently selected unit

EX2: ...
send following Disting ops to unit 2 ignoring the currently selected unit

EX3: ...
send following Disting ops to unit 3 ignoring the currently selected unit

EX4: ...
send following Disting ops to unit 4 ignoring the currently selected unit

EX.PRESET / EX.PRESET x EX.PRE
load preset x or get the currently loaded preset

EX.SAVE x
save to preset x

EX.RESET
reset the currently loaded preset

EX.ALG / EX.ALG x EX.A
get or set the current algorithm to x (single algorithms only)

EX.CTRL x y EX.C
set I2C controller x to value y

EX.PARAM x / EX.PARAM x y EX.P
set parameter x to value y or get the current parameter value

EX.PV x y
set parameter x using a value determined by scaling y from 0..16384 range.

EX.MIN x
get the minimum possible value for parameter x

EX.MAX x
get the maximum possible value for parameter x

EX.VOX x y z EX.V
send a note to voice x using pitch y and velocity z

EX.VOX.P x y EX.VP
set voice x to pitch y

EX.VOX.O x EX.VO
send a note off to voice x

EX.CH x EX.#
select default note channel (for multi channel algorithms like Poly FM)

EX.NOTE x y EX.N
send a note using pitch x and velocity y (voice allocated by the Disting)

EX.N# x y z
send a note to channel x using pitch y and velocity z (voice allocated by the Disting)

EX.NOTE.O x EX.NO
send a note off using pitch x

EX.NO# x y
send a note off to channel x using pitch y

EX.ALLOFF EX.AO
all notes off

EX.T x
send a trigger to voice x with medium velocity (use with SD Triggers algo)

EX.TV x y
send a trigger to voice x using velocity y (use with SD Triggers algo)

EX.REC x
control WAV recorder recording: 1 to start, 0 to stop

EX.PLAY x
control WAV recorder playback: 1 to start, 0 to stop

EX.AL.P x
set Augustus Loop pitch to value x

EX.AL.CLK
send clock to Augustus Loop

EX.LP x
get current state for loop x

EX.LP.REC x
toggle recording for loop x

EX.LP.PLAY x
toggle playback for loop x

EX.LP.CLR x
clear loop x

EX.LP.REV x
toggle reverse for loop x

EX.LP.REV? x
returns 1 if loop x is reversed, 0 otherwise

EX.LP.DOWN x
toggle octave down for loop x

EX.LP.DOWN? x
return 1 if loop x is transposed octave down, 0 otherwise

EX.M.CH / EX.M.CH x
get or set the currently selected MIDI channel (1-16)

EX.M.N x y
send MIDI Note On message for note x (0..127) and velocity y (0..127)

EX.M.N# x y z
send MIDI Note On message on channel x for note y (0..127) and velocity z (0..127)

EX.M.NO x
send MIDI Note off message for note x (0..127)

EX.M.NO# x y
send MIDI Note off message on channel x for note y (0..127)

EX.M.CC x y
send MIDI CC message for controller x (0..127) and value y (0..127)

EX.M.CC# x y z
send MIDI CC message on channel x for controller y (0..127) and value z (0..127)

EX.M.PB x
send MIDI Pitchbend message

EX.M.PRG x
send MIDI Program Change message

EX.M.CLK
send MIDI clock message

EX.M.START
send MIDI Start message

EX.M.STOP
send MIDI Stop message

EX.M.CONT
send MIDI Continue message

EX.SB.CH / EX.SB.CH x
get or set the currently selected Select Bus channel (1-16)

EX.SB.N x y
send Select Bus Note On message for note x (0..127) and velocity y (0..127)

EX.SB.NO x
send Select Bus Note off message for note x (0..127)

EX.SB.PB x
send Select Bus Pitchbend message

EX.SB.CC x y
send Select Bus CC message for controller x (0..127) and value y (0..127)

EX.SB.PRG x
send Select Bus Program Change message

EX.SB.CLK
send Select Bus clock message

EX.SB.START
send Select Bus Start message

EX.SB.STOP
send Select Bus Stop message

EX.SB.CONT
send Select Bus Continue message

EX.A1 / EX.A1 x
get or set the left dual algorithm

EX.A2 / EX.A2 x
get or set the right dual algorithm

EX.A12 x y
set both dual algorithms

EX.P1 x / EX.P1 x y
get left algorithm parameter x or set it to value y

EX.P2 x / EX.P2 x y
get right algorithm parameter x or set it to value y

EX.PV1 x y
set left algorithm parameter x using a value determined by scaling y from 0..16384 range

EX.PV2 x y
set right algorithm parameter x using a value determined by scaling y from 0..16384 range

EX.MIN1 x
get left algorithm parameter minimum value

EX.MAX1 x
get left algorithm parameter maximum value

EX.MIN2 x
get right algorithm parameter minimum value

EX.MAX2 x
get right algorithm parameter maximum value

EX.Z1 / EX.Z1 x
get left Z knob value or set left Z parameter (0..127 range)

EX.Z01
restore control for left Z knob and input

EX.Z2 / EX.Z2 x
get right Z knob value or set right Z parameter (0..127 range)

EX.Z02
restore control for right Z knob and input

EX.PRE1 x
load left preset from x slot

EX.PRE2 x
load right preset from x slot

EX.SAVE1 x
save left preset to x slot

EX.SAVE2 x
save right preset to x slot

Matrixarchate

MA.SELECT x
select the default matrixarchate module, default 1

MA.STEP
advance program sequencer

MA.RESET
reset program sequencer

MA.PGM pgm
select the current program (1-based)

MA.ON x y
connect row x and column y in the current program (rows/columns are 0-based)

MA.PON pgm x y
connect row x and column y in program `pgm`

MA.OFF x y
disconnect row x and column y in the current program

MA.POFF x y pgm
connect row x and column y in program `pgm`

MA.SET x y state
set the connection at row x and column y to `state` (1 - on, 0 - off)

MA.PSET pgm x y state
set the connection at row x and column y in program `pgm` to `state` (1 - on, 0 - off)

MA.COL col / MA.COL col value
get or set column `col` (as a 16 bit unsigned value where each bit represents a connection)

MA.PCOL pgm col / MA.PCOL pgm col value
get or set column `col` in program `pgm`

MA.ROW row / MA.ROW row value
get or set row `row`

MA.PROW pgm row / MA.PROW pgm row value
get or set row `row` in program `pgm`

MA.CLR
clear all connections

MA.PCLR pgm
clear all connections in program `pgm`

I2C2MIDI

I2M.CH / I2M.CH x I2M.#
Get currently set MIDI channel / Set MIDI channel x (1..16 for TRS, 17..32 for USB) for MIDI out

I2M.TIME / I2M.TIME x I2M.T
Get current note duration / Set note duration of MIDI notes to x ms (0..32767) for current channel

I2M.T# ch / I2M.T# ch x
Get current note duration / Set note duration of MIDI notes to x ms (0..32767) for channel ch (0..32).

I2M.SHIFT / I2M.SHIFT x I2M.S
Get current transposition / Set transposition of MIDI notes to x semitones (-127..127) for current channel

I2M.S# ch / I2M.S# ch x
Get current transposition / Set transposition of MIDI notes to x semitones (-127..127) for channel ch (0..32)

I2M.MIN x y
Set minimum note number for MIDI notes to x (0..127), using mode y (0..3), for current channel

I2M.MIN# ch x y
Set minimum note number for MIDI notes to x (0..127), using mode y (0..3), for channel ch (0..32)

I2M.MAX x y
Set maximum note number for MIDI notes to x (0..127), using mode y (0..3), for current channel

I2M.MAX# ch x y
Set maximum note number for MIDI notes to x (0..127), using mode y (0..3), for channel ch (0..32)

I2M.REP / I2M.REP x
Get current repetition / Set repetition of MIDI notes to x repetitions (1..127) for current channel

I2M.REP# ch x
Get current repetition / Set repetition of MIDI notes to x repetitions (1..127) for channel ch (0..32)

I2M.RAT / I2M.RAT x
Get current ratcheting / Set ratcheting of MIDI notes to x ratchets (1..127) for current channel

I2M.RAT# ch x
Get current ratcheting / Set ratcheting of MIDI notes to x ratchets (1..127) for channel ch (0..32)

I2M.MUTE / I2M.MUTE x
Get mute state / Set mute state of current MIDI channel to x (0..1)

I2M.MUTE# / I2M.MUTE# ch x
Get mute state / Set mute state of MIDI channel ch to x (0..1)

I2M.SOLO / I2M.SOLO x
Get solo state / Set solo state of current MIDI channel to x (0..1)

I2M.SOLO# / I2M.SOLO# ch x
Get solo state / Set solo state of MIDI channel ch to x (0..1)

I2M.NOTE x y I2M.N
Send MIDI Note On message for note number x (0..127) with velocity y (1..127) on current channel

I2M.N# ch x y
Send MIDI Note On message for note number x (0..127) with velocity y (1..127) on channel ch (1..32)

I2M.NOTE.O x I2M.NO
Send a manual MIDI Note Off message for note number x (0..127)

I2M.NO# ch x
Send a manual MIDI Note Off message for note number x (0..127) on channel ch (1..32)

I2M.NT x y z
Send MIDI Note On message for note number x (0..127) with velocity y (1..127) and note duration z ms (0..32767)

I2M.NT# ch x y z
Send MIDI Note On message for note number x (0..127) with velocity y (1..127) and note duration z ms (0..32767) on channel ch (1..32)

I2M.CC x y
Send MIDI CC message for controller x (0..127) with value y (0..127)

I2M.CC# ch x y
Send MIDI CC message for controller x (0..127) with value y (0..127) on channel ch (1..32)

I2M.CC.SET x y
Send MIDI CC message for controller x (0..127) with value y (0..127), bypassing any slew settings

I2M.CC.SET# ch x y
Send MIDI CC message for controller x (0..127) with value y (0..127) on channel ch (1..32), bypassing any slew settings

I2M.CCV x y
Send MIDI CC message for controller x (0..127) with volt value y (0..16383, 0..+10V)

I2M.CCV# ch x y
Send MIDI CC message for controller x (0..127) with volt value y (0..16383, 0..+10V) on channel ch (1..32)

I2M.CC.OFF x / I2M.CC.OFF x y
Get current offset / Set offset of values of controller x (0..127) to y (-127..127)

I2M.CC.OFF# ch x / I2M.CC.OFF# ch x y
Get current offset / Set offset of values of controller x (0..127) to y (-127..127) for channel ch (1..32)

I2M.CC.SLEW x / I2M.CC.SLEW x y
Get current slew time for controller x / Set slew time for controller x (0..127) to y ms (0..32767)

I2M.CC.SLEW# ch x / I2M.CC.SLEW# ch x y
Get current slew time for controller x / Set slew time for controller x (0..127) to y ms (0..32767) for channel ch (1..32)

I2M.NRPN x y z
Send MIDI NRPN message (high-res CC) for parameter MSB x and LSB y with value y (0..16383)

I2M.NRPN# ch x y z
Send MIDI NRPN message (high-res CC) for parameter MSB x and LSB y with value y (0..16383) on channel ch (1..32)

I2M.NRPN.OFF x y / I2M.NRPN.OFF x y z
Get current offset / Set offset of values of NRPN messages to z (-16384..16383)

I2M.NRPN.OFF# ch x y / I2M.NRPN.OFF# ch x y z
Get current offset / Set offset of values of NRPN messages to z (-16384..16383) for channel ch (1..32)

I2M.NRPN.SLEW x y / I2M.NRPN.SLEW x y z
Get current slew time / Set slew time for NRPN messages to z ms (0..32767)

I2M.NRPN.SLEW# ch x y / I2M.NRPN.SLEW# ch x y z
Get current slew time / Set slew time for NRPN messages to z ms (0..32767) for channel ch (1..32)

I2M.NRPN.SET x y z
Send MIDI NRPN message for parameter MSB x and LSB y with value y (0..16383), bypassing any slew settings

I2M.NRPN.SET# ch x y z
Send MIDI NRPN message for parameter MSB x and LSB y with value y (0..16383) on channel ch (1..32), bypassing any slew settings

I2M.PRG x
Send MIDI Program Change message for program x (0..127)

I2M.PB x
Send MIDI Pitch Bend message with value x (-8192..8191)

I2M.AT x
Send MIDI After Touch message with value x (0..127)

I2M.CLK
Send MIDI Clock message, this still needs improvement ...

I2M.START
Send MIDI Clock Start message

I2M.STOP
Send MIDI Clock Stop message

I2M.CONT
Send MIDI Clock Continue message

I2M.CHORD x y z I2M.C
Play chord x (1..8) with root note y (-127..127) and velocity z (1..127)

I2M.C# ch x y z
Play chord x (1..8) with root note y (-127..127) and velocity z (1..127) on channel ch (1..32)

I2M.C.ADD x y I2M.C+
Add relative note y (-127..127) to chord x (0..8), use `x = 0` to add to all chords

I2M.C.RM x y I2M.C-
Remove note y (-127..127) from chord x (0..8), use `x = 0` to remove from all chords

I2M.C.INS x y z
Add note z (-127..127) to chord x (0..8) at index y (0..7), with z relative to the root note; use `x = 0` to insert into all chords

I2M.C.DEL x y
Delete note at index y (0..7) from chord x (0..8), use `x = 0` to delete from all chords

I2M.C.SET x y z
Set note at index y (0..7) in chord x (0..8) to note z (-127..127), use `x = 0` to set in all chords

I2M.C.B x y
Clear and define chord x (0..8) using reverse binary notation (`R...`)

I2M.C.CLR x
Clear chord x (0..8), use `x = 0` to clear all chords

I2M.C.L x / I2M.C.L x y
Get current length / Set length of chord x (0..8) to y (1..8), use `x = 0` to set length of all chords

I2M.C.SC x y
Set scale for chord x (0..8) based on chord y (0..8), use `x = 0` to set for all chords, use `y = 0` to remove scale

I2M.C.REV x y
Set reversal of notes in chord x (0..8) to y. `y = 0` or an even number means not reversed, y = 1 or an uneven number means reversed. Use `x = 0` to set for all chords.

I2M.C.ROT x y
Set rotation of notes in chord x (0..8) to y steps (-127..127), use `x = 0` to set for all chords

I2M.C.TRP x y
Set transposition of chord x (0..8) to y (-127..127), use `x = 0` to set for all chords

I2M.C.DIS x y z
Set distortion of chord x (0..8) to y (-127..127) with anchor point z (0..16), use `x = 0` to set for all chords

I2M.C.REF x y z
Set reflection of chord x (0..8) to y iterations (-127..127) with anchor point z (0..16), use `x = 0` to set for all chords

I2M.C.INV x y
Set inversion of chord x (0..8) to y (-32..32), use `x = 0` to set for all chords

I2M.C.STR x y
Set strumming of chord x (0..8) to x ms (0..32767), use `x = 0` to set for all chords

I2M.C.VCUR w x y z I2M.C.V~
Set velocity curve for chord w (0..8) with curve type x (0..5), start value y% (0..32767) and end value z% (0..32767), use `w = 0` to set for all chords, use `x = 0` to turn off

I2M.C.TCUR w x y z I2M.C.T~
Set time curve to strumming for chord w (0..8) with curve type x (0..5), start value y% (0..32767) and end value z% (0..32767), use `w = 0` to set for all chords, use `x = 0` to turn off

I2M.C.DIR x y
Set play direction for chord x (0..8) to direction y (0..8)

I2M.C.QN x y z
Get the transformed note number of a chord note for chord x (1..8) with root note y (-127..127) at index z (0..7)

I2M.C.QV x y z
Get the transformed note velocity of a chord note for chord x (1..8) with root velocity y (1..127) at index z (0..7)

I2M.B.R x
Turn recording of notes into the buffer on or off

I2M.B.L x
Set the length of the buffer to x ms (0..32767)

I2M.B.START x
Add an offset of x ms (0..32767) to the start of the buffer

I2M.B.END x
Add a negative offset of x ms (0..32767) to the end of the buffer

I2M.B.DIR x
Set the play direction x (0..2) of the buffer

I2M.B.SPE x
Set the playing speed x (1..32767) of the buffer. `x = 100` is equivalent to 'normal speed', `x = 50` means double the speed, `x = 200` means half the speed, etc.

I2M.B.FB x
Set the feedback length x (0..255) of the buffer

I2M.B.NSHIFT x
Set the note shift of recorded notes to x semitones (-127..127)

I2M.B.VSHIFT x
Set the velocity shift of recorded notes to x (-127..127)

I2M.B.TSHIFT x
Set the note duration shift ('time shift') of recorded notes to x ms (-16384..16383)

I2M.B.NOFF x
Set the note offset of recorded notes to x semitones (-127..127)

I2M.B.VOFF x
Set the velocity offset of recorded notes to x (-127..127)

I2M.B.TOFF x
Set the note duration offset ('time offset') of recorded notes to x ms (-16384..16383)

I2M.B.CLR
Clear the buffer, erasing all recorded notes in the buffer

I2M.B.MODE x
Set the buffer mode to x (0..1). 1) Digital 2) Tape

I2M.Q.CH / I2M.Q.CH x `I2M.Q.#`
Get currently set MIDI channel / Set MIDI channel x (1..16) for MIDI in

I2M.Q.LATCH x
Turn on or off 'latching' for MIDI notes received via MIDI in

I2M.Q.NOTE x `I2M.Q.N`
Get x (0..7) last note number (0..127) received via MIDI in

I2M.Q.VEL x `I2M.Q.V`
Get x (0..7) last note velocity (1..127) received via MIDI in

I2M.Q.CC x
Get current value (0..127) of controller x (0..127) received via MIDI in

I2M.Q.LCH
Get the latest channel (1..16) received via MIDI in

I2M.Q.LN
Get the note number (0..127) of the latest Note On received via MIDI in

I2M.Q.LV
Get the velocity (1..127) of the latest Note On received via MIDI in

I2M.Q.LO
Get the note number (0..127) of the latest Note Off received via MIDI in

I2M.Q.LC
Get the latest controller number (0..127) received via MIDI in

I2M.Q.LCC
Get the latest controller value (0..127) received via MIDI in

I2M.PANIC
Send MIDI Note Off messages for all notes on all channels, and reset note duration, shift, repetition, ratcheting, min/max