Variables

A / A x
get / set the variable A, default 1

B / B x
get / set the variable B, default 2

C / C x
get / set the variable C, default 3

D / D x
get / set the variable D, default 4

DRUNK / DRUNK x
changes by -1, 0, or 1 upon each read saving its state, setting will give it a new value for the next read

DRUNK.MIN / DRUNK.MIN x
set the lower bound for DRUNK, default 0

DRUNK.MAX / DRUNK.MAX x
set the upper bound for DRUNK, default 255

DRUNK.WRAP / DRUNK.WRAP x
should DRUNK wrap around when it reaches it's bounds, default 0

FLIP / FLIP x
returns inverted state (0 or 1) on each read (also settable)

I / I x
get / set the variable I

O / O x
auto-increments *after* each access, can be set, starting value 0

O.INC / O.INC x
how much to increment O by on each invocation, default 1

O.MIN / O.MIN x
the lower bound for O, default 0

O.MAX / O.MAX x
the upper bound for O, default 63

O.WRAP / O.WRAP x
should O wrap when it reaches its bounds, default 1

T / T x
get / set the variable T, typically used for time, default 0

TIME / TIME x
timer value, counts up in ms., wraps after 32s, can be set

TIME.ACT / TIME.ACT x
enable or disable timer counting, default 1

LAST x
get value in milliseconds since last script run time

X / X x
get / set the variable X, default 0

Y / Y x
get / set the variable Y, default 0

Z / Z x
get / set the variable Z, default 0

J / J x
get / set the variable J

K / K x
get / set the variable K

Hardware

CV x / CV x y
CV target value

CV.OFF x / CV.OFF x y
CV offset added to output

CV.SET x
Set CV value

CV.SLEW x / CV.SLEW x y
Get/set the CV slew time in ms

IN
Get the value of IN jack (0-16383)

IN.SCALE min max
Set static scaling of the IN CV to between min and max.

PARAM PRM
Get the value of PARAM knob (0-16383)

PARAM.SCALE min max
Set static scaling of the PARAM knob to between min and max.

IN.CAL.MIN
Reads the input CV and assigns the voltage to the zero point

IN.CAL.MAX
Reads the input CV and assigns the voltage to the max point

IN.CAL.RESET
Resets the input CV calibration

PARAM.CAL.MIN
Reads the Parameter Knob minimum position and assigns a zero value

PARAM.CAL.MAX
Reads the Parameter Knob maximum position and assigns the maximum point

PARAM.CAL.RESET
Resets the Parameter Knob calibration

TR x / TR x y
Set trigger output x to y (0-1)

TR.POL x / TR.POL x y
Set polarity of trigger output x to y (0-1)

TR.TIME x / TR.TIME x y
Set the pulse time of trigger x to y ms

TR.TOG x
Flip the state of trigger output x

TR.PULSE x TR.P
Pulse trigger output x

MUTE x / MUTE x y
Disable trigger input x

STATE x
Read the current state of input x

DEVICE.FLIP
Flip the screen/inputs/outputs

LIVE.OFF LIVE.O
Show the default live mode screen

LIVE.VARS LIVE.V
Show variables in live mode

LIVE.GRID LIVE.G
Show grid visualizer in live mode

LIVE.DASH x LIVE.D
Show the dashboard with index x

PRINT x / PRINT x y PRT
Print a value on a live mode dashboard or get the printed value

Patterns

P.N / P.N x
get/set the pattern number for the working pattern, default 0

P x / P x y
get/set the value of the working pattern at index x

PN x y / PN x y z
get/set the value of pattern x at index y

P.L / P.L x
get/set pattern length of the working pattern, non-destructive to data

PN.L x / PN.L x y
get/set pattern length of pattern x. non-destructive to data

P.WRAP / P.WRAP x
when the working pattern reaches its bounds does it wrap (0/1), default 1 (enabled)

PN.WRAP x / PN.WRAP x y
when pattern x reaches its bounds does it wrap (0/1), default 1 (enabled)

P.START / P.START x
get/set the start location of the working pattern, default 0

PN.START x / PN.START x y
get/set the start location of pattern x, default 0

P.END / P.END x
get/set the end location of the working pattern, default 63

PN.END x / PN.END x y
get/set the end location of the pattern x, default 63

P.I / P.I x
get/set index position for the working pattern.

PN.I x / PN.I x y
get/set index position for pattern x

P.HERE / P.HERE x
get/set value at current index of working pattern

PN.HERE x / PN.HERE x y
get/set value at current index of pattern x

P.NEXT / P.NEXT x
increment index of working pattern then get/set value

PN.NEXT x / PN.NEXT x y
increment index of pattern x then get/set value

P.PREV / P.PREV x
decrement index of working pattern then get/set value

PN.PREV x / PN.PREV x y
decrement index of pattern x then get/set value

P.INS x y
insert value y at index x of working pattern, shift later values down, destructive to loop length

PN.INS x y z
insert value z at index y of pattern x, shift later values down, destructive to loop length

P.RM x
delete index x of working pattern, shift later values up, destructive to loop length

PN.RM x y
delete index y of pattern x, shift later values up, destructive to loop length

P.PUSH x
insert value x to the end of the working pattern (like a stack), destructive to loop length

PN.PUSH x y
insert value y to the end of pattern x (like a stack), destructive to loop length

P.POP
return and remove the value from the end of the working pattern (like a stack), destructive to loop length

PN.POP x
return and remove the value from the end of pattern x (like a stack), destructive to loop length

P.MIN
find the first minimum value in the pattern between the START and END for the working pattern and return its index

PN.MIN x
find the first minimum value in the pattern between the START and END for pattern x and return its index

P.MAX
find the first maximum value in the pattern between the START and END for the working pattern and return its index

PN.MAX x
find the first maximum value in the pattern between the START and END for pattern x and return its index

P.SHUF
shuffle the values in active pattern (between its START and END)

PN.SHUF x
shuffle the values in pattern x (between its START and END)

P.ROT n
rotate the values in the active pattern n steps (between its START and END, negative rotates backward)

PN.ROT x n
rotate the values in pattern x (between its START and END, negative rotates backward)

P.REV
reverse the values in the active pattern (between its START and END)

PN.REV x
reverse the values in pattern x

P.RND
return a value randomly selected between the start and the end position

PN.RND x
return a value randomly selected between the start and the end position of pattern x

P.+ x y
increase the value of the working pattern at index x by y

PN.+ x y z
increase the value of pattern x at index y by z

P.- x y
decrease the value of the working pattern at index x by y

PN.- x y z
decrease the value of pattern x at index y by z

P.+W x y a b
increase the value of the working pattern at index x by y and wrap it to a..b range

PN.+W x y z a b
increase the value of pattern x at index y by z and wrap it to a..b range

P.-W x y a b
decrease the value of the working pattern at index x by y and wrap it to a..b range

PN.-W x y z a b
decrease the value of pattern x at index y by z and wrap it to a..b range

P.MAP: ...
apply the 'function' to each value in the active pattern, I takes each pattern value

PN.MAP x: ...
apply the 'function' to each value in pattern x, I takes each pattern value

Control flow

IF x: ...
if x is not zero execute command

ELIF x: ...
if all previous IF / ELIF fail, and x is not zero, execute command

ELSE: ...
if all previous IF / ELIF fail, excute command

L x y: ...
run the command sequentially with I values from x to y

W x: ...
run the command while condition x is true

EVERY x: ... EV
run the command every x times the command is called

SKIP x: ...
run the command every time except the xth time.

OTHER: ...
runs the command when the previous EVERY/SKIP did not run its command.

SYNC x
synchronizes *all* EVERY and SKIP counters to offset x.

PROB x: ...
potentially execute command with probability x (0-100)

SCRIPT / SCRIPT x $
get current script number, or execute script x (1-10), recursion allowed

SCRIPT.POL x / SCRIPT.POL x p
$.POL
get script x trigger polarity, or set polarity p (1 rising edge, 2 falling, 3 both)

SCENE / SCENE x
get the current scene number, or load scene x (0-31)

SCENE.G x
load scene x (0-31) without loading grid control states

SCENE.P x
load scene x (0-31) without loading pattern state

KILL
clears stack, clears delays, cancels pulses, cancels slews, disables metronome

BREAK BRK
halts execution of the current script

INIT
clears all state data

INIT.CV x
clears all parameters on CV associated with output x

INIT.CV.ALL
clears all parameters on all CV's

INIT.DATA
clears all data held in all variables

INIT.P x
clears pattern associated with pattern number x

INIT.P.ALL
clears all patterns

INIT.SCENE
loads a blank scene

INIT.SCRIPT x
clear script number x

INIT.SCRIPT.ALL
clear all scripts

INIT.TIME x
clear time on trigger x

INIT.TR x
clear all parameters on trigger associated with TR x

INIT.TR.ALL
clear all triggers

Maths

ADD x y +
add x and y together

SUB x y -
subtract y from x

MUL x y *
multiply x and y together

DIV x y /
divide x by y

MOD x y %
find the remainder after division of x by y

RAND x RND
generate a random number between 0 and x inclusive

RRAND x y RRND
generate a random number between x and y inclusive

TOSS
randomly return 0 or 1

? x y z
if condition x is true return y, otherwise return z

MIN x y
return the minimum of x and y

MAX x y
return the maximum of x and y

LIM x y z
limit the value x to the range y to z inclusive

WRAP x y z WRP
limit the value x to the range y to z inclusive, but with wrapping

QT x y
round x to the closest multiple of y (quantise)

QT.S x r s
quantize 1V/OCT signal x to scale s (0-8, reference N.S scales) with root 1V/OCT pitch r

QT.CS x r s d c
quantize 1V/OCT signal x to chord c (1-7) from scale s (0-8, reference N.S scales) at degree d (1-7) with root 1V/OCT pitch r

QT.B x
quantize 1V/OCT signal x to scale defined by N.B

QT.BX x i
quantize 1V/OCT signal x to scale defined by N.BX in scale index i

AVG x y
the average of x and y

EQ x y ==
does x equal y

NE x y != XOR
x is not equal to y

LT x y <
x is less than y

GT x y >
x is greater than y

LTE x y <=
x is less than or equal to y

GTE x y >=
x is greater than or equal to y

INR l x h ><
x is greater than l and less than h (within range)

OUTR l x h <>
x is less than l or greater than h (out of range)

INRI l x h >=<
x is greater than or equal to l and less than or equal to h (within range, inclusive)

OUTRI l x h <=>
x is less than or equal to l or greater than or equal to h (out of range, inclusive)

EZ x !
x is 0, equivalent to logical NOT

NZ x
x is not 0

LSH x y <<
left shift x by y bits, in effect multiply x by 2 to the power of y

RSH x y >>
right shift x by y bits, in effect divide x by 2 to the power of y

LROT x y <<<
circular left shift x by y bits, wrapping around when bits fall off the end

RROT x y >>>
circular right shift x by y bits, wrapping around when bits fall off the end

| x y
bitwise or x | y

& x y
bitwise and x & y

x y
bitwise xor x ^ y

~ x
bitwise not, i.e.: inversion of x

BSET x y
set bit y in value x

BGET x y
get bit y in value x

BCLR x y
clear bit y in value x

BTOG x y
toggle bit y in value x

BREV x
reverse bit order in value x

ABS x
absolute value of x

AND x y &&
logical AND of x and y

AND3 x y z &&&
logical AND of x, y and z

AND4 x y z a &&&&
logical AND of x, y, z and a

OR x y ||
logical OR of x and y

OR3 x y z |||
logical OR of x, y and z

OR4 x y z a ||||
logical OR of x, y, z and a

JI x y
just intonation helper, precision ratio divider normalised to 1V

SCALE a b x y i SCL
scale i from range a to b to range x to y, i.e. i * (y - x) / (b - a)

ER f l i
Euclidean rhythm, f is fill (1-32), l is length (1-32) and i is step (any value), returns 0 or 1

NR p m f s
Numeric Repeater, p is prime pattern (0-31), m is & mask (0-3), f is variation factor (0-16) and s is step (0-15), returns 0 or 1

BPM x
milliseconds per beat in BPM x

N x
converts an equal temperament note number to a value usable by the CV outputs (x in the range -127 to 127)

VN x
converts 1V/OCT value x to an equal temperament note number

HZ x
converts 1V/OCT value x to Hz/Volt value, useful for controlling non-euro synths like Korg MS-20

N.B d / N.B r s
get degree d of scale/set scale root to r, scale to s, s is either bit mask (s >= 1) or scale preset (s < 1)

N.BX i d / N.BX i r s
multi-index version of N.B, scale at i (index) 0 is shared with N.B

N.S r s d
Note Scale operator, r is the root note (0-127), s is the scale (0-8) and d is the degree (1-7), returns a value from the N table.

N.C r c d
Note Chord operator, r is the root note (0-127), c is the chord (0-12) and d is the degree (0-3), returns a value from the N table.

N.CS r s d c
Note Chord Scale operator, r is the root note (0-127), s is the scale (0-8), d is the scale degree (1-7) and c is the chord component (0-3), returns a value from the N table.

V x
converts a voltage to a value usable by the CV outputs (x between 0 and 10)

VV x
converts a voltage to a value usable by the CV outputs (x between 0 and 1000, 100 represents 1V)

EXP x
exponentiation table lookup. 0-16383 range (V 0-10)

SGN x
sign function: 1 for positive, -1 for negative, 0 for 0

CHAOS x
get next value from chaos generator, or set the current value

CHAOS.R x
get or set the R parameter for the CHAOS generator

CHAOS.ALG x
get or set the algorithm for the CHAOS generator. 0 = LOGISTIC, 1 = CUBIC, 2 = HENON, 3 = CELLULAR

R / R x
get a random number/set R.MIN and R.MAX to same value x (effectively allowing R to be used as a global variable)

R.MIN x
set the lower end of the range from -32768 – 32767, default: 0

R.MAX x
set the upper end of the range from -32768 – 32767, default: 16383

Metronome

M / M x
get/set metronome interval to x (in ms), default 1000, minimum value 25

M! / M! x
get/set metronome to experimental interval x (in ms), minimum value 2

M.ACT / M.ACT x
get/set metronome activation to x (0/1), default 1 (enabled)

M.RESET
hard reset metronome count without triggering

Delay

DEL x: ...
Delay command by x ms

DEL.CLR
Clear the delay buffer

DEL.X x delay_time: ...
Delay x commands at delay_time ms intervals

DEL.R x delay_time: ...
Trigger the command following the colon once immediately, and delay x - 1 commands at delay_time ms intervals

DEL.G x delay_time num denom: ...
Trigger the command once immediately and x - 1 times at ms intervals of delay_time * (num/denom)^n where n ranges from 0 to x - 1.

DEL.B delay_time bitmask: ...
Trigger the command up to 16 times at intervals of delay_time ms. Active intervals set in 16-bit bitmask, LSB = immediate.

Stack

S: ...
Place a command onto the stack

S.CLR
Clear all entries in the stack

S.ALL
Execute all entries in the stack

S.POP
Execute the most recent entry

S.L
Get the length of the stack

Queue

Q / Q x
Modify the queue entries

Q.N / Q.N x
The queue length

Q.AVG / Q.AVG x
Return the average of the queue

Q.CLR / Q.CLR x
Clear queue

Q.GRW / Q.GRW x
Get/set grow state

Q.SUM / Q.SUM x
Get sum of elements

Q.MIN / Q.MIN x
Get/set minimum value

Q.MAX / Q.MAX x
Get/set maximum value

Q.RND / Q.RND x
Get random element/randomize elements

Q.SRT / Q.SRT
Sort all or part of queue

Q.REV
Reverse queue

Q.SH / Q.SH x
Shift elements in queue

Q.ADD x / Q.ADD x i
Perform addition on elements in queue

Q.SUB x / Q.SUB x i
Perform subtraction on elements in queue

Q.MUL x / Q.MUL x i
Perform multiplication on elements in queue

Q.DIV x / Q.DIV x i
Perform division on elements in queue

Q.MOD x / Q.MOD x i
Perform module (%) on elements in queue

Q.I i / Q.I i x
Get/set value of elements at index

Q.2P / Q.2P i
Copy queue to current pattern/copy queue to pattern at index i

Q.P2 / Q.P2 i
Copy current pattern to queue/copy pattern at index i to queue

Seed

SEED / SEED x
get / set the random number generator seed for all SEED ops

RAND.SEED / RAND.SEED x RAND.SD
R.SD
get / set the random number generator seed for R, RRAND, and RAND ops

TOSS.SEED / TOSS.SEED x TOSS.SD
get / set the random number generator seed for the TOSS op

PROB.SEED / PROB.SEED x PROB.SD
get / set the random number generator seed for the PROB mod

DRUNK.SEED / DRUNK.SEED x DRUNK.SD
get / set the random number generator seed for the DRUNK op

P.SEED / P.SEED x P.SD
get / set the random number generator seed for the P.RND and PN.RND ops

Turtle

@ / @ x
get or set the current pattern value under the turtle

@X / @X x
get the turtle X coordinate, or set it to x

@Y / @Y x
get the turtle Y coordinate, or set it to x

@MOVE x y
move the turtle x cells in the X axis and y cells in the Y axis

@F x1 y1 x2 y2
set the turtle's fence to corners x1,y1 and x2,y2

@FX1 / @FX1 x
get the left fence line or set it to x

@FX2 / @FX2 x
get the right fence line or set it to x

@FY1 / @FY1 x
get the top fence line or set it to x

@FY2 / @FY2 x
get the bottom fence line or set it to x

@SPEED / @SPEED x
get the speed of the turtle's @STEP in cells per step or set it to x

@DIR / @DIR x
get the direction of the turtle's @STEP in degrees or set it to x

@STEP
move @SPEED/100 cells forward in @DIR, triggering @SCRIPT on cell change

@BUMP / @BUMP 1
get whether the turtle fence mode is BUMP, or set it to BUMP with 1

@WRAP / @WRAP 1
get whether the turtle fence mode is WRAP, or set it to WRAP with 1

@BOUNCE / @BOUNCE 1
get whether the turtle fence mode is BOUNCE, or set it to BOUNCE with 1

@SCRIPT / @SCRIPT x
get which script runs when the turtle changes cells, or set it to x

@SHOW / @SHOW 0/1
get whether the turtle is displayed on the TRACKER screen, or turn it on or off

Grid

G.RST
full grid reset

G.CLR
clear all LEDs

G.DIM level
set dim level

G.ROTATE x
set grid rotation

G.KEY x y action
emulate grid press

G.GRP / G.GRP id
get/set current group

G.GRP.EN id / G.GRP.EN id x
enable/disable group or check if enabled

G.GRP.RST id
reset all group controls

G.GRP.SW id
switch groups

G.GRP.SC id / G.GRP.SC id script
get/set group script

G.GRPI
get last group

G.LED x y / G.LED x y level
get/set LED

G.LED.C x y
clear LED

G.REC x y w h fill border
draw rectangle

G.RCT x1 y1 x2 y2 fill border
draw rectangle

G.BTN id x y w h type level script
initialize button

G.GBT group id x y w h type level script
initialize button in group

G.BTX id x y w h type level script columns rows
initialize multiple buttons

G.GBX group id x y w h type level script columns rows
initialize multiple buttons in group

G.BTN.EN id / G.BTN.EN id x
enable/disable button or check if enabled

G.BTN.X id / G.BTN.X id x
get/set button x coordinate

G.BTN.Y id / G.BTN.Y id y
get/set button y coordinate

G.BTN.V id / G.BTN.V id value
get/set button value

G.BTN.L id / G.BTN.L id level
get/set button level

G.BTNI
id of last pressed button

G.BTNX / G.BTNX x
get/set x of last pressed button

G.BTNY / G.BTNY y
get/set y of last pressed button

G.BTNV / G.BTNV value
get/set value of last pressed button

G.BTNL / G.BTNL level
get/set level of last pressed button

G.BTN.SW id
switch button

G.BTN.PR id action
emulate button press/release

G.GBTN.V group value
set value for group buttons

G.GBTN.L group odd_level even_level
set level for group buttons

G.GBTN.C group
get count of currently pressed

G.GBTN.I group index
get id of pressed button

G.GBTN.W group
get button block width

G.GBTN.H group
get button block height

G.GBTN.X1 group
get leftmost pressed x

G.GBTN.X2 group
get rightmost pressed x

G.GBTN.Y1 group
get highest pressed y

G.GBTN.Y2 group
get lowest pressed y

G.FDR id x y w h type level script
initialize fader

G.GFD grp id x y w h type level script
initialize fader in group

G.FDX id x y w h type level script columns rows
initialize multiple faders

G.GFX group id x y w h type level script columns rows
initialize multiple faders in group

G.FDR.EN id / G.FDR.EN id x
enable/disable fader or check if enabled

G.FDR.X id / G.FDR.X id x
get/set fader x coordinate

G.FDR.Y id / G.FDR.Y id y
get/set fader y coordinate

G.FDR.N id / G.FDR.N id value
get/set fader value

G.FDR.V id / G.FDR.V id value
get/set scaled fader value

G.FDR.L id / G.FDR.L id level
get/set fader level

G.FDRI
id of last pressed fader

G.FDRX / G.FDRX x
get/set x of last pressed fader

G.FDRY / G.FDRY y
get/set y of last pressed fader

G.FDRN / G.FDRN value
get/set value of last pressed fader

G.FDRV / G.FDRV value
get/set scaled value of last pressed fader

G.FDRL / G.FDRL level
get/set level of last pressed fader

G.FDR.PR id value
emulate fader press

G.GFDR.N group value
set value for group faders

G.GFDR.V group value
set scaled value for group faders

G.GFDR.L group odd_level even_level
set level for group faders

G.GFDR.RN group min max
set range for group faders

MIDI In

MI.$ x / MI.$ x y
assign MIDI event type x to script y

MI.LE
get the latest event type

MI.LCH
get the latest channel (1..16)

MI.LN
get the latest Note On (0..127)

MI.LNV
get the latest Note On scaled to teletype range (shortcut for `N MI.LN`)

MI.LV
get the latest velocity (0..127)

MI.LVV
get the latest velocity scaled to 0..16383 range (0..+10V)

MI.LO
get the latest Note Off (0..127)

MI.LC
get the latest controller number (0..127)

MI.LCC
get the latest controller value (0..127)

MI.LCCV
get the latest controller value scaled to 0..16383 range (0..+10V)

MI.NL
get the number of Note On events

MI.NCH
get the Note On event channel (1..16) at index specified by variable `I`

MI.N
get the Note On (0..127) at index specified by variable `I`

MI.NV
get the Note On scaled to 0..+10V range at index specified by variable `I`

MI.V
get the velocity (0..127) at index specified by variable `I`

MI.VV
get the velocity scaled to 0..10V range at index specified by variable `I`

MI.OL
get the number of Note Off events

MI.OCH
get the Note Off event channel (1..16) at index specified by variable `I`

MI.O
get the Note Off (0..127) at index specified by variable `I`

MI.CL
get the number of controller events

MI.CCH
get the controller event channel (1..16) at index specified by variable `I`

MI.C
get the controller number (0..127) at index specified by variable `I`

MI.CC
get the controller value (0..127) at index specified by variable `I`

MI.CCV
get the controller value scaled to 0..+10V range at index specified by variable `I`

MI.CLKD / MI.CLKD x
set clock divider to x (1-24) or get the current divider

MI.CLKR
reset clock counter

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)

Whitewhale

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.

W/

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

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

Fader

FADER x FB
reads the value of the FADER slider x; default return range is from 0 to 16383

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

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

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 with the portamento rate determined by the TO.OSC.SLEW value; y is 1v/oct translated from the standard range (1-16384); a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.SET x y
set oscillation for CV output x to y (ignores CV.OSC.SLEW); y is 1v/oct translated from the standard range (1-16384); a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.QT x y
targets oscillation for CV output x to y with the portamento rate determined by the TO.OSC.SLEW value; y is 1v/oct translated from the standard range (1-16384) and quantized to current OSC.SCALE; a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.QT.SET x y
set oscillation for CV output x to y (ignores CV.OSC.SLEW); y is 1v/oct translated from the standard range (1-16384) and quantized to current OSC.SCALE; a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.N x y
targets oscillation for CV output x to note y with the portamento rate determined by the TO.OSC.SLEW value; see quantization scale reference for y; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.N.SET x y
sets oscillation for CV output x to note y (ignores CV.OSC.SLEW); see quantization scale reference for y; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.FQ x y
targets oscillation for CV output x to frequency y with the portamento rate determined by the TO.OSC.SLEW value; y is in Hz; a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.FQ.SET x y
sets oscillation for CV output x to frequency y (ignores CV.OSC.SLEW); y is in Hz; a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.LFO x y
targets oscillation for CV output x to LFO frequency y with the portamento rate determined by the TO.OSC.SLEW value; y is in mHz (millihertz: 10^-3 Hz); a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

TO.OSC.LFO.SET x y
sets oscillation for CV output x to LFO frequency y (ignores CV.OSC.SLEW); y is in mHz (millihertz: 10^-3 Hz); a value of 0 disables oscillation; CV amplitude is used as the peak for oscillation and needs to be > 0 for it to be perceivable

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 values range 0-4500. There are 45 different waveforms, values translate to sine (0), triangle (100), saw (200), pulse (300) all the way to random/noise (4500); oscillator shape between values is a blend of the pure waveforms

TO.OSC.RECT x y
rectifies the polarity of the oscillator for output x to y; range for y is -2 to 2; default is 0 (no rectification); 1 & -1 are partial rectification - omitting all values on the other side of the sign; 2 & -2 are full rectification - inverting values from the other pole

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
This parameter essentially allows output x to act as a gate between the 0 and 1 state. Changing this value from 0 to 1 causes the envelope to trigger the attack phase and hold at the peak CV value; changing this value from 1 to 0 causes the decay stage of the envelope to be triggered.

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
fires a PULSE at the End of Rise to the unit-local trigger output 'n' for the envelope on CV output x; n refers to trigger output 1-4 on the same TXo as CV output 'y'

TO.ENV.EOC x n
fires a PULSE at the End of Cycle to the unit-local trigger output 'n' for the envelope on CV output x; n refers to trigger output 1-4 on the same TXo as CV output 'y'

TO.ENV.LOOP x y
causes the envelope on CV output x to loop for y times; a y of 0 will cause the envelope to loop infinitely; setting y to 1 (default) disables looping and (if currently looping) will cause it to finish its current cycle and cease

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)

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

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

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

EX.M.PB x
send MIDI Pitchbend message

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

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

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.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)

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!

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

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

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

EX.M.PB x
send MIDI Pitchbend message

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

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

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)`.