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

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 0

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

Hardware

CV / CV x
CV target value

CF.OFF x / CF.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
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

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

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

TR.POL x / TR.POL y x y
Set polarity of trigger output to 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

Patterns

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

P x / P 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)

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

PN.START x / PN.START x y
set the start location of pattern x

PN.END x / PN.END x y
set the end location of pattern x

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

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

PN.WRAP x / PN.WRAP x y
non-destructive to date

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

PN.MAX x
return the maximum value of pattern between the start and end of pattern

PN.MIN x
find the first minimum value in the pattern between the start and end of pattern

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

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

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

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

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

P.PUSH x
insert value x to the end of the working pattern

PN.PUSH x y
insert value y to the end of the working pattern

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

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

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

PN.PUSH x
insert value x to the end of the working pattern

PN.PUSH x y
insert value y to the end of the working pattern

PN.PUSH x y z
insert value y at index x 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

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

P.PUSH x
insert value x to the end of the working pattern

PN.PUSH x y
insert value y to the end of the working pattern

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

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, execute command

RUN 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: ...
run the command every x times the command is called

IF 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 (1-8), recursion allowed

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

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

BREAK
halts execution of the current script

INIT
clears all state data
**Maths**

- **ADD** a x y +
- **SUB** x y -
- **MUL** x y *
- **DIV** x y / divide by y
- **MOD** x y %
- **RAND** x RND generate a random number between 0 and x inclusive
- **RRAND** x y RND generate a random number between x and y inclusive
- **TOSST** randomly return 0 or 1

- 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

- **SCALE a b c x** SCL scale from a to c to range x to y, i.e. 1 (* y - x) / (b - a)
- **ER f l** Euclidean rhythm, f is fill (1-32), l is length (1-32) and i is step (any value), returns 0 or 1

**Trajectory**

- **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 (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

**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, GRP** RPT get last group
- **G, LED x y / G, LED y x level** get/set LED
- **G, LED, C x y clear LED
- **G, REC** x y h fill border draw rectangle
- **G, RCT x y x2 y2 fill border draw rectangle
- **G, BGN id x y h type level script** initialize button
**Ansible**

- **KR.PRE** / **KR.PRE**  x
  - Get/set 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 x, 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 z / **KR.L.LEN** x y
  - Get/set length of track
- **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
  - Get/set current preset / load preset x
- **Cy.RES** x
  - Reset channel x (0 = all)
- **Cy.POS** x y / **Cy.POS** x y z
  - 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** c
  - 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)
- **LV.POS** x
  - Get / set position of channel x (0 = all)
- **LV.RES** x
  - Reset position to loop start for track x, parameter y
- **LV.CV** x
  - Get/set current CV value for channel x
- **Kr.MUTE** x / **Kr.MUTE** x y
  - Get/set mute state for channel x, parameter y / set to z
- **Kr.POS** x y / **Kr.POS** x y z / **Kr.POS** x y
  - Get/set current scale
- **Kr.VOL** / **Kr.VOL** x
  - Get/set current volume
- **Kr.TMUTE** x
  - Toggle mute state for channel
- **Kr.MUTE2**
  - Get/set mute state for channel
- **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, parameter y / set to z
- **Kr.RST** x
  - Stop track x (1-4)
- **Orca**
- **OR.CLOCK** x
  - Advance track x (1-4)
- **OR.RST** x
  - Reset track x (1-4)
- **OR.GRST** x / **OR.GRST** x y
  - Global reset (x can be any value)
- **OR.Trim** x
  - Choose track (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 (8-16)
- **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)
- **Orca**
- **Orca**
  - Select scale (1-16)
- **OR.BANK** x
  - Select preset bank x (1-8)
- **Orca**
  - Reset preset or bank (1-8)
- **OR.RELOAD** x
  - Reload preset or bank (8 - current preset, 1 - individual channels, 2 - all banks)
- **OR.ROTS** x
  - Rotate weights by x (1-3)
- **Orca**
  - Select tracks for CV A where x is a binary number representing the tracks
- **Orca**
  - Select tracks for CV B where x is a binary number representing the tracks

**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.TRIPE** x
  - Recall triple shape (1-4)
- **Es.MAGIC** x
  - Magic shape (1= halfspeed, 2=doublespeed, 3=linearize)
Just Friends

**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.

**JF.RUN**
x
Send a 'voltage' to the RUN input. Requires JF.RMODE 1 to have been executed, or a physical cable & control voltage to set the RUN parameter. Use JF.RUN V to set to x volts. The expected range is V -5 to V 5

**JFSHIFT**
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 velocity. 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, y 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 divider for the tuning ratio), z = denominator (set the divisor for the tuning ratio).

**JF.MODE**
x
Set the current choice of standard functionality, or Just Type alternate modes. 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
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).

**JF.NOTE**
x y
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.

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

**JF.TR**
x y
Simulate a TRIGGER input. x is channel (0 = all) 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)

**Fader**

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

**Matrixarchate**

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

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

**MA.COL**
col
set the connection at row x and column y in the current program (rows/columns are 0-based)

**MA.PCOL**
col pgm
set the connection at row x and column y in program pgm

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

**MA.PGM**
pgm
select the current program in program pgm

**MA.PGM**
pgm
connect row x and column y in the current program

**MA.PGM**
pgm
connect row x and column y in program pgm

**MA.PGM**
pgm
connect row x and column y in program pgm

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

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

**MA.COL**
col
set the connection at row x and column y in the current program (rows/columns are 0-based)

**MA.PCOL**
col pgm
set the connection at row x and column y in program pgm

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

**MA.PGM**
pgm
connect row x and column y in the current program

**MA.PGM**
pgm
connect row x and column y in program pgm

**MA.PGM**
pgm
connect row x and column y in program pgm

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

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

**MA.COL**
col
set the connection at row x and column y in the current program (rows/columns are 0-based)

**MA.PCOL**
col pgm
set the connection at row x and column y in program pgm

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

**MA.PGM**
pgm
connect row x and column y in the current program

**MA.PGM**
pgm
connect row x and column y in program pgm

**MA.PGM**
pgm
connect row x and column y in program pgm

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

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

**MA.COL**
col
set the connection at row x and column y in the current program (rows/columns are 0-based)

**MA.PCOL**
col pgm
set the connection at row x and column y in program pgm

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

**MA.PGM**
pgm
connect row x and column y in the current program

**MA.PGM**
pgm
connect row x and column y in program pgm

**MA.PGM**
pgm
connect row x and column y in program pgm

**MA.PGM**
pgm
connect row x and column y in program pgm

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

**MA.STEP**
advances 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.ON**
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**
x y state
set the connection at row x and column y in program pgm to state (1 - on, 0 - off)
TI.IN.QT x

returns the quantized value for PARAM knob x; default return range from 0 to 16383; return range can be altered by the TI.PRM.MAP command

TI.PRAM.QT x

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

TI.PRAM.N x

TI.PRM.N return the quantized note number for PARAM knob x using the scale set by TI.PRAM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.IN x

TI.IN 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

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

TI.IN.N x

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

TI.IN.SCALE x

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

TI.IN.MAP x y z

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

TI.PARAM.x

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

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

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

TI.PARAM.x

TI.PARAM.x 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 returns the quantized value for PARAM knob x using the scale set by TI.PRM.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.PRM.SCALE, select scale # y for PARAM knob x; scales listed in full description

TI.PARAM.MAP x y z

TI.PARAM.MAP maps the PARAM values for input x across the range y - z (defaults 0-16383)
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO.OSC.CYC.x y</td>
<td>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</td>
</tr>
<tr>
<td>TO.OSC.CYC.SET.x y</td>
<td>sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in seconds</td>
</tr>
<tr>
<td>TO.OSC.CYC.M x y</td>
<td>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</td>
</tr>
<tr>
<td>TO.OSC.CYC.M.SET.x y</td>
<td>sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in minutes</td>
</tr>
<tr>
<td>TO.OSC.SLEW.x y</td>
<td>the oscillator on CV output sets the frequency slew time (portamento) for CV output x with the portamento rate determined by the TO.OSC.SLEW value; y is in seconds</td>
</tr>
<tr>
<td>TO.OSC.SLEW.S.x y</td>
<td>sets the oscillator cycle length to y for CV output x (ignores CV.OSC.SLEW); y is in seconds</td>
</tr>
<tr>
<td>TO.OSC.WIDTH.x y</td>
<td>sets the waveform for output x to y; y values range 0-4999; values translate to sine (0), triangle (1000), saw (2000), pulse (3000), or noise (4000); oscillator shape between values is a blend of the pure waveforms</td>
</tr>
<tr>
<td>TO.OSC.RECT.x y</td>
<td>rectifies the polarity of the oscillator for output x to y; range for y is -2 to 2; default is 0 (no rectification); 1 &amp; -1 are partial rectification - omitting all values on the other side of the sign; 2 &amp; -2 are full rectification - inverting values from the other pole</td>
</tr>
<tr>
<td>TO.OSC.PHASE.x y</td>
<td>resets the phase of the oscillator on CV output x to y (0 to 16383); y is the range of one cycle</td>
</tr>
<tr>
<td>TO.OSC.SLEW.x y</td>
<td>sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in milliseconds</td>
</tr>
<tr>
<td>TO.OSC.SLEW.S.x y</td>
<td>sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in milliseconds</td>
</tr>
<tr>
<td>TO.OSC.SLEW.M.x y</td>
<td>sets the frequency slew time (portamento) for the oscillator on CV output x to y; y in minutes</td>
</tr>
<tr>
<td>TO.OSC.CTR.x y</td>
<td>centers the oscillator on CV output x to y; y values are bipolar (-16384 to +16383) and map to -10 to +10</td>
</tr>
<tr>
<td>TO.ENV.ACT.x y</td>
<td>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 &gt; 8 for the envelope to be perceivable</td>
</tr>
<tr>
<td>TO.ENV.x y</td>
<td>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.</td>
</tr>
<tr>
<td>TO.ENV.TRIG.x y</td>
<td>triggers the envelope at CV output x to cycle; CV amplitude is used as the peak for the envelope and needs to be &gt; 8 for the envelope to be perceivable</td>
</tr>
<tr>
<td>TO.ENV.DEC.x y</td>
<td>set the envelope decay time to y for CV output x; y in milliseconds</td>
</tr>
<tr>
<td>TO.ENV.DEC.M.x y</td>
<td>set the envelope attack time to y for CV output x; y in seconds</td>
</tr>
<tr>
<td>TO.ENV.DEC.S.x y</td>
<td>set the envelope decay time to y for CV output x; y in seconds</td>
</tr>
<tr>
<td>TO.ENV.DEC.M.x y</td>
<td>set the envelope attack time to y for CV output x; y in minutes</td>
</tr>
<tr>
<td>TO.ENV.EOC.x n</td>
<td>fires a PULSE at the End of Rise to the unit-local trigger output 'y' for the envelope on CV output x; n refers to trigger output 1-4 on the same TXo as CV output 'y'</td>
</tr>
<tr>
<td>TO.ENV.EOR.x n</td>
<td>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'</td>
</tr>
<tr>
<td>TO.ENV.LEE.x y</td>
<td>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</td>
</tr>
<tr>
<td>TO.TR.INIT.x y</td>
<td>initializes TR output x back to the default boot settings and behaviors; neutralizes metronomes, dividers, pulse counters, etc.</td>
</tr>
<tr>
<td>TO.CV.INIT.x y</td>
<td>initializes CV output x back to the default boot settings and behaviors; neutralizes offsets, slew, envelopes, oscillator, etc.</td>
</tr>
<tr>
<td>TO.INIT.d</td>
<td>initializes all of the TR and CV outputs for device number d (1-8)</td>
</tr>
<tr>
<td>TO.KILL.d</td>
<td>cancels all TR pulses and CV slews for device number d (1-8)</td>
</tr>
</tbody>
</table>