Command:SCAN (SN)

MS2000 or RM2000 syntax and Function

Shortcut	SN
Format	SCAN [X=scan_axis_x] [Y=scan_axis_y] [Z=scan_axis_z] [F=pattern]
Units	Integer
Remembered	Using SS Z
Firmware Required	SCAN MODULE or ARRAY MODULE

Sets which axes are to be used for 2D raster scan. The same axis settings **also apply to the ARRAY module**. The fast-scanned raster axis (horizontal) is defined by `scan_axis = 1`; the slow-scanned axis (vertical) is defined by `scan_axis = 2`. Single axis scans (1D) requires setting the unused axes `scan_axis = 0`, and the driven axis as `scan_axis = 1`. **Note: this is different from the Tiger usage of pseudoaxes X/Y/Z.**

To set multiple axes in the SCAN command, you will need to set all of the axes to 0 first.

If you try to set the axis index and that index is already assigned to another axis, then the value will not change. For example, if you use **SCAN X=2** and the Y axis is already set to index 2, nothing will happen. The solution is to set **SCAN Y=0** and then you will be able to change the X axis index.

No Arguments: The command SCAN initiates (or stops) a scan using parameters set with the SCANR and SCANV commands.

X: `[scan_axis_x]` Set or query the scan axis for the X axis.

Y: `[scan_axis_y]` Set or query the scan axis for the Y axis.

Z: `[scan_axis_z]` Set or query the scan axis for the Z axis.

Scan Axis		
Code	**Name**	**Description**
0	None	No Axis
1	Horz	Fast Axis
2	Vert	Slow Axis

F: `[pattern]` The scan pattern may be set to 0 for RASTER scans or 1 for SERPENTINE scans. This setting defaults to RASTER for firmware with the SCAN MODULE included and SERPENTINE for firmware with the ARRAY MODULE included. When the number of lines (SCANV Z) is set to the default of 1 then the behavior is the same for both raster and serpentine scans.

Scan Pattern
0 - RASTER
1 - SERPENTINE

```
SCAN X? Y? Z?
:A X=Horz Y=Vert Z=None
SCAN X=0 Y=0 Z=0
:A
SCAN X=2 Y=1 Z=0
:A
SCAN X? Y? Z?
:A X=Vert Y=Horz Z=None
```

Tiger (motorized) syntax

Shortcut	SN
Format	[addr#]SCAN [X?] [Y=fast_axis_id] [Z=slow_axis_id] [F=pattern]
Units	Integer
Type	Card-Addressed
Remembered	Using [addr#]SS Z
Firmware Required	SCAN MODULE or ARRAY MODULE

Note multiple small changes in usage from MS-2000 command set.

No Arguments: The command SCAN initiates (or stops) a scan using parameters set with the SCANR and SCANV commands.

X: `[scan_state]` Specifying an argument for the pseudoaxis X in decimal sets the state directly (see table below; the value is simply the decimal representation of the corresponding state character). Note that the firmware expects only certain states to be set by the user (marked as “OK to set” in the table); setting to a different state may yield unpredictable results. Querying the pseudoaxis X value returns the character associated with the current state.

Y/Z: `[axis_id]` Specify the cards axis ID as the `fast_axis_id` or `slow_axis_id` (axis IDs are obtainable using Z2B query; they are 0 and 1 for X and Y axes respectively on a two-axis motor card). If `slow_axis_id` is set to 9 then a true single-axis scan will be executed (not even anti-backlash moves on the slow axis); if `slow_axis_id` is specified but the slow axis start and stop positions (NV X and NV Y) are equal then a single-axis scan will execute but the slow axis position will be checked at the scan turnaround points. **Note: this is different from the MS-2000 usage of pseudoaxes X/Y/Z.**

F: `[pattern]` The scan pattern may be set to 0 for RASTER scans or 1 for SERPENTINE scans.

Scan states (SCAN_MODULE firmware)			
Char	**Dec**	**OK to set?**	**State**
I		No	Idle/disabled
S	83	Yes	Starts state machine
P	80	Yes	Stop (goes to idle state after cleanup)
a		No	Waits until slow axis move complete
b		No	Starts move along fast axis
c		No	Waits for fast axis move to finish
d		No	Starts serpentine slow axis move
e		No	Waits until serpentine move complete
f		No	Used in XF_SPIM only

Scan states (SCAN_MODULE firmware)			
Char	**Dec**	**OK to set?**	**State**
g		No	Waits until retrace is complete
h		No	Scan complete, cleans up
t		No	Scan init, if `scan_overshoot` is not 0 (NV T)

Tiger (micro-mirror) syntax

Shortcut	SN
Format	[addr#]SCAN [X=state]
Units	Integer
Type	Card-Addressed
Remembered	Using [addr#]SS Z
Firmware Required	MM_SPIM

No Arguments: Starts or arms the SPIM state machine or terminates it if running or armed (starts state machine execution for Micro-mirror, puts in arm state for piezo). By so doing, any active single-axis functions will be stopped and the SPIM positions/steps will be calculated according to the active parameters (e.g. SAA, SA0, NR, NV, RT).

X: `[state]` Specifying an argument for the pseudo-axis X in decimal sets the state directly (see table below; the value is simply the decimal representation of the corresponding state character). Note that the firmware expects only certain states to be set by the user (marked as “OK to set” in the table); setting to a different state may yield unpredictable results. Querying the pseudo-axis X value returns the character associated with the current state.

Micro-mirror SPIM states (MM_SPIM firmware)			
Char	**Dec**	**OK to set?**	**State**
I		No	Idle/disabled
S	83	Yes	Starts main acquisition state machine
a	97	Yes	Arm for trigger (goes to state 'A')
A		No	Armed and waiting
T	84	Yes	Trigger from state 'A' (Requires firmware v3.37+)
P	80	Yes	Stop (goes to state 'I')
M		No	In middle of sheet (executing per-sheet scan/camera/laser state machines)
s		No	Starting sheet
c		No	Incrementing sheet
R		No	Starting side
y		No	Delay between sides
Y		No	Delay between repeats

Note: Other undocumented states may be used during SPIM state machine execution.

scan, array, commands, tiger, ms2000