OCT Engine Serial Command Reference

ENG-0060

Revision History

Date	Author	Changes	Rev.
4/26/21	J. Price	Added introduction section	K
12/1/2020	J. Price	Combine Rev I Changes and 110587 command differences from 110062	J
6/29/2020	J. Price	Added mirror command, a_div/b_scans details, remove watermark	I
6/26/2018	J. Traud	Added Pause and Pause Timeout commands	H
5/23/2018	J. Traud	Added Rotating Cross Command	G
1/8/2018	J. Traud	Added stepper motor acceleration and speed commands.	F
1/8/2018	J. Traud	Added figures to support sweep commands and scan patterns.	E
11/6/2017	J. Traud	Re-specified the “repeated scan” parameter to be “total scans”	D
5/3/2017	J. Price	Added new commands in OCTX firmware	C
1/25/2016	J. Traud	Updated pramp command inputs	B
12/2/2015	J. Traud	Revised Document Layout and updated command description for linear ramping modes.	A

Contents

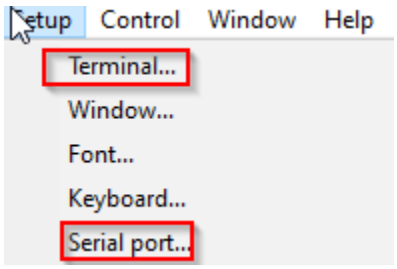
Revision History 1
Introduction 2
System Commands 3
Sweep Commands 4
Motor Commands 18

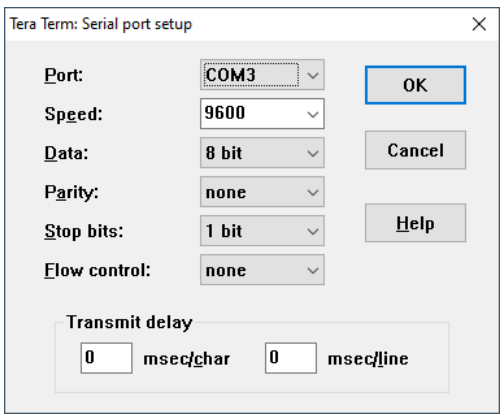
Introduction

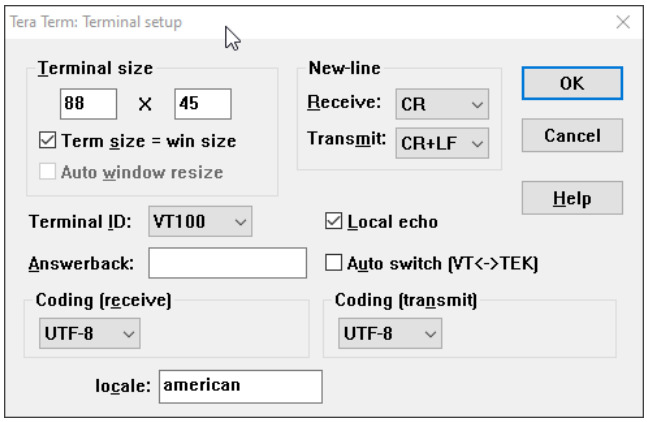
The 110062 and 110587 OCT control boards are controlled via serial commands over a USB STMicroelectronics Virtual COM Port. These commands can be sent through a terminal program like Tera Term or from the setup mode in Spark OCT software.

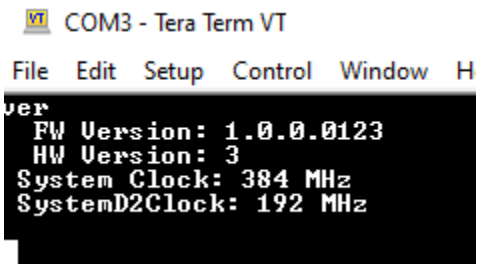
https://osdn.net/projects/ttssh2/releases/

Configure the serial port in TeraTerm as shown below in the "Terminal" and "Serial port…" setup windows. The save and restore options can be used to save presets for TeraTerm.









System Commands

Command:	ver
Response:	Ver:#.## A
Remarks:	The **Version** function responds with the firmware version.

Command:	reset
Response:	none
Remarks:	The **Reset** function resets the microcontroller. This does not reset your computer's USB connection and will require you to plug and unplug USB cable to reconnect.

Command:	dfu
Response:	A
Remarks:	The **DFU** function places the electronics into the Device Firmware Update mode. Will last 3 seconds before deactivating.

Command:	ping
Response:	A
Remarks:	The **Ping** function tests to see if the board is active. Command only responds with an acknowledge.

Command:	foci <value>
Response:	A
Remarks:	The **foci** function provides voltage control for the liquid lens. Can be any value between 0 and 255. Will respond with setting if value not entered.

Command:	out1 <value> out2 <value>
Response:	A
Remarks:	The **Output** functions controls the state of OUT1 and OUT2 on the control board. These are 3.3V outputs. A value of 0 turns the output OFF while a non-zero value will turn the output ON. Output 1 – Typically connected to an onboard SLD. Output 2 – Typically connected to a sample illuminator.

Sweep Commands

Command:	<Variable> <#####>
Response:	"ok.\n"
Remarks:	Will respond with current setting if value not entered.

Variable	Range	Default	Function
a_scans	2-65535	1000	Sets the number of trigger steps for each sweep. Set before filling x ramp and y ramp.
b_scans	0-65534	0	Sets the number of scan paths in a 2D scan in even numbers. If set to 0 while performing an XY scan the X and Y values will sweep in unison. When set greater than 1 this will perform an area raster scan (3D). Multiples of 2.
delay	0-65535	50	(In µs) Time between pulses sent to the camera. Minimum value of 3. *For 110587: quarter microsecond per value
pulse	0-65535	5	(In µs) Duration of Camera Pulse. If set to zero no trigger pulses occur. *For 110587: quarter microsecond per value
t_ret	0-255	7	(In µs) Duration of return pulse + delay (but no trigger) *For 110587: quarter microsecond per value
a_div	0-65535	1	On 110062 if supported in the firmware version, this Scales the return path time. The return path plots the same amount of points as the scanning path. The a_div register acts as a divisor for the amount of counts. If your a_scans is set to 1024 and a_div is set to 2, then you will plot 1024 points when scanning and 512 points while returning. For 110587: a_div represents the total number of points for the return path so it can stay constant even when a_scans is increased.
phase	0-65535	5	Adjusts phase relationship between mirror positioning and triggering. This compensates for delays in motor movement in respect to triggering pulses. **Should not be greater than trdelay.**
trigger	0-255	0	Return trigger enable. If set to zero than no triggers appear on the return sweep and the return time is active. If non-zero, triggers occur on both parts of the sweep.
a_hold	0-65535	0	Number of triggerless pulses between a scans. For stabilization of mechanics.
b_hold	0-65535	0	Number of triggerless pulses between b scans. For stabilization of mechanics.
trdelay	0-65535	0	Added counts to the beginning and end of a scanning path to ensure position accuracy and stability.
trdmode	0, 1	1	A value of 1 adds the trigger delay to the beginning and end of a sweep.

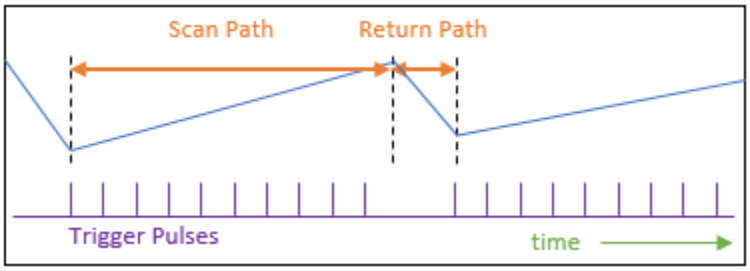


Figure 1 – The register *a_scans* represents the amount of "acquisitions" or "triggers" sent to the camera during a scan path. The image to the left shows an *a_scans* value of 10.

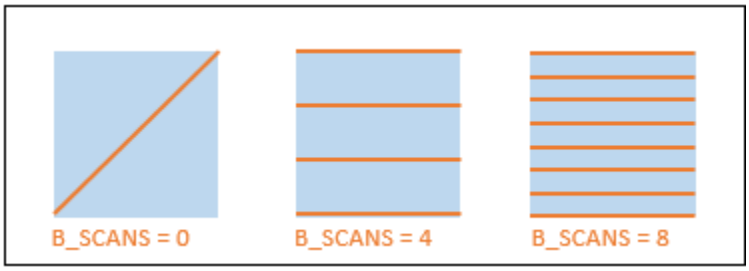


Figure 2 – The register b_scans defines the amount of scan paths used when performing an area scan.

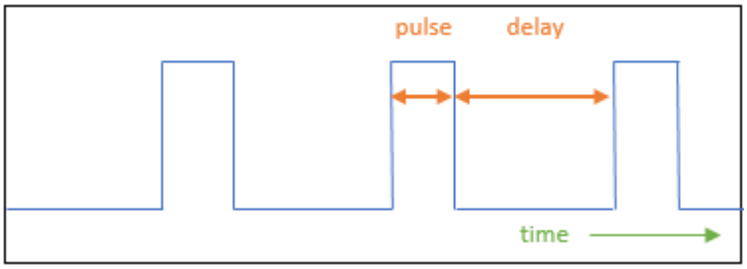


Figure 3 – The timing for the Camera trigger is defined by the *pulse* and the *delay* registers where pulse defines the width of the pulse sent to the camera and the delay is the spacing in between pulses.

The sum of these two registers also defines the scanning speed. The scanning path takes `(a_scans+2*trdelay)*(pulse+delay)` us to complete.

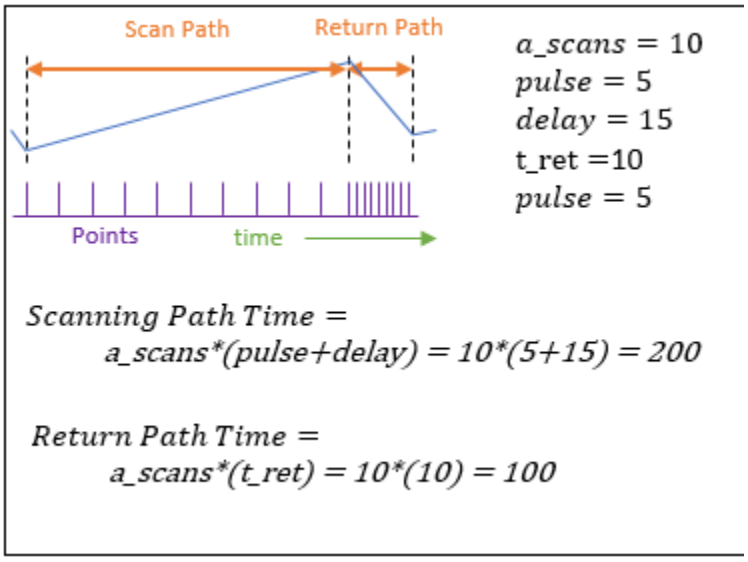


Figure 4 – The timing for the return path is defined by the `t_ret` register. Likewise with the timing of the scanning path, the total return path time is `(a_scans+2*trdelay)*(tret)`.

For 110587: a_div*(t_ret) represents return path

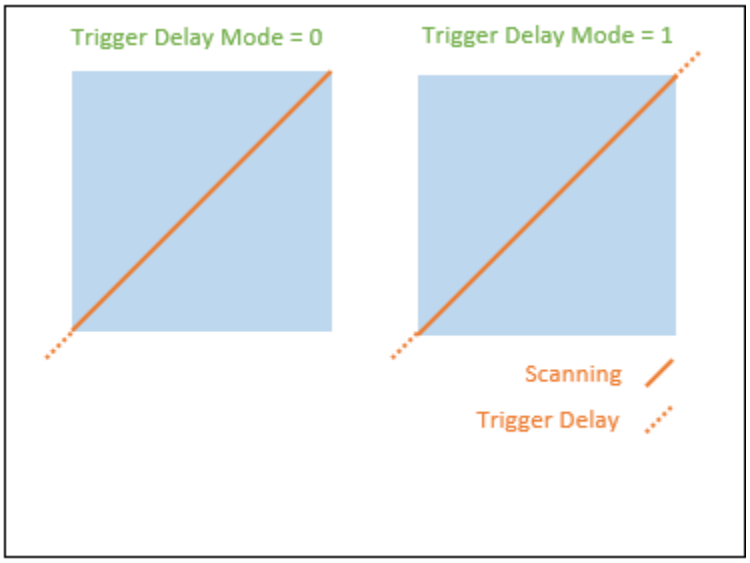


Figure 5 – The `trdelay` register defines the amount of points plotted prior to the scanning path. This extends our path outside of the defined scanning area to ensure position accuracy.

The `trdmode` register defines if the added points are added at the beginning of a path or at both the beginning and the end.

Command:	xramp <start> <stop> yramp <start> <stop>
Response:	A
Remarks:	The **X Ramp** and **Y Ramp** functions perform a linear scan defined by the input parameters. **Start** -- The starting X or Y coordinate. Acceptable range is from 0 to 65,535. **Stop** -- The starting X or Y coordinate. Acceptable range is from 0 to 65,535.

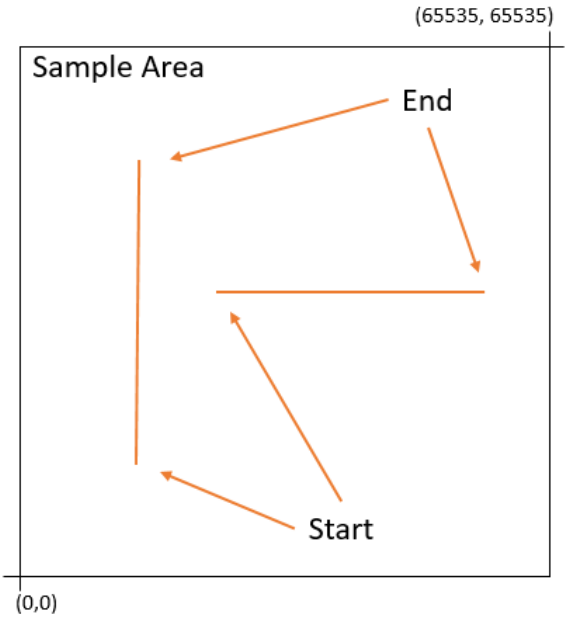


Figure 6 – Defining the Scan Parameters

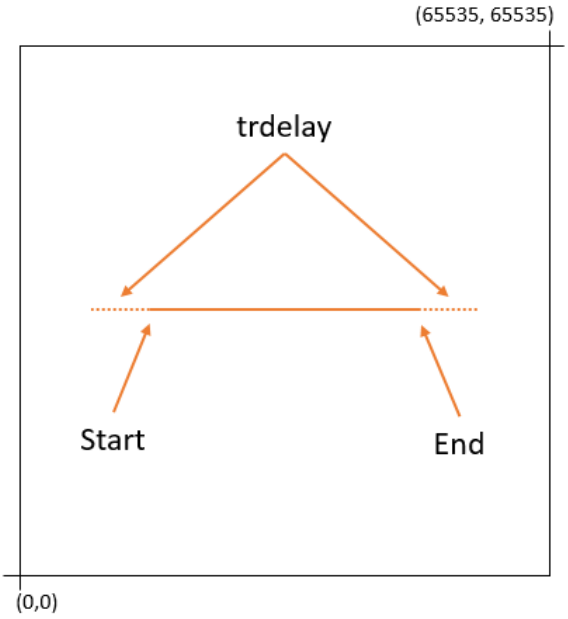


Figure 7 – Impact of Trigger Delay

Command:	xy_ramp <X start> <X stop> <Y start> <Y stop> xy_ramp <X start> <X stop> <Y start> <Y stop> <passes>
Response:	A
Remarks:	The **XY Ramp** function performs either a linear scan defined by the input parameters when the b_scans register is 0 or performs an area scan when the b_scans register is non-zero. **Start** -- The starting X or Y coordinate. Acceptable range is from 0 to 65,535. **Stop** -- The starting X or Y coordinate. Acceptable range is from 0 to 65,535. **Passes** – Total number of scans for the specified path. A value greater than 1 specifies that the scan path is repeated and resulting data is averaged in software. When the b_scans register is equal to 0 then a single linear scan is performed using the X and Y, Start and End parameters. When the b_scans register is greater than 1 we perform an area scan. An area scan is a series of X Ramp functions with the Y value equally distributed across the defined vertical space. The Trigger Delay and Trigger Delay Modes impact the scan pattern in the same manner as the X Ramp and Y Ramp functions. The Trigger Delay adds points to the scan line outside of the designated area. The Trigger Delay Mode defines if points are added just to the beginning or to the beginning and end of the scan.

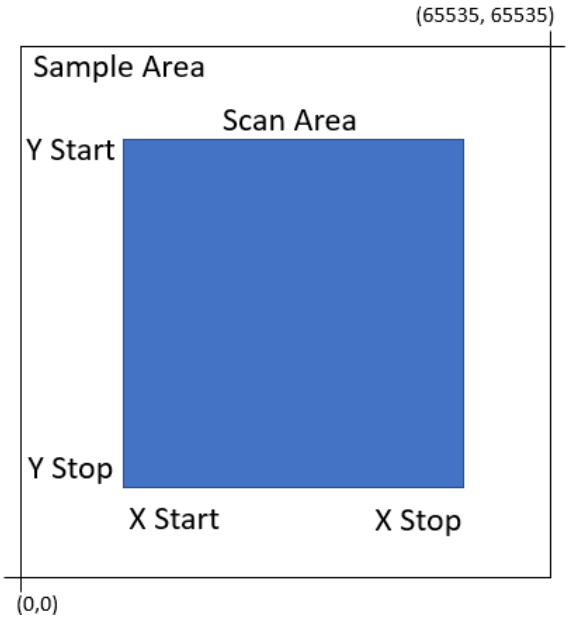


Figure 8 – Defining the scan area

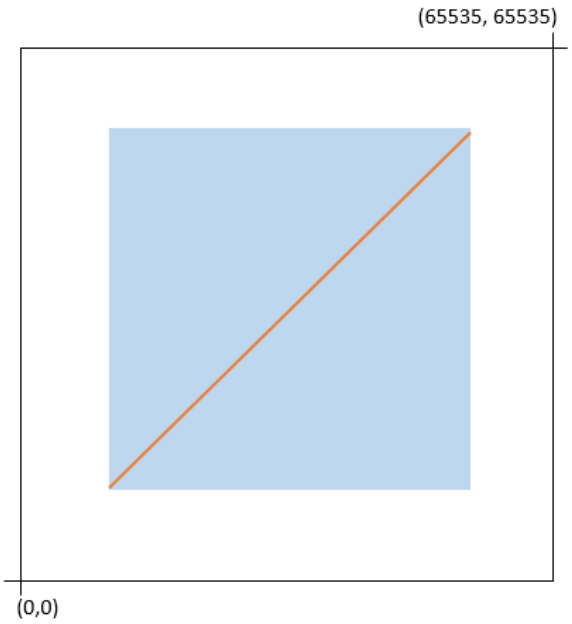


Figure 9 – Impact of b_scans = 0

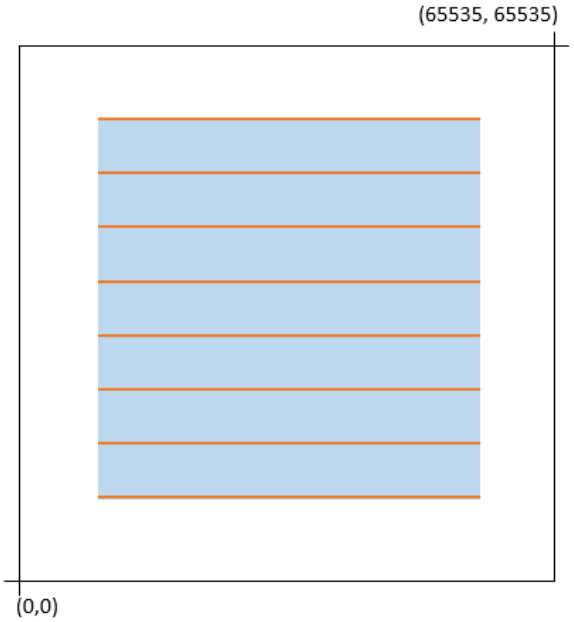


Figure 10 – Impact of b_scans = 8

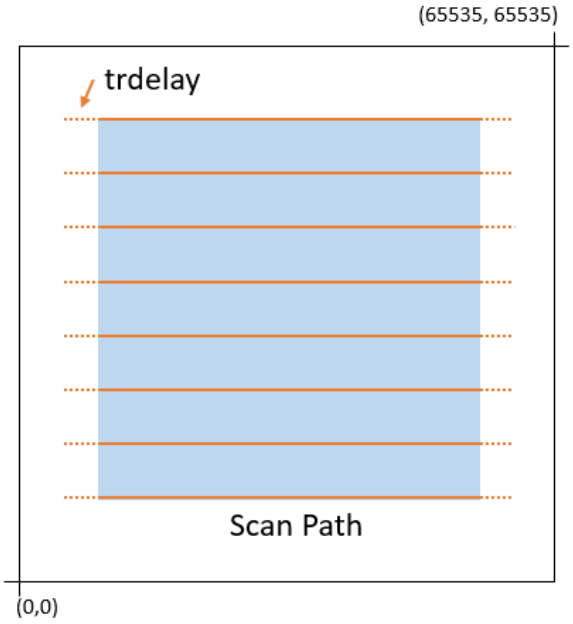


Figure 11 – Impact of trdelay

Command:	pramp <Center X> <Center Y> <radius> <passes>
Response:	A
Remarks:	The **Polar Ramp** performs a scan in the form of concentric circles. **Center –**The center values will indicate the origin position for the scan in DAC values **Radius** – Initial and most outward concentric circle **Passes –** The amount of times each individual circle is scanned before the next The scan pattern begins by creating a circle using the defined center and radius defined by the command. Subsequent circles have a reduced radius defined by the initial radius divided by the total number of b_scans. Trigger delay is used here by beginning each new circle prior to the starting location. When the parameter “passes” is greater than 1 each individual circle is repeated before the radius is decremented for the next circle. In between each circle the trdelay value is ignored. Instead the pattern continues to scan around the circle.

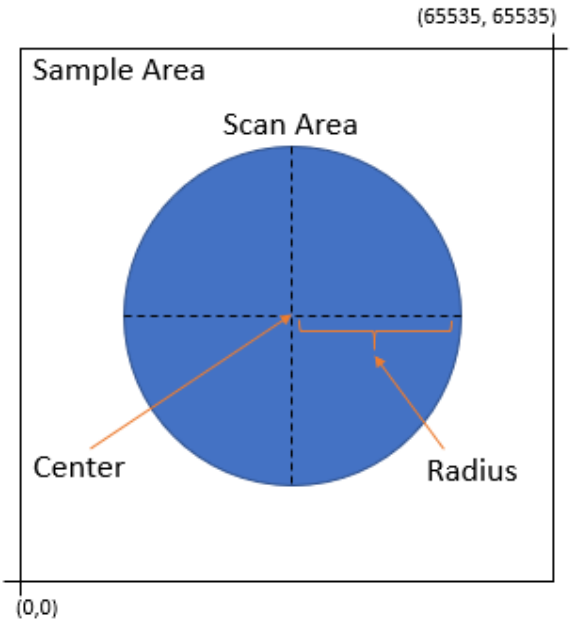


Figure 12 – Defining the Scan Parameters

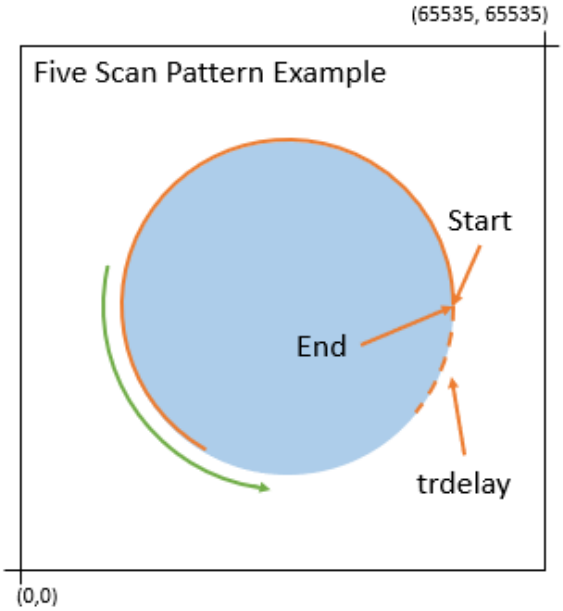


Figure 13 – Impact of Trigger Delay

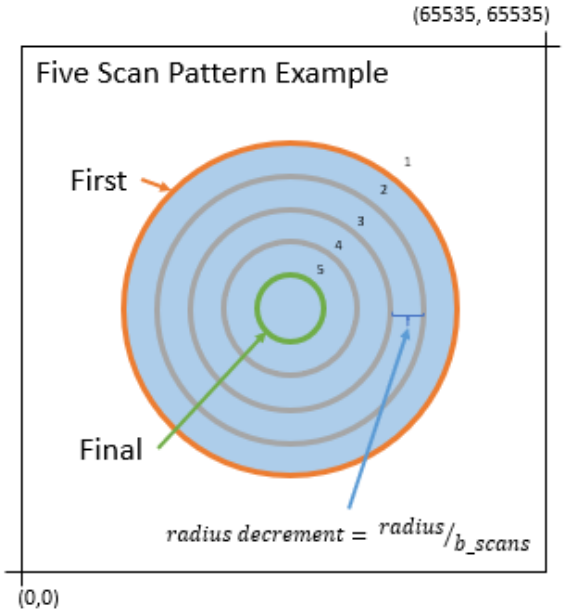


Figure 14 – Impact of b_scans

Command:	sramp <Center X> <Center Y> <radius>
Response:	A
Remarks:	The **Spiral Ramp** function performs an Archimedean spiral with the defined origin. **Center –**The center values will indicate the origin position for the scan in DAC values **Radius** – Initial, outward most concentric circle The total number of spirals is defined by the separate b_scans register. The total number of points scanned is equal to the square of b_scans with each individual scanned point equally distributed over the total arc length of the spiral.

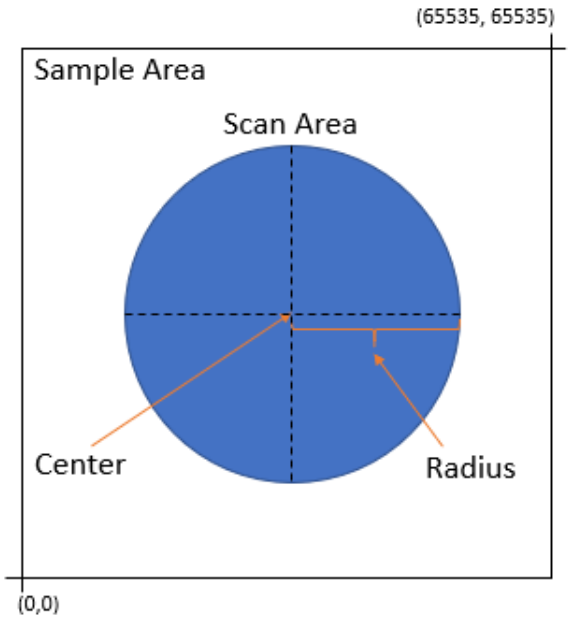


Figure 15 – Defining the Scan Parameters

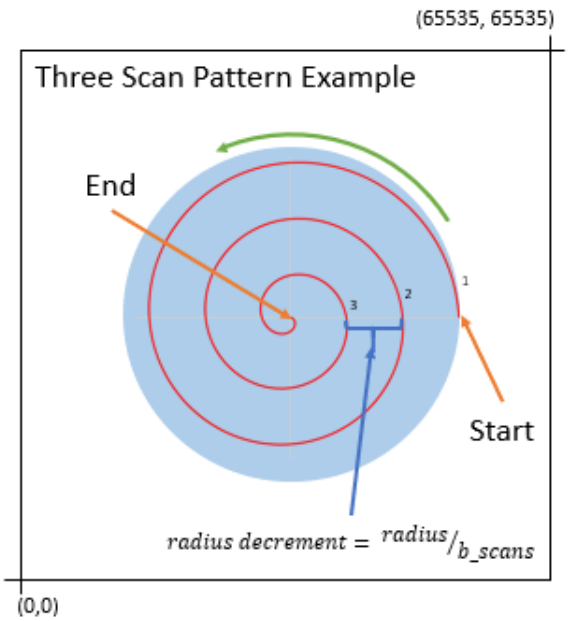


Figure 16 – Impact of b_scans

Command:	rramp <Center X> <Center Y> <radius> <slices> <passes>
Response:	A
Remarks:	The **Radial Ramp** function performs a scanning pattern in which we achieve a circular scan by performing a series of linear "slices" through the origin. **Center –**The center values will indicate the origin position for the scan in DAC values **Radius** – Initial, outward most concentric circle **Slices –** The total number of scans through the origin **Passes –** The amount of times each individual slices is scanned before the next The initial linear scan begins at an x value of the radius and ends at the negative radius. The incremented angle between each subsequent linear scanning path is defined by pi/slices. This means that each scan will originate in either the first or second quadrant. When a value greater than 1 is entered for "passes" each individual line is repeated before proceeding to the next. Trigger delay is used in the same fashion as the X/Y Ramp and XY Ramp scanning patters with trdmode forced ON.

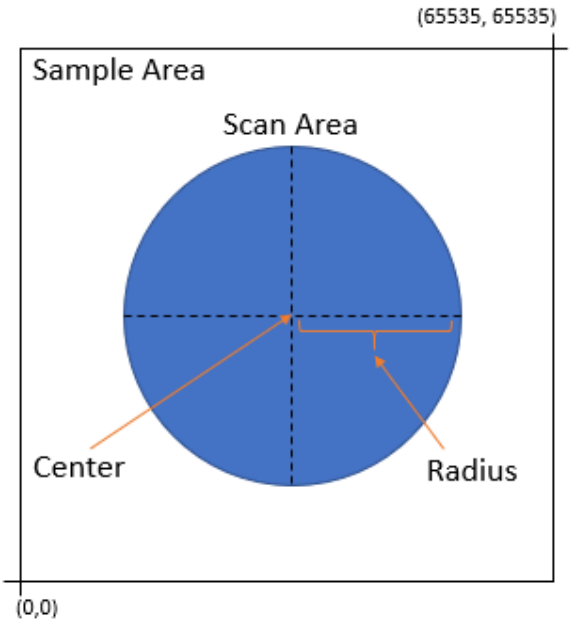


Figure 17 – Defining the Scan Parameters

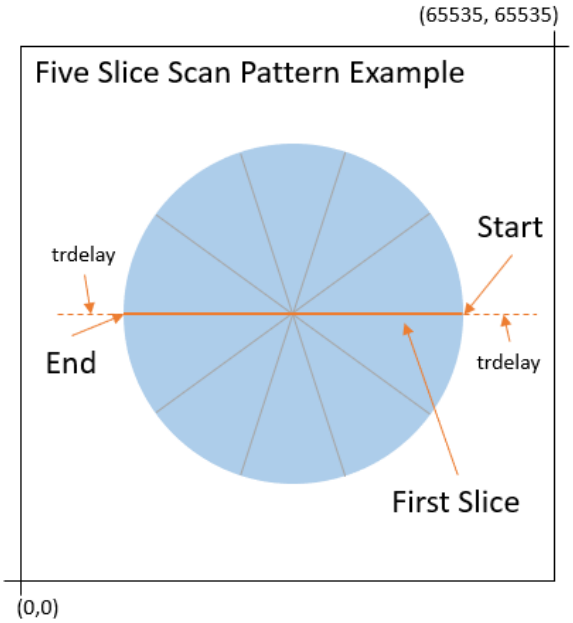


Figure 18 – Impact of Trigger Delay

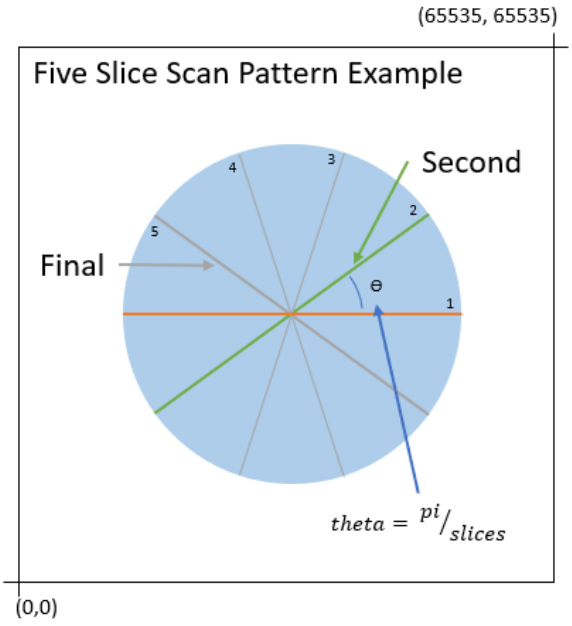


Figure 19 – Impact of b_scans

Command:	rotcross <Center X> <Center Y> <radius> <crosses> <passes> < Theta> <dTheta>
Response:	A
Remarks:	The **Rotating Cross** function performs a series of crosses through the specified origin. The fly-back path is altered on this scan. **Center –**The center values will indicate the origin position for the scan in DAC values. **Radius** – Scanning radius based on the specified center of the scanning pattern. **Crosses –** The total number of crosses through the origin. **Passes –** The amount of times each individual cross is scanned before the next. **Theta –** Starting angle of first cross. Units are in degrees. **dTheta** – The incremental increase in angle between each cross. Units in degrees. ***NOTE1:*** The Fly-back path is altered on this scan. The destination of the flyback for this scan is the starting point of the next line instead of the starting point of the current line. ***NOTE2:*** Trigger delay is used in the same fashion as the X/Y Ramp and XY Ramp scanning patterns with trdmode forced ON.

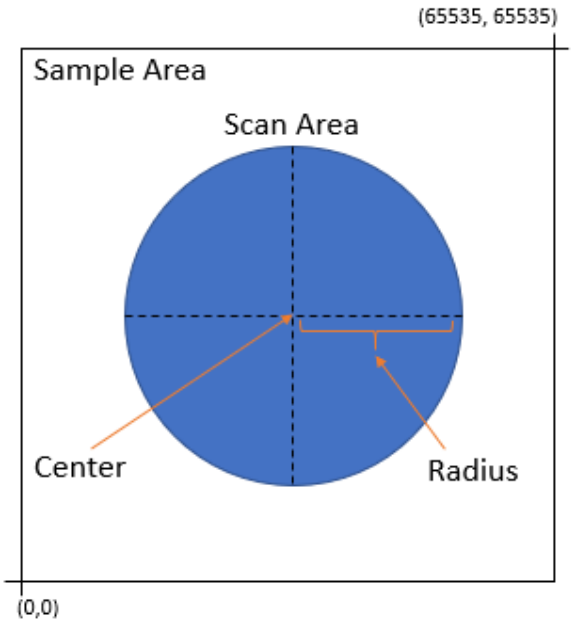


Figure 20 – Defining the Scan Parameters

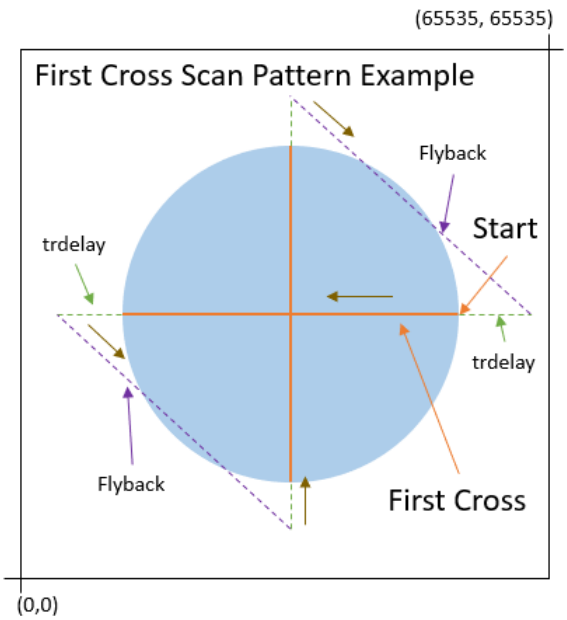


Figure 21 – Trigger Delay and Flybacks

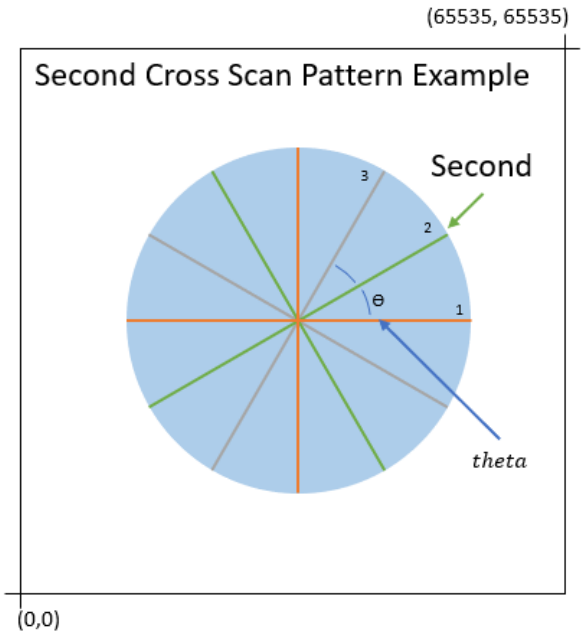


Figure 22 – Next Scan Progression

Command:	scan <count>
Response:	A
Remarks:	Initiates a scan. If no arguments or if count is negative , the system scans indefinitely, Otherwise, the system will scan the number of times specified by the count. In vector mode, count is the number of single b-scans to complete. In either of the two raster modes, count signifies the number of c-scans to complete. Count can be set to +/- 2,147,483,648.

Command:	ntscan
Response:	A
Remarks:	As above but without triggers

Command:	stop
Response:	A
Remarks:	Stops scanning at the end of the current b scan. Will turn off mirrors.

Command:	pause
Response:	#A Done Timeout
Remarks:	Stops the current waveform at the end of the pattern. The paused pattern will timeout as specified by the **ptimeout** command. When the waveform stops at the end of a pattern the system will respond with **Done** and will respond with **Timeout** when the timeout counter specified by the **ptimeout** command expires.

Command:	ptimeout <value>
Response:	A
Remarks:	This is a value in milliseconds for the waveform to hold position after a **pause** event has occurred. When the timeout expires, the system will execute a **stop** command and then respond with the ascii string **Timeout** over the USB serial port.

Command:	mirror <##>
Response:	A
Remarks:	Turns on frequency generator for mirror filter. Value can be from 0-80 which sets the cut off frequency in 10s of Hz. For example “mirror 40” sets the cut off frequency to 400 Hz, and “mirror 40 1” is the same as 40.1 or 401 Hz. Setting it to 0 turns off the

	filter generator and the HV supply. Will respond with current setting if value not entered.

Motor Commands

Motor	Description
q	Quarter wave stepper motor
p	Path motor
h	Half wave stepper motor
a	All motors

Command:	mgr <motor> <value>
Response:	A
Remarks:	The **Motor Go Relative** function sends the motor to the specified relative location. This applies to motors q, p, or h

Command:	mg2 <motor> <value>
Response:	A
Remarks:	The **Motor Go To** function sends the motor to the specified absolute location This applies to motors q, p, or h

Command:	mgh <motor>
Response:	A
Remarks:	The **Motor Home** function send motor re-homes the specified motor. This applies to motors q, p, h, or a

Command:	mstop <motor>
Response:	A
Remarks:	The **Motor Stop** command stops the specified motor. This applies to motors q, p, h, or a

Command:	minfo
Response:	A
Remarks:	The **Motor Information** function responds with location and destination for each motor.

Command:	mih <motor>
Response:	A
Remarks:	The **Motor Is Home** function responds with 0 if the specified motor is not home else unique hexadecimal bit if home. This applies to motors q, p, h, or a

Command:	maset <motor> <value>
Response:	A
Remarks:	The **Motor Acceleration Set** function allows for fine tuning of the acceleration of the specified stepper motor. Values are in steps per second squared.

Motor	Default Value
p	5000
q	5000
h	5000

Command:	msset <motor> <value>
Response:	A
Remarks:	The **Motor Speed Set** function allows for fine tuning of the speed of the specified stepper motor. Values are in steps per second.

Motor	Default Value
p	1450
q	5000
h	5000
